Avoiding Catastrophe in Online Learning by Asking for Help

Benjamin Plaut [1] **Hanlin Zhu** [1] **Stuart Russell** [1]

Abstract

Most learning algorithms with formal regret guarantees assume that all mistakes are recoverable and essentially rely on trying all possible behaviors. This approach is problematic when some mistakes are *catastrophic*, i.e., irreparable. We propose an online learning problem where the goal is to minimize the chance of catastrophe. Specifically, we assume that the payoff in each round represents the chance of avoiding catastrophe in that round and try to maximize the product of payoffs (the overall chance of avoiding catastrophe) while allowing a limited number of queries to a mentor. We also assume that the agent can transfer knowledge between similar inputs. We first show that in general, any algorithm either queries the mentor at a linear rate or is nearly guaranteed to cause catastrophe. However, in settings where the mentor policy class is learnable in the standard online model, we provide an algorithm whose regret and rate of querying the mentor both approach 0 as the time horizon grows. Although our focus is the product of payoffs, we provide matching bounds for the typical additive regret. Conceptually, if a policy class is learnable in the absence of catastrophic risk, it is learnable in the presence of catastrophic risk if the agent can ask for help.

1. Introduction

There has been mounting concern over catastrophic risk from AI, including but not limited to autonomous weapon accidents (Abaimov & Martellini, 2020), bioterrorism (Mouton et al., 2024), cyberattacks on critical infrastructure (Guembe et al., 2022), and loss of control (Bengio et al., 2024). See Critch & Russell (2023) and Hendrycks et al. (2023) for taxonomies of societal-scale AI risks. In this paper, we use "catastrophe" to refer to any kind of irreparable harm. In addition to the large-scale risks above, our definition also covers smaller-scale (yet still unacceptable) incidents such as serious medical errors (Rajpurkar et al., 2022), crashing a robotic vehicle (Kohli & Chadha, 2020), and discriminatory sentencing (Villasenor & Foggo, 2020).

The gravity of these risks contrasts starkly with the dearth of theoretical understanding of how to avoid them. Nearly all of learning theory explicitly or implicitly assumes that no single mistake is too costly. We focus on *online learning*, where an agent repeatedly interacts with an unknown environment and uses its observations to gradually improve its performance. Most online learning algorithms essentially try all possible behaviors and see what works well. We do not want autonomous weapons or surgical robots to try all possible behaviors.

More precisely, trial-and-error-style algorithms only work when catastrophe is assumed to be impossible. This assumption can take multiple forms, such as that the agent's actions do not affect future inputs (e.g., Slivkins, 2011), that no action has irreversible effects (e.g., Jaksch et al., 2010) or that the environment is reset at the start of each "episode" (e.g., Azar et al., 2017). One could train an agent entirely in a controlled lab setting where one of those assumptions does hold, but we argue that sufficiently general agents will inevitably encounter novel scenarios when deployed in the real world. Machine learning models often behave unpredictably in unfamiliar environments (see, e.g., Quiñonero-Candela et al., 2022), and we do not want AI biologists or robotic vehicles to behave unpredictably.

The goal of this paper is to understand the conditions under which it is possible to formally guarantee avoidance of catastrophe in online learning. Certainly some conditions are necessary, because the problem is hopeless if the agent must rely purely on trial-and-error: any untried action could lead to paradise or disaster and the agent has no way to predict which. In the real world, however, one need not learn through pure trial-and-error: one can also ask for help. We think it is critical for high-stakes AI applications to employ a designated supervisor who can be asked for help. Examples include a human doctor supervising AI doctors, a robotic vehicle with a human driver who can take over in emergencies, autonomous weapons with a human operator, and many more. We hope that our work constitutes a step in the

[1] University of California, Berkeley. Correspondence to: Benjamin Plaut <plaut@berkeley.edu>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).

direction of practical safety guarantees for such applications.

