
Online Estimation via Offline Estimation: An Information-Theoretic Framework

Dylan J. Foster

dylanfoster@microsoft.com

Yanjun Han

yanjunhan@nyu.edu

Jian Qian

jianqian@mit.edu

Alexander Rakhlin

rakhlin@mit.edu

Abstract

The classical theory of statistical estimation aims to estimate a parameter of interest under data generated from a fixed design (“offline estimation”), while the contemporary theory of online learning provides algorithms for estimation under adaptively chosen covariates (“online estimation”). Motivated by connections between estimation and interactive decision making, we ask: *is it possible to convert offline estimation algorithms into online estimation algorithms in a black-box fashion?* We investigate this question from an information-theoretic perspective by introducing a new framework, *Oracle-Efficient Online Estimation* (OEOE), where the learner can only interact with the data stream indirectly through a sequence of *offline estimators* produced by a black-box algorithm operating on the stream. Our main results settle the statistical and computational complexity of online estimation in this framework.

1. *Statistical complexity.* We show that information-theoretically, there exist algorithms that achieve near-optimal online estimation error via black-box offline estimation oracles, and give a nearly-tight characterization for minimax rates in the OEOE framework.
2. *Computational complexity.* We show that the guarantees above cannot be achieved in a computationally efficient fashion in general, but give a refined characterization for the special case of conditional density estimation: computationally efficient online estimation via black-box offline estimation is possible whenever it is possible via unrestricted algorithms.

Finally, we apply our results to give offline oracle-efficient algorithms for interactive decision making.

1 Introduction

Consider a general framework for statistical estimation specified by a tuple $(\mathcal{X}, \mathcal{Y}, \mathcal{Z}, \mathcal{K}, \mathcal{F})$, which we will show encompasses classification, regression, and conditional density estimation. The learner is given a parameter space \mathcal{F} (typically a function class), where each parameter $f \in \mathcal{F}$ is a map from the space of *covariates* \mathcal{X} to the space of *values* \mathcal{Z} . For an integer $T \geq 1$, the learner is given a dataset $(x^1, y^1), \dots, (x^T, y^T)$, where x^1, \dots, x^T are *covariates* and y^1, \dots, y^T are *outcomes* generated via $y^t \sim \mathcal{K}(\cdot \mid f^*(x^t))$, where $f^* \in \mathcal{F}$ is an unknown target parameter that the learner wishes to estimate; here \mathcal{K} is a probability kernel that assigns to each value $z \in \mathcal{Z}$ a distribution $\mathcal{K}(\cdot \mid z)$ on the space of outcomes \mathcal{Y} . We adopt the shorthand $\mathcal{K}(z) = \mathcal{K}(\cdot \mid z)$ throughout.

The classical theory of statistical estimation typically assumes that the *covariates* x^1, \dots, x^T are an arbitrary fixed design, and is concerned with estimating the target parameter $f^* \in \mathcal{F}$ well *in-distribution* [64, 40, 62]. Formally, for a *loss function* $\mathsf{D} : \mathcal{Z} \times \mathcal{Z} \rightarrow \mathbb{R}_{\geq 0}$ on the space of values \mathcal{Z} , the goal of the learner is to output an estimator \hat{f} based on $(x^1, y^1), \dots, (x^T, y^T)$ such that the in-distribution error

$$\mathbf{Est}_{\mathsf{D}}^{\text{Off}}(T) := \sum_{t=1}^T \mathsf{D}(\hat{f}(x^t), f^*(x^t)) \tag{1}$$

is small; we refer to this as an *offline estimation* guarantee. Canonical examples include:

- Classification (i.e., distribution-free PAC learning [38, 63]), where $\mathcal{Z} = \mathcal{Y} = \{0, 1\}$, $\mathcal{K}(f^*(x)) = \mathbb{1}_{f^*(x)}$,¹ and $D_{0/1}(\hat{f}(x), f^*(x)) = \mathbb{1}\{\hat{f}(x) \neq f^*(x)\}$ is the 0/1-loss.
- Regression with a well-specified model [62, 66], where $\mathcal{Z} = \mathcal{Y} = \mathbb{R}$, $\mathcal{K}(f^*(x)) = \mathcal{N}(f^*(x), \sigma^2)$, and $D_{\text{sq}}(\hat{f}(x), f^*(x)) = (\hat{f}(x) - f^*(x))^2$ is the square loss.
- Conditional density estimation [12], where \mathcal{Y} is an arbitrary alphabet, $\mathcal{Z} = \Delta(\mathcal{Y})$, $\mathcal{K}(f^*(x)) = f^*(x)$, and $D_{\text{H}}^2(\cdot, \cdot)$ is squared Hellinger distance; see Appendix C for details.

In parallel to statistical estimation, the contemporary theory of online learning [17, 49] provides estimation error algorithms that support *adaptively chosen* sequences of covariates, a meaningful form of *out-of-distribution* guarantee. Here, the examples (x^t, y^t) arrive one at a time. For each step $t \in [T]$, the learner produces an estimator $\hat{f}^t : \mathcal{X} \rightarrow \mathcal{Z}$ based on the data $(x^1, y^1), \dots, (x^{t-1}, y^{t-1})$ observed so far. The covariate x^t is then chosen in an arbitrary fashion, and the outcome is generated via $y^t \sim \mathcal{K}(f^*(x^t))$ and revealed to the learner. The quality of the estimators is measured via²

$$\text{Est}_{\text{D}}^{\text{On}}(T) := \sum_{t=1}^T D(\hat{f}^t(x^t), f^*(x^t)). \quad (2)$$

We refer to this as an *online estimation* guarantee; classical examples include online classification in the mistake-bound model [41], online regression [51], and online conditional density estimation [11]. Online estimation provides a non-trivial out-of-distribution guarantee, as it requires (on average) that the learner achieves non-trivial estimation performance on covariates x^t that can be arbitrarily far from the previous examples x^1, \dots, x^{t-1} . This property has many applications in algorithm design, notably in the context of *interactive decision making*, where it has recently found extensive use for problems including contextual bandits [25, 58, 24], reinforcement learning [27, 28], and imitation learning [56, 55].

In this paper, we investigate the relative power of online and offline estimation through a new information-theoretic perspective. It is well known that any algorithm for online estimation can be used *as-is* to solve offline estimation through *online-to-batch conversion*, a standard technique in learning theory and statistics [3, 9, 16, 61, 36, 8]. The converse is less apparent, as online estimation requires non-trivial algorithm design techniques that go well beyond classical estimators like least-squares or maximum likelihood [17]. In the case of regression with a finite class \mathcal{F} , least squares achieves optimal offline estimation error $\text{Est}_{\text{D}}^{\text{Off}}(T) \leq O(\log|\mathcal{F}|)$,³ and while it is possible to achieve a similar rate $\text{Est}_{\text{D}}^{\text{On}}(T) \leq O(\log|\mathcal{F}|)$ for online estimation, this requires Vovk's aggregating algorithm or exponential weights [65]; directly applying least squares or other standard offline estimators leads to vacuous guarantees. This leads us to ask: *Is it possible to convert offline estimation algorithms into online estimation algorithms in a black-box fashion?*

Computationally speaking, this question has practical significance, since online estimation algorithms are typically far less efficient than their offline counterparts (the classical exponential weights algorithm maintains a separate weight for every $f \in \mathcal{F}$, which is exponentially less memory-efficient than empirical risk minimization). In fact, at first glance this seems like a *purely* computational question: if the learner has access to an offline estimator, nothing is stopping them (information-theoretically) from throwing the estimator away and using the data to run an online estimation algorithm.⁴ Yet, for aforementioned applications in interactive decision making [25, 58, 24, 27, 28, 56, 55], estimation algorithms—particularly online estimators—play a deeper information-theoretic role, and can be viewed as compressing the data stream into a succinct, operational representation that directly informs downstream decision making. With these applications in mind, the first contribution of this paper is to introduce a new protocol, *Oracle-Efficient Online Estimation*, which provides an *information-theoretic* abstraction of the role of online versus offline estimation, analogous to the framework of information-based complexity in optimization [43, 60, 48, 2] and statistical query complexity in theoretical computer science [13, 37, 21, 22].

¹We use $\mathbb{1}_y$ to indicate the delta distribution that places probability mass 1 on y .

²For technical reasons, it is also common to consider randomized estimators where $\hat{f}^t \sim \mu^t$, and measure error by $\text{Est}_{\text{D}}^{\text{On}}(T) := \sum_{t=1}^T \mathbb{E}_{\hat{f}^t \sim \mu^t} [D(\hat{f}^t(x^t), f^*(x^t))]$.

³We consider *unnormalized* estimation error in Eq. (1), following the convention of online learning. For normalized estimation error, we have $\frac{1}{T} \text{Est}_{\text{D}}^{\text{Off}}(T) \leq \frac{\log|\mathcal{F}|}{T}$, following the convention of statistical estimation.

⁴Related computational questions have already been studied, with negative results [14, 33].

Protocol 1 Oracle-Efficient Online Estimation (OEOE)

- 1: **for** $t = 1, \dots, T$ **do**
- 2: Learner receives estimator $\hat{f}^t = \text{Alg}_{\text{Off}}^t(x^1, \dots, x^{t-1}, y^1, \dots, y^{t-1})$ from offline est. oracle.
- 3: Based on x^1, \dots, x^{t-1} and $\hat{f}^1, \dots, \hat{f}^t$, learner produces an estimator $\bar{f}^t \in \mathcal{Z}^{\mathcal{X}}$, which may be randomized according to a distribution μ^t .
- 4: Based on μ^t , nature selects covariate $x^t \in \mathcal{X}$ and outcome $y^t \sim \mathcal{K}(f^*(x^t))$, but does not directly reveal them to the learner.

1.1 Our protocol: oracle-efficient online estimation

In the Oracle-Efficient Online Estimation (OEOE) framework, the aim is to perform *online* estimation in the sense of Eq. (2), with the twist that the learner does not directly observe the outcomes y^1, \dots, y^T ; rather, they interact with the environment *indirectly* through a sequence of *offline* estimators produced by a black-box algorithm operating on the historical data. We formalize this black-box algorithm as an *estimation oracle* $\text{Alg}_{\text{Off}} = \{\text{Alg}_{\text{Off}}^t\}_{t=1}^T$ (e.g., Foster and Rakhlin [26]), which is a mapping from histories to estimators that enjoy bounded offline estimation error.

Definition 1.1 (Offline estimation oracle). *An offline estimation oracle $\text{Alg}_{\text{Off}} = \{\text{Alg}_{\text{Off}}^t\}_{t=1}^T$ for a statistical estimation instance $(\mathcal{X}, \mathcal{Y}, \mathcal{Z}, \mathcal{K}, \mathcal{F})$ and loss D is a mapping $\text{Alg}_{\text{Off}}^t : (\mathcal{X} \times \mathcal{Y})^{t-1} \rightarrow (\mathcal{X} \rightarrow \mathcal{Z})$ such that for any sequence $(x^1, y^1), \dots, (x^T, y^T)$ with $y^t \sim \mathcal{K}(f^*(x^t))$, the sequence of estimators $\hat{f}^t = \text{Alg}_{\text{Off}}^t(x^1, \dots, x^{t-1}, y^1, \dots, y^{t-1})$ satisfies $\text{Est}_D^{\text{Off}}(t) := \sum_{s=1}^{t-1} D(\hat{f}^s(x^s), f^*(x^s)) \leq \beta_{\text{Off}}$ for all $t \in [T]$ almost surely; we allow x^t to be selected adaptively based on y^1, \dots, y^{t-1} and $\hat{f}^1, \dots, \hat{f}^{t-1}$. We refer to $\beta_{\text{Off}} \geq 0$ as the offline estimation parameter.*

This definition simply asserts that the estimators \hat{f}^t produced by the offline estimation oracle satisfy the guarantee in Eq. (1), even when the covariates are selected adaptively. Examples include standard algorithms like least-squares for regression and maximum likelihood for conditional density estimation, which guarantee $\beta_{\text{Off}} \leq O(\log|\mathcal{F}|)$ with high probability whenever \mathcal{F} is a finite class; see Appendix C.1 for further background.⁵ Throughout the paper, we assume for simplicity that $\beta_{\text{Off}} > 0$ is known in advance.

With this definition, we present the Oracle-Efficient Online Estimation protocol in Protocol 1. In the protocol, a learner aims to perform online estimation, but at each step t , the only information available is the covariates x^1, \dots, x^{t-1} and the estimators $\hat{f}^1, \dots, \hat{f}^t$ generated by an offline estimation oracle satisfying Definition 1.1; the outcomes y^1, \dots, y^T are not directly observed. Based on this information, the learner produces a new estimator \bar{f}^t such that the online estimation error $\text{Est}_D^{\text{On}}(T) = \sum_{t=1}^T \mathbb{E}_{\bar{f} \sim \mu^t} [D(\bar{f}(x^t), f^*(x^t))]$ in Eq. (2) is minimized.⁶ An algorithm is termed *oracle-efficient* if it attains low online estimation error (2) in the OEOE framework. Note that while the learner cannot directly observe the outcomes y^1, \dots, y^T , the covariates x^1, \dots, x^T are observed; we prove that without this ability, it is impossible to achieve non-trivial estimation performance (Section 3).

The OEOE framework abstracts away the property that oracle-efficient algorithms implicitly interact with the environment through a compressed, potentially lossy channel (the estimation oracle Alg_{Est}). We believe this property merits deeper investigation: it is shared by essentially all algorithms from recent research that reduces interactive decision making and reinforcement learning to estimation oracles [25, 58, 27, 28, 56, 55], yet the relative power of offline oracles and analogously defined online oracles is poorly understood in this context. By providing an information-theoretic abstraction to study oracle-efficiency, the OEOE framework plays a role similar to information-based complexity in optimization [43, 60, 48, 2] and statistical query complexity in theoretical computer science [13, 37, 21, 22], both of which provide rich frameworks for designing and evaluating iterative algorithms that interact with the environment in a structured fashion. We expect that this abstraction will find broader use for more complex domains (e.g., decision making and active learning) as a means to guide algorithm design and prove lower bounds against natural classes of algorithms.

⁵Most algorithms only ensure that the guarantee in Definition 1.1 holds with high probability. We assume an almost sure bound to simplify exposition, but our results trivially extend. Likewise, our results immediately extend to handle the case in which β_{Off} is allowed to grow as a (sublinear) function of t .

⁶For technical reasons, we allow the learner to randomize the estimator \bar{f}^t via a distribution μ^t .

Let us first build some intuition. Familiar readers may recognize that the classical *halving algorithm* for binary classification (e.g., Cesa-Bianchi and Lugosi [17]) can be viewed as oracle-efficient in our framework. Specifically, for binary classification with $\mathcal{Y} = \mathcal{Z} = \{0, 1\}$ and loss function $D_{0/1}(\hat{f}(x), f^*(x)) = \mathbb{1}\{\hat{f}(x) \neq f^*(x)\}$, the halving algorithm can use any offline oracle with $\beta_{\text{Off}} = 0$ to achieve $\text{Est}_D^{\text{On}}(T) = O(\log|\mathcal{F}|)$, which is optimal. However, little is known for noisy oracles with $\beta_{\text{Off}} > 0$, or more general outcome spaces and loss functions (e.g., regression or density estimation). In addition, the halving algorithm—while oracle-efficient—is computationally inefficient, as it requires maintaining an explicit version space. This leads us to restate our central question formally, in two parts:

1. *Can we design oracle-efficient algorithms with near-optimal online estimation error (2), up to polynomial factors (for general instances $(\mathcal{X}, \mathcal{Y}, \mathcal{Z}, \mathcal{K}, \mathcal{F})$ and $\beta_{\text{Off}} > 0$)?*
2. *Can we do so in a computationally efficient fashion?*

1.2 Contributions

For a general class of losses D , referred to as *metric-like*, we settle the statistical and computational complexity of performing online estimation via black-box offline estimation oracles up to mild gaps, answering questions (1) and (2) above.

Statistical complexity. Our first result concerning statistical complexity focuses on finite classes \mathcal{F} , where the optimal rates for offline and online estimation with standard losses $D(\cdot, \cdot)$ both scale as $\Theta(\log|\mathcal{F}|)$. For this setting, we show (Theorem 3.1) that there exists an oracle-efficient online estimation algorithm that achieves $\text{Est}_D^{\text{On}}(T) = O((\beta_{\text{Off}} + 1) \min\{\log|\mathcal{F}|, |\mathcal{X}|\})$ in the OOE framework, and that this is optimal (Theorem 3.2). This provides an affirmative answer to question (1), and characterizes the statistical complexity of oracle-efficient online estimation with finite classes \mathcal{F} .

In the general OOE framework, the learner can use the entire history of offline estimators $\hat{f}^1, \dots, \hat{f}^t$ and covariates x^1, \dots, x^{t-1} to produce the online estimator \bar{f}^t for step t . As a secondary result, we study a restricted class of *memoryless* oracle-efficient algorithms that choose \bar{f}^t only based on the most recent offline estimator \hat{f}^t , and show (Theorem 3.3) that it is impossible for such algorithms to achieve low online estimation error.

Lastly, we give a more general approach to deriving oracle-efficient reductions (Theorem D.1) that is based on *delayed online learning* [67, 42, 35, 47]. Using this result, we give a characterization of learnability with *infinite classes* for binary classification in the OOE framework (Theorem D.2), proving that finite Littlestone dimension is necessary and sufficient for oracle-efficient learnability.

Computational complexity. On the computational side, we provide a negative answer to question (2), showing (Theorem 4.1) that under standard conjectures in computational complexity, there do not exist polynomial-time algorithms with non-trivial online estimation error in OOE framework. In spite of this negative result, we provide a fine-grained perspective for the statistical problem of *conditional density estimation*, a general task that subsumes classification and regression and has immediate applications to reinforcement learning and interactive decision making [27, 28]. Here we show, perhaps surprisingly (Theorem 4.2), that online estimation in the OOE framework is no harder computationally than online estimation with arbitrary, unrestricted algorithms. This result is salient in light of the applications we discuss below.

Implications for interactive decision making. As the preceding discussion has alluded to, our interest in studying oracle-efficient online estimation is largely motivated by a connection to the problem of *interactive decision making*. Foster et al. [27, 28], Foster and Rakhlin [26] propose a general framework for interactive decision making called *Decision Making with Structured Observations* (DMSO), which subsumes contextual bandits, bandit problems with structured rewards, and reinforcement learning with general function approximation. They show that for any decision making problem in the DMSO framework, there exists an algorithm that, given access to an online estimation algorithm (or, “oracle”) for conditional density estimation for an appropriate class \mathcal{F} , it is possible to achieve *near-optimal regret*. The results above critically make use of *online estimation* oracles, as they require achieving low estimation error for adaptively chosen sequences of covariates, and it is natural to ask whether similar guarantees can be achieved using only offline estimation oracles. However, positive results are only known for certain special cases [18–20, 58], with scant results for reinforcement learning

in particular. In this context, our results have the following implication (Corollary E.1): *Information-theoretically, it is possible to achieve near-optimal regret for any interactive decision making problem using an algorithm that accesses the data stream only through offline estimation oracles.*

Additional results. *Due to space constraints, the following results are deferred to the appendix: (i) detailed examples for our statistical estimation framework (Appendix C); (ii) additional results concerning statistical complexity of the OEOE framework (Appendix D); and (iii) detailed results for our application to interactive decision making (Appendix E).*

2 Preliminaries

Unless otherwise stated, our results assume the loss function D has *metric-like* structure.

Definition 2.1 (Metric-like loss). *A loss function $D : \mathcal{Z} \times \mathcal{Z} \rightarrow [0, 1]$ is metric-like on the set \mathcal{Z} if it is symmetric and satisfies (i) $D(z_1, z_2) \geq 0$ for any $z_1, z_2 \in \mathcal{Z}$ and $D(z, z) = 0$ for all $z \in \mathcal{Z}$; and (ii) $D(z_1, z_2) \leq C_D \cdot (D(z_1, z_3) + D(z_3, z_2))$ for all $z_1, z_2, z_3 \in \mathcal{Z}$, for an absolute constant $C_D \geq 1$.*

Throughout the paper, we focus on three canonical applications, outlined in the introduction: Classification with the indicator loss $D_{0/1}$ ($C_D = 1$), regression with the square loss D_{sq} ($C_D = 2$), and conditional density estimation with squared Hellinger distance D_H^2 ($C_D = 2$). See Appendix C for detailed examples and discussion. (omitted for space).

Finite versus infinite classes. The majority of our results focus on finite classes \mathcal{F} . We believe this captures the essential difficulty of the problem, but we expect that most of our sample complexity results (which typically scale with $\log |\mathcal{F}|$) can be extended to infinite classes by combining our techniques with appropriate notions of complexity for the function class (Littlestone dimension for classification, sequential Rademacher complexity, and sequential covering numbers [50, 41, 52]). For the canonical settings of classification, regression, and conditional density estimation, there exist algorithms that achieve $\text{Est}_D^{\text{Off}}(T) = O(\log |\mathcal{F}|)$ and $\text{Est}_D^{\text{On}}(T) = O(\log |\mathcal{F}|)$ for arbitrary finite classes; see Appendix C for details.

We defer additional notation and related work to Appendices A and B

3 Statistical complexity of oracle-efficient online estimation

This section presents our main results concerning the statistical complexity of oracle-efficient online estimation. In Section 3.1, we focus on finite classes \mathcal{F} and present an oracle-efficient algorithm that achieves near-optimal online estimation error (Theorem 3.1). We then provide a lower bound that shows that our reduction is near optimal (Theorem 3.2). In Section 3.2, we turn our attention to memoryless oracle-efficient algorithms, proving strong impossibility results (Theorem 3.3).

3.1 Minimax sample complexity for oracle-efficient algorithms

In this section, we present our main statistical conclusion for the OEOE framework: *For any finite class \mathcal{F} , it is possible to transform any black-box offline estimation algorithm into an online estimation algorithm with near-optimal error* (up to a logarithmic factor that we show is unavoidable).

Algorithm and minimax upper bound. Our results are achieved through a new algorithm, *Version Space Averaging*, described in Algorithm 1. At each round t , the algorithm uses estimators $\hat{f}^1, \dots, \hat{f}^t$ produced by an offline estimation oracle Alg_{Off} , along with the previous covariates x^1, \dots, x^{t-1} , to construct a *version space* $\mathcal{F}_t \subseteq \mathcal{F}$ in Eq. (3). Informally, \mathcal{F}_t consists of all $f \in \mathcal{F}$ that are *consistent* with the estimators $\hat{f}^1, \dots, \hat{f}^t$ in the sense that for all $s \in [t]$, the offline estimation error relative to \hat{f}^s is small; as long as the offline estimation oracle Alg_{Off} has offline estimation error β_{Off} (Definition 1.1), it follows immediately that the construction in Eq. (3) satisfies $f^* \in \mathcal{F}_t$. Given the version space \mathcal{F}_t , Algorithm 1 predicts by uniformly sampling: $\hat{f}^t \sim \mu^t := \text{Unif}(\mathcal{F}_t)$, then proceeds to the next round.⁷ The main guarantee for Algorithm 1 is stated in Theorem 3.1.

⁷For realizable binary classification with $D_{0/1}(\hat{f}(x), f^*(x)) = \mathbb{1}(\hat{f}(x) \neq f^*(x))$ and $\beta_{\text{Off}} = 0$, the version space construction in Eq. (3) coincides with that of the halving algorithm (e.g., Cesa-Bianchi and Lugosi [17]), and its estimation error bound (Theorem 3.1) matches the halving algorithm up to absolute constants. As such, Algorithm 1 can be viewed as a noisy/error-tolerant generalization of the halving algorithm, which may find broader use.

Algorithm 1 Version Space Averaging

- 1: **input:** parameter space \mathcal{F} , offline estimation oracle Alg_{Off} with parameter $\beta_{\text{Off}} \geq 0$.
- 2: **for** $t = 1, 2, \dots, T$ **do**
- 3: Receive $\hat{f}^t = \text{Alg}_{\text{Off}}^t(x^1, \dots, x^{t-1}, y^1, \dots, y^{t-1})$.
- 4: Calculate version space:
$$\mathcal{F}_t = \left\{ f \in \mathcal{F} \mid \forall s \in [t], \sum_{\tau < s} D(\hat{f}^s(x^\tau), f(x^\tau)) \leq \beta_{\text{Off}} \right\}. \quad (3)$$
- 5: Predict $\bar{f}^t \sim \mu^t := \text{Unif}(\mathcal{F}_t)$ and receive x^t . *// Nature draws $y^t \sim \mathcal{K}(f^*(x^t))$ and passes to Alg_{Off} .*

Theorem 3.1 (Main upper bound for OOE). *For any instance $(\mathcal{X}, \mathcal{Y}, \mathcal{Z}, \mathcal{K}, \mathcal{F})$, any metric-like loss D , and any offline estimator Alg_{Off} with parameter $\beta_{\text{Off}} \geq 0$, Algorithm 1 is oracle-efficient and achieves*

$$\text{Est}_D^{\text{On}}(T) \leq O(C_D \cdot (\beta_{\text{Off}} + 1) \cdot \min \{\log |\mathcal{F}|, |\mathcal{X}| \log T\}).$$

Most notably, Algorithm 1 achieves $\text{Est}_D^{\text{On}}(T) \leq O(C_D \cdot (\beta_{\text{Off}} + 1) \cdot \log |\mathcal{F}|)$; that is, up to a $O(\log |\mathcal{F}|)$ factor, the reduction achieves online estimation rates in the OOE framework that are no worse than the minimax rate for offline estimation. For classification, regression, and density estimation with generic finite classes \mathcal{F} (Appendix C), the best possible offline estimation error rate is $\beta_{\text{Off}} = O(\log |\mathcal{F}|)$, so this shows that price of oracle-efficiency is at most quadratic.

Minimax lower bound. Next, we show that the upper bound in Theorem 3.1 is nearly tight, giving a lower bound that matches up to logarithmic factors.

Theorem 3.2 (Main lower bound for OOE). *Consider the binary classification setting with $\mathcal{Z} = \mathcal{Y} = \{0, 1\}$ and loss $D_{0/1}(\cdot, \cdot)$. For any $N \in \mathbb{N}$ and $\beta_{\text{Off}} > 0$, there exists an instance $(\mathcal{X}, \mathcal{Y}, \mathcal{Z}, \mathcal{K}, \mathcal{F})$ with $\log |\mathcal{F}| = |\mathcal{X}| = N$ such that for any oracle-efficient algorithm, there is a sequence of covariates (x^1, \dots, x^T) and offline oracle with parameter β_{Off} such that $\mathbb{E}[\text{Est}_D^{\text{On}}(T)] \geq \Omega(\min \{(\beta_{\text{Off}} + 1)N, T\})$.*

This result states that for a generic finite class \mathcal{F} and offline estimation oracle Alg_{Off} , any oracle-efficient online estimator must have

$$\mathbb{E}[\text{Est}_D^{\text{On}}(T)] \geq \Omega(\min \{(\beta_{\text{Off}} + 1) \log |\mathcal{F}|, (\beta_{\text{Off}} + 1) |\mathcal{X}|, T\})$$

in the worst case. This implies that the $\log |\mathcal{F}|$ factor we pay for offline to online conversion is unavoidable, and that Theorem 3.1 is optimal up to a $\log T$ factor, giving a near-optimal characterization for the minimax rate for online estimation in the OOE framework. We conclude with two remarks: (i) The $(\beta_{\text{Off}} + 1)$ scaling (as opposed to say, β_{Off}) in Theorem 3.1 is unavoidable, as witnessed by the optimality of the halving algorithm for noiseless binary classification [17]; (ii) if the space \mathcal{Z} and the loss D are convex, then we can change Algorithm 1 to output a deterministic prediction by using the average of all parameters in \mathcal{F}_t rather than the uniform distribution on \mathcal{F}_t . See Lemma G.1 for details.

General reductions and infinite classes. Algorithm 1 is somewhat specialized to finite classes. In Appendix D (deferred to the appendix for space), we provide a more general approach to designing oracle-efficient algorithms based on *delayed online learning* (Theorem D.1), and use it to derive a characterization of oracle-efficient learnability for classification with infinite classes \mathcal{F} (Theorem D.2).

Full memory vs. finite memory. Algorithm 1 requires full memory of all past offline estimators. The more general approach proposed in Appendix D can use N most recent offline estimators to obtain an estimation error bound of $O(C_D(N + \beta_{\text{Off}}T/N) + N \cdot \log |\mathcal{F}|)$ (Corollary D.1) for any integer $N > 0$.

3.2 Impossibility of memoryless oracle-efficient algorithms

In the general OOE framework, the learner can use the entire history of estimators $\hat{f}^1, \dots, \hat{f}^t$ and covariates x^1, \dots, x^{t-1} to produce the online estimator \bar{f}^t for step t ; notably the Version Space Averaging algorithm with which our upper bounds in the prequel are derived uses the entire history. In this section, we show that for *memoryless* oracle-efficient algorithms (Definition 3.1) that select the estimator \bar{f}^t only as a function of the most recent offline estimator \hat{f}^t , similar guarantees are impossible.

Definition 3.1 (Memoryless algorithm). *An online estimation algorithm is memoryless if there exists a map $F^t(\cdot)$ such that we can write $\mu^t = F^t(\hat{f}^t)$, where $\hat{f}^t = \mathbf{Alg}_{\text{Off}}^t(x^1, \dots, x^{t-1}, y^1, \dots, y^{t-1})$ and μ^t is the randomization distribution for the online estimator \hat{f}^t .⁸*

Memoryless algorithms are more practical than arbitrary algorithms, since they do not require storing past estimators or covariates in memory. Our motivation for studying memoryless algorithms arises from recent work in interactive decision making [25, 27], which shows that there exist near-optimal algorithms for contextual bandits and reinforcement learning that use estimation oracles in memoryless fashion. We show that unfortunately, it is not possible to convert offline estimators into memoryless online estimation algorithms with non-trivial error.