1.1. Our model

We propose an online learning model of avoiding catastrophe with mentor help. On each time step, the agent observes an input, selects an action or queries the mentor, and obtains a payoff. Each payoff represents the probability of avoiding catastrophe on that time step (conditioned on no prior catastrophe). The agent's goal is to maximize the *product* of payoffs, which is equal to the overall probability of avoiding catastrophe.[1] As is standard in online learning, we consider the product of payoffs obtained while learning, not the product of payoffs of some final policy.

The (possibly suboptimal) mentor has a stationary policy, and when queried, the mentor illustrates their policy's action for the current input. We want the agent's regret – defined as the gap between the mentor's performance and the agent's performance – to go to zero as the time horizon T grows. In other words, with enough time, the agent should avoid catastrophe nearly as well as the mentor. We also expect the agent to become self-sufficient over time: the number of queries to the mentor should be sublinear in T, or equivalently, the rate of querying the mentor should go to zero.

1.2. Our assumptions

The agent needs some way to make inferences about unqueried inputs in order to decide when to ask for help. Much past work has used Bayesian inference, which suffers from tractability issues in complex environments.[2] We instead assume that the mentor policy satisfies a novel property that we call *local generalization*: informally, if the mentor told us that an action was safe for a similar input, then that action is probably also safe for the current input. For example, if it is safe to ignore a 3 mm spot on an X-ray, it is likely (but not certainly) also safe to ignore a 3.1 mm spot with the same density, location, etc. Unlike Bayesian inference, local generalization only requires computing distances and is compatible with any input space that admits a distance metric. See Section 5.2 for further discussion of local generalization.

Unlike the standard online learning model, we assume that the agent does not observe payoffs. This is because the payoff in our model represents the chance of avoiding catastrophe on that time step. In the real world, one only observes whether catastrophe occurred, not its probability.[3]

Table 1: Comparison between the standard online learning model and our model.

	Standard model	Our model
Objective	Sum of payoffs	Product of payoffs
Regret goal	Sublinear	Subconstant
Feedback	Every time step	Only from queries
Mentor	No	Yes
Local generalization	No	Yes

1.3. Standard online learning

To properly understand our results, it is important to understand standard online learning. In the standard model, the agent observes an input on each time step and must choose an action. An adversary then reveals the correct action, which results in some payoff to the agent. The goal is sublinear regret with respect to the sum of payoffs, or equivalently, the average regret per time step should go to 0 as $T \to \infty$. Table 1 delineates the precise differences between the standard model and our model.

If the adversary's choices are unconstrained, the problem is hopeless: if the adversary determines the correct action on each time step randomly and independently, the agent can do no better than random guessing. However, sublinear regret becomes possible if (1) the hypothesis class has finite Littlestone dimension (Littlestone, 1988), or (2) the hypothesis class has finite VC dimension (Vapnik & Chervonenkis, 1971) and the input is *σ-smooth*[4] (Haghtalab et al., 2024).

The goal of sublinear regret in online learning implicitly assumes catastrophe is impossible: the agent can make arbitrarily many (and arbitrarily costly) mistakes as long as the *average* regret per time step goes to 0. In contrast, we demand *subconstant* regret: the *total* probability of catastrophe should go to 0. Furthermore, standard online learning allows the agent to observe payoffs on every time step, while our agent only receives feedback on time steps with queries. However, the combination of a mentor and local generalization allows our agent to learn without trying actions directly, which is enough to offset all of the above disadvantages.

1.4. Our results

At a high level, we show that avoiding catastrophe with the help of a mentor and local generalization is no harder than online learning without catastrophic risk.

We first show that in general, any algorithm with sublinear queries to the mentor has unbounded regret in the worst-case (Theorem 4.1). As a corollary, even when the mentor

[1] Conditioning on no prior catastrophe means we do not need to assume that these probabilities are independent (and if catastrophe has already occurred, this time step does not matter). This is due to the chain rule of probability.

[2] For the curious reader, Betancourt (2018) provides a thorough treatment. See also Section 2.