Theorem 3.3 (Impossibility of memoryless algorithms for OOE). *Consider the binary classification setting with $\mathcal{Z} = \mathcal{Y} = \{0, 1\}$ and loss $D_{0/1}(\cdot, \cdot)$. For any $N \in \mathbb{N}$ and $\beta_{\text{Off}} \geq 0$, there exists an instance $(\mathcal{X}, \mathcal{Y}, \mathcal{Z}, \mathcal{K}, \mathcal{F})$ with $|\mathcal{F}| = |\mathcal{X}| = N$ such that for any memoryless oracle-efficient algorithm, there exists a sequence of covariates (x^1, \dots, x^T) and a (potentially improper) offline oracle $\mathbf{Alg}_{\text{Off}}$ with parameter β_{Off} such that $\mathbb{E}[\mathbf{Est}_D^{\text{On}}(T)] \geq \Omega(\min\{N(\beta_{\text{Off}} + 1), T\})$. This conclusion still holds when the online estimation algorithm remembers $\hat{f}^1, \dots, \hat{f}^{t-1}$ but not x^1, \dots, x^{t-1} .*

This result shows that in the worst case, any memoryless oracle-efficient algorithm must have

$$\mathbb{E}[\mathbf{Est}_D^{\text{On}}(T)] \geq \Omega((\beta_{\text{Off}} + 1) \min\{|\mathcal{X}|, |\mathcal{F}|\}).$$

This precludes an online estimation error bound scaling with $(\beta_{\text{Off}} + 1) \log |\mathcal{F}|$ as in Theorem 3.1, and shows that the gap between general and memoryless oracle-efficient algorithms can be exponential.

Interestingly, the lower bound in Theorem 3.3 holds even if the online estimation algorithm is allowed to remember $\hat{f}^1, \dots, \hat{f}^{t-1}$, but not x^1, \dots, x^{t-1} . The intuition here is that without covariate information, it is not possible to aggregate the predictions of previous estimators or otherwise use them to reduce uncertainty. This provides post-hoc motivation for our decision to incorporate covariate memory into the OOE protocol in Section 1.1.

The proof of Theorem 3.3 uses that the estimators \hat{f}^t produced by the offline estimation oracle may be *improper* (i.e., $\hat{f}^t \notin \mathcal{F}$). We defer a variant of the result that holds even if the estimation oracle is *proper* under additional assumptions as well as the complementary upper bound to Appendix D.2

4 Computational complexity of oracle-efficient online estimation

In this section, we turn our attention to the computational complexity of oracle-efficient online estimation in the OOE framework. In Section 4.1, we show (Theorem 4.1) that in general, it is not possible to transform black-box offline estimation algorithms into online estimation algorithms in a computationally efficient fashion. Then, in Section 4.2, we provide a more fine-grained perspective, showing (Theorem 4.2) that for conditional density estimation, online estimation in the OOE framework is no harder computationally than online estimation with unrestricted algorithms.

4.1 Computational hardness of oracle-efficient estimation

Our main upper bounds (Section 3.1) show that online estimation error $\mathbf{Est}_D^{\text{On}}(T) \leq O((\beta_{\text{Off}} + 1) \log |\mathcal{F}|)$ can be achieved in an oracle-efficient fashion for any finite class \mathcal{F} , but the algorithm (Algorithm 1) is not computationally efficient. We now show that this is fundamental: There exist classes \mathcal{F} for which offline estimation can be performed in polynomial time, yet no oracle-efficient algorithm running in polynomial-time algorithm can achieve sublinear online estimation error.

Computational model. To present our results, we must formalize a computational model for oracle-efficient online estimation, and in particular, define a notion of *input length* for oracle-efficient online algorithms. To do so, we restrict our attention to noiseless binary classification, and consider a sequence of classification instances indexed by $n \in \mathbb{N}$, with $\mathcal{Z} = \mathcal{Y} = \{0, 1\}$, $\mathcal{X}_n := \{0, 1\}^n$, $\mathcal{K}(z) = \mathbb{1}_z$, and indicator loss $D_{0/1}(\cdot, \cdot)$. We consider a sequence of classes \mathcal{F}_n that have polynomial description length i.e. $\log |\mathcal{F}_n|$ is polynomial in n , so that $f \in \mathcal{F}_n$ can be described in $\text{poly}(n)$ bits. In particular, we assume that $f \in \mathcal{F}_n$ is represented as a Boolean circuit of size $\text{poly}(n)$ so that $f(x)$ can be computed in $\text{poly}(n)$ time for $x \in \mathcal{X}_n$; we refer to such sequences as *polynomially computable*.

⁸There are many natural variants of this protocol. For example, algorithms could select μ^t based on \hat{f}^t and B bits of auxiliary memory. We hope future work will explore these variants.

To allow for offline estimators that are improper, we assume that for all t and all sequences $(x^1, y^1), \dots, (x^T, y^T)$, the output $\hat{f}^t : \{0, 1\}^n \rightarrow \{0, 1\}$ returned by $\mathbf{Alg}_{\text{Off}}^t(x^1, \dots, x^{t-1}, y^1, \dots, y^{t-1})$ is a Boolean circuit of size $\text{poly}(n)$; we refer to such oracles as having $\text{poly}(n)$ -output description length.⁹ Likewise, to allow the online estimation algorithm itself to be improper and randomized, we restrict to algorithms for which computing $\bar{f}^t(x)$ for $\bar{f}^t \sim \mu^t$ can be implemented as $\bar{f}^t(x, r)$ for a random bit string $r \sim \text{Unif}(\{0, 1\}^B)$, where $B = \text{poly}(n)$; we refer to the online estimator as having $\text{poly}(n)$ -output description length if $f^t(\cdot, \cdot)$ is itself a Boolean circuit of size $\text{poly}(n)$. We refer to the online estimation algorithm *polynomial time* if it runs in time $\text{poly}(n)$ for any sequence of inputs t, x^1, \dots, x^{t-1} , and $\hat{f}^1, \dots, \hat{f}^{t-1}$, and has $\text{poly}(n)$ -output description length.¹⁰

Main lower bound. Our main computational lower bound is as follows.

Theorem 4.1 (Computational lower bound for OEOE). *Assume the existence of one-way functions.¹¹ There exists a sequence of polynomially computable classes $(\mathcal{F}_1, \mathcal{F}_2, \dots, \mathcal{F}_n, \dots)$, along with a sequence of $\text{poly}(n)$ -output description length offline oracles with $\beta_{\text{Off}} = 0$ associated with each \mathcal{F}_n , such that for any fixed polynomials $p, q : \mathbb{N} \rightarrow \mathbb{N}$ and all $n \in \mathbb{N}$ sufficiently large, any oracle-efficient online estimation algorithm with runtime bounded by $p(n)$ must have $\mathbb{E}[\text{Est}_{\text{D}}^{\text{On}}(T)] \geq T/4$ for all $1 \leq T \leq q(n)$. At the same time, there exists an inefficient algorithm that achieves $\mathbb{E}[\text{Est}_{\text{D}}^{\text{On}}(T)] \leq O(\sqrt{n})$ for all $T \in \mathbb{N}$.*

Informally, Theorem 4.1 shows that there exist a class \mathcal{F} and offline estimation oracles $\mathbf{Alg}_{\text{Off}}$ for which no oracle-efficient online estimation algorithm that runs in time

$$\text{poly}(\text{len}(\mathcal{X}), \text{len}(\mathcal{F}), \max_t \text{len}(\hat{f}^t), T)$$

can achieve sublinear estimation error, where $\text{len}(\mathcal{X})$ and $\text{len}(\mathcal{F})$ denote the number of bits required to describe $x \in \mathcal{X}$ and $f \in \mathcal{F}$, and $\text{len}(\hat{f}^t)$ denotes the size of the circuit required to compute $\hat{f}^t(x)$. Yet, low online estimation error *can* be achieved by an inefficient algorithm, and there exist efficient offline estimators with $\beta_{\text{Off}} = 0$ as well. The result is essentially a corollary of Blum [14]; we refer to Appendix H.1 for the detailed proof. We mention in passing that Hazan and Koren [33] also give lower bounds against reducing online learning to offline oracles, but in a somewhat different computational model; see Appendix B for detailed discussion.

Theorem 4.1 is slightly disappointing, since one of the main motivations for studying oracle-efficiency is to leverage offline estimators as a computational primitive. Combined with our results in Section 3, Theorem 4.1 shows that even though it is possible to be oracle-efficient information-theoretically, it is not possible to achieve this computationally. Nonetheless, we are optimistic that our abstraction can (i) aid in designing computationally efficient algorithms for learning settings beyond online estimation, and (ii) continue to serve as a tool to formalize lower bounds against natural classes of algorithms, as we have done here; see Section 5 for further discussion.

Remark 4.1. *Theorem 4.1 relies on the fact that the offline estimator may be improper (i.e., $\hat{f}^t \notin \mathcal{F}$). An interesting open problem is whether one can attain $\text{poly}(\log |\mathcal{F}_n|) \cdot o(T)$ online estimation error with runtime $\text{poly}(\log |\mathcal{F}_n|)$ given a proper offline estimation oracle with parameter $\beta_{\text{Off}} = 0$. \triangleleft*

4.2 Conditional density estimation: computationally efficient algorithms

In spite of the negative result in the prequel, which shows that efficient computation in the OEOE framework is not possible in general, we can provide a more fine-grained perspective on the computational complexity of oracle-efficient estimation for the problem *conditional density estimation*, a general task which subsumes classification and regression, and has immediate applications to reinforcement learning and interactive decision making [27, 28].

Conditional density estimation. Recall that conditional density estimation [71, 12] is the special case of the online estimation framework in Section 1 in which \mathcal{X} and \mathcal{Y} are arbitrary, $\mathcal{Z} = \Delta(\mathcal{Y})$, and the kernel is $\mathcal{K}(z) = z$; that is sampling $y \sim \mathcal{K}(f^*(x))$ is equivalent to sampling $y \sim f^*(x)$. For

⁹See, e.g., Arora and Barak [6], Definition 6.1

¹⁰The precise computational model for runtime under consideration (e.g., Turing machines or Boolean circuits) does not change the nature of our results.

¹¹Existence of one-way functions is a standard and widely believed complexity-theoretic assumption, which forms the basis of modern cryptography [30].

the loss, we use squared Hellinger distance: $D_H^2(f(x), f^*(x)) = \frac{1}{2} \int (\sqrt{f(y|x)} - \sqrt{f^*(y|x)})^2$; with online estimation error given by $\text{Est}_H^{\text{On}}(T) = \sum_{t=1}^T D_H^2(\hat{f}^t(x^t), f^*(x^t))$.

Base algorithm. Our result is based on a reduction. We assume access to a base algorithm \mathcal{A}_{CDE} for online estimation in the Conditional Density Estimation (CDE) framework, which is *unrestricted* in the sense that it is not necessarily oracle-efficient. That is, \mathcal{A}_{CDE} can directly use the full data stream $(x^1, y^1), \dots, (x^{t-1}, y^{t-1})$ at step t . For parameters $R_{\text{CDE}}(T)$ and $C_{\mathcal{F}} \geq 1$, we assume that for any $f^* \in \mathcal{F}$, the base algorithm \mathcal{A}_{CDE} ensures that for all $\delta \in (0, e^{-1})$, with probability at least $1 - \delta$,

$$\text{Est}_H^{\text{On}}(T) \leq R_{\text{CDE}}(T) + C_{\mathcal{F}} \cdot \log(1/\delta). \quad (4)$$

We define the total runtime for \mathcal{A}_{CDE} across all rounds as $\text{Time}(\mathcal{F}, T)$.

Main result. Our main result shows that any algorithm \mathcal{A}_{CDE} satisfying the guarantee above can be transformed into an oracle-efficient algorithm with only polynomial blowup in runtime. For technical reasons, we assume that $V := e \vee \sup_{f, f' \in \mathcal{F}, x \in \mathcal{X}, y \in \mathcal{Y}} \frac{f(y|x)}{f'(y|x)}$ is bounded; our sample complexity bounds scale only logarithmically with respect to this parameter. In addition, we assume that all $f \in \mathcal{F}$ and $x \in \mathcal{X}$ have $O(1)$ description length, and that one can sample $y \sim f(x)$ in $O(1)$ time.

Theorem 4.2. *Let \mathcal{A}_{CDE} be an arbitrary (unrestricted) online estimation algorithm that satisfies Eq. (4) and has runtime $\text{Time}(\mathcal{F}, T)$. Then for any $N \in \mathbb{N}$, there exists an oracle-efficient online estimation algorithm that achieves estimation error*

$$\mathbb{E}[\text{Est}_H^{\text{On}}(T)] \leq \tilde{O}(C_{\mathcal{F}} \log V \cdot \beta_{\text{Off}} T / N + N \cdot (R_{\text{CDE}}(T) + C_{\mathcal{F}} \log V))$$

with runtime $\text{poly}(\text{Time}(\mathcal{F}, T), \log |\mathcal{F}|, \log |\mathcal{X}|, T)$, where $\beta_{\text{Off}} \geq 0$ is the offline estimation parameter. The distributions μ^1, \dots, μ^T produced by the algorithm have support size $\text{poly}(\log |\mathcal{F}|, \log |\mathcal{X}|, T)$. As a special case, if the online estimation guarantee for the base algorithm holds with $R_{\text{CDE}}(T) \leq C'_{\mathcal{F}} \log T$ for some problem-dependent constant $C'_{\mathcal{F}} \geq 1$, then by choosing N appropriately, we achieve $\mathbb{E}[\text{Est}_H^{\text{On}}(T)] \leq \tilde{O}((C_{\mathcal{F}}(C_{\mathcal{F}} + C'_{\mathcal{F}})\beta_{\text{Off}})^{1/2} \log V \cdot T^{1/2} + (C_{\mathcal{F}} + C'_{\mathcal{F}}) \log V)$.

Note that the estimation error bound in Theorem 4.2 is sublinear whenever the rate $R_{\text{CDE}}(T)$ is. This implies that for squared Hellinger distance, online estimation in the OEOE framework is no harder computationally than online estimation with arbitrary, unrestricted algorithms.

The proof of Theorem 4.2 is algorithmic, and is based on several layers of reductions. The main reason why the result is specialized to conditional density estimation is as follows: If we have an estimator \hat{f} for which $D_H^2(\hat{f}(x), f^*(x))$ is small for some x , we can simulate $y \sim f^*(x)$ up to low statistical error by sampling $y \sim \hat{f}(x)$ instead, as \hat{f} and f^* are close in distribution. This allows us to implement a scheme based on simulating outcomes and feeding them to the base algorithm.

5 Discussion

Our work introduces the Oracle-Efficient Online Estimation protocol as an information-theoretic framework to study the relative power of online and offline estimators and gives a nearly complete characterization of the statistical and computational complexity of learning in this framework. In what follows, we discuss broader implications for our information-abstraction of oracle-efficiency.

Oracle-efficient learning as a general framework for analysis of algorithms. One of the most important contributions of this work is to formalize oracle-efficient algorithms as mappings that act upon a sequence of estimators but do not directly act on historical outcomes. While the computational lower bounds we provide for oracle-efficient learning are somewhat disappointing, we are optimistic that—similar to statistical query complexity in TCS and information-based complexity in optimization—our abstraction can (i) aid in designing computationally efficient algorithms for learning settings beyond online estimation, and (ii) continue to serve as a tool to formalize lower bounds against natural classes of algorithms, for estimation and beyond. That is, we envision *oracle-efficient learning* as a more general framework to study oracle-based algorithms in any type of interactive learning problem. We remark that one need not restrict to offline oracles; it is natural to study oracle-efficient algorithms based on online estimation oracles or other types of oracles through our information-theoretic abstraction as well. For concreteness, let us mention a couple of natural settings where our information-theoretic abstraction can be applied.

Oracle-efficient interactive decision making. For interactive decision making problems like bandits and reinforcement learning (more broadly, the DMSO framework described in Appendix E.1), it is natural to formalize *oracle-efficient* algorithms as algorithms that do not directly observe rewards (bandits) or trajectories (reinforcement learning), and instead must select their decision based on an (online or offline) estimator (e.g., regression for bandits or conditional density estimation for RL). Foster and Rakhlin [25] and Foster et al. [27] et seq. provide algorithms with this property for contextual bandits and RL, respectively, but the power of offline oracles in this context is not well understood.

Oracle-efficient active learning. For active learning, it is natural to consider algorithms that decide whether to query the label for a point in an oracle-efficient fashion (e.g., Krishnamurthy et al. [39]). For concreteness, consider pool-based active learning [32]. Suppose the learner is given a pool $\mathcal{P} = \{x_1, \dots, x_n\}$ of covariates and a parameter space \mathcal{F} . The learner can repeatedly choose $x^t \in \mathcal{P}$ and call the offline oracle to obtain an estimator \hat{f}^t such $\text{Est}_D^{\text{Off}}(t) := \sum_{i < t} D(\hat{f}^t(x^i), f^*(x^i)) \leq \beta_{\text{Off}}$ (in contrast to an unrestricted algorithm that observes $y^t = f^*(x^t)$). The aim is to learn a hypothesis with low classification error using the smallest number of queries possible. Can we design oracle-efficient algorithms that do so with near-optimal label complexity?

5.1 Further Directions

We close with some additional directions for future research.

Refined notions of estimation oracles. This work considers generic offline estimation algorithms that satisfy the statistical guarantee in Definition 1.1 but can otherwise be arbitrary. Understanding the power of offline estimators that satisfy more refined (e.g., problem-dependent) guarantees is an interesting direction for future research.

Open questions for proper versus improper learning. Our results leave some interesting gaps in the power of proper versus improper oracles. First, the computational lower bounds in Section 4.1, leave open the possibility of attaining $\text{poly}(\log |\mathcal{M}_n|) \cdot o(T)$ online estimation error with runtime $\text{poly}(\log |\mathcal{M}_n|)$ given access to a *proper* offline estimation oracle with parameter $\beta_{\text{Off}} = 0$. Second, our results in Section 3.2 leave open the possibility of bypassing the $\Omega(|\mathcal{X}|(\beta_{\text{Off}} + 1))$ lower bound for memoryless algorithms under the assumption that the offline oracle is proper.

Acknowledgments and Disclosure of Funding

We acknowledge support from the ARO through award W911NF-21-1-0328 and from the Simons Foundation and the NSF through award DMS-2031883.

References

- [1] Anders Aamand, Justin Y Chen, Huy Lê Nguyen, and Sandeep Silwal. Improved space bounds for learning with experts. *arXiv preprint arXiv:2303.01453*, 2023.
- [2] Alekh Agarwal, Peter L Bartlett, Pradeep Ravikumar, and Martin J Wainwright. Information-theoretic lower bounds on the oracle complexity of stochastic convex optimization. *IEEE Transactions on Information Theory*, 5(58):3235–3249, 2012.
- [3] Aizerman, Braverman, and Rozonoer. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and remote control*, 25:821–837, 1964.
- [4] Philip Amortila, Dylan J Foster, Nan Jiang, Ayush Sekhari, and Tengyang Xie. Harnessing density ratios for online reinforcement learning. *International Conference on Learning Representations (ICLR)*, 2024.
- [5] Yossi Arjevani, Yair Carmon, John C Duchi, Dylan J Foster, Nathan Srebro, and Blake Woodworth. Lower bounds for non-convex stochastic optimization. *Mathematical Programming*, 199(1-2):165–214, 2023.
- [6] Sanjeev Arora and Boaz Barak. *Computational complexity: a modern approach*. Cambridge University Press, 2009.
- [7] Idan Attias, Steve Hanneke, Alkis Kalavasis, Amin Karbasi, and Grigoris Velezgas. Optimal learners for realizable regression: Pac learning and online learning. *arXiv preprint arXiv:2307.03848*, 2023.

- [8] Jean-Yves Audibert. Fast learning rates in statistical inference through aggregation. *The Annals of Statistics*, 37(4):1591–1646, 2009.
- [9] Andrew R Barron. Are bayes rules consistent in information? In *Open problems in communication and computation*, pages 85–91. Springer, 1987.
- [10] Shai Ben-David, Dávid Pál, and Shai Shalev-Shwartz. Agnostic online learning. In *COLT*, volume 3, page 1, 2009.
- [11] Blair Bilodeau, Dylan J Foster, and Daniel Roy. Tight bounds on minimax regret under logarithmic loss via self-concordance. In *International Conference on Machine Learning*, 2020.
- [12] Blair Bilodeau, Dylan J Foster, and Daniel M Roy. Minimax rates for conditional density estimation via empirical entropy. *The Annals of Statistics*, 51(2):762–790, 2023.
- [13] Avrim Blum, Merrick Furst, Jeffrey Jackson, Michael Kearns, Yishay Mansour, and Steven Rudich. Weakly learning dnf and characterizing statistical query learning using fourier analysis. In *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*, pages 253–262, 1994.
- [14] Avrim L Blum. Separating distribution-free and mistake-bound learning models over the boolean domain. *SIAM Journal on Computing*, 23(5):990–1000, 1994.
- [15] Mark Bun. A computational separation between private learning and online learning. *Advances in Neural Information Processing Systems*, 33:20732–20743, 2020.
- [16] Olivier Catoni. The mixture approach to universal model selection. 1997.
- [17] Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.
- [18] Nicolo Cesa-Bianchi, Alex Conconi, and Claudio Gentile. On the generalization ability of on-line learning algorithms. *IEEE Transactions on Information Theory*, 50(9):2050–2057, 2004.
- [19] Constantinos Daskalakis and Vasilis Syrgkanis. Learning in auctions: Regret is hard, envy is easy. In *2016 ieee 57th annual symposium on foundations of computer science (focs)*, pages 219–228. IEEE, 2016.
- [20] Miroslav Dudík, Nika Haghtalab, Haipeng Luo, Robert E Schapire, Vasilis Syrgkanis, and Jennifer Wortman Vaughan. Oracle-efficient online learning and auction design. *Journal of the ACM (JACM)*, 67(5):1–57, 2020.
- [21] Vitaly Feldman. A complete characterization of statistical query learning with applications to evolvability. *Journal of Computer and System Sciences*, 78(5):1444–1459, 2012.
- [22] Vitaly Feldman. A general characterization of the statistical query complexity. In *Conference on Learning Theory*, pages 785–830. PMLR, 2017.
- [23] Vitaly Feldman, Cristobal Guzman, and Santosh Vempala. Statistical query algorithms for mean vector estimation and stochastic convex optimization. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1265–1277. SIAM, 2017.
- [24] Dylan J Foster and Akshay Krishnamurthy. Efficient first-order contextual bandits: Prediction, allocation, and triangular discrimination. *Neural Information Processing Systems (NeurIPS)*, 2021.
- [25] Dylan J Foster and Alexander Rakhlin. Beyond UCB: Optimal and efficient contextual bandits with regression oracles. *International Conference on Machine Learning (ICML)*, 2020.
- [26] Dylan J Foster and Alexander Rakhlin. Foundations of reinforcement learning and interactive decision making. *arXiv preprint arXiv:2312.16730*, 2023.
- [27] Dylan J Foster, Sham M Kakade, Jian Qian, and Alexander Rakhlin. The statistical complexity of interactive decision making. *arXiv preprint arXiv:2112.13487*, 2021.

- [28] Dylan J Foster, Noah Golowich, and Yanjun Han. Tight guarantees for interactive decision making with the decision-estimation coefficient. *Conference on Learning Theory (COLT)*, 2023.
- [29] Dylan J Foster, Noah Golowich, Jian Qian, Alexander Rakhlin, and Ayush Sekhari. Model-free reinforcement learning with the decision-estimation coefficient. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.
- [30] Oded Goldreich et al. Foundations of cryptography—a primer. *Foundations and Trends® in Theoretical Computer Science*, 1(1):1–116, 2005.
- [31] Alon Gonen, Elad Hazan, and Shay Moran. Private learning implies online learning: An efficient reduction. *Advances in Neural Information Processing Systems*, 32, 2019.
- [32] Steve Hanneke. Theory of disagreement-based active learning. *Foundations and Trends® in Machine Learning*, 7(2-3):131–309, 2014.
- [33] Elad Hazan and Tomer Koren. The computational power of optimization in online learning. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 128–141, 2016.
- [34] TS Jayram. Hellinger strikes back: A note on the multi-party information complexity of and. In *International Workshop on Approximation Algorithms for Combinatorial Optimization*, pages 562–573. Springer, 2009.
- [35] Pooria Joulani, Andras Gyorgy, and Csaba Szepesvári. Online learning under delayed feedback. In *International Conference on Machine Learning*, pages 1453–1461. PMLR, 2013.
- [36] Anatoli Juditsky, Philippe Rigollet, and Alexandre B Tsybakov. Learning by mirror averaging. *The Annals of Statistics*, 36(5):2183–2206, 2008.
- [37] Michael Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM (JACM)*, 45(6):983–1006, 1998.
- [38] Michael Kearns, Ming Li, Leonard Pitt, and Leslie Valiant. On the learnability of boolean formulae. In *Proceedings of the nineteenth annual ACM symposium on Theory of computing*, pages 285–295, 1987.
- [39] Akshay Krishnamurthy, Alekh Agarwal, Tzu-Kuo Huang, Hal Daumé III, and John Langford. Active learning for cost-sensitive classification. In *International Conference on Machine Learning*, pages 1915–1924, 2017.
- [40] Erich L Lehmann and George Casella. *Theory of point estimation*. Springer Science & Business Media, 2006.
- [41] Nick Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine learning*, 2(4):285–318, 1988.
- [42] Jon Christian Mesterharm. *Improving on-line learning*. Rutgers The State University of New Jersey, School of Graduate Studies, 2007.
- [43] Arkadii Nemirovski, David Borisovich Yudin, and Edgar Ronald Dawson. Problem complexity and method efficiency in optimization. 1983.
- [44] Binghui Peng and Aviad Rubinstein. Near optimal memory-regret tradeoff for online learning. *arXiv preprint arXiv:2303.01673*, 2023.
- [45] Binghui Peng and Fred Zhang. Online prediction in sub-linear space. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1611–1634. SIAM, 2023.
- [46] Yury Polyanskiy and Yihong Wu. Lecture notes on information theory. 2014.
- [47] Kent Quanrud and Daniel Khashabi. Online learning with adversarial delays. *Advances in neural information processing systems*, 28, 2015.

[48] Maxim Raginsky and Alexander Rakhlin. Information-based complexity, feedback and dynamics in convex programming. *IEEE Transactions on Information Theory*, 57(10):7036–7056, 2011.

[49] Alexander Rakhlin and Karthik Sridharan. Statistical learning and sequential prediction, 2012. Available at http://www.mit.edu/~rakhlin/courses/stat928/stat928_notes.pdf.

[50] Alexander Rakhlin and Karthik Sridharan. Statistical learning theory and sequential prediction. *Lecture Notes in University of Pennsylvania*, 44, 2012.

[51] Alexander Rakhlin and Karthik Sridharan. Online nonparametric regression. In *Conference on Learning Theory*, 2014.

[52] Alexander Rakhlin and Karthik Sridharan. Online nonparametric regression with general loss functions. *arXiv preprint arXiv:1501.06598*, 2015.

[53] Alexander Rakhlin, Karthik Sridharan, and Ambuj Tewari. Online learning: Random averages, combinatorial parameters, and learnability. *Advances in Neural Information Processing Systems* 23, pages 1984–1992, 2010.

[54] Alexander Rakhlin, Karthik Sridharan, and Ambuj Tewari. Online learning via sequential complexities. *Journal of Machine Learning Research*, 2014.

[55] Stephane Ross and J Andrew Bagnell. Reinforcement and imitation learning via interactive no-regret learning. *arXiv preprint arXiv:1406.5979*, 2014.

[56] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.

[57] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[58] David Simchi-Levi and Yunzong Xu. Bypassing the monster: A faster and simpler optimal algorithm for contextual bandits under realizability. *Mathematics of Operations Research*, 2021.

[59] Vaidehi Srinivas, David P Woodruff, Ziyu Xu, and Samson Zhou. Memory bounds for the experts problem. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1158–1171, 2022.

[60] Joseph F Traub, Grzegorz W Wasilkowski, and Henryk Woźniakowski. Information-based complexity. 1988.

[61] Alexandre B Tsybakov. Optimal rates of aggregation. In *Learning Theory and Kernel Machines*, pages 303–313. Springer, 2003.

[62] Alexandre B Tsybakov. *Introduction to Nonparametric Estimation*. Springer Publishing Company, Incorporated, 2008.

[63] Leslie G Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

[64] S. A. van de Geer. *Empirical Processes in M-Estimation*. Cambridge University Press, 2000.

[65] Vladimir Vovk. Competitive on-line linear regression. In *NIPS '97: Proceedings of the 1997 conference on Advances in neural information processing systems 10*, pages 364–370, Cambridge, MA, USA, 1998. MIT Press.

[66] Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press, 2019.

[67] Marcelo J Weinberger and Erik Ordentlich. On delayed prediction of individual sequences. *IEEE Transactions on Information Theory*, 48(7):1959–1976, 2002.

- [68] David P Woodruff, Fred Zhang, and Samson Zhou. Streaming algorithms for learning with experts: Deterministic versus robust. *arXiv preprint arXiv:2303.01709*, 2023.
- [69] Tengyang Xie, Dylan J Foster, Yu Bai, Nan Jiang, and Sham M Kakade. The role of coverage in online reinforcement learning. *International Conference on Learning Representations (ICLR)*, 2023.
- [70] Yunbei Xu and Assaf Zeevi. Upper counterfactual confidence bounds: a new optimism principle for contextual bandits. *arXiv preprint arXiv:2007.07876*, 2020.
- [71] Yuhong Yang and Andrew Barron. Information-theoretic determination of minimax rates of convergence. *Annals of Statistics*, pages 1564–1599, 1999.

Contents of Appendix

I Additional discussion and examples	16
A Additional notation	16
B Additional related work	16
C Examples of estimation problems and loss functions	17
C.1 Examples of offline oracles	18
II Omitted results	20
D General reductions for oracle-efficient online estimation	20
D.1 Characterization of oracle-efficient learnability for classification	22
D.2 Additional lower and upper bounds for memoryless oracle-efficient algorithms . . .	22
E Application to interactive decision making	23
E.1 Offline oracle-efficient algorithms for interactive decision making	23
E.2 Bypassing impossibility of memoryless algorithms via coverability	25
III Proofs	28
F Technical tools	28
G Proofs from Section 3	29
G.1 Proofs from Section 3.1	29
G.2 Proofs from Section 3.2	32
G.3 Proofs from Appendix D	36
H Proofs from Section 4	40
H.1 Proofs from Section 4.1	40
H.2 Proofs from Section 4.2	41
I Proofs from Appendix E	48

Part I

Additional discussion and examples

A Additional notation

We denote $\mathbb{R}_{\geq 0} = [0, \infty)$. For any $a, b \in \mathbb{R}$, $a \wedge b := \min \{a, b\}$ and $a \vee b := \max \{a, b\}$. For any integer $N > 0$, $[N] = \{1, \dots, N\}$. For any set \mathcal{X} , $\Delta(\mathcal{X})$ is the space of all distributions on \mathcal{X} . For any integer T , the notation $x^{1:T}$ will be the shorthand notation for the sequence x^1, \dots, x^T . For any real number $x \in \mathbb{R}$, denote by $\lfloor x \rfloor$ the largest integer that is smaller than or equal to x and by $\lceil x \rceil$ the smallest integer that is greater than or equal to x . The indicator function is denoted by $\mathbb{1}(\cdot)$. We define $O(\cdot)$, $\Omega(\cdot)$, $o(\cdot)$, $\Theta(\cdot)$, $\tilde{O}(\cdot)$, $\tilde{\Omega}(\cdot)$, $\tilde{\Theta}(\cdot)$ following standard non-asymptotic big-oh notation. We use the binary relation $x \lesssim y$ to indicate that $x \leq O(y)$.

B Additional related work

In this section we discuss related work not already covered in detail.

Computational lower bounds for online learning. Beyond Blum [14], another work that considers computational lower bounds for online learning is Hazan and Koren [33]. This work proves lower bounds for online learning in a model where the learner has access to an ERM oracle that can minimize the training loss for an arbitrary dataset $(x^1, y^1), \dots, (x^T, y^T)$. Their lower bound does not fit in our computational model due to details around the way description length is formalized. In particular, the main focus of [33] is to obtain a lower bound on the *number of oracle calls* any online learning algorithm must make to an ERM oracle.

Similar to the setup for Theorem 4.1, Hazan and Koren [33] consider a sequence of classification instances with $\mathcal{X}_n = \{0, 1\}^n$ and classes \mathcal{F}_n of the size of $\Omega(2^{\sqrt{|\mathcal{X}_n|}})$, and show that any online learning algorithm requires $\Omega(\sqrt{|\mathcal{X}_n|})$ oracle calls to achieve low regret for this class. However, the estimators $f \in \mathcal{F}_n$ returned by the oracle in their construction have $\Omega(\sqrt{|\mathcal{X}_n|})$ description length themselves, meaning that they do not satisfy the $\text{poly}(n)$ -description length required by the model described in Section 4.1 (in other words, the result is not meaningful as a lower bound on runtime, because simply reading in the output of the offline oracle takes exponential time). For completeness, we restate the example proposed by Hazan and Koren [33, Theorem 22] in our framework below.

Hard case from [33]. For any integer $n \geq 1$, consider a binary classification problem with $\mathcal{X}_{2n} = \{0, 1\}^{2n}$, $\mathcal{Z} = \mathcal{Y} = \{0, 1\}$, $\mathcal{D} = \mathcal{D}_{0/1}$ and $\mathcal{K}(z) = \mathbb{1}_z$. Let $N := 2^{2n}$, and let \mathcal{S} be the collection of all sets $S = \{s_1, \dots, s_{2^n}\} \subset \{0, 1\}^{2n}$ where $\{0, 1\}^{2n}$ is also treated as the integer set of $\{0, \dots, 2^{2n} - 1\}$ in left-to-right order and $s_i \in \{2^n(i-1), \dots, 2^ni - 1\}$ for each $i = 1, \dots, 2^n$. We define a class $\mathcal{F}_{2n} = \{f_{S,\tau} : S \in \mathcal{S}, \tau \leq 2^n\}$, where

$$f_{S,\tau}(x) = \begin{cases} 0 & \text{if } x \in S \text{ and } x \geq s_\tau, \\ 1 & \text{otherwise.} \end{cases}$$

For this class, we reduce back to the Theorem 22 of Hazan and Koren [33] which states that any algorithm with runtime $o(\sqrt{N})$ has to suffer online estimation error at least $t/2$ for all $1 \leq t \leq 2^n$. The issue with this example for our computation model is that $|\mathcal{F}_{2n}| = \Omega(2^{\sqrt{|\mathcal{X}_{2n}|}})$. Any sufficient description for this parameter space in bit strings (or, e.g., boolean circuits) will scale with $\Omega(\sqrt{|\mathcal{X}_{2n}|})$. Thus, the description length required to return \hat{f}^t is too large (already larger than the lower bound obtained).

Online learning with memory constraints. A number of recent works focus on memory-regret tradeoffs in online learning [59, 45, 44, 1, 68]. Here, the learner can observe the full data stream $(x^1, y^1), \dots, (x^T, y^T)$, but is constrained to B bits of memory. This framework is incomparable to the OOE framework, but it would be interesting to explore whether there are deeper connections (e.g., any memoryless OOE algorithm inherently has memory no larger than that of the offline oracle).

Gaps between offline and online. A long line of work aims to characterize the optimal regret for online learning, developing complexity measures (Littlestone dimension, sequential Rademacher complexity) that parallel classical complexity measures like VC dimension and Rademacher complexity for offline learning and estimation [10, 53, 54, 7]. It is well known that in general, the optimal rates for online learning can be significantly worse than those for offline learning. Our work primarily focuses on finite classes \mathcal{F} , where there is no gap, but for infinite classes, any conversion from offline to online estimation will inevitably lead to a loss in the estimation error rate that scales with appropriate complexity measures for online learning (cf. Appendix D.1).

Other restricted computational models. Our information-theoretic formulation of oracle-efficiency is inspired by statistical query complexity in theoretical computer science and information complexity in optimization, both of which can be viewed as restricted computational models with an information-theoretic flavor. The statistical query model is a framework in which the learner can only access the environment through an oracle that outputs noise estimates (“statistical queries”) for a target parameter of interest [13, 37, 21–23]. Information complexity in optimization is a model in which algorithms can only access the parameter of interest through (potentially noisy) local queries to gradients or other information [43, 60, 48, 2, 5].

C Examples of estimation problems and loss functions

In what follows, we give detailed background on three canonical examples of the general estimation framework discussed in Section 1: Binary classification, square loss regression, and conditional density estimation.

Classification [38, 63]. For binary classification, we take $\mathcal{Z} = \mathcal{Y} = \{0, 1\}$ with the binary loss $D_{0/1}(z_1, z_2) = \mathbb{1}(z_1 \neq z_2)$ for $z_1, z_2 \in \mathcal{Z}$ and kernel $\mathcal{K}(z) = \mathbb{1}_z$, which is noiseless. The binary loss is metric-like with $C_D = 1$.

For offline estimation, observe that with covariates x^1, \dots, x^T and outcomes $y^t = f^*(x^t)$ for all $t \in \{1, \dots, T\}$, any empirical risk minimizer \hat{f} that sets $\hat{f}(x^t) = y^t$ obtains

$$\sum_{t=1}^T D_{0/1}(\hat{f}(x^t), f^*(x^t)) = 0. \quad (5)$$

For online estimation, the halving algorithm [17] achieves

$$\text{Est}_{0/1}^{\text{On}}(T) = \sum_{t=1}^T D_{0/1}(\hat{f}^t(x^t), f^*(x^t)) \leq \log(|\mathcal{F}|). \quad (6)$$

We mention in passing that another natural classification setting we do not explore in detail in this paper is *noisy classification*, where the setting is as above, except that we set $\mathcal{Z} = [0, 1]$, $\mathcal{K}(f^*(x)) = \text{Ber}(f^*(x))$, and take $D_{\text{abs}}(z_1, z_2) = |z_1 - z_2|$ as the absolute loss for all $z_1, z_2 \in \mathcal{Z}$.

Square loss regression [62, 66]. For real-valued regression, we take $\mathcal{Z} = \mathcal{Y} = \mathbb{R}$ with the square loss $D_{\text{sq}}(z_1, z_2) = (z_1 - z_2)^2$ for $z_1, z_2 \in \mathcal{Z}$ and the kernel $\mathcal{K}(f^*(x)) = \mathcal{N}(f^*(x), 1)$ or another subGaussian distribution. Note that the square loss is a metric-like loss with $C_D = 2$.

For offline estimation, with covariates x^1, \dots, x^T and outcomes $y^t \sim f^*(x^t) + \varepsilon^t$ for all $t \in \{1, \dots, T\}$, the classical Empirical Risk Minimization (ERM) $\hat{f} := \arg \min_{f \in \mathcal{F}} \sum_{t=1}^T (f(x^t) - y^t)^2$ gives

$$\text{Est}_{\text{sq}}^{\text{Off}}(T) = \sum_{t=1}^T D_{\text{sq}}(\hat{f}(x^t), f^*(x^t)) \leq \log(|\mathcal{F}| \delta^{-1}), \quad (7)$$

with probability at least $1 - \delta$ (cf. Lemma C.1 below).

For online estimation, the exponential weights algorithm [17], with decision space \mathcal{F} and the loss at each round chosen to be $\ell^t(f) = (f(x^t) - y^t)^2$, achieves

$$\text{Est}_{\text{sq}}^{\text{On}}(T) = \sum_{t=1}^T D_{\text{sq}}(\hat{f}^t(x^t), f^*(x^t)) \leq \log(|\mathcal{F}| \delta^{-1}), \quad (8)$$

with probability at least $1 - \delta$ (cf. Foster and Rakhlin [25] for a proof).

Conditional density estimation [12]. For conditional density estimation, we consider an arbitrary outcome space \mathcal{Y} and take $\mathcal{Z} = \Delta(\mathcal{Y})$ with *squared Hellinger distance* D_H^2 given by¹²

$$D_H^2(\hat{f}(x), f^*(x)) = \frac{1}{2} \int \left(\sqrt{\hat{f}(y \mid x)} - \sqrt{f^*(y \mid x)} \right)^2 dy. \quad (9)$$

and $\mathcal{K}(z) = z$ for all $z \in \mathcal{Z}$. Note that squared Hellinger distance is a metric-like loss with $C_D = 2$.

For offline estimation, with covariates x^1, \dots, x^T and outcomes $y^t \sim f^*(x^t)$ for all $t \in \{1, \dots, T\}$, the classical Maximum Likelihood Estimator (MLE) $\hat{f} := \arg \max_{f \in \mathcal{F}} \sum_{t=1}^T \log f(y^t \mid x^t)$ gives

$$\text{Est}_H^{\text{Off}}(T) = \sum_{t=1}^T D_H^2(\hat{f}(x^t), f^*(x^t)) \leq \log(|\mathcal{F}| \delta^{-1}), \quad (10)$$

with probability at least $1 - \delta$ (cf. Lemma C.2 below).

For online estimation, the exponential weights algorithm [17], with decision space \mathcal{F} and the loss at each round chosen to be $\ell^t(f) = -\log f(y^t \mid x^t)$, achieves

$$\text{Est}_H^{\text{On}}(T) = \sum_{t=1}^T D_H^2(\hat{f}^t(x^t), f^*(x^t)) \leq \log(|\mathcal{F}| \delta^{-1}), \quad (11)$$

with probability at least $1 - \delta$ (cf. Foster et al. [27] for a proof).

C.1 Examples of offline oracles

For completeness, below we prove the offline estimation (fixed design) guarantees for square loss empirical risk minimization and maximum likelihood estimation mentioned in the prequel.

Empirical risk minimization for square loss regression. For any square loss regression instance $(\mathcal{X}, \mathcal{Y}, \mathcal{Z}, \mathcal{K}, \mathcal{F})$ defined as in Appendix C. The Empirical Risk Minimizer (ERM) estimator \hat{f} is defined as

$$\hat{f} = \arg \min_{f \in \mathcal{F}} \sum_{t=1}^T (f(x^t) - y^t)^2.$$

We have the following bounds on the offline estimation error for the squared loss.

Lemma C.1. *For any target parameter $f^* \in \mathcal{F}$, with probability at least $1 - \delta$, we have,*

$$\sum_{t=1}^T (\hat{f}(x^t) - f^*(x^t))^2 \leq 8 \log(|\mathcal{F}|/\delta).$$

Proof of Lemma C.1. Observe that the ERM estimator satisfies

$$\begin{aligned} \sum_{t=1}^T (f^*(x^t) - \hat{f}(x^t))^2 &= \sum_{t=1}^T \left((\hat{f}(x^t) - y^t)^2 - (f^*(x^t) - y^t)^2 + 2(\hat{f}(x^t) - f^*(x^t))(y^t - f^*(x^t)) \right) \\ &\leq 2 \sum_{t=1}^T (\hat{f}(x^t) - f^*(x^t)) \varepsilon^t, \end{aligned} \quad (12)$$

where the last inequality is by the definition of \hat{f} , and $\varepsilon^t := y^t - f^*(x^t)$ is a standard normal distribution for all $t \in [T]$. Then by the Gaussian tail bound, we have with probability at least $1 - \delta$,

$$\sum_{t=1}^T (\hat{f}(x^t) - f^*(x^t)) \varepsilon^t \leq \sqrt{2 \sum_{t=1}^T (\hat{f}(x^t) - f^*(x^t))^2 \log(|\mathcal{F}|/\delta)}.$$

¹²More generally, if ν is a common dominating measure, then $D_H^2(\mathbb{P}, \mathbb{Q}) = \frac{1}{2} \int \left(\sqrt{\frac{d\mathbb{P}}{d\nu}} - \sqrt{\frac{d\mathbb{Q}}{d\nu}} \right)^2 d\nu$, where $\frac{d\mathbb{P}}{d\nu}$ and $\frac{d\mathbb{Q}}{d\nu}$ are Radon-Nikodym derivatives. The notation in Eq. (9) reflects that this quantity is invariant under the choice of ν .

Plug the above inequality back into Eq. (12) and reorganize, we obtain the desired bound of

$$\sum_{t=1}^T (\hat{f}(x^t) - f^*(x^t))^2 \leq 8 \log(|\mathcal{F}|/\delta).$$

□

Maximum likelihood estimation for conditional density estimation. For any conditional density estimation instance $(\mathcal{X}, \mathcal{Y}, \mathcal{Z}, \mathcal{K}, \mathcal{F})$ defined as in Appendix C. The Maximum Likelihood Estimator (MLE) \hat{f} is defined as

$$\hat{f} = \arg \max_{f \in \mathcal{F}} \sum_{t=1}^T \log f(y^t | x^t).$$

We have the following bounds on the offline estimation error for squared Hellinger distance.

Lemma C.2. *For any target parameter $f^* \in \mathcal{F}$, with probability at least $1 - \delta$, we have,*

$$\sum_{t=1}^T D_H^2(\hat{f}(x^t), f^*(x^t)) \leq \log(|\mathcal{F}|/\delta).$$

Proof of Lemma C.2. For any parameter $f \in \mathcal{F}$, define

$$Z_t = -\frac{1}{2}(\log f^*(y^t | x^t) - \log f(y^t | x^t)).$$

Then for any $f \in \mathcal{F}$, by Lemma A.4 of Foster et al. [27], with probability at least $1 - \delta/|\mathcal{F}|$, we have,

$$\sum_{t=1}^T -\frac{1}{2}(\log f^*(y^t | x^t) - \log f(y^t | x^t)) \leq \sum_{t=1}^T \log \left(\mathbb{E} \left[\sqrt{\frac{f(y^t | x^t)}{f^*(y^t | x^t)}} \right] \right) + \log(|\mathcal{F}|/\delta).$$

We further have by the inequality of $\log(1 + x) \leq x$ that

$$\log \left(\mathbb{E} \left[\sqrt{\frac{f(y^t | x^t)}{f^*(y^t | x^t)}} \right] \right) \leq \mathbb{E} \left[\sqrt{\frac{f(y^t | x^t)}{f^*(y^t | x^t)}} - 1 \right].$$

Since $y^t \sim f^*(x^t)$, we have by standard calculus and the definition of the squared Hellinger distance that

$$\mathbb{E} \left[\sqrt{\frac{f(y^t | x^t)}{f^*(y^t | x^t)}} - 1 \right] = -D_H^2(f^*(x^t), f(x^t)).$$

Altogether, we have obtained for any $f \in \mathcal{F}$, with probability at least $1 - \delta/|\mathcal{F}|$

$$\sum_{t=1}^T -\frac{1}{2}(\log f^*(y^t | x^t) - \log f(y^t | x^t)) \leq -\sum_{t=1}^T D_H^2(f^*(x^t), f(x^t)) + \log(|\mathcal{F}|/\delta).$$

Thus, by union bound, the above inequality holds for all $f \in \mathcal{F}$ with probability at least $1 - \delta$. Thus the MLE \hat{f} satisfies

$$\begin{aligned} \sum_{t=1}^T D_H^2(f^*(x^t), \hat{f}(x^t)) &\leq \sum_{t=1}^T \frac{1}{2}(\log f^*(y^t | x^t) - \log \hat{f}(y^t | x^t)) + \log(|\mathcal{F}|/\delta) \\ &\leq \log(|\mathcal{F}|/\delta), \end{aligned}$$

where the second inequality is by the definition of MLE. □

Algorithm 2 Reduction from OOE to Online Learning with Delayed Feedback

- 1: **input:** Offline estimation oracle Alg_{Off} with parameter $\beta_{\text{Off}} \geq 0$, delay parameter $N \in \mathbb{N}$, delayed online learning algorithm \mathcal{A}_{DOL} for class \mathcal{F} .
- 2: **for** $t = 1, 2, \dots, T$ **do**
- 3: Receive $\hat{f}^t = \text{Alg}_{\text{Off}}(x^1, \dots, x^{t-1}, y^1, \dots, y^{t-1})$.
- 4: **if** $t > N$ **then**
- 5: Let $\tilde{f}^{t-N} := \frac{1}{N} \sum_{i=t-N+1}^t \hat{f}^i$.
- 6: Let $\ell^{t-N}(f) := D(\tilde{f}^{t-N}(x^{t-N}), f(x^{t-N}))$ and pass $\ell^{t-N}(\cdot)$ to \mathcal{A}_{DOL} as the delayed feedback.
- 7: Let $\mu^t = \mathcal{A}_{\text{DOL}}^t(\ell^1, \dots, \ell^{t-N})$ be the delayed online learner's prediction distribution.
- 8: Predict with $\hat{f}^t \sim \mu^t$ and receive x^t .

Part II

Omitted results

D General reductions for oracle-efficient online estimation

The oracle-efficient online estimation algorithm in Section 3.1, Algorithm 1, is somewhat specialized to finite classes. In this section, we provide a more general approach to designing oracle-efficient algorithms based on *delayed online learning*, and use it to derive a characterization of oracle-efficient learnability for classification with infinite classes \mathcal{F} .

For the results in this section, we assume that \mathcal{Z} and D are convex, which covers regression and conditional density estimation; variants of our result for classification are given in Appendix G.3.

Online learning with delayed feedback. Before introducing our algorithm, we first introduce an abstract *delayed online learning* framework [67, 42, 35, 47]. In our framework, the learner is given a class $\mathcal{F} \subseteq \mathcal{Z}^{\mathcal{X}}$. Their goal is to choose a sequence of parameters $\bar{f}^1, \dots, \bar{f}^T$ that minimizes regret against the class \mathcal{F} for an adversarially chosen sequence of loss functions ℓ^1, \dots, ℓ^T , with the twist being that the loss ℓ^t is not revealed immediately at step t , and instead becomes available at step $t + N$ for a *delay* parameter $N \in \mathbb{N}$.

In more detail, the interaction between the learner and the environment proceeds as follows:

- For $t = 1, \dots, T$:
- The learner picks $\bar{f}^t \sim \mu^t \in \Delta(\mathcal{Z}^{\mathcal{X}})$.
- Learner incurs loss $\ell^t(\bar{f}^t)$ and adversary reveals loss function $\ell^{t-N} : \mathcal{Z}^{\mathcal{X}} \rightarrow [0, 1]$ for round $t - N$ (if $t < N + 1$, nothing is revealed).

The goal of the learner is to minimize regret in the sense that

$$R_{\text{DOL}}(T, N, \gamma) := \sum_{t=1}^T \mathbb{E}_{\bar{f}^t \sim \mu^t} [\ell^t(\bar{f}^t)] - \gamma \cdot \min_{f \in \mathcal{F}} \sum_{t=1}^T \ell^t(f)$$

is small, where $\gamma \geq 1$ is a parameter. For $\gamma = 1$, this definition coincides with the standard notion of regret in online learning (e.g., Cesa-Bianchi and Lugosi [17]), but allowing for $\gamma > 1$ will prove useful for our technical results.

Algorithm and online estimation error bound. Algorithm 2 describes our reduction from oracle-efficient online estimation to delayed online learning. In addition to an offline oracle Alg_{Off} , the algorithm takes as input a *delay parameter* $N \in \mathbb{N}$ and a delayed online learning algorithm \mathcal{A}_{DOL} for the class \mathcal{F} (Algorithm 2 does not explicitly take the class \mathcal{F} as an argument, as the algorithm only implicitly makes use of \mathcal{F} through \mathcal{A}_{DOL}).

The basic premise behind Algorithm 2 is that for any sequence of consistent offline estimators, we can average to improve the predictions. Consider a step $t \in [T]$. Suppose $\hat{f}^1, \dots, \hat{f}^T$ are produced by an

offline oracle $\mathbf{Alg}_{\text{Off}}$ with parameter $\beta_{\text{Off}} \geq 0$, where we augment the sequence by setting $\hat{f}^{T+s} = \hat{f}^T$ for all $s \in \mathbb{N}$. Then we can use an argument based on convexity (cf. proof of Theorem D.1) to show that for any $N \in \mathbb{N}$, the averaged parameters

$$\tilde{f}^t := \frac{1}{N} \sum_{i=t+1}^{t+N} \hat{f}^i \quad (13)$$

satisfies

$$\begin{aligned} \sum_{t=1}^T D\left(\tilde{f}^t(x^t), f^*(x^t)\right) &\leq N + \frac{1}{N} \sum_{t=1}^{T-N} \sum_{i=t+1}^{t+N} D\left(\hat{f}^i(x^t), f^*(x^t)\right) \\ &= N + \frac{1}{N} \sum_{t=2}^T \sum_{i < t} D\left(\hat{f}^i(x^i), f^*(x^i)\right) \leq N + \beta_{\text{Off}} T/N. \end{aligned}$$

In particular, as we increase N , the quality of the predictions increases, and we achieve sublinear estimation error as soon as $N = \omega(T)$. Of course, the catch here is that \tilde{f}^t depends on the predictions of future estimators, and cannot be computed at step t . However, \tilde{f}^t can be computed at step $t+N+1$, with a delay of N . This leads us to appeal to delayed online learning. In particular, at each step $t \geq N+1$, Algorithm 2 proceeds as follows. Using the new offline estimator \hat{f}^t from $\mathbf{Alg}_{\text{Off}}$, the algorithm computes the averaged estimator $\tilde{f}^{t-N} := \frac{1}{N} \sum_{i=t-N+1}^t \hat{f}^i$ corresponding to the estimator in Eq. (13) for step $t-N$. The algorithm then defines a loss function

$$\ell^{t-N}(f) := D\left(\tilde{f}^{t-N}(x^{t-N}), f(x^{t-N})\right)$$

and feeds it into the delayed online learning algorithm \mathcal{A}_{DOL} as the feedback for step $t-N$. Finally, Algorithm 2 uses the prediction distribution μ^t produced by \mathcal{A}_{DOL} to sample the final estimator f^t , then proceeds to the next step. Our main theorem shows that as long \mathcal{A}_{DOL} achieves low regret for delayed online learning, this strategy leads to low online estimation error.

Theorem D.1 (Reduction from oracle-efficient online estimation to delayed online learning). *Let D be any convex, metric-like loss. Suppose we run Algorithm 2 with delay parameter $N \in \mathbb{N}$ and a delayed online learning algorithm \mathcal{A}_{DOL} for the class \mathcal{F} . Then for all $\gamma \geq 1$, Algorithm 2 ensures that*

$$\mathbb{E}[\text{Est}_D^{\text{On}}(T)] \leq O(C_D \gamma (N + \beta_{\text{Off}} T/N) + R_{\text{DOL}}(T, N, \gamma)), \quad (14)$$

with any offline oracle $\mathbf{Alg}_{\text{Off}}$ with parameter $\beta_{\text{Off}} \geq 0$, where

$$R_{\text{DOL}}(T, N, \gamma) := \sum_{t=1}^T \mathbb{E}_{\bar{f}^t \sim \mu^t} [\ell^t(\bar{f}^t)] - \gamma \cdot \min_{f \in \mathcal{F}} \sum_{t=1}^T \ell^t(f) \quad (15)$$

is the regret of \mathcal{A}_{DOL} for the sequence of losses constructed in Algorithm 2.

The parameter N controls a sort of bias-variance tradeoff in Theorem D.1. The first term in Eq. (14) (corresponding to the bias of the averaged estimators) decreases with the delay N , while the second term (corresponding to the regret of \mathcal{A}_{DOL}) increases; the optimal choice for N will balance these terms. To make this concrete, we revisit finite classes as a warmup.

Example: Finite classes. Delayed online learning is well-studied, and optimal algorithms are known for many classes of interest [67, 42, 35, 47]. The following standard result (a proof is given in Appendix G.3 for completeness) gives a delayed regret bound for arbitrary finite classes.

Lemma D.1. *Consider the delayed online learning setting with a delay parameter N . There exists an algorithm that achieves*

$$R_{\text{DOL}}(T, N, 2) = \sum_{t=1}^T \mathbb{E}_{\bar{f}^t \sim \mu^t} [\ell^t(\bar{f}^t)] - 2 \cdot \min_{f \in \mathcal{F}} \sum_{t=1}^T \ell^t(f) \leq 2N \cdot \log |\mathcal{F}|$$

for any sequences of losses $\ell^1, \dots, \ell^T \in [0, 1]$.

Combining Theorem D.1 with Lemma D.1, we can obtain the following upper bound for oracle-efficient online estimation.

Corollary D.1 (Oracle-efficient online estimation for finite classes via delayed online learning). *Consider an arbitrary instance $(\mathcal{X}, \mathcal{Y}, \mathcal{Z}, \mathcal{K}, \mathcal{F})$ and metric-like loss D , and assume \mathcal{Z} is convex. By choosing \mathcal{A}_{DOL} as in Lemma D.1, Algorithm 2 ensures that for any $N \geq 1$,*

$$\mathbb{E}[\text{Est}_D^{\text{On}}(T)] \leq O(C_D(N + \beta_{\text{Off}}T/N) + N \cdot \log |\mathcal{F}|).$$

with any offline oracle Alg_{Off} with parameter β_{Off} .

By choosing $N = \sqrt{\frac{C_D \beta_{\text{Off}} \cdot T}{C_D + \log |\mathcal{F}|}} \vee 1$, Corollary D.1 gives an upper bound of $O(\sqrt{C_D \beta_{\text{Off}}(C_D + \log |\mathcal{F}|) \cdot T} + \log |\mathcal{F}|)$. While the rate in Corollary D.1, is worse than Theorem 3.1 in terms of dependence on T , the reduction has two advantages: (1) It does not require a-priori knowledge of the offline estimation parameter β_{Off} : If we choose $N = \sqrt{\frac{C_D \cdot T}{C_D + \log |\mathcal{F}|}} \vee 1$, Corollary D.1 obtains $O((\beta_{\text{Off}} + 1)(\sqrt{C_D(C_D + \log |\mathcal{F}|) \cdot T} + C_D + \log |\mathcal{F}|))$; (2) The reduction by Algorithm 2 is more flexible, and allows for guarantees beyond finite classes, as we now illustrate.

D.1 Characterization of oracle-efficient learnability for classification

As an application of Algorithm 2 and Theorem D.1, we give a characterization for oracle-efficient learnability in the OEOE framework. To state the result, we define $\text{Ldim}(\mathcal{F})$ as the Littlestone dimension for a binary function class \mathcal{F} (e.g., Ben-David et al. [10]).