[3] One may be able to detect "close calls" in some cases, but observing the precise probability seems unrealistic.

[4] Informally, the adversary chooses a distribution over inputs instead of a precise input. See Section 3 for the formal definition.

can avoid catastrophe with certainty, any algorithm either needs extensive supervision or is nearly guaranteed to cause catastrophe (Corollary 4.1.1).

Our primary result is a simple algorithm whose total regret and rate of querying the mentor both go to 0 as $T \to \infty$ when either (1) the mentor policy class has finite Littlestone dimension or (2) the mentor policy class has finite VC dimension and the input sequence is σ-smooth (Theorem 5.2). Conceptually, the algorithm has two components: (1) for "in-distribution" inputs, run a standard online learning algorithm (adjusted to account for only receiving feedback in response to queries), and (2) for "out-of-distribution" inputs, ask for help. Our algorithm can handle an unbounded input space and does not need to know the local generalization constant.

Although we focus on the product of payoffs, we show that the results above (both positive and negative) hold for the typical additive regret as well. In fact, we show that multiplicative regret and additive regret are tightly related in our setting (Lemma A.1).

In summary, the combination of local generalization and a mentor allows us to reduce the regret by an entire factor of T, resulting in subconstant regret (multiplicative or additive) instead of the typical sublinear regret.

2. Related work

Learning with irreversible costs. Despite the ubiquity of irreversible costs in the real world, theoretical work on this topic remains limited. This may be due to the fundamental modeling question of how the agent should learn about novel inputs or actions without trying them directly.

The most common approach is to allow the agent to ask for help. This alone is insufficient, however: the agent must have some way to decide *when* to ask for help. A popular solution is to perform Bayesian inference on the world model, but this has two tricky requirements: (1) a prior distribution which contains the true world model (or an approximation), and (2) an environment where computing (or approximating) the posterior is tractable. A finite set of possible environments satisfies both conditions but is unrealistic in many real-world scenarios. In contrast, our algorithm can handle an uncountable policy class and a continuous unbounded input space, which is crucial for many real-world scenarios in which one never sees the exact same input twice.

Bayesian inference combined with asking for help is studied by Cohen et al. (2021); Cohen & Hutter (2020); Kosoy (2019); Mindermann et al. (2018). We also mention Hadfield-Menell et al. (2017); Moldovan & Abbeel (2012); Turchetta et al. (2016), who utilize Bayesian inference in the context of safe (online) reinforcement learning without asking for help (and without regret bounds).

We are only aware of two papers that theoretically address irreversibility without Bayesian inference: Grinsztajn et al. (2021) and Maillard et al. (2019). The former proposes to sample trajectories and learn reversibility based on temporal consistency between states: intuitively, if s_1 always precedes s_2, we can infer that s_1 is unreachable from s_2. Although the paper theoretically grounds this intuition, there is no formal regret guarantee. The latter presents an algorithm which asks for help in the form of rollouts from the current state. However, the regret bound and number of rollouts are both linear in the worst case, due to the dependence on the γ^* parameter which roughly captures how bad an irreversible action can be. In contrast, our algorithm achieves good regret even when actions are maximally bad.

To our knowledge, we are the first to provide an algorithm which formally guarantees avoidance of catastrophe (with high probability) without Bayesian inference. We are also not aware of prior results comparable to our negative result, including in the Bayesian regime.

Safe reinforcement learning (RL). The safe RL problem is typically formulated as a constrained Markov Decision Process (CMDP) (Altman, 2021). In CMDPs, the agent must maximize reward while also satisfying safety constraints. See Gu et al. (2024); Zhao et al. (2023); Wachi et al. (2024) for surveys. The two most relevant safe RL papers are Liu et al. (2021) and Stradi et al. (2024), both of which provide algorithms guaranteed to satisfy initially unknown safety constraints. Since neither paper allows external help, they require strong assumptions to make the problem tractable: the aforementioned results assume that the agent (1) knows a strictly safe policy upfront (i.e., a policy which satisfies the safety constraints with slack), (2) is periodically reset, and (3) observes the safety costs. In contrast, our agent has no prior knowledge, is never reset, and never observes payoffs.