Theorem D.2 (Characterization of oracle-efficient learnability for binary classification). *Consider a binary classification instance $(\mathcal{X}, \mathcal{Y}, \mathcal{Z}, \mathcal{K}, \mathcal{F})$ with $\mathcal{Z} = \mathcal{Y} = \{0, 1\}$, $D = D_{0/1}$ and $\mathcal{K}(z) = \mathbb{1}_z$. For any class \mathcal{F} and $\beta_{\text{Off}} \geq 0$, there exists an oracle-efficient algorithm that achieves online estimation error $O(\sqrt{\beta_{\text{Off}} \text{Ldim}(\mathcal{F}) \cdot T \log T} + \text{Ldim}(\mathcal{F}) \log T)$. On the other hand, in the worst-case any algorithm must suffer at least $\Omega(\text{Ldim}(\mathcal{F}))$ online estimation error.*

The main idea behind this result is to show that we can create a delayed online learner for the reduction in Algorithm 2 that achieves low regret for Littlestone classes.¹³

D.2 Additional lower and upper bounds for memoryless oracle-efficient algorithms

The proof of Theorem 3.3 uses that the estimators \hat{f}^t produced by the offline estimation oracle may be *improper* (i.e., $\hat{f}^t \notin \mathcal{F}$). We next provide a variant of the result that holds even if the estimation oracle is *proper*, under the additional assumptions that (i) the learner is itself proper in the sense that $\mu^t \in \Delta(\mathcal{F})$, and (ii) the learner is *time-invariant* (i.e., the learner sets $\mu^t = F(\hat{f}^t)$ for all t).

Theorem 3.3' (Impossibility of memoryless algorithms for OEOE; proper variant). *Consider the binary classification setting with $\mathcal{Z} = \mathcal{Y} = \{0, 1\}$ and loss $D_{0/1}(\cdot, \cdot)$. For any $N \in \mathbb{N}$ and $\beta_{\text{Off}} \geq 0$, there exists an instance $(\mathcal{X}, \mathcal{Y}, \mathcal{Z}, \mathcal{K}, \mathcal{F})$ with $|\mathcal{F}| = |\mathcal{X}| = N$ such that for any memoryless oracle-efficient algorithm that is (i) **proper**, and (ii) **time-invariant**, there exists a sequence of covariates (x^1, \dots, x^T) and a **proper** offline oracle Alg_{Off} with parameter β_{Off} such that $\mathbb{E}[\text{Est}_D^{\text{On}}(T)] \geq \Omega(\min\{N(\beta_{\text{Off}} + 1), T\})$.*

A complementary upper bound. For completeness, we conclude by showing that the (large) lower bound in Theorem 3.3 can be achieved with a memoryless oracle-efficient algorithm. We consider the “trivial” algorithm that outputs the estimators produced by the offline oracle as-is.

Proposition D.1 (Upper bound for memoryless OEOE). *For any instance $(\mathcal{X}, \mathcal{Y}, \mathcal{Z}, \mathcal{K}, \mathcal{F})$, metric-like loss D , and offline oracle Alg_{Off} with parameter β_{Off} , the algorithm that returns $\bar{f}^t = \hat{f}^t := \text{Alg}_{\text{Off}}^t(x^1, \dots, x^{t-1}, y^1, \dots, y^{t-1})$ has online estimation error $\text{Est}_D^{\text{On}}(T) \leq O((\beta_{\text{Off}} + 1)|\mathcal{X}| \log T)$.*

¹³Formally, to handle the fact that $\mathcal{Z} = \{0, 1\}$ is not convex, this result requires a slight modification to Line 4 in Algorithm 2 that replaces the average with a majority vote. See the proof for details.

E Application to interactive decision making

In this section, we apply our techniques for oracle-efficient online estimation to the *Decision Making with Structured Observations* (DMSO) framework for interactive decision making introduced by [27]. First, in Appendix E.1, we use our reductions to provide offline oracle-efficient algorithms for interactive decision making. Then, in Appendix E.2, we focus on reinforcement learning and show that it is possible to bypass the impossibility results for memoryless oracle-efficient algorithms (Theorem 3.3) for instances $(\mathcal{X}, \mathcal{Y}, \mathcal{Z}, \mathcal{K}, \mathcal{F})$ corresponding to Markov decision processes that satisfy a structural property known as coverability.

E.1 Offline oracle-efficient algorithms for interactive decision making

In this section, we introduce the setting of Decision Making with Structured Observations (DMSO) and the applications of our results to this setting.

Decision Making with Structured Observations (DMSO). The DMSO framework [27] captures a large class of interactive decision making problems (e.g. contextual bandits and reinforcement learning). In this framework, the learner is given access to a *model class* \mathcal{M} that contains an unknown true model $M^* : \Pi \rightarrow \Delta(\mathcal{R} \times \mathcal{O})$, where Π is the *decision space*, $\mathcal{R} \subseteq \mathbb{R}$ is the *reward space* and \mathcal{O} is the *observation space*. Then the interaction between the learner and the environment proceeds in T rounds, where for each round $t = 1, \dots, T$:

1. The learner selects a *decision* $\pi^t \in \Pi$.
2. Nature selects a *reward* $r^t \in \mathcal{R}$ and *observation* $o^t \in \mathcal{O}$ based on the decision, where the pair (r^t, o^t) is drawn independently from the unknown distribution $M^*(\pi^t)$. The reward and observation is then observed by the learner.

Let $g^M(\pi) := \mathbb{E}^{M, \pi}[r]$ denote the mean reward function and $\pi_M := \arg \max_{\pi \in \Pi} g^M(\pi)$ denote the decision with the greatest expected reward for M . The learner's performance is evaluated in terms of *regret* to the optimal decision for M^* :

$$\text{Reg}_{\text{DM}}(T) := \sum_{t=1}^T \mathbb{E}_{\pi^t \sim p^t} [g^{M^*}(\pi_M) - g^{M^*}(\pi^t)], \quad (16)$$

where $p^t \in \Delta(\Pi)$ is the learner's distribution over decisions at round t .

Background: Reducing DMSO to online estimation. Any DMSO class $(\mathcal{M}, \Pi, \mathcal{O})$ induces an instance $(\mathcal{X}, \mathcal{Y}, \mathcal{Z}, \mathcal{K}, \mathcal{F})$ of the estimation framework in Section 1 as follows. We associate $\mathcal{F} = \mathcal{M}$, $\mathcal{X} = \Pi$, $\mathcal{Y} = \mathcal{O} \times \mathcal{R}$, $\mathcal{Z} = \Delta(\mathcal{O} \times \mathcal{R})$, and $\mathcal{K}(M^*(\pi)) = M^*(\pi)$. That is, we have a conditional density estimation problem in which the covariates are decisions $\pi \in \Pi$ and the outcomes are observation-reward pairs drawn from the underlying model $M^*(\pi)$. In particular for a sequence of decisions π^1, \dots, π^T and a sequence of estimators $\widehat{M}^1, \dots, \widehat{M}^T$, we define the online estimation error for a loss D as

$$\text{Est}_D^{\text{On}}(T) = \sum_{t=1}^T D(\widehat{M}^t(\pi^t), M^*(\pi^t)). \quad (17)$$

We refer to any algorithm Alg_{On} that ensures that $\text{Est}_D^{\text{On}}(T) \leq \beta_{\text{On}}$ almost surely given access to $\{(\pi^t, o^t, r^t)\}_{t=1}^T$ with $(o^t, r^t) \sim M^*(\pi^t)$ as an *online estimation oracle* with parameter β_{On} .

Foster et al. [27, 28] give an algorithm, *Estimation-to-Decisions* (E2D), that provides bounds on the regret in Eq. (16) given access to an online estimation oracle Alg_{On} . The algorithm is (online) oracle-efficient and memoryless, in the sense that the decision π^t at each step t is a measurable function of the oracle's output \widehat{M}^t . To restate their result, we define the *Decision-Estimation Coeficient* for the class \mathcal{M} as

$$\text{dec}_\gamma^D(\mathcal{M}, \bar{M}) = \inf_{p \in \Delta(\Pi)} \sup_{M \in \mathcal{M}} \mathbb{E}_{\pi \sim p} \left[g^M(\pi_M) - g^M(\pi) - \gamma \cdot D(M(\pi), \bar{M}(\pi)) \right] \quad (18)$$

for a reference model \bar{M} and any losses D . We further define $\text{dec}_\gamma^D(\mathcal{M}) = \sup_{\bar{M} \in \mathcal{M}} \text{dec}_\gamma^D(\mathcal{M}, \bar{M})$. With this notation, the main regret bound for E2D is as follows.

Algorithm 3 Estimation to Decisions Meta-Algorithm with Offline Oracles (E2D.Off)

- 1: **parameters**:
 - Offline estimation oracle $\mathbf{Alg}_{\text{Off}}$ with parameter β_{Off} .
 - Oracle-efficient online estimation algorithm $\mathbf{Alg}_{\text{OEOE}}$.
 - Exploration parameter $\gamma > 0$.
- 2: **for** $t = 1, 2, \dots, T$ **do**
- 3: Compute estimate $\widehat{M}^t = \mathbf{Alg}_{\text{Off}}^t(\pi^1, \dots, \pi^{t-1}, o^1, \dots, o^{t-1})$.
- 4: Feed \widehat{M}^t to OEOE algorithm $\mathbf{Alg}_{\text{OEOE}}$ and obtain μ^t .
- 5: Let

$$p^t = \arg \min_{p \in \Delta(\Pi)} \sup_{M \in \mathcal{M}} \mathbb{E}_{\pi \sim p, \widehat{M} \sim \mu^t} \left[g^M(\pi_M) - g^M(\pi) - \gamma \cdot D(\widehat{M}(\pi), M(\pi)) \right]. \quad (20)$$
- 6: Sample decision $\pi^t \sim p^t$ and $o^t \sim M^*(\pi^t)$ and feed π^t to OEOE algorithm $\mathbf{Alg}_{\text{OEOE}}$.

Proposition E.1 (Theorem 4.3 of [27]). *For any model class $(\mathcal{M}, \Pi, \mathcal{O})$ and metric-like loss D , any $\gamma > 0$, and any online estimation oracle \mathbf{Alg}_{On} with parameter $\beta_{\text{On}} > 0$, the E2D algorithm ensures that*

$$\mathbf{Reg}_{\text{DM}}(T) \lesssim \sup_{\mu \in \Delta(\mathcal{M})} \text{dec}_\gamma^D(\mathcal{M}, \mu) \cdot T + \gamma \cdot \beta_{\text{On}}. \quad (19)$$

For the case of squared Hellinger distance where $D = D_H^2$, the Decision-Estimation Coefficient $\text{dec}_\gamma^D(\mathcal{M})$ was shown to be a *lower bound* on the minimax optimal regret for any class \mathcal{M} . Hence, Proposition E.1 shows that it is possible to achieve near-optimal regret for any interactive decision making problems whenever an online estimation oracle is available. However, it was unclear whether similar results could be achieved based on offline estimation oracles.

Making E2D offline oracle-efficient. Algorithm 3 (E2D.Off) invokes the E2D algorithm of Foster et al. [27] with any oracle-efficient online estimation algorithm $\mathbf{Alg}_{\text{OEOE}}$ (which can be any of the algorithms we provide, e.g. Theorems 3.1 and 4.2), along with an offline estimation oracle $\mathbf{Alg}_{\text{Off}}$, to provide offline oracle-efficient guarantees for interactive decision making. Invoking the algorithm with Version Space Averaging (via Theorem 3.1) leads to the following corollary.

Corollary E.1. *Consider any DMSO class $(\mathcal{M}, \Pi, \mathcal{O})$ and metric-like loss D . Algorithm 3, with exploration parameter $\gamma > 0$ and $\mathbf{Alg}_{\text{OEOE}}$ chosen to be Algorithm 1, ensures that*

$$\mathbb{E}[\mathbf{Reg}_{\text{DM}}] \leq O(\log T) \cdot \max \left\{ \text{dec}_\gamma^D(\mathcal{M}) \cdot T, \gamma \cdot (\beta_{\text{Off}} + 1) \log |\mathcal{M}| \right\},$$

for any offline estimation oracle $\mathbf{Alg}_{\text{Off}}$ with parameter β_{Off} .

This result shows that information-theoretically, it is possible to achieve low regret in the DMSO framework with offline oracles, though the result is not computationally efficient.

As an example, in the case of square loss regression (Appendix C) which is used for contextual bandits, an offline guarantee of $\beta_{\text{Off}} = O(\log |\mathcal{M}|)$ is achievable. Meanwhile, it is known that $\text{dec}_\gamma^{D_{\text{sq}}}(\mathcal{M}) \lesssim |\mathcal{A}|/\gamma$ [27]. Thus Corollary E.1 achieves a bound of $\tilde{O}(\sqrt{|\mathcal{A}|T} \cdot \log |\mathcal{M}|)$ with an appropriate choice of γ . The best known regret guarantee for contextual bandit is $\tilde{O}(\sqrt{|\mathcal{A}|T \log |\mathcal{M}|})$ [58, 70]. The bound from Corollary E.1 matches the state-of-the-art result up to a factor $\tilde{O}(\sqrt{\log |\mathcal{M}|})$. How to remove this suboptimality is an interesting direction for future work.

Naturally, the other reductions for oracle-efficient online estimation developed in this paper can be combined with Algorithm 3 as well. In particular, by combining with Theorem 4.2 we derive the following corollary for squared Hellinger distance.

Corollary (Informal). *Whenever online conditional density estimation can be performed efficiently with access to the full history, and whenever the minimax problem in Eq. (18) can be solved efficiently, there exists a computationally efficient and offline oracle-efficient algorithm with near-optimal regret in the DMSO framework.*

E.2 Bypassing impossibility of memoryless algorithms via coverability

Recall that our results in Section 3.2 show that in general, it is impossible to obtain low online estimation error through memoryless oracle-efficient algorithms. In this section, we revisit memoryless algorithms for the *Markov decision processes* a particular type of class $(\mathcal{M}, \Pi, \mathcal{O})$ (or equivalently, $(\mathcal{X}, \mathcal{Y}, \mathcal{Z}, \mathcal{K}, \mathcal{F})$). We prove that for any class of Markov decision processes for which a structural parameter called *coverability* [69] is small, any offline estimator can be directly converted into an online estimator.

Markov decision processes. We consider classes $(\mathcal{M}, \Pi, \mathcal{O})$ that correspond to an episodic finite-horizon reinforcement learning setting, following Foster et al. [27]. With $H \in \mathbb{N}$ denoting the horizon, each model $M \in \mathcal{M}$ specifies a non-stationary Markov decision process as a tuple $M = \{\{\mathcal{S}_h\}_{h=1}^H, \mathcal{A}, \{P_h^M\}_{h=1}^H, \{R_h^M\}_{h=1}^H, d_1\}$, where \mathcal{S}_h is the state space for layer h , \mathcal{A} is the action space, $P_h^M : \mathcal{S}_h \times \mathcal{A} \rightarrow \Delta(\mathcal{S}_{h+1})$ is the probability transition kernel for layer h , $R_h^M : \mathcal{S}_h \times \mathcal{A} \rightarrow \Delta(\mathbb{R})$ is the reward distribution for layer h , and $d_1 \in \Delta(\mathcal{S}_1)$ is the initial state distribution. We allow the reward distribution and transition kernel to vary across models in \mathcal{M} and assume that the initial state distribution is fixed.

We set $\Pi \subset \Pi_{\text{RNS}}$, which denotes the set of all randomized, non-stationary policies $\pi = (\pi_1, \dots, \pi_H) \in \Pi_{\text{RNS}}$, where $\pi_h : \mathcal{S}_h \rightarrow \Delta(\mathcal{A})$. For a fixed MDP $M \in \mathcal{M}$ and $\pi \in \Pi$, the observation $o \sim M(\pi)$ is a trajectory $(s_1, a_1, r_1), \dots, (s_H, a_H, r_H)$ that is generated through the following process, beginning from $s_1 \sim d_1$. For $h = 1, \dots, H$:

- $a_h \sim \pi_h(s_h)$.
- $r_h \sim R_h^M(s_h, a_h)$ and $s_{h+1} \sim P_h^M(\cdot | s_h, a_h)$.

So the obseravtion space $\mathcal{O} = \mathcal{S}_1 \times \mathcal{A} \times \mathbb{R} \times \dots \times \mathcal{S}_H \times \mathcal{A} \times \mathbb{R}$. For notational convenience, we take s_{H+1} to be a deterministic terminal state. We use $\mathbb{P}^{M, \pi}$ and $\mathbb{E}^{M, \pi}[\cdot]$ to denote the probability law and expectation over trajectories induced by $M(\pi)$. In addition, we define $\bar{P}_h^M(\cdot | s_h, a_h)$ as the conditional distribution on s_{h+1}, r_h given s_h, a_h under M for $h \in [H]$.

The guarantees we provide apply to any loss that has a particular *layer-wise* structure tailored to reinforcement learning.

Definition E.1 (Layer-wise loss). *For any sequence of losses $\{D_h\}_{h \in [H]}$ bounded by $[0, 1]$, where $D_h : \Delta(\mathcal{S}_h \times \mathbb{R}) \times \Delta(\mathcal{S}_h \times \mathbb{R}) \rightarrow [0, 1]$ for all $h \in [H]$, we define the layer-wise loss D^{RL} on $\Delta(\mathcal{O})$ as¹⁴*

$$D^{\text{RL}}(M(\pi) \| M'(\pi)) = \sum_{h=1}^H \mathbb{E}^{M', \pi} \left[D_h \left(\bar{P}^{M'}(\cdot | s_h, a_h) \| \bar{P}^M(\cdot | s_h, a_h) \right) \right],$$

for any pair of MDPs $M, M' \in \mathcal{M}$ and policy $\pi \in \Pi$.¹⁵

Examples of the layer-wise loss are scaled reverse KL-divergence (which is bounded by $[0, 1]$ whenever the density ratios under consideration are upper and lower bounded with an appropriate scaling) [27] and the squared Bellman error [29]. Another useful example is the the sum of layer-wise squared Hellinger distances given by

$$D_H^{\text{RL}}(M(\pi) \| M'(\pi)) = \sum_{h=1}^H \mathbb{E}^{M', \pi} \left[D_H^2 \left(\bar{P}^M(\cdot | s_h, a_h), \bar{P}^{M'}(\cdot | s_h, a_h) \right) \right]. \quad (21)$$

This loss coincides with the global squared Hellinger distance $D_H^2(M(\pi), M'(\pi))$ up to an $O(H)$ factor.

¹⁴The ordering of M and M' on the right-hand side of the definition is due to the following technical reason: Theorem E.1 only works when the expectation on the right-hand side to be taken with respect to $M^*(\pi)$, which shows up as the second argument in the offline oracle guarantee (??).

¹⁵For the results in this section, it will be useful to work with asymmetric losses, and in this case we use the notation $D(\cdot \| \cdot)$ instead of $D(\cdot, \cdot)$.

Coverability. We provide memoryless oracle-efficient algorithms for online estimation for any layer-wise loss D^{RL} when the underlying MDP M^* has bounded *coverability* [69].

Definition E.2 (Coverability). *For an MDP M^* and a policy π , we define $d_h^\pi(s, a) \equiv \mathbb{E}^{M^*, \pi}[\mathbb{1}(s_h, a_h = s, a)]$. The coverability coefficient C_{cov} for a policy class Π for the MDP M^* is given by*

$$C_{\text{cov}}(M^*) := \inf_{\nu_1, \dots, \nu_H \in \Delta(\mathcal{S} \times \mathcal{A})} \sup_{\pi \in \Pi, h \in [H]} \left\| \frac{d_h^\pi}{\nu_h} \right\|_\infty.$$

It is immediate to see that $C_{\text{cov}} \leq |\Pi|$, but in general it can be much smaller. Examples of MDP classes with low coverability include Block MDPs, Low-Rank MDPs, and exogenous block MDPs [69, 4].

Offline-to-online conversion under coverability. Our main result shows that under coverability, the outputs of any offline estimation oracle Alg_{Off} satisfy an online estimation guarantee *as-is*.

Theorem E.1 (Offline-to-online conversion under coverability). *For any layer-wise loss D^{RL} and MDP class $(\mathcal{M}, \Pi, \mathcal{O})$ and $M^* \in \mathcal{M}$, the sequence of estimators $(\widehat{M}^1, \dots, \widehat{M}^T)$ produced by any offline estimation oracle Alg_{Off} for D^{RL} with parameter β_{Off} satisfy*

$$\sum_{t=1}^T D^{\text{RL}}\left(\widehat{M}^t(\pi^t) \| M^*(\pi^t)\right) \leq O\left(\sqrt{H C_{\text{cov}}(M^*) \beta_{\text{Off}} T \log T} + H C_{\text{cov}}(M^*)\right).$$

This result is based on a variant of the proof technique in Theorem 1 of Xie et al. [69]. An important application of the result, which can be applied in tandem with the guarantees in Foster et al. [27], concerns squared Hellinger distance.

Corollary E.2. *For any MDP class $(\mathcal{M}, \Pi, \mathcal{O})$ and $M^* \in \mathcal{M}$, the sequence of estimators $(\widehat{M}^1, \dots, \widehat{M}^T)$ produced by any offline estimation oracle Alg_{Off} for squared Hellinger distance D_H^2 with parameter β_{Off} satisfy*

$$\text{Est}_H^{\text{On}}(T) = \sum_{t=1}^T D_H^2\left(\widehat{M}^t(\pi^t), M^*(\pi^t)\right) \leq O\left(H \sqrt{C_{\text{cov}}(M^*) \beta_{\text{Off}} T \log T} + H^2 C_{\text{cov}}(M^*)\right)$$

This result follows by using that the layer-wise squared Hellinger distance in Eq. (21) is equivalent to $D_H^2(M(\pi), M'(\pi))$ up to $O(H)$ factors.

Application to interactive decision making. We apply Theorem E.1 to decision making via Algorithm 3.

Corollary E.3. *Consider any layer-wise loss $D = D^{\text{RL}}$ and MDP class $(\mathcal{M}, \Pi, \mathcal{O})$, and let $C_{\text{cov}} := \sup_{M \in \mathcal{M}} C_{\text{cov}}(M)$. Algorithm 3 with exploration parameter $\gamma > 0$ and $\text{Alg}_{\text{OE}}^{\text{OE}}$ chosen to be the identity map ensures that*

$$\mathbb{E}[\text{Reg}_{\text{DM}}] \leq O(\log T) \cdot \max \left\{ \sup_{\mu \in \Delta(\mathcal{M})} \text{dec}_\gamma^D(\mathcal{M}, \mu) \cdot T, \gamma \cdot \left(\sqrt{H C_{\text{cov}} \beta_{\text{Off}} T \log T} + H C_{\text{cov}} \right) \right\},$$

for any offline estimation oracle Alg_{Off} with parameter β_{Off} .

Contextual bandits and optimality of offline-to-online conversion. Another implication for Theorem E.1 concerns the special case of contextual bandits (that is, MDPs with horizon one). For the contextual bandit setting we abbreviate $\mathcal{S} = \mathcal{S}_1$, and refer to $d_1 \in \Delta(\mathcal{S})$ as the *context distribution*. We define $g^M(s, a) = \mathbb{E}_{r \sim R_1^M(s, a)}[r]$ as the expected reward function under a model M , and following Foster and Rakhlin [25], use the squared error between mean reward functions as our divergence:

$$D_{\text{CB}}(M(\pi), M'(\pi)) := \mathbb{E}_{s \sim d_1, a \sim \pi(s)} [D_{\text{sq}}(g^M(s, a), g^{M'}(s, a))]. \quad (22)$$

For this setting, the coverability coefficient C_{cov} is always bounded by the number of actions $|\mathcal{A}|$, which leads to the following corollary.

Corollary E.4. *For any contextual bandit class $(\mathcal{M}, \Pi, \mathcal{O})$ and $M^* \in \mathcal{M}$, the sequence of estimators $(\widehat{M}^1, \dots, \widehat{M}^T)$ produced by any offline estimation oracle $\mathbf{Alg}_{\text{Off}}$ for D_{CB} with parameter β_{Off} satisfy*

$$\mathbf{Est}_{\text{D}}^{\text{On}}(T) = \sum_{t=1}^T D_{\text{CB}}\left(\widehat{M}^t(\pi^t), M^*(\pi^t)\right) \leq O\left(\sqrt{|\mathcal{A}|T\beta_{\text{Off}} \log T} + |\mathcal{A}|\right),$$

Recall that Foster and Rakhlin [25] show that any algorithm for online estimation with the divergence in Eq. (22) with $\mathbf{Est}_{\text{D}}^{\text{On}}(T) \leq \beta_{\text{On}}$ can be lifted to a contextual bandit algorithm with regret $O(\sqrt{|\mathcal{A}|T \cdot \beta_{\text{On}}})$ via the *inverse gap weighting strategy*, even if contexts are chosen adversarially. Subsequent work of Simchi-Levi and Xu [58] shows that for stochastic contexts, the inverse gap weighting strategy also yields regret $O(\sqrt{|\mathcal{A}|T \cdot \beta_{\text{Off}}})$ given access to an offline oracle with parameter β_{Off} . On the other hand, combining Corollary E.4 with the guarantee from Foster and Rakhlin [25] gives regret $O(|\mathcal{A}|^{1/4}T^{3/4}\beta_{\text{Off}}^{1/4})$. This does not recover the result from Simchi-Levi and Xu [58], but nonetheless gives an alternative proof that sublinear offline estimation error suffices for sublinear regret.

The guarantee in Theorem E.1 leads to a degradation in rate from β_{Off} to $\sqrt{T\beta_{\text{Off}}}$ (suppressing problem-dependent parameters). Our next result shows that this is tight in general.

Proposition E.2 (Tightness of offline-to-online conversion). *For any integer $T \geq 1$ and $\beta_{\text{Off}} > 0$, there exists a contextual bandit class $(\mathcal{M}, \Pi = \mathcal{A}^{\mathcal{S}}, \mathcal{O})$ with $|\mathcal{A}| = 2$, a distribution $d_1 \in \Delta(\mathcal{S})$, a sequence (π^1, \dots, π^T) and an offline oracle $\mathbf{Alg}_{\text{Off}}$ for D_{CB} with parameter β_{Off} such that the oracle's outputs $(\widehat{M}^1, \dots, \widehat{M}^T)$ satisfy*

$$\mathbf{Est}_{\text{D}}^{\text{On}}(T) = \sum_{t=1}^T D_{\text{CB}}\left(\widehat{M}^t(\pi^t), M^*(\pi^t)\right) \geq \Omega\left(\sqrt{T\beta_{\text{Off}}}\right).$$

Part III

Proofs

F Technical tools

Lemma F.1. *For any non-increasing sequence $x_1 \geq x_2 \geq \dots \geq x_{T+1} \geq 1$,*

$$\sum_{t=1}^T \frac{x_t - x_{t+1}}{x_t} \leq \log(x_1).$$

Proof of Lemma F.1. Since $\log(1+a) \leq a$ for all $a > -1$, for any $t \in [T]$, we have

$$-\log(x_t/x_{t+1}) = \log\left(1 + \left(\frac{x_{t+1}}{x_t} - 1\right)\right) \leq \left(\frac{x_{t+1}}{x_t} - 1\right).$$

Summing up over $t \in [T]$, we obtain

$$\sum_{t=1}^T \frac{x_t - x_{t+1}}{x_t} = \sum_{t=1}^T 1 - \frac{x_{t+1}}{x_t} \leq \sum_{t=1}^T \log(x_t/x_{t+1}) = \log(x_1/x_{T+1}) \leq \log(x_1).$$

□

The following lemma gives an improvement to Lemma A.13 of Foster et al. [27] that removes a logarithmic factor. This shows that up to an absolute constant, squared Hellinger distance obeys a one-sided version of the chain rule for KL divergence.

Lemma F.2 (Subadditivity for squared Hellinger distance). *Let $(\mathcal{X}_1, \mathcal{F}_1), \dots, (\mathcal{X}_n, \mathcal{F}_n)$ be a sequence of measurable spaces, and let $\mathcal{X}^i = \prod_{t=1}^i \mathcal{X}_t$ and $\mathcal{F}^i = \bigotimes_{t=1}^i \mathcal{F}_t$. For each i , let $\mathbb{P}^i(\cdot | \cdot)$ and $\mathbb{Q}^i(\cdot | \cdot)$ be probability kernels from $(\mathcal{X}^{i-1}, \mathcal{F}^{i-1})$ to $(\mathcal{X}_i, \mathcal{F}_i)$. Let \mathbb{P} and \mathbb{Q} be the laws of X_1, \dots, X_n under $X_i \sim \mathbb{P}^i(\cdot | X_{1:i-1})$ and $X_i \sim \mathbb{Q}^i(\cdot | X_{1:i-1})$ respectively. Then it holds that*

$$\mathbb{D}_H^2(\mathbb{P}, \mathbb{Q}) \leq 7 \cdot \mathbb{E}_{\mathbb{P}} \left[\sum_{i=1}^n \mathbb{D}_H^2(\mathbb{P}^i(\cdot | X_{1:i-1}), \mathbb{Q}^i(\cdot | X_{1:i-1})) \right].$$

Proof of Lemma F.2. We appeal to the *cut-and-paste property* of [34], defining a collection of distributions indexed by a hypercube $\{0, 1\}^n$ with the property that the vertices $(0, \dots, 0)$ and $(1, \dots, 1)$ correspond to the distribution \mathbb{P} and \mathbb{Q} . Concretely, for any vertex $v \in \{0, 1\}^n$ of the hypercube, we define a probability distribution

$$\mathfrak{P}_v := \prod_{i \in [n]} R_{v_i}(\cdot | X_{1:i-1}), \quad \text{where } R_{v_i}(\cdot | X_{1:i-1}) = \begin{cases} \mathbb{P}^i(\cdot | X_{1:i-1}) & \text{if } v_i = 0, \\ \mathbb{Q}^i(\cdot | X_{1:i-1}) & \text{if } v_i = 1. \end{cases}$$

Observe that $\mathfrak{P}_{(0, \dots, 0)} = \mathbb{P}$ and $\mathfrak{P}_{(1, \dots, 1)} = \mathbb{Q}$. Now, consider any four vertices $a, b, c, d \in \{0, 1\}^n$ with the property that $\{a_i, b_i\} = \{c_i, d_i\}$ for each $i \in [n]$ (with $\{\cdot\}$ interpreted as a multi-set). Then for any measure $\nu := \prod_{i=1}^n \nu_i(\cdot | X_{1:i-1})$, where $\nu_i(\cdot | X_{1:i-1})$ is any common dominating conditional measure¹⁶ for $\mathbb{P}^i(\cdot | X_{1:i-1})$ and $\mathbb{Q}^i(\cdot | X_{1:i-1})$, by the definition of squared Hellinger distance we have

$$\begin{aligned} \mathbb{D}_H^2(\mathfrak{P}_a, \mathfrak{P}_b) &= 1 - \int \sqrt{\prod_{i=1}^n \frac{dR_{a_i}(\cdot | X_{1:i-1})}{d\nu_i} \frac{dR_{b_i}(\cdot | X_{1:i-1})}{d\nu_i}} d\nu \\ &= 1 - \int \sqrt{\prod_{i=1}^n \frac{dR_{c_i}(\cdot | X_{1:i-1})}{d\nu_i} \frac{dR_{d_i}(\cdot | X_{1:i-1})}{d\nu_i}} d\nu \\ &= \mathbb{D}_H^2(\mathfrak{P}_c, \mathfrak{P}_d). \end{aligned} \tag{23}$$

¹⁶For example, we can take $\nu_i(\cdot | X_{1:i}) = (\mathbb{P}^i(\cdot | X_{1:i-1}) + \mathbb{Q}^i(\cdot | X_{1:i-1}))/2$. for \mathbb{P}^i and \mathbb{Q}^i . The result is independent of the choice of ν .