Online learning. See Cesa-Bianchi & Lugosi (2006) and Chapter 21 of Shalev-Shwartz & Ben-David (2014) for introductions to online learning. A classical result states that sublinear regret is possible if and only if the hypothesis class has finite Littlestone dimension (Littlestone, 1988). However, even some simple hypothesis classes have infinite Littlestone dimension, such as the class of thresholds on $[0, 1]$ (Example 21.4 in Shalev-Shwartz & Ben-David, 2014). Recently, Haghtalab et al. (2024) showed that if the adversary only chooses a distribution over inputs rather than the precise input, only finite VC dimension (Vapnik & Chervonenkis, 1971) is needed for sublinear regret. Specifically, they assume that each input is sampled from a distribution whose concentration is upper bounded by $\frac{1}{\sigma}$ times the uniform distribution. This framework is known as *smoothed analysis*, originally due to Spielman & Teng (2004).

Multiplicative objectives. Although online learning traditionally studies the sum of payoffs, there is some work which aims to maximize the product of payoffs (or equivalently, the sum of logarithms). See, e.g., Chapter 9 of Cesa-Bianchi & Lugosi (2006). However, these regret bounds are still sublinear in T, in comparison to our subconstant regret bounds. Also, like most online learning work, those results assume that payoffs are observed on every time step. In contrast, our agent only receives feedback in response to queries (Table 1) and never observes payoffs. Barman et al. (2023) studied a multiplicative objective in a multi-armed bandit context, but their objective is the geometric mean of payoffs instead of the product. Interpreted in our context, their regret bounds imply that the *average* chance of catastrophe goes to zero, while we guarantee that the *total* chance of catastrophe goes to zero. This distinction is closely related to the difference between subconstant and sublinear regret.

Active learning and imitation learning. Our assumption that the agent only receives feedback in response to queries falls under the umbrella of active learning (Hanneke, 2014). This contrasts with passive learning, where the agent receives feedback automatically. The way our agent learns from the mentor is also reminiscent of imitation learning (Osa et al., 2018). Although ideas from these areas could be useful in our setting, we are not aware of any results from that literature which account for irreversible costs.

3. Model

Inputs. Let $\mathbb{N}$ denote the strictly positive integers and let $T \in \mathbb{N}$ be the time horizon. Let $\boldsymbol{x} = (x_1, x_2, \dots, x_T) \in \mathcal{X}^T$ be the sequence of inputs. In the fully adversarial setting, each x_t can have arbitrary (possibly randomized) dependence on the events of prior time steps. In the smoothed setting, the adversary only chooses the distribution $\mathcal{D}_t$ from which x_t is sampled. Formally, a distribution $\mathcal{D}$ over $\mathcal{X}$ is σ-smooth if for any $S \subseteq \mathcal{X}$, $\mathcal{D}(S) \le \frac{1}{\sigma} U(S)$. (In the smoothed setting, we assume that $\mathcal{X}$ supports a uniform distribution U.[5]) If each x_t is sampled from a σ-smooth $\mathcal{D}_t$, we say that $\boldsymbol{x}$ is σ-smooth. The sequence $\boldsymbol{\mathcal{D}} = \mathcal{D}_1, \dots, \mathcal{D}_T$ can still be adaptive, i.e., the choice of $\mathcal{D}_t$ can depend on the events of prior time steps.