Let k be the maximum integer such that $2^k \leq n$. Then since Eq. (23) holds, by Theorem 7 of Jayram [34] applied with the pairwise disjoint collection $A_j = \{i \leq n \mid i \bmod 2^k = j\}$ for all $j \in [2^k]$ where $|A_j| \leq 2$ for all j , we have

$$\begin{aligned} D_H^2(\mathbb{P}, \mathbb{Q}) \cdot \prod_{i=1}^n (1 - 1/2^i) &\leq \sum_{j=1}^{2^k} D_H^2 \left(\mathbb{P}, \prod_{l \in A_j} \mathbb{Q}^l(\cdot \mid X_{1:l-1}) \prod_{l' \notin A_j} \mathbb{P}^{l'}(\cdot \mid X_{1:l'-1}) \right) \\ &\leq 2 \mathbb{E}_{\mathbb{P}} \left[\sum_{i=1}^n D_H^2(\mathbb{P}^i(\cdot \mid X_{1:i-1}), \mathbb{Q}^i(\cdot \mid X_{1:i-1})) \right]. \end{aligned}$$

To conclude, we note that $\prod_{i=1}^n (1 - 1/2^i) > 2/7$.

□

G Proofs from Section 3

G.1 Proofs from Section 3.1

Theorem 3.1 (Main upper bound for OEOE). *For any instance $(\mathcal{X}, \mathcal{Y}, \mathcal{Z}, \mathcal{K}, \mathcal{F})$, any metric-like loss D , and any offline estimator Alg_{Off} with parameter $\beta_{\text{Off}} \geq 0$, Algorithm 1 is oracle-efficient and achieves*

$$\text{Est}_D^{\text{On}}(T) \leq O(C_D \cdot (\beta_{\text{Off}} + 1) \cdot \min \{\log |\mathcal{F}|, |\mathcal{X}| \log T\}).$$

Proof of Theorem 3.1. Our main technical result is the following lemma, which is proven in the sequel.

Lemma G.1. *Consider any instance $(\mathcal{X}, \mathcal{Y}, \mathcal{Z}, \mathcal{K}, \mathcal{F})$ and a metric-like loss¹⁷ D on \mathcal{Z} . Let $f^* \in \mathcal{F}$ be the target parameter, and consider a sequence of sets $\mathcal{F} = \mathcal{F}_1 \supseteq \mathcal{F}_2 \supseteq \dots \supseteq \mathcal{F}_T \supseteq \{f^*\}$ and sequence of covariates $x^1, \dots, x^T \in \mathcal{X}$ with the property that for all $t \in [T]$, all $f \in \mathcal{F}_t$ satisfy the following offline estimation guarantee:*

$$\sum_{s=1}^{t-1} D(f(x^s), f^*(x^s)) \leq \beta_{\text{Off}}. \quad (24)$$

Then, by defining $\mu^t = \text{Unif}(\mathcal{F}_t)$, we have that

$$\sum_{t=1}^T \mathbb{E}_{f \sim \mu^t} [D(f(x^t), f^*(x^t))] \leq O((\beta_{\text{Off}} + 1) \cdot \min \{\log |\mathcal{F}|, |\mathcal{X}| \log T\}).$$

To invoke Lemma G.1, we observe that the version space construction in Algorithm 1 ensures that for all $t \in [T]$, all $f \in \mathcal{F}_t$ satisfy

$$\sum_{s=1}^{t-1} D(f(x^s), f^*(x^s)) \leq \sum_{s=1}^{t-1} C_D \left(D(f(x^s), \hat{f}^t(x^s)) \right) + D(\hat{f}^t(x^s), f^*(x^s)) \leq 2C_D \beta_{\text{Off}}.$$

In addition, it is immediate to see that $\mathcal{F} = \mathcal{F}_1 \supseteq \mathcal{F}_2 \supseteq \dots \supseteq \mathcal{F}_T$. Thus, by invoking Lemma G.1 with parameter $\beta_{\text{Off}}' = 2C_D \beta_{\text{Off}}$, we have that

$$\text{Est}_D^{\text{On}}(T) = \sum_{t=1}^T \mathbb{E}_{f \sim \mu^t} [D(f(x^t), f^*(x^t))] \leq O((C_D \beta_{\text{Off}} + 1) \cdot \min \{\log |\mathcal{F}|, |\mathcal{X}| \log T\}).$$

To simplify, we note that $(C_D \beta_{\text{Off}} + 1) \leq C_D (\beta_{\text{Off}} + 1)$, since $C_D \geq 1$.

□

¹⁷For this lemma, D need not be a metric-like loss; it suffices that D is bounded and has $D(z, z) = 0$ for all $z \in \mathcal{Z}$.

Proof of Lemma G.1. We begin by proving that $\text{Est}_D^{On}(T) \leq O(C_D \cdot (\beta_{\text{Off}} + 1) \cdot \log |\mathcal{F}|)$. Let us adopt the convention that $\mathcal{F}_{T+1} = \{f^*\}$, so that $\mathcal{F}_{T+1} \subseteq \mathcal{F}_T \subseteq \dots \subseteq \mathcal{F}_1$. For any parameter $f \in \mathcal{F} \setminus \{f^*\}$, define $t_f := \min \{t : f \notin \mathcal{F}_{t+1}\}$. It is immediate to see that for all $f \in \mathcal{F} \setminus \{f^*\}$, $1 \leq t_f \leq T$ and for all $t \in [T]$, $|\{f : t_f = t\}| = |\mathcal{F}_t \setminus \mathcal{F}_{t+1}|$. Using that $D(f^*(x), f^*(x)) = 0$ for all $x \in \mathcal{X}$, we re-write the online estimation error as

$$\begin{aligned} \sum_{t=1}^T \mathbb{E}_{f \sim \mu^t} [D(f(x^t), f^*(x^t))] &= \sum_{t=1}^T \frac{1}{|\mathcal{F}_t|} \sum_{f \in \mathcal{F}_t} D(f(x^t), f^*(x^t)) = \sum_{t=1}^T \frac{1}{|\mathcal{F}_t|} \sum_{f \in \mathcal{F}_t, f \neq f^*} D(f(x^t), f^*(x^t)) \\ &= \sum_{t=1}^T \sum_{f: t_f=t} \sum_{s \leq t} \frac{1}{|\mathcal{F}_s|} D(f(x^s), f^*(x^s)) \end{aligned}$$

Using that $\mathcal{F}_T \subseteq \mathcal{F}_{T-1} \subseteq \dots \subseteq \mathcal{F}_1$, we can upper bound this quantity by

$$\sum_{t=1}^T \sum_{f: t_f=t} \sum_{s \leq t} \frac{1}{|\mathcal{F}_s|} D(f(x^s), f^*(x^s)) \leq \sum_{t=1}^T \sum_{f: t_f=t} \frac{1}{|\mathcal{F}_t|} \sum_{s \leq t} D(f(x^s), f^*(x^s)).$$

To proceed, observe that for any function $f \in \mathcal{F}$, if $t_f = t$, then $f \in \mathcal{F}_t$. It follows from the assumed bound in Eq. (24) that if $t_f = t$, then

$$\begin{aligned} \sum_{s \leq t} D(f(x^s), f^*(x^s)) &= D(f(x^t), f^*(x^t)) + \sum_{s \leq t-1} D(f(x^s), f^*(x^s)) \\ &\leq 1 + \beta_{\text{Off}}, \end{aligned}$$

where we have used the fact that the loss D is bounded by 1. Using this fact, and recalling that for all $t \in [T]$, $|\{f : t_f = t\}| = |\mathcal{F}_t \setminus \mathcal{F}_{t+1}|$, we bound

$$\sum_{t=1}^T \sum_{f: t_f=t} \frac{1}{|\mathcal{F}_t|} \sum_{s \leq t} D(f(x^s), f^*(x^s)) \leq (\beta_{\text{Off}} + 1) \sum_{t=1}^T \sum_{f: t_f=t} \frac{1}{|\mathcal{F}_t|} \leq (\beta_{\text{Off}} + 1) \sum_{t=1}^T \frac{|\mathcal{F}_t \setminus \mathcal{F}_{t+1}|}{|\mathcal{F}_t|}.$$

Finally, by Lemma F.1, we have that

$$\sum_{t=1}^T \frac{|\mathcal{F}_t \setminus \mathcal{F}_{t+1}|}{|\mathcal{F}_t|} = \sum_{t=1}^T \frac{|\mathcal{F}_t| - |\mathcal{F}_{t+1}|}{|\mathcal{F}_t|} \leq \log |\mathcal{F}_1| = \log |\mathcal{F}|.$$

We conclude that

$$\sum_{t=1}^T \mathbb{E}_{f \sim \mu^t} [D(f(x^t), f^*(x^t))] \leq (\beta_{\text{Off}} + 1) \log |\mathcal{F}|.$$

We now prove the bound $\text{Est}_D^{On}(T) \leq O(C_D \cdot (\beta_{\text{Off}} + 1) \cdot |\mathcal{X}| \log T)$. For each $x \in \mathcal{X}$, define

$$N_{t-1}(x) = \sum_{s=1}^{t-1} \mathbb{1}(x^s = x). \text{ Then we can write the online estimation error as}$$

$$\sum_{t=1}^T \mathbb{E}_{f \sim \mu^t} [D(f(x^t), f^*(x^t))] = \sum_{t=1}^T \sum_{x \in \mathcal{X}} \frac{\mathbb{1}(x = x^t)}{N_{t-1}(x) \vee 1} \cdot \mathbb{E}_{f \sim \mu^t} [(N_{t-1}(x) \vee 1) D(f(x), f^*(x))].$$

From the definition of μ^t , we have that

$$\begin{aligned} \mathbb{E}_{f \sim \mu^t} [(N_{t-1}(x) \vee 1) D(f(x), f^*(x))] &= \frac{1}{|\mathcal{F}_t|} \sum_{f \in \mathcal{F}_t} (N_{t-1}(x) \vee 1) D(f(x), f^*(x)) \\ &\leq \frac{1}{|\mathcal{F}_t|} \sum_{f \in \mathcal{F}_t} (N_{t-1}(x) + 1) D(f(x), f^*(x)). \end{aligned}$$

Now, from the offline guarantee assumed in Eq. (24), we have for all $x \in \mathcal{X}$ and $f \in \mathcal{F}_t$,

$$(N_{t-1}(x) + 1) D(f(x), f^*(x)) \leq \sum_{s \leq t-1} D(f(x^s), f^*(x^s)) + D(f(x), f^*(x)) \leq \beta_{\text{Off}} + 1.$$

Combining these observations, we have

$$\sum_{t=1}^T \mathbb{E}_{f \sim \mu^t} [\mathsf{D}(f(x^t), f^*(x^t))] \leq (\beta_{\text{Off}} + 1) \sum_{t=1}^T \sum_{x \in \mathcal{X}} \frac{\mathbb{1}(x = x^t)}{N_{t-1}(x) \vee 1}.$$

Now, for each $x \in \mathcal{X}$, define $t_x = \min \{t \leq T : x^t = x\}$ if this set is not empty, and set $t_x = T$ otherwise. From this definition and the fact that $1 + 1/2 + \dots + 1/T \leq 1 + \log T$, we have that

$$\begin{aligned} \sum_{x \in \mathcal{X}} \sum_{t=1}^T \frac{\mathbb{1}(x = x^t)}{N_{t-1}(x) \vee 1} &= \sum_{x \in \mathcal{X}} \left(\sum_{t=1}^{t_x} \frac{\mathbb{1}(x = x^t)}{N_{t-1}(x) \vee 1} + \sum_{t=t_x+1}^T \frac{\mathbb{1}(x = x^t)}{N_{t-1}(x) \vee 1} \right) \\ &\leq \sum_{x \in \mathcal{X}} \left(1 + \sum_{i=1}^{N_{T-1}(x)} \frac{1}{i} \right) \\ &\leq 2|\mathcal{X}| + |\mathcal{X}| \log T \leq 3|\mathcal{X}| \log T. \end{aligned}$$

We conclude that

$$\sum_{t=1}^T \mathbb{E}_{f \sim \mu^t} [\mathsf{D}(f(x^t), f^*(x^t))] \leq 3(\beta_{\text{Off}} + 1) \cdot |\mathcal{X}| \log T.$$

□

Theorem 3.2 (Main lower bound for OEOE). *Consider the binary classification setting with $\mathcal{Z} = \mathcal{Y} = \{0, 1\}$ and loss $\mathsf{D}_{0/1}(\cdot, \cdot)$. For any $N \in \mathbb{N}$ and $\beta_{\text{Off}} > 0$, there exists an instance $(\mathcal{X}, \mathcal{Y}, \mathcal{Z}, \mathcal{K}, \mathcal{F})$ with $\log |\mathcal{F}| = |\mathcal{X}| = N$ such that for any oracle-efficient algorithm, there is a sequence of covariates (x^1, \dots, x^T) and offline oracle with parameter β_{Off} such that $\mathbb{E}[\text{Est}_D^{\text{On}}(T)] \geq \Omega(\min \{(\beta_{\text{Off}} + 1)N, T\})$.*

Proof of Theorem 3.2. Let $N \geq 1$ be given, and consider the model class where $\mathcal{X} = \{x_1, \dots, x_N\}$ is an arbitrary discrete set, $\mathcal{Z} = \mathcal{Y} = \{0, 1\}$, $\mathcal{F} = \{0, 1\}^{\mathcal{X}}$, $\mathsf{D}(z_1, z_2) = \mathsf{D}_{0/1}(z_1, z_2) = \mathbb{1}\{z_1 \neq z_2\}$, and $\mathcal{K}(z) = \mathbb{1}_z$.

We first specify the offline estimation oracle, then specify an adversarially chosen covariate sequence. Fix $T \in \mathbb{N}$, and for any $1 \leq t \leq T$ and sequence of covariates x^1, \dots, x^t define $N_t(x) := \sum_{s=1}^t \mathbb{1}(x^s = x)$. For any target parameter $f^* \in \mathcal{F}$ and offline estimation parameter $\beta_{\text{Off}} > 0$, we consider the oracle $\text{Alg}_{\text{Off}}^t(\cdot; f^*)$ for the sequence x^1, \dots, x^t that returns

$$\widehat{f}^t(x) = \begin{cases} 0 & \text{if } N_{t-1}(x) < \beta_{\text{Off}}, \\ f^*(x) & \text{otherwise.} \end{cases}$$

To complete the construction, we consider sequence (x^1, \dots, x^T) in which

$$x^t = x_{\min \{\lceil t/\lceil \beta_{\text{Off}} \rceil \rceil, N\}}.$$

Equivalently, and perhaps more intuitively, we set

$$(x^1, \dots, x^T) = (\underbrace{x_1, \dots, x_1}_{\lceil \beta_{\text{Off}} \rceil}, \underbrace{x_2, \dots, x_2}_{\lceil \beta_{\text{Off}} \rceil}, \dots, \underbrace{x_N, \dots, x_N}_{\lceil \beta_{\text{Off}} \rceil}, x_N, x_N, \dots),$$

stopping earlier if $T \leq N\lceil \beta_{\text{Off}} \rceil$.

For any f^* , we now show that $\text{Alg}_{\text{Off}}^t(\cdot; f^*)$ is an offline oracle with parameter β_{Off} on the sequence x^1, \dots, x^T . This is because for any $i < \min \{\lceil t/\lceil \beta_{\text{Off}} \rceil \rceil, N\}$, the covariate x_i is repeated $\lceil \beta_{\text{Off}} \rceil \geq \beta_{\text{Off}}$ times. Thus $\widehat{f}^t(x_i) = f^*(x_i)$. This implies for $t \leq N \cdot \lceil \beta_{\text{Off}} \rceil$ that

$$\sum_{s=1}^{t-1} \mathsf{D}_{0/1}(\widehat{f}^t(x^s), f^*(x^s)) = \sum_{s=t-\lceil t/\lceil \beta_{\text{Off}} \rceil \rceil + 1}^{t-1} \mathsf{D}_{0/1}(\widehat{f}^t(x^s), f^*(x^s)) \leq \lceil \beta_{\text{Off}} \rceil - 1 \leq \beta_{\text{Off}}.$$

If $t > N \cdot \lceil \beta_{\text{Off}} \rceil$, then all the covariates are repeated for more than $\lceil \beta_{\text{Off}} \rceil$ times, thus $\hat{f}^t = f^*$. Overall, we have shown that $\text{Alg}_{\text{Off}}^t(\cdot; f^*)$ is an offline oracle with parameter β_{Off} .

Now, fix any oracle-efficient online estimation algorithm, and consider the expected regret under $f^* \sim \text{Unif}(\mathcal{F})$ (with $\text{Alg}_{\text{Off}}^t(\cdot; f^*)$ as the oracle). If $T \geq N \lceil \beta_{\text{Off}} \rceil$, then regardless of how the algorithm chooses μ^t , since for any block of x_i , $f^*(x_i)$ is independent of $\mu^t(x_i)$, the expected regret is lower bounded as

$$\begin{aligned} & \sum_{t=1}^T \mathbb{E}_{f^* \sim \text{Unif}(\mathcal{F})} \mathbb{E}_{\bar{f} \sim \mu^t} [\mathsf{D}_{0/1}(\bar{f}(x^t), f^*(x^t))] \\ & \geq \sum_{i=1}^N \sum_{j=1}^{\lceil \beta_{\text{Off}} \rceil} \mathbb{E}_{f^* \sim \text{Unif}(\mathcal{F})} \mathbb{E}_{\bar{f} \sim \mu^{i \lceil \beta_{\text{Off}} \rceil + j}} [\mathsf{D}_{0/1}(\bar{f}(x_i), f^*(x_i))] \\ & = \frac{1}{2} N \lceil \beta_{\text{Off}} \rceil \geq \Omega(N(\beta_{\text{Off}} + 1)) = \Omega(\min\{(\beta_{\text{Off}} + 1) \log |\mathcal{F}|, (\beta_{\text{Off}} + 1)|\mathcal{X}|\}). \end{aligned}$$

□

G.2 Proofs from Section 3.2

Theorem 3.3 (Impossibility of memoryless algorithms for OEOE). *Consider the binary classification setting with $\mathcal{Z} = \mathcal{Y} = \{0, 1\}$ and loss $\mathsf{D}_{0/1}(\cdot, \cdot)$. For any $N \in \mathbb{N}$ and $\beta_{\text{Off}} \geq 0$, there exists an instance $(\mathcal{X}, \mathcal{Y}, \mathcal{Z}, \mathcal{K}, \mathcal{F})$ with $|\mathcal{F}| = |\mathcal{X}| = N$ such that for any memoryless oracle-efficient algorithm, there exists a sequence of covariates (x^1, \dots, x^T) and a (potentially improper) offline oracle Alg_{Off} with parameter β_{Off} such that $\mathbb{E}[\text{Est}_D^{\text{On}}(T)] \geq \Omega(\min\{N(\beta_{\text{Off}} + 1), T\})$. This conclusion still holds when the online estimation algorithm remembers $\hat{f}^1, \dots, \hat{f}^{t-1}$ but not x^1, \dots, x^{t-1} .*

Proof of Theorem 3.3. Given a parameter $\beta_{\text{Off}} > 0$ and an integer N , assume without loss of generality that $K := T/(\lfloor \beta_{\text{Off}} \rfloor + 1)$ is an integer. Consider the instance $(\mathcal{X}, \mathcal{Y}, \mathcal{Z}, \mathcal{K}, \mathcal{F})$ with $\mathcal{X} = [N]$, $\mathcal{Z} = \mathcal{Y} = \{0, 1\}$, $\mathcal{D} = \mathsf{D}_{0/1}$, $\mathcal{K}(z) = \mathbb{1}_z$, and parameter space $\mathcal{F} = \{f_i\}_{i \in [N]}$ is defined as

$$f_i(x) = \mathbb{1}\{x = i\}.$$

We consider a sequence of covariates (x^1, \dots, x^T) divided into K blocks, each with length $\lfloor \beta_{\text{Off}} \rfloor + 1$. In each block, the covariates will be chosen to be the same, i.e., $x^1 = \dots = x^{\lfloor \beta_{\text{Off}} \rfloor + 1}$, $x^{\lfloor \beta_{\text{Off}} \rfloor + 2} = \dots = x^{2\lfloor \beta_{\text{Off}} \rfloor + 2}$. We define $\tau_t = \lceil t/(\lfloor \beta_{\text{Off}} \rfloor + 1) \rceil$ as the index of the block the step t belongs to, and we adopt the convention that $x_\tau \in \mathcal{X}$ is value of the covariates for block τ , i.e., $x^t = x_{\tau_t}$ for all t . We leave the precise choice for x_1, \dots, x_K as a free parameter for now.

Fix any memoryless oracle-efficient online estimation algorithm defined by a sequence of maps $\{F^t\}_{t \in [T]}$ (cf. Definition 3.1). We set the true target parameter to be $f^* = f_{i^*}$, where the index $i^* \in [N]$ will be chosen later in the proof (in an adversarial fashion based on the algorithm under consideration); for now, we leave $i^* \in [N]$ as a free parameter.

We first specify the offline estimation oracle under f_{i^*} . For each block index $\tau = 1, \dots, K$, define

$$\mathcal{X}^\tau = \mathcal{X} \setminus (\{x^s\}_{s \leq (\tau-1)(\lfloor \beta_{\text{Off}} \rfloor + 1)} \cup \{i^*\})$$

as the set of covariates in $\mathcal{X} \setminus \{i^*\}$ that have not been observed before block τ , and let $\bar{\mathcal{X}}^\tau := \mathcal{X}^\tau \cup \{i^*\}$. We define

$$\hat{f}^t(x) = \mathbb{1}\{x \in \bar{\mathcal{X}}^{\tau_t}\}$$

as the estimator returned by the oracle at round t . It is immediate from this construction that regardless of how x^1, \dots, x^T are chosen, the offline estimation error is bounded by

$$\forall t \in [T], \sum_{s < t} \mathsf{D}_{0/1}(f^*(x^s), \hat{f}^t(x^s)) \leq \sum_{s=(\tau_t-1)(\lfloor \beta_{\text{Off}} \rfloor + 1)+1}^{t-1} \mathsf{D}_{0/1}(f^*(x^s), \hat{f}^t(x^s)) \leq \lfloor \beta_{\text{Off}} \rfloor \leq \beta_{\text{Off}},$$

since the value of \hat{f}^t differs from f_{i^*} only for covariates that have not been observed before block τ_t .

It remains to lower bound the algorithm's online estimation error. To start, note that $D_{0/1}$ coincides with D_{sq} on the set $\{0, 1\}$ and D_{sq} is convex. Hence, by Jensen's inequality, it suffices to lower bound

$$\sum_{t=1}^T (f^t(x^t) - f_{i^*}(x^t))^2,$$

where $f^t(x^t)$ is the mean under $\bar{f}^t(x^t)$ where $\bar{f}^t \sim \mu^t = F^t(\hat{f}^1, \dots, \hat{f}^{t-1}, \hat{f}^t)$.

To proceed, we specify the sequence x^1, \dots, x^T , choosing x^t as a measurable function of $\bar{\mathcal{X}}^t$ and i^* (recall that i^* itself has yet to be chosen). Fix a round t , and suppose that $\mathcal{X}^t \neq \emptyset$. We choose x^t to lower bound the estimation error by considering two cases. In the process, we will also define a function $\hat{j}^\tau : (2^{[N]})^{\otimes \tau} \rightarrow [N] \cup \{\perp\}$ for each $\tau \in [K]$, where $(2^{[N]})^{\otimes \tau} = \underbrace{2^{[N]} \times \dots \times 2^{[N]}}_{\tau \text{ copies}}$. Let

$\tau \in [K]$ be fixed.

- If $\sum_{p=1}^{\lfloor \beta_{\text{off}} \rfloor + 1} \mathbb{1}(f^{\tau(\lfloor \beta_{\text{off}} \rfloor + 1) + p}(x) \leq 1/2) \geq \frac{\lfloor \beta_{\text{off}} \rfloor + 1}{2}$ for all $x \in \bar{\mathcal{X}}^\tau$, we set $x_\tau = i^*$ (equivalently, $x^{\tau(\lfloor \beta_{\text{off}} \rfloor + 1) + p} = i^*$ for $p = 1, \dots, \lfloor \beta_{\text{off}} \rfloor + 1$), so that

$$\sum_{t=\tau(\lfloor \beta_{\text{off}} \rfloor + 1) + 1}^{(\tau+1)(\lfloor \beta_{\text{off}} \rfloor + 1)} \max_{x^t} (f^t(x^t) - f_{i^*}(x^t))^2 \geq \frac{\lfloor \beta_{\text{off}} \rfloor + 1}{8}.$$

In this case, we define $\hat{j}^\tau(\bar{\mathcal{X}}^1, \dots, \bar{\mathcal{X}}^\tau) = \perp$.

- If there exists $j \in \bar{\mathcal{X}}^\tau$ such that $\sum_{p=1}^{\lfloor \beta_{\text{off}} \rfloor + 1} \mathbb{1}(f^{\tau(\lfloor \beta_{\text{off}} \rfloor + 1) + p}(j) \leq 1/2) < \frac{\lfloor \beta_{\text{off}} \rfloor + 1}{2}$, we set $x_\tau = j$ (equivalently, $x^{\tau(\lfloor \beta_{\text{off}} \rfloor + 1) + p} = j$ for $p = 1, \dots, \lfloor \beta_{\text{off}} \rfloor + 1$) for the least such j , so that

$$\sum_{t=\tau(\lfloor \beta_{\text{off}} \rfloor + 1) + 1}^{(\tau+1)(\lfloor \beta_{\text{off}} \rfloor + 1)} \max_{x^t} (f^t(x^t) - f_{i^*}(x^t))^2 \geq \frac{\lfloor \beta_{\text{off}} \rfloor + 1}{8} \mathbb{1}\{i^* \neq j\}.$$

In this case, we define $\hat{j}^\tau(\bar{\mathcal{X}}^1, \dots, \bar{\mathcal{X}}^\tau) = j$ as well.

Note that since f^t is a measurable function of $\bar{\mathcal{X}}^1, \hat{j}^1, \dots, \bar{\mathcal{X}}^{\tau_t}, \hat{j}^{\tau_t}$ is well-defined.

Combining these cases, it follows that for any choice of i^* , choosing x_1, \dots, x_K in the fashion described above ensures that

$$\sum_{t=1}^T (f^t(x^t) - f_{i^*}(x^t))^2 \geq \frac{\lfloor \beta_{\text{off}} \rfloor + 1}{8} \sum_{\tau=1}^K \mathbb{1}\{\hat{j}^\tau(\bar{\mathcal{X}}^1, \dots, \bar{\mathcal{X}}^\tau) \neq i^*\}.$$

We now state and prove the following technical lemma, which asserts that there exists a choice of i^* for which the right-hand side above is large.

Lemma G.2. *For any algorithm, there exists a choice for $i^* \in [N]$ such that*

$$\min\left\{\tau : \hat{j}^\tau(\bar{\mathcal{X}}^1, \dots, \bar{\mathcal{X}}^\tau) = i^*\right\} \geq \Omega(N).$$

Proof of Lemma G.2. Consider a more abstract process, which we claim captures the evolution of $\bar{\mathcal{X}}^\tau$. Let $i^* \in [N]$, and let $A^1 = [N]$. We consider a sequence of sets $\{A^\tau\}_{\tau \geq 1}$ evolving according to the following process, parameterized by a sequence of functions $\{g^\tau : (2^{[N]})^{\otimes \tau} \rightarrow [N] \cup \{\perp\}\}_{\tau \geq 1}$ and index $i^* \in [N]$.

For $\tau \geq 1$:

- If $g^\tau(A^1, \dots, A^\tau) = \perp$, $A^{\tau+1} \leftarrow A^\tau$.
- If $g^\tau(A^1, \dots, A^\tau) \neq \perp$, let $A^{\tau+1} = A^\tau \setminus \{g^\tau(A^1, \dots, A^\tau)\}$ if $g^\tau(A^1, \dots, A^\tau) \neq i^*$, and let $A^{\tau+1} \leftarrow A^\tau$ otherwise.

We claim that there exists $i^* \in [N]$ such that $g^s(A^1, \dots, A^s) \neq i^*$ for all $s < N$ under this process. To see this define a set X^τ inductively: Starting from $X^1 = [N]$, set $X^{\tau+1} \leftarrow X^\tau \setminus g^\tau(X^1, \dots, X^\tau)$ if $g^\tau(X^1, \dots, X^\tau) \neq \perp$, and set $X^{\tau+1} \leftarrow X^\tau$ otherwise; note that this process does not depend on the choice i^* , since g^τ itself does not depend on i^* .

For any τ , observe that for any $i \in X^\tau$, if we set $i^* = i$, then $g^s(A^1, \dots, A^s) \neq i^*$ for all $s < \tau$, and so $A^\tau = X^\tau$. It follows that as long as $X^\tau \neq \emptyset$, we can choose i^* such that $g^s(A^1, \dots, A^s) \neq i^*$ for all $s < \tau$. Since X^τ shrinks by at most one element per iteration, it follows that this is possible for all $\tau < N$.

□

It follows immediately from Lemma G.2 that by choosing i^* as guaranteed by the lemma, we have

$$\begin{aligned} \sum_{t=1}^T (f^t(x^t) - f_{i^*}(x^t))^2 &\geq \frac{\lfloor \beta_{\text{Off}} \rfloor + 1}{8} \cdot \Omega(\min\{N, K\}) \\ &\geq \Omega(\min\{N(\beta_{\text{Off}} + 1), T\}). \end{aligned}$$

□

Theorem 3.3' (Impossibility of memoryless algorithms for OEOE; proper variant). *Consider the binary classification setting with $\mathcal{Z} = \mathcal{Y} = \{0, 1\}$ and loss $D_{0/1}(\cdot, \cdot)$. For any $N \in \mathbb{N}$ and $\beta_{\text{Off}} \geq 0$, there exists an instance $(\mathcal{X}, \mathcal{Y}, \mathcal{Z}, \mathcal{K}, \mathcal{F})$ with $|\mathcal{F}| = |\mathcal{X}| = N$ such that for any memoryless oracle-efficient algorithm that is (i) **proper**, and (ii) **time-invariant**, there exists a sequence of covariates (x^1, \dots, x^T) and a **proper** offline oracle Alg_{Off} with parameter β_{Off} such that $\mathbb{E}[\text{Est}_D^{\text{On}}(T)] \geq \Omega(\min\{N(\beta_{\text{Off}} + 1), T\})$.*

Proof of Theorem 3.3'. Given a parameter $N \in \mathbb{N}$, we consider the instance $(\mathcal{X}, \mathcal{Y}, \mathcal{Z}, \mathcal{K}, \mathcal{F})$ given by $\mathcal{X} = \{x_i\}_{i \in [N]}$, $\mathcal{Z} = \mathcal{Y} = \{0, 1\}$, $D = D_{0/1}$ and $\mathcal{K}(z) = \mathbb{1}_z$, with parameter space $\mathcal{F} := \{f_i\}_{i \in [N]}$ given by

$$f_i(x_j) = \mathbb{1}(j \geq i).$$

Let any memoryless oracle-efficient algorithm defined by prediction map $F^1 = \dots = F^T = F$ be given. We lower bound the algorithm's online estimation error by considering two cases.

Case 1: *There exists a parameter f_i such that the distribution $\mu_i = F(f_i)$ satisfies $\mu_i(f_i) < 1/2$.* We consider two sub-cases of Case 1. The first subcase is where $\mu_i(\{f_j : j > i\}) > 1/4$. In this case, we choose the sequence of covariates as $x^1 = \dots = x^T = x_i$, set $f^* = f_i$, and choose Alg_{Off} to be the offline estimation oracle that sets $\hat{f}^1 = \dots = \hat{f}^T = f_i$. With this choice, the offline estimation error for the oracle satisfies

$$\forall t \in [T], \sum_{s < t} D_{0/1}\left(f^*(x^s), \hat{f}^t(x^s)\right) = \sum_{s < t} D_{0/1}(f_i(x^s), f_i(x^s)) = 0.$$

However, the online estimation error satisfies

$$\begin{aligned} \sum_{t=1}^T \mathbb{E}_{\bar{f} \sim \mu^t} [D_{0/1}(f^*(x^t), \bar{f}(x^t))] &= \sum_{t=1}^T \mathbb{E}_{\bar{f} \sim \mu_i} [D_{0/1}(f_i(x_i), \bar{f}(x_i))] \\ &\geq \sum_{t=1}^T \sum_{j > i} \mu_i(f_j) D_{0/1}(f_i(x_i), f_j(x_i)) \\ &= \sum_{t=1}^T \sum_{j > i} \mu_i(f_j) D_{0/1}(f_i(x_i), f_{i+1}(x_i)) \\ &\geq \frac{1}{4} \sum_{t=1}^T D_{0/1}(f_i(x_i), f_{i+1}(x_i)) = T/4. \end{aligned}$$

The second sub-case of Case 1 is where $\mu_i(\{f_j : j > i\}) \leq 1/4$. This, combined with the fact that $\mu_i(f_i) < 1/2$, gives $\mu_i(\{f_j : j < i\}) > 1/4$. In this sub-case, we choose the sequence of covariates as $x^1 = \dots = x^T = x_{i-1}$, set $f^* = f_i$, and choose Alg_{Off} to be the offline estimation oracle that sets $f^1 = \dots = f^T = f_i$. In this case, the offline estimation error is zero:

$$\forall t \in [T], \sum_{s < t} \mathsf{D}_{0/1} \left(f^*(x^s), \hat{f}^t(x^s) \right) = \sum_{s < t} \mathsf{D}_{0/1} (f_i(x^s), f_i(x^s)) = 0.$$

In addition, the online estimation error is lower bounded by

$$\begin{aligned} \sum_{t=1}^T \mathbb{E}_{\bar{f} \sim \mu^t} [\mathsf{D}_{0/1} (f^*(x^t), \bar{f}(x^t))] &= \sum_{t=1}^T \mathbb{E}_{\bar{f} \sim \mu_i} [\mathsf{D}_{0/1} (f_i(x_{i-1}), \bar{f}(x_{i-1}))] \\ &\geq \sum_{t=1}^T \sum_{j < i} \mu_i(f_j) \mathsf{D}_{0/1} (f_i(x_{i-1}), f_j(x_{i-1})) \\ &= \sum_{t=1}^T \sum_{j < i} \mu_i(f_j) \mathsf{D}_{0/1} (f_i(x_{i-1}), f_{i-1}(x_{i-1})) \\ &\geq \frac{1}{4} \sum_{t=1}^T \mathsf{D}_{0/1} (f_i(x_{i-1}), f_{i-1}(x_{i-1})) = T/4. \end{aligned}$$

Case 2: For all parameter $f_i \in \mathcal{F}$, the distribution $\mu_i = F(f_i)$ has $\mu_i(f_i) \geq 1/2$.

In this case, we choose x^1, \dots, x^T by repeating each of the covariates $\lfloor \beta_{\text{Off}} \rfloor + 1$ number of times in increasing order by their index, and choose the offline estimation oracle Alg_{Off} to return f_1, \dots, f_N in the same block-wise but with the index offset by 1.

Formally, let $\tau_t = \lceil t/(\lfloor \beta_{\text{Off}} \rfloor + 1) \rceil$ denote the index of the block that round t belongs to, so that $\tau_1 = \dots = \tau_{\lfloor \beta_{\text{Off}} \rfloor + 1} = 1, \tau_{\lfloor \beta_{\text{Off}} \rfloor + 2} = \dots = \tau_{2\lfloor \beta_{\text{Off}} \rfloor + 2} = 2$ and so on. We choose $x^t = x_{\min\{\tau_t, N\}}$, and choose the offline oracle Alg_{Off} to set $\hat{f}^t = f_{\min\{\tau_t, N\}}$. Finally, we set $f^* = f_N$.

We have that for all t , the offline estimation error of the oracle is bounded as.

$$\begin{aligned} \sum_{s < t} \mathsf{D}_{0/1} \left(\hat{f}^t(x^s), f^*(x^s) \right) &= \sum_{s < t} \mathsf{D}_{0/1} (f_N(x_{\min\{\tau_s, N\}}), f_{\min\{\tau_s, N\}}(x_{\min\{\tau_s, N\}})) \\ &= \sum_{s < t, \tau_s = \tau_t} 1 \leq \lfloor \beta_{\text{Off}} \rfloor \leq \beta_{\text{Off}}. \end{aligned}$$

However, the online estimation error is lower bounded by

$$\begin{aligned} \sum_{t=1}^T \mathbb{E}_{\bar{f} \sim \mu^t} [\mathsf{D}_{0/1} (f^*(x^t), \bar{f}(x^t))] &\geq \frac{1}{2} \sum_{t=1}^T \mathsf{D}_{0/1} (f_N(x_{\min\{\tau_t, N\}}), f_{\min\{\tau_t, N\}}(x_{\min\{\tau_t, N\}})) \\ &\geq \Omega(\min\{T, N(\beta_{\text{Off}} + 1)\}). \end{aligned}$$

□

Proposition D.1 (Upper bound for memoryless OOE). *For any instance $(\mathcal{X}, \mathcal{Y}, \mathcal{Z}, \mathcal{K}, \mathcal{F})$, metric-like loss D , and offline oracle Alg_{Off} with parameter β_{Off} , the algorithm that returns $\hat{f}^t = \hat{f}^t := \text{Alg}_{\text{Off}}^t(x^1, \dots, x^{t-1}, y^1, \dots, y^{t-1})$ has online estimation error $\mathbf{Est}_{\mathsf{D}}^{\text{On}}(T) \leq O((\beta_{\text{Off}} + 1)|\mathcal{X}| \log T)$.*

Proof of Proposition D.1. The proof is very similar to the second part of the proof of Lemma G.1.

For each $x \in \mathcal{X}$, define $N_{t-1}(x) = \sum_{s=1}^{t-1} \mathbb{1}(x^s = x)$. Then we can write the online estimation error as

$$\sum_{t=1}^T \mathsf{D} \left(\hat{f}^t(x^t), f^*(x^t) \right) = \sum_{t=1}^T \sum_{x \in \mathcal{X}} \frac{\mathbb{1}(x = x^t)}{N_{t-1}(x) \vee 1} \cdot (N_{t-1}(x) \vee 1) \cdot \mathsf{D} \left(\hat{f}^t(x), f^*(x) \right),$$

As a consequence of the offline estimation guarantee for \hat{f}^t , we have that

$$(N_{t-1}(x) \vee 1) \mathsf{D}(\hat{f}^t(x), f^*(x)) \leq \left(\sum_{s=1}^{t-1} \mathsf{D}(\hat{f}^t(x^s), f^*(x^s)) \right) \vee 1 \leq \beta_{\text{Off}} + 1.$$

Combining this with the preceding inequality gives

$$\sum_{t=1}^T \mathsf{D}(\hat{f}^t(x^t), f^*(x^t)) \leq (\beta_{\text{Off}} + 1) \sum_{t=1}^T \sum_{x \in \mathcal{X}} \frac{\mathbb{1}(x = x^t)}{N_{t-1}(x) \vee 1}.$$

Now, for any $x \in \mathcal{X}$, define $t_x := \min \{t \leq T : x^t = x\}$ if this set is not empty, and let $t_x = T$ otherwise. From this definition and the fact that $1 + 1/2 + \dots + 1/T \leq 1 + \log T$, we have that

$$\begin{aligned} \sum_{x \in \mathcal{X}} \sum_{t=1}^T \frac{\mathbb{1}(x = x^t)}{N_{t-1}(x) \vee 1} &= \sum_{x \in \mathcal{X}} \left(\sum_{t=1}^{t_x} \frac{\mathbb{1}(x = x^t)}{N_{t-1}(x) \vee 1} + \sum_{t=t_x+1}^T \frac{\mathbb{1}(x = x^t)}{N_{t-1}(x) \vee 1} \right) \\ &\leq \sum_{x \in \mathcal{X}} \left(1 + \sum_{i=1}^{N_{T-1}(x)} \frac{1}{i} \right) \\ &\leq 2|\mathcal{X}| + |\mathcal{X}| \log T \leq 3|\mathcal{X}| \log T. \end{aligned}$$

We conclude that

$$\sum_{t=1}^T \mathsf{D}(f^*(x^t), \hat{f}^t(x^t)) \leq 3(\beta_{\text{Off}} + 1) \cdot |\mathcal{X}| \log T.$$

□

G.3 Proofs from Appendix D

Theorem D.1 (Reduction from oracle-efficient online estimation to delayed online learning). *Let D be any convex, metric-like loss. Suppose we run Algorithm 2 with delay parameter $N \in \mathbb{N}$ and a delayed online learning algorithm \mathcal{A}_{DOL} for the class \mathcal{F} . Then for all $\gamma \geq 1$, Algorithm 2 ensures that*

$$\mathbb{E}[\mathbf{Est}_{\mathsf{D}}^{\text{On}}(T)] \leq O(C_{\mathsf{D}}\gamma(N + \beta_{\text{Off}}T/N) + R_{\text{DOL}}(T, N, \gamma)), \quad (14)$$

with any offline oracle $\mathbf{Alg}_{\text{Off}}$ with parameter $\beta_{\text{Off}} \geq 0$, where

$$R_{\text{DOL}}(T, N, \gamma) := \sum_{t=1}^T \mathbb{E}_{\bar{f}^t \sim \mu^t} [\ell^t(\bar{f}^t)] - \gamma \cdot \min_{f \in \mathcal{F}} \sum_{t=1}^T \ell^t(f) \quad (15)$$

is the regret of \mathcal{A}_{DOL} for the sequence of losses constructed in Algorithm 2.

Proof of Theorem D.1. Using the metric-like loss property, we can bound the online estimation error of Algorithm 2 by

$$\begin{aligned} &\sum_{t=1}^T \mathbb{E}_{\bar{f} \sim \mu^t} [\mathsf{D}(\bar{f}(x^t), f^*(x^t))] \\ &\leq C_{\mathsf{D}} \cdot \sum_{t=1}^T \mathbb{E}_{\bar{f} \sim \mu^t} [\mathsf{D}(\bar{f}(x^t), \tilde{f}^t(x^t))] + C_{\mathsf{D}} \cdot \sum_{t=1}^T \mathsf{D}(\tilde{f}^t(x^t), f^*(x^t)). \end{aligned}$$

By the regret guarantee for the delayed online learning algorithm \mathcal{A}_{DOL} , we have

$$\sum_{t=1}^T \mathbb{E}_{\bar{f} \sim \mu^t} [\mathsf{D}(\bar{f}(x^t), \tilde{f}^t(x^t))] \leq \gamma \cdot \sum_{t=1}^T \mathsf{D}(\tilde{f}^t(x^t), f^*(x^t)) + R_{\text{DOL}}(T, N, \gamma)$$

since $f^* \in \mathcal{F}$. Combining these observations, we have that

$$\sum_{t=1}^T \mathbb{E}_{\bar{f} \sim \mu^t} [\mathsf{D}(\bar{f}(x^t), f^*(x^t))] \leq C_{\mathsf{D}}(\gamma + 1) \sum_{t=1}^T \mathsf{D}(\tilde{f}^t(x^t), f^*(x^t)) + R_{\mathsf{DOL}}(T, N, \gamma).$$

Finally, from the definition of \tilde{f}^t and the convexity of the loss D , we have

$$\begin{aligned} \sum_{t=1}^T \mathsf{D}(\tilde{f}^t(x^t), f^*(x^t)) &\leq N + \frac{1}{N} \sum_{t=1}^{T-N} \sum_{i=t+1}^{t+N} \mathsf{D}(\hat{f}^i(x^t), f^*(x^t)) \\ &= N + \frac{1}{N} \sum_{t=2}^T \sum_{i < t} \mathsf{D}(\hat{f}^t(x^i), f^*(x^i)) \\ &\leq N + \frac{\beta_{\mathsf{Off}} T}{N}, \end{aligned}$$

where the final line uses the offline estimation guarantee for $\mathsf{Alg}_{\mathsf{Off}}$. This completes the proof. \square

Lemma D.1. *Consider the delayed online learning setting with a delay parameter N . There exists an algorithm that achieves*

$$R_{\mathsf{DOL}}(T, N, 2) = \sum_{t=1}^T \mathbb{E}_{\bar{f}^t \sim \mu^t} [\ell^t(\bar{f}^t)] - 2 \cdot \min_{f \in \mathcal{F}} \sum_{t=1}^T \ell^t(f) \leq 2N \cdot \log |\mathcal{F}|$$

for any sequences of losses $\ell^1, \dots, \ell^T \in [0, 1]$.

Proof of Lemma D.1. This result follows using Lemma G.3 with $\gamma = 2$, choosing $\mathcal{A}_{\mathsf{OL}}$ to be the exponential weights algorithm described in Corollary 2.3 of Cesa-Bianchi and Lugosi [17], which has

$$R_{\mathsf{OL}}(T, \gamma) \leq O(\log |\mathcal{F}|)$$

for all $T \in \mathbb{N}$ and $\gamma \geq 1$. \square

Theorem D.2 (Characterization of oracle-efficient learnability for binary classification). *Consider a binary classification instance $(\mathcal{X}, \mathcal{Y}, \mathcal{Z}, \mathcal{K}, \mathcal{F})$ with $\mathcal{Z} = \mathcal{Y} = \{0, 1\}$, $\mathsf{D} = \mathsf{D}_{0/1}$ and $\mathcal{K}(z) = \mathbb{1}_z$. For any class \mathcal{F} and $\beta_{\mathsf{Off}} \geq 0$, there exists an oracle-efficient algorithm that achieves online estimation error $O(\sqrt{\beta_{\mathsf{Off}} \mathsf{Ldim}(\mathcal{F}) \cdot T \log T} + \mathsf{Ldim}(\mathcal{F}) \log T)$. On the other hand, in the worst-case any algorithm must suffer at least $\Omega(\mathsf{Ldim}(\mathcal{F}))$ online estimation error.*

Proof of Theorem D.2. For the lower bound we recall that for $\beta_{\mathsf{Off}} = 0$, Lemma 21.6 of [57] states that any algorithm (oracle-efficient or not) has to suffer $\Omega(\mathsf{Ldim}(\mathcal{F}))$ online estimation error in the worst case.

For the remainder of the proof, we focus on establishing the upper bound. For any set of parameters $\mathcal{F} : \mathcal{X} \rightarrow \Delta(\{0, 1\})$, define the majority vote function $\mathsf{Majority}(\mathcal{F})$ for a class \mathcal{F} via

$$\mathsf{Majority}(\mathcal{F})(x) = \mathbb{1} \left\{ \sum_{f \in \mathcal{F}} f(1 \mid x) \geq \sum_{f \in \mathcal{F}} f(0 \mid x) \right\}$$

for all $x \in \mathcal{X}$. We will show that Algorithm 4 (a variant of Algorithm 2 that replaces averaging with a majority vote), with a properly chosen delayed online learning algorithm $\mathcal{A}_{\mathsf{DOL}}$, can obtain

$$O\left(\sqrt{\beta_{\mathsf{Off}} \mathsf{Ldim}(\mathcal{F}) \cdot T \log T} + \mathsf{Ldim}(\mathcal{F}) \log T\right)$$

online estimation error.

Algorithm 4 Reduction to delayed online learning for binary loss

- 1: **input:** Offline estimation oracle $\mathbf{Alg}_{\text{Off}}$ with parameter $\beta_{\text{Off}} \geq 0$, delay parameter $N \in \mathbb{N}$, delayed online learning algorithm \mathcal{A}_{DOL} for class \mathcal{F} .
- 2: **for** $t = 1, 2, \dots, T$ **do**
- 3: Receive \tilde{f}^t from offline estimation algorithm.
- 4: **if** $t > N$ **then**
- 5: Let $\tilde{f}^{t-N} = \text{Majority}(\{\hat{f}^i\}_{i=t-N+1}^t)$. *// This is the only different step compared to Algorithm 2.*
- 6: Let $\ell^{t-N}(f) = D_{0/1}(\tilde{f}^{t-N}(x^{t-N}), f(x^{t-N}))$ and pass $\ell^{t-N}(\cdot)$ to \mathcal{A}_{DOL} as the delayed feedback.
- 7: Let $\mu^t = \mathcal{A}_{\text{DOL}}^t(\ell^1, \dots, \ell^{t-N})$ be the delayed online learner's prediction distribution.
- 8: Predict with $\tilde{f}^t \sim \mu^t$ and receive x^t .

Let $\gamma \geq 1$ be fixed, and consider any delayed online learning algorithm \mathcal{A}_{DOL} that achieves

$$\sum_{t=1}^T \mathbb{E}_{\bar{f}^t \sim \mu^t} [\ell^t(\bar{f}^t)] - \gamma \cdot \min_{f \in \mathcal{F}} \sum_{t=1}^T \ell^t(f) \leq R_{\text{DOL}}(T, N, \gamma).$$

for any sequence of losses in the delayed online learning setting with delay N (i.e., where we receive loss ℓ^t at time $t + N$ for some $N \geq 0$).

We proceed to bound the regret of Algorithm 4. Since the loss $D_{0/1}$ is metric-like, the online estimation error is upper bounded by

$$\sum_{t=1}^T \mathbb{E}_{\bar{f} \sim \mu^t} [D_{0/1}(\bar{f}(x^t), f^*(x^t))] \leq \sum_{t=1}^T \mathbb{E}_{\bar{f} \sim \mu^t} [D_{0/1}(\bar{f}(x^t), \tilde{f}^t(x^t))] + \sum_{t=1}^T D_{0/1}(\tilde{f}^t(x^t), f^*(x^t)).$$

Next, the guarantee of \mathcal{A}_{DOL} ensures that

$$\sum_{t=1}^T \mathbb{E}_{\bar{f} \sim \mu^t} [D_{0/1}(\bar{f}(x^t), \tilde{f}^t(x^t))] \leq \gamma \sum_{t=1}^T D_{0/1}(\tilde{f}^t(x^t), f^*(x^t)) + R_{\text{DOL}}(T, N, \gamma),$$

since $f^* \in \mathcal{F}$. Combining these observations, we have that

$$\sum_{t=1}^T \mathbb{E}_{\bar{f} \sim \mu^t} [D_{0/1}(\bar{f}(x^t), f^*(x^t))] \leq (\gamma + 1) \sum_{t=1}^T D_{0/1}(\tilde{f}^t(x^t), f^*(x^t)) + R_{\text{DOL}}(T, N, \gamma).$$

Finally, we observe that for each step t , if $D_{0/1}(\tilde{f}^t(x^t), f^*(x^t)) = 1$, it means that least $N/2$ of the predictors $\hat{f}^{t+1}, \dots, \hat{f}^{t+N}$ must have predicted $f^*(x^t)$ incorrectly. This implies that

$$D_{0/1}(\tilde{f}^t(x^t), f^*(x^t)) \leq \frac{2}{N} \sum_{i=t+1}^{t+N} D_{0/1}(\hat{f}^i(x^t), f^*(x^t)).$$

But since the offline estimation assumption states that

$$\sum_{i < t} D_{0/1}(\hat{f}^i(x^i), f^*(x^i)) \leq \beta_{\text{Off}},$$

this implies that

$$\begin{aligned} \sum_{t=1}^T D_{0/1}(\tilde{f}^t(x^t), f^*(x^t)) &\leq N + \frac{2}{N} \sum_{t=1}^{T-N} \sum_{i=t+1}^{t+N} D_{0/1}(\hat{f}^i(x^t), f^*(x^t)) \\ &= N + \frac{2}{N} \sum_{t=2}^T \sum_{i < t} D_{0/1}(\hat{f}^i(x^t), f^*(x^t)) \\ &\leq N + \frac{2\beta_{\text{Off}}T}{N}. \end{aligned}$$

We conclude that

$$\sum_{t=1}^T \mathbb{E}_{\bar{f}^t \sim \mu^t} [\mathcal{D}_{0/1}(\bar{f}^t(x^t), f^*(x^t))] \leq O(\gamma(N + \beta_{\text{Off}} T/N) + R_{\text{DOL}}(T, N, \gamma)).$$

To complete the proof, we set $\gamma = 2$ and choose \mathcal{A}_{OL} to be the algorithm described in Theorem 21.10 of Shalev-Shwartz and Ben-David [57], which (by incorporating the same technique¹⁸ as Corollary 2.3 of Cesa-Bianchi and Lugosi [17]), ensures that

$$R_{\text{OL}}(T/N, 2) \leq O(\text{Ldim}(\mathcal{F}) \log T).$$

Then by Lemma G.3 with $\gamma = 2$, we have

$$R_{\text{DOL}}(T, N, 2) \leq O(N \cdot \text{Ldim}(\mathcal{F}) \log T).$$

Setting $N = \sqrt{\beta_{\text{Off}} T / (\text{Ldim}(\mathcal{F}) \log T)} \vee 1$, this yields

$$\sum_{t=1}^T \mathbb{E}_{\bar{f}^t \sim \mu^t} [\mathcal{D}(\bar{f}^t(x^t), f^*(x^t))] \leq O\left(\sqrt{\beta_{\text{Off}} \text{Ldim}(\mathcal{F}) \cdot T \log T} + \text{Ldim}(\mathcal{F}) \log T\right).$$

□

G.3.1 Supporting Lemmas

Algorithm 5 Reduction from delayed online learning to non-delayed online learning

```

1: input: Delay parameter  $N \in \mathbb{N}$ , base online learning algorithm  $\mathcal{A}_{\text{OL}}$ .
2: Initialize  $N$  copies  $\mathcal{A}_{\text{OL}}^1, \dots, \mathcal{A}_{\text{OL}}^N$  of the base algorithm.
3: for  $t = 1, \dots, T$  do
4:   if  $t \leq N$  then
5:     Let  $\mu^t = \mathcal{A}_{\text{OL}}^t(\emptyset)$ .
6:   else
7:     Let  $i \equiv t \pmod{N}$  where  $i \in [N]$ .
8:     Receive loss  $\ell^{t-N}$ .
9:     Feed  $\ell^{t-N}$  to  $\mathcal{A}_{\text{OL}}^i$ .
10:    Let  $\mu^t = \mathcal{A}_{\text{OL}}^i(\ell^i, \ell^{i+N}, \dots, \ell^{t-N})$ .
11:   Play  $\bar{f}^t \sim \mu^t$ .

```

The following lemma is a standard result [67, 42, 35, 47] which shows that the delayed online learning problem setting in Appendix D can be generically reduced to non-delayed online learning. The idea behind the reduction, which is displayed in Algorithm 5, is as follows. Given a delay parameter $N \in \mathbb{N}$, we run N copies $\mathcal{A}_{\text{OL}}^1, \dots, \mathcal{A}_{\text{OL}}^N$ of a given “base” online learning algorithm \mathcal{A}_{OL} for a class \mathcal{F} over disjoint subsequences of rounds. The following lemma gives a guarantee for this reduction

Lemma G.3 (Delayed online learning reduction). *Let \mathcal{A}_{OL} be a base online learning algorithm for the class \mathcal{F} with the property that for any sequence of losses ℓ^1, \dots, ℓ^T in the non-delayed online learning setting and any $\gamma \geq 1$,*

$$\sum_{t=1}^T \mathbb{E}_{\bar{f}^t \sim \mu^t} [\ell^t(\bar{f}^t)] - \gamma \cdot \min_{f \in \mathcal{F}} \sum_{t=1}^T \ell^t(f) \leq R_{\text{OL}}(T, \gamma).$$

If we run Algorithm 5 with delay parameter $N \in \mathbb{N}$, then for all $\gamma \geq 1$, the algorithm ensures that

$$\begin{aligned} R_{\text{DOL}}(T, N, \gamma) &\leq \sum_{i=1}^N \left(\sum_{j=1}^{T/N} \mathbb{E}_{\bar{f}^{i+N \cdot j} \sim \mu^{i+N \cdot j}} [\ell^{i+N \cdot j}(\bar{f}^{i+N \cdot j})] - \gamma \cdot \min_{f \in \mathcal{F}} \sum_{j=1}^{T/N} \ell^{i+N \cdot j}(f) \right) \\ &\leq N \cdot R_{\text{OL}}(T/N, \gamma) \end{aligned}$$

for online learning with delay N .

¹⁸The algorithm described in Theorem 21.10 of Shalev-Shwartz and Ben-David [57] applies the exponential weights algorithm to a specialized class of experts, and the guarantee obtained is for $R_{\text{OL}}(T, 1)$. The analysis from Corollary 2.3 of Cesa-Bianchi and Lugosi [17] shows that for $\gamma > e/(e-1)$, the same algorithm obtains $R_{\text{OL}}(T, 2)$ scaling with $O(\text{Ldim}(\mathcal{F}) \log T)$. We omit the details here since it is a standard argument.