Actions and payoffs. Let $\mathcal{Y}$ be a finite set of actions. There also exists a special action $\tilde{y}$ which corresponds to querying the mentor. For $k \in \mathbb{N}$, let $[k] = \{1, 2, \dots, k\}$. On each time step $t \in [T]$, the agent must select action $y_t \in \mathcal{Y} \cup \{\tilde{y}\}$, which generates a payoff. Let $\boldsymbol{y} = (y_1, \dots, y_T)$. We allow the payoff function to vary between time steps: let $\boldsymbol{\mu} = (\mu_1, \dots, \mu_T) \in (\mathcal{X} \times \mathcal{Y} \to [0,1])^T$ be the sequence of payoff functions. Then $\mu_t(x_t, y_t) \in [0,1]$ is the agent's payoff at time t. Like $\boldsymbol{\mathcal{D}}$, we allow $\boldsymbol{\mu}$ to be adaptive. Unless otherwise noted, all expectations are over any randomization in the agent's decisions, any randomization in $\boldsymbol{x}$, and any randomization in the adaptive choice of $\boldsymbol{\mu}$.

Asking for help. The mentor is endowed with a (possibly suboptimal) policy $\pi^m : \mathcal{X} \to \mathcal{Y}$. When action $\tilde{y}$ is chosen, the mentor informs the agent of the action $\pi^m(x_t)$ and the agent obtains payoff $\mu_t(x_t, \pi^m(x_t))$. For brevity, let $\mu_t^m(x) = \mu_t(x, \pi^m(x))$. The agent never observes payoffs: the only way to learn about $\boldsymbol{\mu}$ is by querying the mentor.

We would like an algorithm which becomes "self-sufficient" over time: the rate of querying the mentor should go to 0 as $T \to \infty$, or equivalently, the cumulative number of queries should be sublinear in T. Formally, let $Q_T(\boldsymbol{\mu}, \pi^m) = \{t \in [T] : y_t = \tilde{y}\}$ be the random variable denoting the set of time steps with queries. Then we say that the (expected) number of queries is sublinear in T if $\sup_{\boldsymbol{\mu}, \pi^m} \mathbb{E}[|Q_T(\boldsymbol{\mu}, \pi^m)|] \in o(T)$. In other words, there must exist $g : \mathbb{N} \to \mathbb{N}$ which does not depend on $\boldsymbol{\mu}$ or π^m such that $g(T) \in o(T)$ and $\sup_{\boldsymbol{\mu}, \pi^m} \mathbb{E}[|Q_T(\boldsymbol{\mu}, \pi^m)|] \le g(T)$. For brevity, we will usually write $Q_T = Q_T(\boldsymbol{\mu}, \pi^m)$.

Local generalization. We assume that $\boldsymbol{\mu}$ and π^m satisfy *local generalization*. Informally, if the agent is given an input x, taking the mentor action for a similar input x' is almost as good as taking the mentor action for x. Formally, we assume $\mathcal{X} \subseteq \mathbb{R}^n$ and there exists $L > 0$ such that for all $x, x' \in \mathcal{X}$ and $t \in [T]$, $|\mu_t^m(x) - \mu_t(x, \pi^m(x'))| \le L||x - x'||$, where $||\cdot||$ denotes Euclidean distance. This represents the ability to transfer knowledge between similar inputs:

$$\Big| \underbrace{\mu_t(x, \pi^m(x))}_{\text{Taking the right action}} - \underbrace{\mu_t(x, \pi^m(x'))}_{\text{Using what you learned in } x'} \Big| \le \underbrace{L||x - x'||}_{\text{Input similarity}}$$

This ability seems fundamental to intelligence and is well-understood in psychology (e.g., Esser et al., 2023) and education (e.g., Hajian, 2019). Note that the input space $\mathcal{X} \subseteq \mathbb{R}^n$ can be any encoding of the agent's situation, not just its physical positioning. See Section 5.2 for further discussion.

All suprema over $\boldsymbol{\mu}, \pi^m$ pairs are assumed to be restricted to $\boldsymbol{\mu}, \pi^m$ pairs which satisfy local generalization.