Proof of Lemma G.3. By the guarantee of \mathcal{A}_{OL} , we have that for all $i \in [N]$,

$$\sum_{j=1}^{T/N} \mathbb{E}_{\bar{f}^{i+N \cdot j} \sim \mu^{i+N \cdot j}} [\ell^{i+N \cdot j}(\bar{f}^{i+N \cdot j})] - \gamma \cdot \min_{f \in \mathcal{F}} \sum_{j=1}^{T/N} \ell^{i+N \cdot j}(f) \leq R_{\text{OL}}(T/N, \gamma).$$

Summing up over all $i \in [N]$, we obtain

$$\begin{aligned} N \cdot R_{\text{OL}}(T/N, \gamma) &= \sum_{i=1}^N \sum_{j=1}^{T/N} \mathbb{E}_{\bar{f}^{i+N \cdot j} \sim \mu^{i+N \cdot j}} [\ell^{i+N \cdot j}(\bar{f}^{i+N \cdot j})] - \gamma \cdot \sum_{i=1}^N \min_{f \in \mathcal{F}} \sum_{j=1}^{T/N} \ell^{i+N \cdot j}(f) \\ &\geq \sum_{i=1}^N \sum_{j=1}^{T/N} \mathbb{E}_{\bar{f}^{i+N \cdot j} \sim \mu^{i+N \cdot j}} [\ell^{i+N \cdot j}(\bar{f}^{i+N \cdot j})] - \gamma \cdot \min_{f \in \mathcal{F}} \sum_{i=1}^N \sum_{j=1}^{T/N} \ell^{i+N \cdot j}(f) \\ &= \sum_{t=1}^T \mathbb{E}_{\bar{f}^t \sim \mu^t} [\ell^t(\bar{f}^t)] - \gamma \cdot \min_{f \in \mathcal{F}} \sum_{t=1}^T \ell^t(f). \end{aligned}$$

□

H Proofs from Section 4

H.1 Proofs from Section 4.1

Theorem 4.1 (Computational lower bound for OEOE). *Assume the existence of one-way functions.¹⁹ There exists a sequence of polynomially computable classes $(\mathcal{F}_1, \mathcal{F}_2, \dots, \mathcal{F}_n, \dots)$, along with a sequence of $\text{poly}(n)$ -output description length offline oracles with $\beta_{\text{Off}} = 0$ associated with each \mathcal{F}_n , such that for any fixed polynomials $p, q : \mathbb{N} \rightarrow \mathbb{N}$ and all $n \in \mathbb{N}$ sufficiently large, any oracle-efficient online estimation algorithm with runtime bounded by $p(n)$ must have $\mathbb{E}[\text{Est}_{\text{D}}^{\text{On}}(T)] \geq T/4$ for all $1 \leq T \leq q(n)$. At the same time, there exists an inefficient algorithm that achieves $\mathbb{E}[\text{Est}_{\text{D}}^{\text{On}}(T)] \leq O(\sqrt{n})$ for all $T \in \mathbb{N}$.*

Proof of Theorem 4.1. We frame the example proposed by Blum [14] in their Theorem 3.2 (see also Bun [15]) in our setting. For any integer $n \geq 1$, let the covariate space \mathcal{X}_n be $\mathcal{X}_n = \{0, 1\}^n$, and set $\mathcal{Z} = \mathcal{Y} = \{0, 1\}$ and $\mathcal{K}(z) = \mathbb{1}_z$. We define a class $\mathcal{F}_n = \{f_s : s \in \{0, 1\}^{\sqrt{n}}\}$ with

$$f_s(x) := \begin{cases} 1 & \text{if } x \in c_s, \\ 0 & \text{otherwise,} \end{cases}$$

for a certain collection of subsets $\{c_s \in \mathcal{X}_n\}_{s \in \{0, 1\}^{\sqrt{n}}}$ defined in Definition 2 of Blum [14], which is constructed based on cryptographic functions using the assumption of existence of one-way functions. The precise definition will not be important. The properties we will use are:

1. The value $f_s(x)$ can be computed in $\text{poly}(n)$ time for any $x \in \mathcal{X}_n$.
2. For any polynomials $p(n), q(n)$, any (possibly randomized) online estimation algorithm (oracle-efficient or not) which runs in time $p(n)$, and any time step $T \leq q(n)$, for sufficiently large n where $q(n) \ll 2^{\sqrt{n}}$,²⁰ there exists $s \in \{0, 1\}^{\sqrt{n}}$ and a sequence $x_s^1, \dots, x_s^{2^{\sqrt{n}}-1}, x_s^{2^{\sqrt{n}}}$ (the specific definition of this sequence can be found in Blum [14]) such that the online estimation error under this sequence when $f^* = f_s$ is at least $T/4$ in expectation. Our lower bound construction for any oracle efficient online estimation algorithm with runtime bounded by $p(n)$ in time step bounded by $1 \leq T \leq q(n)$ will choose the aforementioned covariate sequence as the covariates revealed with the aforementioned function as the true parameter, i.e., $x^\tau = x_s^\tau$ for $\tau \in [T]$ and $f^* = f_s$.

¹⁹Existence of one-way functions is a standard and widely believed complexity-theoretic assumptions, which forms the basis of modern cryptography [30].

²⁰The argument is essentially asymptotic, since the choice of n is determined by the power of the one-way function.

It is straightforward to see that the sequence $(\mathcal{F}_1, \mathcal{F}_2, \dots, \mathcal{F}_n, \dots)$ admits polynomial description length as claimed, since $\log |\mathcal{F}_n| = \sqrt{n}$. We are left to verify that there is an offline oracle that achieves $\beta_{\text{Off}} = 0$ with $\text{poly}(n)$ -output description length, yet does not provide any information not already available to the learner in the setting of Blum [14] (recall that in the protocol of Blum [14], the learner gets to see the covariates x^1, \dots, x^t and the true labels y^1, \dots, y^{t-1} at time step t before making their prediction, but does not receive any other feedback).

Consider the following offline oracle $\text{Alg}_{\text{Off}}^t(x^1, \dots, x^{t-1}, y^1, \dots, y^{t-1})$. The oracle output \hat{f}^t is a circuit that, on input x , compares x sequentially with x^1, \dots, x^t . If x is ever equal to x^τ for some $\tau \in [t-1]$, the circuit will output y^τ . If x is not equal to any x^τ for $\tau \in [t-1]$, the circuit outputs 0. Such a Boolean circuit can be constructed with polynomial size in n because each x^τ has length n for $\tau \in [t-1]$ and $t \leq T \leq q(n)$ by assumption. It is easy to see that this oracle achieves $\beta_{\text{Off}} = 0$, yet does not provide any additional information about target parameter f^* beyond what is available in the model of Blum [14]. Combining all the above, we complete our lower bound proof.

Lastly, we observe that since the setting we consider is an instance of noiseless binary classification, the classical halving algorithm achieves an online estimation error bound of $O(\log |\mathcal{F}_n|) = O(\sqrt{n})$ [17].

□

H.2 Proofs from Section 4.2

In this section, we prove Theorem 4.2 through four layers of reductions through different variants of the online estimation setting. In Appendix H.2.1, we first introduce the relevant settings and the describe reductions through them. We then combine these reductions to prove Theorem 4.2. Finally, in Appendix H.2.2, we prove each of the four reduction results.

H.2.1 Proof of Theorem 4.2

Theorem 4.2. *Let \mathcal{A}_{CDE} be an arbitrary (unrestricted) online estimation algorithm that satisfies Eq. (4) and has runtime $\text{Time}(\mathcal{F}, T)$. Then for any $N \in \mathbb{N}$, there exists an oracle-efficient online estimation algorithm that achieves estimation error*

$$\mathbb{E}[\text{Est}_{\text{H}}^{\text{On}}(T)] \leq \tilde{O}(C_{\mathcal{F}} \log V \cdot \beta_{\text{Off}} T / N + N \cdot (R_{\text{CDE}}(T) + C_{\mathcal{F}} \log V))$$

with runtime $\text{poly}(\text{Time}(\mathcal{F}, T), \log |\mathcal{F}|, \log |\mathcal{X}|, T)$, where $\beta_{\text{Off}} \geq 0$ is the offline estimation parameter. The distributions μ^1, \dots, μ^T produced by the algorithm have support size $\text{poly}(\log |\mathcal{F}|, \log |\mathcal{X}|, T)$. As a special case, if the online estimation guarantee for the base algorithm holds with $R_{\text{CDE}}(T) \leq C'_{\mathcal{F}} \log T$ for some problem-dependent constant $C'_{\mathcal{F}} \geq 1$, then by choosing N appropriately, we achieve $\mathbb{E}[\text{Est}_{\text{H}}^{\text{On}}(T)] \leq \tilde{O}((C_{\mathcal{F}}(C_{\mathcal{F}} + C'_{\mathcal{F}})\beta_{\text{Off}})^{1/2} \log V \cdot T^{1/2} + (C_{\mathcal{F}} + C'_{\mathcal{F}}) \log V)$.

Proof of Theorem 4.2. The proof of Theorem 4.2 is algorithmic, and is based on several layers of reductions.

- First, using the scheme in Appendix D, we reduce the problem of oracle-efficient online estimation to delayed online learning with the loss function $\ell^{t-N}(f) = D(\tilde{f}^{t-N}(x^{t-N}), f(x^{t-N}))$ defined in Algorithm 2, where $\tilde{f}^{t-N} = \frac{1}{N} \sum_{i=t-N+1}^t \hat{f}^i$ is an average of offline estimators and $N \in \mathbb{N}$ is a delay parameter.
- Then, using a standard reduction [67, 42, 35, 47], we reduce the delayed online learning problem above to a sequence of N non-delayed online learning problems, with the same sequence of loss functions; both this and the preceding step are computationally efficient.
- To complete the reduction, we argue that the base algorithm can be used to solve the online learning problem above in an oracle-efficient fashion. To do this, we simulate interaction with the environment by sampling fictitious outcomes $y^t \sim \hat{f}^t(x^t)$ from the averaged offline estimators and passing them into the base algorithm. We argue that the fictitious outcomes approximate the true outcomes well through a change-of-measure argument.

Combining the above, we conclude that given any base algorithm that efficiently performs online estimation with outcomes sampled from the target parameter f^* , we can efficiently construct a computationally efficient and oracle-efficient algorithm. In more detail, we introduce four layers of reduction in reverse order from CDE to OEOE.

Conditional Density Estimation with Reference Outcomes (CDEwRO). The bottom-most reduction we consider is from a setting we refer to as *Conditional Density Estimation with Reference Outcomes (CDEwRO)* to the (realizable) CDE setting. CDEwRO is similar to CDE, but with the following difference. Instead of receiving outcomes y^1, \dots, y^T sampled from the true model $f^*(x^t)$ directly, in CDEwRO, the outcome is sampled from a *reference parameter* $\tilde{f}^t(x^t)$ which is guaranteed to be close to f^* in a certain sense. Moreover, the covariates and the reference parameters $\tilde{f}^1, \dots, \tilde{f}^T$ are selected obliviously (i.e. the entire sequence is chosen by the adversary before the online learning protocol begins).

Algorithm 6 Reduction from CDEwRO to CDE

- 1: **input:** Time $T \in \mathbb{N}$, base algorithm \mathcal{A}_{CDE} .
- 2: Nature selects T covariates x^1, \dots, x^T along with the reference parameters $\tilde{f}^1, \dots, \tilde{f}^T$.
- 3: **for** $t = 1, \dots, T$ **do**
- 4: Learner predicts $\bar{f}^t \sim \mu^t = \mathcal{A}_{\text{CDE}}(x^1, \dots, x^{t-1}, y^1, \dots, y^{t-1})$.
- 5: Outcome $y^t \sim \tilde{f}^t(x^t)$ is sampled and revealed to the learner with the covariate x^t .

Our reduction from CDEwRO to CDE is given in Algorithm 6. The main guarantee for this reduction is as follows.

Lemma H.1. *For any fixed $\zeta \geq 0$, suppose $\sum_{t=1}^T D_H^2(\tilde{f}^t(x^t), f^*(x^t)) \leq \zeta$. Let*

$$R_{\text{CDEwRO}}(T, \zeta) := 3C_{\mathcal{F}} \log V \cdot \zeta + R_{\text{CDE}}(T) + 2C_{\mathcal{F}} \cdot \log(C_{\mathcal{F}} T), \quad (25)$$

where $R_{\text{CDE}}(T)$ is defined as in Eq. (4) by the assumption on \mathcal{A}_{CDE} . Then Algorithm 6 achieves an expected online estimation error upper bound of

$$\sum_{t=1}^T \mathbb{E}[D_H^2(\bar{f}^t(x^t), f^*(x^t))] \leq R_{\text{CDEwRO}}(T, \zeta)$$

in the CDEwRO setting, and has runtime $\text{poly}(\text{Time}(\mathcal{F}, T), T)$.

The key technique in the proof of this lemma is a change of measure argument based on Donsker-Varadhan [46].

Conditional Density Estimation with Reference Parameters (CDEwRP). The next reduction in our stack is from a setting we refer to as *Conditional Density Estimation with Reference Parameters (CDEwRP)* to the CDEwRO setting above. CDEwRP is identical to CDEwRO, except that in the former setting, the learner directly observes the reference parameter \tilde{f}^t instead of observing $y^t \sim \tilde{f}^t(x^t)$ as in CDEwRO.

A second difference is that we allow the adversary in the CDEwRP setting to be adaptive, while our definition of the CDEwRO setting only allows for oblivious adversaries. Thus, the reduction we consider serves two purposes:

- Simulating the CDEwRO feedback model through sampling.
- Reducing the adaptive adversary to an oblivious one.

The reduction from adaptive adversaries to oblivious follows and improves upon the result from [31], and may be of independent interest.

Our reduction from CDEwRP to CDEwRO is displayed in Algorithm 6, and takes as input an algorithm $\mathcal{A}_{\text{CDEwRO}}(\cdot; \cdot)$ for the CDEwRO setting, where $\mathcal{A}_{\text{CDEwRO}}^t(T; \cdot)$ denotes the algorithm's output at round $t \leq T$ as a function of the history. The main guarantee for the algorithm is as follows.

Lemma H.2. *For any fixed $\zeta \geq 0$, suppose $\sum_{t=1}^T D_H^2(\tilde{f}^t(x^t), f^*(x^t)) \leq \zeta$. Let*

$$R_{\text{CDEwRP}}(T, \zeta, \varepsilon) := 2R_{\text{CDEwRO}}(T, \zeta) + \varepsilon, \quad (26)$$

Algorithm 7 Reduction from CDEwRP to CDEwRO

- 1: **input:** Time $T \in \mathbb{N}$, accuracy parameter $\varepsilon > 0$, algorithm $\mathcal{A}_{\text{CDEwRO}}$.
- 2: Let $L = O(T \log(T|\mathcal{F}| |\mathcal{X}|) / \varepsilon)$.
- 3: **for** $t = 1, \dots, T$ **do**
- 4: **for** $i = 1, \dots, L$ **do**
- 5: **for** $s = 1, \dots, t - 1$ **do**
- 6: Learner samples $y_i^{s,t} \sim \tilde{f}^s(x^s)$.
- 7: Learner computes $\bar{f}_i^t = \mathcal{A}_{\text{CDEwRO}}(T; x_i^{1,t}, \dots, x_i^{t-1,t}, y_i^{1,t}, \dots, y_i^{t-1,t})$.
- 8: Learner predicts via $\bar{f}^t \sim \mu^t = \text{Unif}(\{\bar{f}_i^t\}_{i \in [L]})$.
- 9: Nature selects and reveals the covariate x^t and the reference parameter \bar{f}^t based on μ^t .

where $R_{\text{CDEwRO}}(T, \zeta)$ is defined as in Eq. (25). Then Algorithm 7 with parameter $\varepsilon > 0$ achieves an expected online estimation error upper bound of

$$\sum_{t=1}^T \mathbb{E}[\mathsf{D}_{\mathsf{H}}^2(\bar{f}^t(x^t), f^*(x^t))] \leq R_{\text{CDEwRP}}(T, \zeta, \varepsilon)$$

in the CDEwRP setting, and runs in time $\text{poly}(\text{Time}(\mathcal{F}, T), T, \log |\mathcal{F}|, \log |\mathcal{X}|, 1/\varepsilon)$. The distributions μ^1, \dots, μ^T produced by the algorithm have support size $\text{poly}(\log |\mathcal{F}|, \log |\mathcal{X}|, T, 1/\varepsilon)$.

Conditional Density Estimation with Delayed Reference Parameters (CDEwDRP). Our next reduction is from a setting we refer to as *Conditional Density Estimation with Delayed Reference Parameters (CDEwDRP)* to the CDEwRP setting. CDEwDRP is identical to CDEwRP, except that the reference function \tilde{f}^t is revealed only at round $t + N$ instead of at round t , for a delay parameter $N \in \mathbb{N}$.

Algorithm 8 Reduction from CDEwDRP to CDEwRP

- 1: **input:** Time $T \in \mathbb{N}$, delay time $N \in \mathbb{N}$, algorithm $\mathcal{A}_{\text{CDEwRP}}$.
- 2: Initialize N copies of the algorithm $\mathcal{A}_{\text{CDEwRP}}$ as $\mathcal{A}_{\text{CDEwRP}}^1, \dots, \mathcal{A}_{\text{CDEwRP}}^N$.
- 3: **for** $t = 1, \dots, T$ **do**
- 4: Learner predicts $\bar{f}^t \sim \mu^t = \mathcal{A}_{\text{CDEwRP}}^i(T/N, 1/N; x^i, x^{i+N}, \dots, x^{t-N}, \tilde{f}^i, \tilde{f}^{i+N}, \dots, \tilde{f}^{t-N})$
where $i \equiv t \pmod{N}$.
- 5: Nature selects and reveals the covariate x^t and the reference parameter \tilde{f}^{t-N} based on μ^t .

Our reduction from CDEwDRP to CDEwRP is displayed in Algorithm 7, and takes as input an algorithm $\mathcal{A}_{\text{CDEwRP}}(\cdot; \cdot)$ for the CDEwRP setting, where $\mathcal{A}_{\text{CDEwRP}}^t(T, \varepsilon; \cdot)$ denotes the algorithm's output at round $t \leq T$ with accuracy parameter $\varepsilon > 0$ (cf. Algorithm 7), as a function of the history. The main guarantee for the algorithm is as follows.

Lemma H.3. For any fixed $\zeta \geq 0$, suppose $\sum_{t=1}^T \mathbb{E}[\mathsf{D}_{\mathsf{H}}^2(\bar{f}^t(x^t), f^*(x^t))] \leq \zeta$. Let

$$R_{\text{CDEwDRP}}(T, N, \zeta) := \sup_{\zeta_1, \dots, \zeta_N \geq 0, \sum_{i=1}^N \zeta_i \leq \zeta} \sum_{i=1}^N R_{\text{CDEwRP}}(T/N, \zeta_i, 1/N), \quad (27)$$

where $R_{\text{CDEwRP}}(T/N, \zeta_i, 1/N)$ is defined as in Eq. (26). Then Algorithm 8 achieves an expected online estimation error upper bound of

$$\sum_{t=1}^T \mathbb{E}[\mathsf{D}_{\mathsf{H}}^2(\bar{f}^t(x^t), f^*(x^t))] \leq R_{\text{CDEwDRP}}(T, N, \zeta)$$

in the CDEwDRP setting, and has runtime $\text{poly}(\text{Time}(\mathcal{F}, T), T, \log |\mathcal{F}|, \log |\mathcal{X}|)$. The distributions μ^1, \dots, μ^T produced by the algorithm have support size $\text{poly}(\log |\mathcal{F}|, \log |\mathcal{X}|, T)$.

Algorithm 9 Reduction from OEOE to CDEwDRP

- 1: **input:** Time $T \in \mathbb{N}$, offline estimation oracle $\mathbf{Alg}_{\text{Off}}$ with parameter $\beta_{\text{Off}} \geq 0$, delay parameter $N \in \mathbb{N}$, CDEwDRP algorithm $\mathcal{A}_{\text{ODEwDRP}}$.
- 2: **for** $t = 1, \dots, T$ **do**
- 3: Receive $\hat{f}^t = \mathbf{Alg}_{\text{Off}}^t(x^1, \dots, x^{t-1}, y^1, \dots, y^{t-1})$.
- 4: Learner computes reference parameter $\tilde{f}^{t-N} = \frac{1}{N} \sum_{i=t-N+1}^t \hat{f}^i$.
- 5: Learner predicts $\bar{f}^t \sim \mu^t = \mathcal{A}_{\text{ODEwDRP}}(T, N; x^1, x^2, \dots, x^t, \tilde{f}^1, \tilde{f}^2, \dots, \tilde{f}^{t-N})$.
- 6: Nature selects and reveals the covariate x^t based on μ^t .

Oracle-Efficient Online Estimation (OEOE). Our final reduction reduces the Oracle-Efficient Online Estimation setting (OEOE) to the CDEwDRP setting described above. This reduction, which is displayed in Algorithm 8, is a variant of the approach used in Theorem D.1. The reduction takes as input a CDEwDRP algorithm $\mathcal{A}_{\text{ODEwDRP}}(\cdot; \cdot)$, where $\mathcal{A}_{\text{ODEwDRP}}^t(T, N; \cdot)$ denotes the algorithm's output at round $t \leq T$ with delay parameter N , as a function of the history.

Lemma H.4. *Algorithm 9 achieves an expected online estimation error upper bound of*

$$\sum_{t=1}^T \mathbb{E}[\mathbf{D}_{\mathbb{H}}^2(\bar{f}^t(x^t), f^*(x^t))] \leq R_{\text{CDEwDRP}}(T, N, N + \beta_{\text{Off}}T/N)$$

in the OEOE setting, and has runtime $\text{poly}(\text{Time}(\mathcal{F}, T), T, \log |\mathcal{F}|, \log |\mathcal{X}|)$. The distributions μ^1, \dots, μ^T produced by the algorithm have support size $\text{poly}(\log |\mathcal{F}|, \log |\mathcal{X}|, T)$.

Completing the proof of Theorem 4.2. To prove Theorem 4.2, we compose all of the preceding reductions, with N left as a free parameter temporarily. We first apply Lemma H.4 to reduce from the OEOE setting to the CDEwDRP setting, with the guarantee that

$$\sum_{t=1}^T \mathbb{E}[\mathbf{D}_{\mathbb{H}}^2(\bar{f}^t(x^t), f^*(x^t))] \leq R_{\text{CDEwDRP}}(T, N, N + \beta_{\text{Off}}T/N).$$

Then by Lemma H.3, we can reduce the CDEwDRP setting to the CDEwRP setting, with the guarantee by Eq. (27) that

$$R_{\text{CDEwDRP}}(T, N, N + \beta_{\text{Off}}T/N) = \sup_{\sum_{i=1}^N \zeta_i \leq N + \beta_{\text{Off}}T/N} \sum_{i=1}^N R_{\text{CDEwRP}}(T/N, \zeta_i, 1/N).$$

Then apply Lemma H.2 N times with T , ζ , and ε in the Lemma chosen to be T/N , ζ_i , and $1/N$ respectively for each $i \in [N]$, we can reduce the CDEwRP setting to the CDEwRO setting with guarantee by Eq. (26) that

$$\sup_{\sum_{i=1}^N \zeta_i \leq N + \beta_{\text{Off}}T/N} \sum_{i=1}^N R_{\text{CDEwRP}}(T/N, \zeta_i, 1/N) = 1 + \sup_{\sum_{i=1}^N \zeta_i \leq N + \beta_{\text{Off}}T/N} 2 \sum_{i=1}^N R_{\text{CDEwRO}}(T/N, \zeta_i).$$

Consequently, apply Lemma H.1 N times with T and ζ in the Lemma chosen to be T/N and ζ_i for each $i \in [N]$, we can reduce the CDEwRO setting to the CDE setting with guarantee by Eq. (25) that

$$\begin{aligned}
& \sup_{\sum_{i=1}^N \zeta_i \leq N + \beta_{\text{off}} T/N} \sum_{i=1}^N R_{\text{CDEwRO}}(T/N, \zeta_i) \\
& \leq \sup_{\sum_{i=1}^N \zeta_i \leq N + \beta_{\text{off}} T/N} \sum_{i=1}^N (3C_{\mathcal{F}} \log V \cdot \zeta_i + R_{\text{CDE}}(T) + 2C_{\mathcal{F}} \cdot \log(C_{\mathcal{F}} T)) \\
& \lesssim C_{\mathcal{F}} \log V \cdot \left(N + \frac{\beta_{\text{off}} T}{N} \right) + N \cdot (R_{\text{CDE}}(T) + C_{\mathcal{F}} \cdot \log(C_{\mathcal{F}} T)) \\
& = C_{\mathcal{F}} \log V \cdot \frac{\beta_{\text{off}} T}{N} + N \cdot (R_{\text{CDE}}(T) + C_{\mathcal{F}} \cdot \log(VC_{\mathcal{F}} T)). \tag{28}
\end{aligned}$$

By choosing $N = \sqrt{\frac{C_{\mathcal{F}} \beta_{\text{off}} T \cdot \log V}{R_{\text{CDE}}(T) + C_{\mathcal{F}} \cdot \log(VC_{\mathcal{F}} T)}} \vee 1$, we can bound the expression in (28) as

$$\begin{aligned}
\mathbb{E}[\text{Est}_{\mathcal{H}}(T, \beta_{\text{off}})] & \lesssim \sqrt{C_{\mathcal{F}} \beta_{\text{off}} T (R_{\text{CDE}}(T) + C_{\mathcal{F}} \cdot \log(VC_{\mathcal{F}} T)) \log V} + R_{\text{CDE}}(T) + C_{\mathcal{F}} \cdot \log(VC_{\mathcal{F}} T) \\
& \leq \sqrt{C_{\mathcal{F}} \beta_{\text{off}} T R_{\text{CDE}}(T) \log V} + C_{\mathcal{F}} \sqrt{\beta_{\text{off}} T \log(VC_{\mathcal{F}} T) \log V} \\
& \quad + R_{\text{CDE}}(T) + C_{\mathcal{F}} \cdot \log(VC_{\mathcal{F}} T).
\end{aligned}$$

Finally, under the assumption that $R_{\text{CDE}}(T) \leq C'_{\mathcal{F}} \log T$, the bound above can be further simplified as

$$\mathbb{E}[\text{Est}_{\mathcal{H}}(T, \beta_{\text{off}})] \lesssim (C_{\mathcal{F}} (C_{\mathcal{F}} + C'_{\mathcal{F}}) \beta_{\text{off}} \log V \log(VC_{\mathcal{F}} T))^{1/2} T^{1/2} + (C_{\mathcal{F}} + C'_{\mathcal{F}}) \log(VC_{\mathcal{F}} T).$$

□

H.2.2 Proofs for supporting lemmas

Proof of Lemma H.1. For Algorithm 6, denote the randomness of the sequence $(y^{1:T}, \mu^{1:T})$ under $f^* \in \mathcal{F}$ and $(\tilde{f}^{1:T})$ by \mathbb{P}^{f^*} and $\mathbb{P}^{\tilde{f}^{1:T}}$ respectively. The data generating process for $(x^{1:T}, y^{1:T})$ in the CDEwRO setting implies that

$$\mathbb{E}_{\mathbb{P}^{\tilde{f}^{1:T}}} \left[\sum_{t=1}^T \mathbb{E}_{\tilde{f}^t \sim \mu^t} [\mathsf{D}_{\mathcal{H}}^2(\tilde{f}^t(x^t), f^*(x^t))] \right] = \mathbb{E} \left[\sum_{t=1}^T \mathbb{E}_{\tilde{f}^t \sim \mu^t} [\mathsf{D}_{\mathcal{H}}^2(\tilde{f}^t(x^t), f^*(x^t))] \right].$$

By Donsker-Varadhan [46], we have that for all $\eta > 0$

$$\begin{aligned}
\frac{1}{\eta} \mathsf{D}_{\mathsf{KL}} \left(\mathbb{P}^{\tilde{f}^{1:T}} \parallel \mathbb{P}^{f^*} \right) & \geq \mathbb{E}_{\mathbb{P}^{\tilde{f}^{1:T}}} \left[\sum_{t=1}^T \mathbb{E}_{\tilde{f}^t \sim \mu^t} [\mathsf{D}_{\mathcal{H}}^2(\tilde{f}^t(x^t), f^*(x^t))] \right] \\
& \quad - \frac{1}{\eta} \log \mathbb{E}_{\mathbb{P}^{f^*}} \exp \left\{ \eta \sum_{t=1}^T \mathbb{E}_{\tilde{f}^t \sim \mu^t} [\mathsf{D}_{\mathcal{H}}^2(\tilde{f}^t(x^t), f^*(x^t))] \right\}. \tag{29}
\end{aligned}$$

For any random variables X, Y, Z , we denote by $\mathsf{D}_{\mathsf{KL}}(\mathbb{P}_X \parallel \mathbb{P}_Y \mid Z) = \mathbb{E}_Z[\mathsf{D}_{\mathsf{KL}}(\mathbb{P}_{X|Z} \parallel \mathbb{P}_{Y|Z})]$. We further note that by the chain rule for KL divergence,

$$\begin{aligned}
\mathsf{D}_{\mathsf{KL}} \left(\mathbb{P}^{\tilde{f}^{1:T}} \parallel \mathbb{P}^{f^*} \right) & = \sum_{t=1}^T \mathbb{E}_{\mathbb{P}^{\tilde{f}^{1:T}}} \left[\mathsf{D}_{\mathsf{KL}} \left(\mathbb{P}_{(x^t, y^t)}^{\tilde{f}^{1:T}} \parallel \mathbb{P}_{(x^t, y^t)}^{f^*} \mid x^{1:t-1}, y^{1:t-1} \right) \right] \\
& = \sum_{t=1}^T \mathsf{D}_{\mathsf{KL}} \left(\tilde{f}^t(x^t) \parallel f^*(x^t) \right),
\end{aligned}$$

where the second equality holds because y^t follows $\tilde{f}^t(x^t)$ and $f^*(x^t)$ respectively, and because the conditional distribution of x^t is identical under both laws due to the oblivious assumption of the

covariates in the setting of CDEwRO. Then by the relation between KL and Hellinger (Lemma A.10 of [27]) and that $1 \leq \log V$, we have

$$\sum_{t=1}^T D_{KL}\left(\tilde{f}^t(x^t) \parallel f^*(x^t)\right) \leq (2 + \log V) \cdot \sum_{t=1}^T D_H^2\left(\tilde{f}^t(x^t), f^*(x^t)\right) \leq 3 \log V \cdot \sum_{t=1}^T D_H^2\left(\tilde{f}^t(x^t), f^*(x^t)\right),$$

where the last inequality is by $V \geq e$. Combining all of the results so far and using (29), we have

$$\begin{aligned} \mathbb{E}\left[\sum_{t=1}^T \mathbb{E}_{\bar{f}^t \sim \mu^t} [D_H^2(\bar{f}^t(x^t), f^*(x^t))]\right] &\leq \frac{3 \log V}{\eta} \cdot \sum_{t=1}^T D_H^2\left(\tilde{f}^t(x^t), f^*(x^t)\right) \\ &\quad + \frac{1}{\eta} \log \mathbb{E}_{\mathbb{P}^{f^*}} \exp\left\{\eta \sum_{t=1}^T \mathbb{E}_{\bar{f}^t \sim \mu^t} [D_H^2(\bar{f}^t(x^t), f^*(x^t))]\right\}. \end{aligned}$$

To proceed, using that for any positive random variable X , $\mathbb{E}[X] = \int_0^\infty \mathbb{P}(X \geq t) dt$, we have

$$\begin{aligned} \frac{1}{\eta} \log \mathbb{E}_{\mathbb{P}^{f^*}} \exp\left\{\eta \sum_{t=1}^T \mathbb{E}_{\bar{f}^t \sim \mu^t} [D_H^2(\bar{f}^t(x^t), f^*(x^t))]\right\} \\ = R_{\text{CDE}}(T) + \frac{1}{\eta} \log \mathbb{E}_{\mathbb{P}^{f^*}} \exp\left\{\eta \left(\sum_{t=1}^T \mathbb{E}_{\bar{f}^t \sim \mu^t} [D_H^2(\bar{f}^t(x^t), f^*(x^t))] - R_{\text{CDE}}(T)\right)\right\} \\ \leq R_{\text{CDE}}(T) + \frac{1}{\eta} \log \mathbb{E}_{\mathbb{P}^{f^*}} \exp\left\{\eta \left(\sum_{t=1}^T \mathbb{E}_{\bar{f}^t \sim \mu^t} [D_H^2(\bar{f}^t(x^t), f^*(x^t))] - R_{\text{CDE}}(T)\right)_+\right\} \\ = R_{\text{CDE}}(T) + \frac{1}{\eta} \log \int_0^\infty \mathbb{P}^{f^*} \left(\left(\sum_{t=1}^T \mathbb{E}_{\bar{f}^t \sim \mu^t} [D_H^2(\bar{f}^t(x^t), f^*(x^t))] - R_{\text{CDE}}(T)\right)_+ \geq \frac{1}{\eta} \log t \right) dt. \end{aligned}$$

Recall the assumption $\sum_{t=1}^T D_H^2(\tilde{f}^t(x^t), f^*(x^t)) \leq \zeta$ and let $\eta = \frac{1}{C_{\mathcal{F}}}$. We have

$$\begin{aligned} \frac{3 \log V}{\eta} \cdot \sum_{t=1}^T D_H^2(\tilde{f}^t(x^t), f^*(x^t)) + \frac{1}{\eta} \log \mathbb{E}_{\mathbb{P}^{f^*}} \exp\left\{\eta \sum_{t=1}^T \mathbb{E}_{\bar{f}^t \sim \mu^t} [D_H^2(\bar{f}^t(x^t), f^*(x^t))]\right\} \\ \leq \frac{3 \log V}{\eta} \cdot \zeta + R_{\text{CDE}}(T) \\ + \frac{1}{\eta} \log \int_0^\infty \mathbb{P}^{f^*} \left(\left(\sum_{t=1}^T \mathbb{E}_{\bar{f}^t \sim \mu^t} [D_H^2(\bar{f}^t(x^t), f^*(x^t))] - R_{\text{CDE}}(T)\right)_+ \geq \frac{1}{\eta} \log t \right) dt \\ \leq 3C_{\mathcal{F}} \log V \cdot \zeta + R_{\text{CDE}}(T) + C_{\mathcal{F}} \log \left(1 + \int_1^{e^{C_{\mathcal{F}} \cdot T}} \frac{1}{t} dt\right) \\ \leq 3C_{\mathcal{F}} \log V \cdot \zeta + R_{\text{CDE}}(T) + 2C_{\mathcal{F}} \cdot \log(C_{\mathcal{F}} T). \end{aligned}$$

where the second inequality uses the assumption Eq. (4) on the algorithm. \square

Proof of Lemma H.2. Our result improves uses the proof technique from the adversarial-to-oblivious reduction in Lemma 11 of Gonen et al. [31], but improves the result by a $O(\log T)$ factor.

Consider the CDEwRP setting. Let $\mathbb{E}^t[\cdot] := \mathbb{E}[\cdot \mid x^{1:t-1}, \tilde{f}^{1:t-1}]$. Let $\mu^{s,t} := \mathbb{E}^t[\mathcal{A}_{\text{CDEwRO}}(y_1^{1,t}, \dots, y_1^{s-1,t}, x_1^{1,t}, \dots, x_1^{s-1,t})]$ for all $1 \leq s \leq t \leq T$ where the expectation is taken over all the random variables $y_1^{1,t}, \dots, y_1^{s-1,t}, x_1^{1,t}, \dots, x_1^{s-1,t}$.

Then by Bernstein's concentration inequality applied to μ^t (interpreted as an empirical approximation to $\mu^{t,t}$), conditioned on $x^{1:t-1}, \tilde{f}^{1:t-1}$, we have with probability at least $1 - \frac{\varepsilon}{2T}$, for all $x' \in \mathcal{X}$ and $f' \in \mathcal{F}$,

$$\mathbb{E}_{f \sim \mu^t} [D_H^2(f(x), f'(x))] \leq 2 \mathbb{E}_{f \sim \mu^t} [D_H^2(f(x), f'(x))] + \varepsilon/(2T). \quad (30)$$

For any fixed $t \in [T]$ and all t' such that $t \leq t' \leq T$, the different trajectories $x_1^{1:t'}, \dots, x_1^{t-1:t'}, y_1^{1:t'}, \dots, y_1^{t-1:t'}$ are i.i.d. conditioned on $x^{1:t-1}, \tilde{f}^{1:t-1}$. Thus, we have

$$\mathbb{E}[\mathbb{E}_{f \sim \mu^{t,t}}^t [\mathsf{D}_H^2(f(x^t), f^*(x^t))]] = \mathbb{E}[\mathbb{E}_{f \sim \mu^{t,t+1}}^{t+1} [\mathsf{D}_H^2(f(x^t), f^*(x^t))]] \quad (31)$$

$$= \dots \quad (32)$$

$$= \mathbb{E}[\mathbb{E}_{f \sim \mu^{t,T}}^T [\mathsf{D}_H^2(f(x^t), f^*(x^t))]]. \quad (33)$$

Finally, for Algorithm 7, we have by the guarantee of $\mathcal{A}_{\text{CDEwRO}}$,

$$\sum_{t=1}^T \mathbb{E}[\mathbb{E}_{f \sim \mu^{t,T}}^T [\mathsf{D}_H^2(f(x^t), f^*(x^t))]] \leq R_{\text{CDEwRO}}(T, \zeta).$$

Combining the three results above, we have

$$\begin{aligned} \sum_{t=1}^T \mathbb{E}[\mathbb{E}_{f \sim \mu^t}^t [\mathsf{D}_H^2(f(x^t), f^*(x^t))]] &\leq 2 \sum_{t=1}^T \mathbb{E}[\mathbb{E}_{f \sim \mu^{t,t}}^t [\mathsf{D}_H^2(f(x^t), f^*(x^t))]] + \varepsilon \\ &= 2 \sum_{t=1}^T \mathbb{E}[\mathbb{E}_{f \sim \mu^{t,T}}^T [\mathsf{D}_H^2(f(x^t), f^*(x^t))]] + \varepsilon \\ &\leq 2R_{\text{CDEwRO}}(T, \zeta) + \varepsilon, \end{aligned}$$

where the first equality is by Eq. (30), the second equality is from Eq. (31), and the final inequality is by the guarantee of $\mathcal{A}_{\text{CDEwRO}}$.

□

Proof of Lemma H.3. Note that Algorithm 8 is a variant of the reduction in Lemma G.3, specialized to squared Hellinger distance, and the proof here will use the same idea as Lemma G.3.

For each $i \in [N]$, let $\zeta_i = \sum_{j=1}^{T/N} \mathsf{D}_H^2(\tilde{f}^{i+N \cdot j}(x^{i+N \cdot j}), f^*(x^{i+N \cdot j}))$. Then by the guarantee of $\mathcal{A}_{\text{CDEwRP}}$, we have for all $i \in [N]$,

$$\sum_{j=1}^{T/N} \mathbb{E}_{\bar{f} \sim \mu^{i+N \cdot j}} [\mathsf{D}_H^2(\bar{f}(x^{i+N \cdot j}), f^*(x^{i+N \cdot j}))] \leq R_{\text{CDEwRP}}(T/N, \zeta_i, 1/N),$$

Summing up over all $i \in [N]$, we obtain

$$\begin{aligned} \sum_{t=1}^T \mathbb{E}_{\bar{f} \sim \mu^t} [\mathsf{D}_H^2(\bar{f}(x^t), f^*(x^t))] &= \sum_{i=1}^N \sum_{j=1}^{T/N} \mathbb{E}_{\bar{f} \sim \mu^{i+N \cdot j}} [\mathsf{D}_H^2(\bar{f}(x^{i+N \cdot j}), f^*(x^{i+N \cdot j}))] \\ &\leq \sum_{i=1}^N R_{\text{CDEwRP}}(T/N, \zeta_i, 1/N). \end{aligned}$$

By the assumption on $\sum_{t=1}^T \mathsf{D}_H^2(\tilde{f}^t(x^t), f^*(x^t))$, we have

$$\sum_{i=1}^N \zeta_i = \sum_{i=1}^N \sum_{j=1}^{T/N} \mathsf{D}_H^2(\tilde{f}^{i+N \cdot j}(x^{i+N \cdot j}), f^*(x^{i+N \cdot j})) = \sum_{t=1}^T \mathsf{D}_H^2(\tilde{f}^t(x^t), f^*(x^t)) \leq \zeta.$$

Finally, we conclude

$$\sum_{t=1}^T \mathbb{E}_{\bar{f} \sim \mu^t} [\mathsf{D}_H^2(\bar{f}(x^t), f^*(x^t))] \leq \sup_{\substack{N \\ \sum_{i=1}^N \zeta_i \leq \zeta}} \sum_{i=1}^N R_{\text{CDEwRP}}(T/N, \zeta_i, 1/N).$$

□

Proof of Lemma H.4. This reduction is arguably the most interesting one. This reduction is by noticing that averaging across the outputs of the offline oracle will generate reference parameters (although delayed) with small online estimation errors as shown later in Eq. (34). By the guarantee of $\mathcal{A}_{\text{ODEwDRP}}$, we have

$$\sum_{t=1}^T \mathbb{E}_{\bar{f} \sim \mu^t} [\mathsf{D}_H^2(\bar{f}(x^t), f^*(x^t))] \leq R_{\text{CDEwDRP}}(T, N, \zeta),$$

where ζ can be chosen to be any upper bound of $\sum_{t=1}^T \mathsf{D}_H^2(\tilde{f}^t(x^t), f^*(x^t))$ since it is unknown to the learner in the setup where we augment the sequence of $\hat{f}^1, \dots, \hat{f}^T$ by setting $\hat{f}^{T+s} = \hat{f}^T$ for all $s \in \mathbb{N}$ and define $\tilde{f}^t := \frac{1}{N} \sum_{i=t+1}^{t+N} \hat{f}^i$ for $t = T-N, T-N+1, \dots, T$. Furthermore, by the definition of \tilde{f}^t , we can obtain

$$\begin{aligned} \sum_{t=1}^T \mathsf{D}_H^2(\tilde{f}^t(x^t), f^*(x^t)) &= N + \frac{1}{N} \sum_{t=1}^{T-N} \sum_{i=t+1}^{t+N} \mathsf{D}_H^2(\bar{f}^i(x^t), f^*(x^t)) \\ &= N + \frac{1}{N} \sum_{t=2}^T \sum_{i < t} \mathsf{D}_H^2(\bar{f}^t(x^i), f^*(x^i)) \\ &\leq N + \frac{\beta_{\text{Off}} T}{N}. \end{aligned} \tag{34}$$

Thus, we conclude Algorithm 9 obtains

$$\sum_{t=1}^T \mathbb{E}_{\bar{f} \sim \mu^t} [\mathsf{D}_H^2(\bar{f}(x^t), f^*(x^t))] \leq R_{\text{CDEwDRP}}(T, N, N + \beta_{\text{Off}} T/N).$$

□

I Proofs from Appendix E

Theorem E.1 (Offline-to-online conversion under coverability). *For any layer-wise loss D^{RL} and MDP class $(\mathcal{M}, \Pi, \mathcal{O})$ and $M^* \in \mathcal{M}$, the sequence of estimators $(\widehat{M}^1, \dots, \widehat{M}^T)$ produced by any offline estimation oracle Alg_{Off} for D^{RL} with parameter β_{Off} satisfy*

$$\sum_{t=1}^T \mathsf{D}^{\text{RL}}(\widehat{M}^t(\pi^t) \| M^*(\pi^t)) \leq O\left(\sqrt{HC_{\text{cov}}(M^*)\beta_{\text{Off}} T \log T} + HC_{\text{cov}}(M^*)\right).$$

Proof of Theorem E.1. This proof closely follows the proof of Theorem 1 in Xie et al. [69]. Define the shorthand $d_h^t(s, a) \equiv d_h^{\pi^t}(s, a)$ and $C_{\text{cov}} = C_{\text{cov}}(M^*)$, and define

$$\tilde{d}_h^t(s, a) := \sum_{\tau=1}^{t-1} d_h^{\tau}(s, a), \quad \text{and} \quad \mu_h^* := \arg \min_{\mu_h \in \Delta(\mathcal{S} \times \mathcal{A})} \sup_{\pi \in \Pi} \left\| \frac{d_h^{\pi}}{\mu_h} \right\|_{\infty}.$$

From the definition of the layer-wise loss, we have

$$\sum_{t=1}^T \mathsf{D}^{\text{RL}}(\widehat{M}^t(\pi^t) \| M^*(\pi^t)) = \sum_{h=1}^H \sum_{t=1}^T \sum_{(s,a) \in \mathcal{S} \times \mathcal{A}} d_h^t(s, a) \mathsf{D}_h\left(\bar{P}_h^{M^*}(s, a) \| \bar{P}_h^{\widehat{M}^t}(s, a)\right).$$

We define a ‘‘burn-in’’ phase for each state-action pair $(s, a) \in \mathcal{S} \times \mathcal{A}$ by defining

$$\tau_h(s, a) = \min \left\{ t \mid \tilde{d}_h^t(s, a) \geq C_{\text{cov}} \cdot \mu_h^*(s, a) \right\}.$$

Let $h \in [H]$ be With this definition, for we can write

$$\begin{aligned} & \sum_{t=1}^T \sum_{(s,a) \in \mathcal{S} \times \mathcal{A}} d_h^t(s,a) \mathsf{D}_h \left(\bar{P}_h^{M^*}(s,a) \| \bar{P}_h^{\bar{M}^t}(s,a) \right) \\ &= \sum_{(s,a) \in \mathcal{S} \times \mathcal{A}} \sum_{t < \tau_h(s,a)} d_h^t(s,a) \mathsf{D}_h \left(\bar{P}_h^{M^*}(s,a) \| \bar{P}_h^{\bar{M}^t}(s,a) \right) + \sum_{(s,a) \in \mathcal{S} \times \mathcal{A}} \sum_{t \geq \tau_h(s,a)} d_h^t(s,a) \mathsf{D}_h \left(\bar{P}_h^{M^*}(s,a) \| \bar{P}_h^{\bar{M}^t}(s,a) \right) \end{aligned}$$

For the first term above, we have

$$\sum_{(s,a) \in \mathcal{S} \times \mathcal{A}} \sum_{t < \tau_h(s,a)} d_h^t(s,a) \mathsf{D}_h \left(\bar{P}_h^{M^*}(s,a) \| \bar{P}_h^{\bar{M}^t}(s,a) \right) \leq \sum_{(s,a) \in \mathcal{S} \times \mathcal{A}} \tilde{d}_h^{\tau_h(s,a)}(s,a) \leq 2C_{\text{cov}} \sum_{(s,a) \in \mathcal{S} \times \mathcal{A}} \mu_h^*(s,a) = 2C_{\text{cov}},$$

where the last inequality holds because

$$\sum_{(s,a) \in \mathcal{S} \times \mathcal{A}} \tilde{d}_h^{\tau_h(s,a)}(s,a) = \sum_{(s,a) \in \mathcal{S} \times \mathcal{A}} \tilde{d}_h^{\tau_h(s,a)-1}(s,a) + d_h^{\tau_h(s,a)}(s,a) \leq 2C_{\text{cov}} \cdot \mu_h^*(s,a).$$

The remaining term is

$$\begin{aligned} & \sum_{h=1}^H \sum_{(s,a) \in \mathcal{S} \times \mathcal{A}} \sum_{t \geq \tau_h(s,a)} d_h^t(s,a) \mathsf{D}_h \left(\bar{P}_h^{M^*}(s,a) \| \bar{P}_h^{\bar{M}^t}(s,a) \right) \\ &= \sum_{h=1}^H \sum_{t=1}^T \sum_{(s,a) \in \mathcal{S} \times \mathcal{A}} d_h^t(s,a) \left(\frac{\tilde{d}_h^t(s,a)}{\tilde{d}_h^t(s,a)} \right)^{1/2} \mathbb{1}\{t \geq \tau_h(s,a)\} \mathsf{D}_h \left(\bar{P}_h^{M^*}(s,a) \| \bar{P}_h^{\bar{M}^t}(s,a) \right) \\ &\leq \sqrt{\sum_{h=1}^H \sum_{t=1}^T \sum_{(s,a) \in \mathcal{S} \times \mathcal{A}} \frac{(\mathbb{1}(t \geq \tau_h(s,a)) d_h^t(s,a))^2}{\tilde{d}_h^t(s,a)}} \cdot \sqrt{\sum_{h=1}^H \sum_{t=1}^T \sum_{(s,a) \in \mathcal{S} \times \mathcal{A}} \tilde{d}_h^t(s,a) \mathsf{D}_h \left(\bar{P}_h^{M^*}(s,a) \| \bar{P}_h^{\bar{M}^t}(s,a) \right)}. \end{aligned}$$

Following the derivation in Theorem 1 of Xie et al. [69], we have

$$\begin{aligned} & \sum_{t=1}^T \sum_{(s,a) \in \mathcal{S} \times \mathcal{A}} \frac{(\mathbb{1}(t \geq \tau_h(s,a)) d_h^t(s,a))^2}{\tilde{d}_h^t(s,a)} \leq 2 \sum_{t=1}^T \sum_{(s,a) \in \mathcal{S} \times \mathcal{A}} \frac{d_h^t(s,a) \cdot d_h^t(s,a)}{\tilde{d}_h^t(s,a) + C_{\text{cov}} \cdot \mu_h^*(s,a)} \\ &\leq 2 \sum_{t=1}^T \sum_{(s,a) \in \mathcal{S} \times \mathcal{A}} \max_{t' \in [T]} d_h^{t'}(s,a) \cdot \frac{d_h^t(s,a)}{\tilde{d}_h^t(s,a) + C_{\text{cov}} \cdot \mu_h^*(s,a)} \\ &\leq 2 \left(\max_{s,a} \sum_{t=1}^T \frac{d_h^t(s,a)}{\tilde{d}_h^t(s,a) + C_{\text{cov}} \cdot \mu_h^*(s,a)} \right) \cdot \left(\sum_{(s,a) \in \mathcal{S} \times \mathcal{A}} \max_{t \in [T]} d_h^t(s,a) \right) \\ &\lesssim C_{\text{cov}} \log T, \end{aligned}$$

where the last inequality follows from Lemmas 3 and 4 of Xie et al. [69]. For the second term, as a consequence of the offline estimation assumption, we have that for all $t \in [T]$,

$$\sum_{h=1}^H \sum_{(s,a) \in \mathcal{S} \times \mathcal{A}} \tilde{d}_h^t(s,a) \mathsf{D}_h \left(\bar{P}_h^{M^*}(s,a) \| \bar{P}_h^{\bar{M}^t}(s,a) \right) = \sum_{s=1}^{t-1} \mathsf{D}^{\text{RL}} \left(\widehat{M}^t(\pi^s) \| M^*(\pi^s) \right) \leq \beta_{\text{Off}}.$$

Altogether, we conclude that

$$\sum_{t=1}^T \mathsf{D}^{\text{RL}} \left(\widehat{M}^t(\pi^t) \| M^*(\pi^t) \right) \leq O \left(\sqrt{HC_{\text{cov}} \beta_{\text{Off}} T \log T} + HC_{\text{cov}} \right).$$

□

Corollary E.2. For any MDP class $(\mathcal{M}, \Pi, \mathcal{O})$ and $M^* \in \mathcal{M}$, the sequence of estimators $(\widehat{M}^1, \dots, \widehat{M}^T)$ produced by any offline estimation oracle Alg_{Off} for squared Hellinger distance D_H^2 with parameter β_{Off} satisfy

$$\text{Est}_{\text{H}}^{\text{On}}(T) = \sum_{t=1}^T D_H^2(\widehat{M}^t(\pi^t), M^*(\pi^t)) \leq O\left(H\sqrt{C_{\text{cov}}(M^*)\beta_{\text{Off}}T\log T} + H^2C_{\text{cov}}(M^*)\right)$$

Proof of Corollary E.2. By Lemma F.2, for any two MDP models M and M' and any $\pi \in \Delta(\Pi_{\text{RNS}})$, we have

$$D_H^2(M(\pi), M'(\pi)) = D_H^2(M'(\pi), M(\pi)) \leq 7 \cdot \sum_{h=1}^H \mathbb{E}^{M', \pi} \left[D_H^2(\bar{P}_h^{M'}(s_h, a_h), \bar{P}_h^M(s_h, a_h)) \right]. \quad (35)$$

On the other hand, for any $h \in [H]$, we have from Lemma A.9 of Foster et al. [27] that

$$\mathbb{E}^{M', \pi} \left[D_H^2(\bar{P}_h^{M'}(s_h, a_h), \bar{P}_h^M(s_h, a_h)) \right] \leq 4D_H^2(M(\pi), M'(\pi)),$$

by choosing $X = (s_h, a_h)$ and $Y = (r_h, s_{h+1})$ in the aforementioned lemma. Summing up over h , we conclude that

$$\sum_{h=1}^H \mathbb{E}^{M', \pi} \left[D_H^2(\bar{P}_h^{M'}(s_h, a_h), \bar{P}_h^M(s_h, a_h)) \right] \leq 4H \cdot D_H^2(M(\pi), M'(\pi)). \quad (36)$$

Consider any sequence of policies π^1, \dots, π^T and outputs $\widehat{M}^1, \dots, \widehat{M}^T$ from an offline oracle with parameter β_{Off} for squared Hellinger distance. By Eq. (36), we have that for all $t \in [T]$,

$$\sum_{\tau=1}^{t-1} \sum_{h=1}^H \mathbb{E}^{M^*, \pi^\tau} \left[D_H^2(\bar{P}_h^{M^*}(s, a), \bar{P}_h^{\widehat{M}^\tau}(s, a)) \right] \leq 4H \sum_{\tau=1}^{t-1} D_H^2(\widehat{M}^\tau(\pi^\tau), M^*(\pi^\tau)) \leq 4\beta_{\text{Off}}H.$$

Theorem E.1 thus implies that

$$\sum_{t=1}^T \sum_{h=1}^H \mathbb{E}^{M^*, \pi^t} \left[D_H^2(\bar{P}_h^{M^*}(s, a), \bar{P}_h^{\widehat{M}^t}(s, a)) \right] \lesssim H\sqrt{C_{\text{cov}}\beta_{\text{Off}}T\log T} + H^2C_{\text{cov}}.$$

Finally, using Eq. (35), we can convert the inequality above into a bound on the square Hellinger distance:

$$\begin{aligned} \sum_{t=1}^T D_H^2(\widehat{M}^t(\pi^t), M^*(\pi^t)) &\lesssim \sum_{t=1}^T \sum_{h=1}^H \mathbb{E}^{M^*, \pi^t} \left[D_H^2(\bar{P}_h^{M^*}(s, a), \bar{P}_h^{\widehat{M}^t}(s, a)) \right] \\ &\lesssim H\sqrt{C_{\text{cov}}\beta_{\text{Off}}T\log T} + H^2C_{\text{cov}}. \end{aligned}$$

□

Proposition E.2 (Tightness of offline-to-online conversion). For any integer $T \geq 1$ and $\beta_{\text{Off}} > 0$, there exists a contextual bandit class $(\mathcal{M}, \Pi = \mathcal{A}^{\mathcal{S}}, \mathcal{O})$ with $|\mathcal{A}| = 2$, a distribution $d_1 \in \Delta(\mathcal{S})$, a sequence (π^1, \dots, π^T) and an offline oracle Alg_{Off} for D_{CB} with parameter β_{Off} such that the oracle's outputs $(\widehat{M}^1, \dots, \widehat{M}^T)$ satisfy

$$\text{Est}_{\text{D}}^{\text{On}}(T) = \sum_{t=1}^T D_{\text{CB}}(\widehat{M}^t(\pi^t), M^*(\pi^t)) \geq \Omega\left(\sqrt{T\beta_{\text{Off}}}\right).$$

Proof of Proposition E.2. Let $\beta_{\text{Off}} > 0$ and T be given, and let $N \in \mathbb{N}$ be chosen such that $T = N \cdot \lfloor N\beta_{\text{Off}} \rfloor$; we assume without loss of generality that N is large enough such that $N\beta_{\text{Off}} > 1$, which implies that $\lfloor N\beta_{\text{Off}} \rfloor \geq N\beta_{\text{Off}}/2$. Define $\mathcal{A} = \{a_0, a_1\}$, $\mathcal{S} = \{s_0, \dots, s_{N-1}\}$, and $g^{M^*}(s, a) \equiv 0$

for all $s \in \mathcal{S}, a \in \mathcal{A}$. Let $d_1 = \text{Unif}(\mathcal{S})$ be the context distribution. For any $t \in \{1, \dots, T\}$, consider the deterministic policy $\pi^t : \mathcal{S} \rightarrow \mathcal{A}$ and the offline estimator \widehat{M}^t defined via

$$\pi^t(s_n) = \begin{cases} a_1, & \text{if } n = \lfloor t/\lfloor N\beta_{\text{Off}} \rfloor \rfloor, \\ a_0, & \text{otherwise.} \end{cases} \quad \text{and} \quad g^{\widehat{M}^t}(s, a) = \begin{cases} 1, & \text{if } s = s_{\lfloor t/\lfloor N\beta_{\text{Off}} \rfloor \rfloor}, a = a_1, \\ 0, & \text{otherwise.} \end{cases}$$

We first verify that $\widehat{M}^1, \dots, \widehat{M}^T$ satisfy the offline oracle requirement. For any $t \in \{1, \dots, T\}$, we have

$$\begin{aligned} & \sum_{\tau=1}^{t-1} \mathbb{E}_{s \sim d_1} [\mathcal{D}_{\text{sq}}(g^M(s, \pi^\tau(s)), g^{\widehat{M}^\tau}(s, \pi^\tau(s)))] \\ &= \frac{1}{N} \sum_{\tau=1}^{t-1} \mathbb{1}\{\pi^\tau(s_{\lfloor t/N \rfloor}) = a_1\} \\ &= \frac{1}{N} \cdot (t - \lfloor t/\lfloor N\beta_{\text{Off}} \rfloor \rfloor \cdot \lfloor N\beta_{\text{Off}} \rfloor) \leq \beta_{\text{Off}}. \end{aligned}$$

However, the online error is

$$\sum_{t=1}^T \mathbb{E}_{s \sim d_1} [\mathcal{D}_{\text{sq}}(g^M(s, \pi^t(s)), g^{\widehat{M}^t}(s, \pi^t(s)))] = \sum_{t=1}^T \frac{1}{N} = \lfloor N\beta_{\text{Off}} \rfloor \geq \frac{1}{2} \sqrt{T\beta_{\text{Off}}}.$$

□

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer **[Yes]** , **[No]** , or **[NA]** .
- **[NA]** means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "**[Yes]**" is generally preferable to "**[No]**", it is perfectly acceptable to answer "**[No]**" provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "**[No]**" or "**[NA]**" is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer **[Yes]** to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading “NeurIPS paper checklist”,**
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers.**

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 claims made in the abstract and introduction accurately reflect the paper’s contributions and scope. The claims are validated by detailed proofs.

Guidelines:

- The answer NA 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 NA 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: See Section 1.2.

Guidelines:

- The answer NA 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: The paper provides detailed assumptions and proofs.

Guidelines:

- The answer NA 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: [NA] .

Justification: This paper does not include experiments.

Guidelines:

- The answer NA 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: [NA]

Justification: This paper does not include experiments.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.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://nips.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, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [NA] .

Justification: This paper does not include experiments.

Guidelines:

- The answer NA 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: [NA] .

Justification: This paper does not include experiments.

Guidelines:

- The answer NA 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: [NA].

Justification: This paper does not include experiments.

Guidelines:

- The answer NA 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, in every respect, with the NeurIPS Code of Ethics

Guidelines:

- The answer NA 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: [NA]

Justification: This is a theoretical work. There is no societal impact on the work performed.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA 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., pretrained language models, image generators, or scraped datasets)?

Answer: [NA] .

Justification: The paper poses no such risks.

Guidelines:

- The answer NA 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: [NA] .

Justification: This paper does not use existing assets.

Guidelines:

- The answer NA 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: [NA].

Justification: This paper does not release new assets.

Guidelines:

- The answer NA 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: [NA].

Justification: This paper does not involve crowdsourcing or research with human subjects.

Guidelines:

- The answer NA 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: [NA].

Justification: This paper does not involve crowdsourcing or research with human subjects.

Guidelines:

- The answer NA 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.