Regret. If $\mu_t(x_t, y_t) \in [0,1]$ is the chance of avoiding catastrophe at time t (conditioned on no prior catastrophe), then by the chain rule of probability, $\prod_{t=1}^T \mu_t(x_t, y_t)$ is the agent's overall chance of avoiding catastrophe. For given $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\mu}, \pi^m$, the agent's *multiplicative regret*[6] is

$$R_T^\times(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\mu}, \pi^m) = \log \prod_{t=1}^T \mu_t^m(x_t) - \log \prod_{t=1}^T \mu_t(x_t, y_t)$$

[5] For example, it suffices for $\mathcal{X}$ to have finite Lebesgue measure. Note that this does not imply boundedness. Alternatively, σ-smoothness can be defined with respect to a different distribution; see Definition 1 of Block et al. (2022).

[6] One could also define the multiplicative regret as $R_T' = \prod_{t=1}^T \mu_t^m(x_t) - \prod_{t=1}^T \mu_t(x_t, y_t)$, but our definition is actually stricter: $\lim_{T\to\infty} R_T^\times \to 0$ implies $\lim_{T\to\infty} R_T' \to 0$, while the reverse is not true. In particular, $\lim_{T\to\infty} R_T' \to 0$ is trivial if $\lim_{T\to\infty} \prod_{t=1}^T \mu_t^m(x_t) \to 0$.

when all payoffs are strictly positive. To handle the case where some payoffs are zero, we assume the existence of $\mu_0^m > 0$ such that $\mu_t^m(x_t) \geq \mu_0^m$ always. Thus only the agent's payoffs can be zero, so we can safely define $R_T^\times(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\mu}, \pi^m) = \infty$ whenever $\mu_t(x_t, y_t) = 0$ for some $t \in [T]$. We write $R_T^\times = R_T^\times(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\mu}, \pi^m)$ for brevity.

The assumption of $\mu_t^m(x_t) \geq \mu_0^m > 0$ means that the mentor cannot be abysmal. In fact, we argue that high-stakes AI applications should employ a mentor who is almost always safe, i.e., $\mu_0^m \approx 1$. If no such mentor exists for some application, perhaps that application should be avoided altogether.

We also define the agent's *additive regret* as

$$R_T^+(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\mu}, \pi^m) = \sum_{t=1}^T \mu_t^m(x_t) - \sum_{t=1}^T \mu_t(x_t, y_t)$$

and similarly write $R_T^+ = R_T^+(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\mu}, \pi^m)$ for brevity. For both objectives, we desire subconstant worst-case regret: the total (not average) expected regret should go to 0 for any $\boldsymbol{\mu}$ and π^m. Formally, we want $\lim_{T\to\infty} \sup_{\boldsymbol{\mu},\pi^m} \mathbb{E}[R_T^\times] = 0$ and $\lim_{T\to\infty} \sup_{\boldsymbol{\mu},\pi^m} \mathbb{E}[R_T^+] = 0$.

VC and Littlestone dimensions. VC dimension (Vapnik & Chervonenkis, 1971) and Littlestone dimension (Littlestone, 1988) are standard measures of learning difficulty which capture the ability of a hypothesis class (in our case, a policy class) to realize arbitrary combinations of labels (in our case, actions). We omit the precise definitions since we only utilize these concepts via existing results. See Shalev-Shwartz & Ben-David (2014) for a comprehensive overview.

Misc. The diameter of a set $S \subseteq \mathcal{X}$ is defined by $\operatorname{diam}(S) = \max_{x,x' \in S} ||x - x'||$. All logarithms and exponents are base e unless otherwise noted. For convenience, we treat $\min_{x\in\emptyset} f(x)$ as ∞ for any function f.

4. Avoiding catastrophe with sublinear queries is impossible in general

We first show that in general, any algorithm with sublinear mentor queries has unbounded regret in the worst-case, even when inputs are i.i.d. on $[0, 1]$ and $\boldsymbol{\mu}$ does not vary over time. The formal proofs are deferred to Appendix A, but we provide intuition and define the construction here.

Theorem 4.1. *Any algorithm with sublinear queries has unbounded worst-case regret (both multiplicative and additive) as $T \to \infty$. Specifically,*

$$\sup_{\boldsymbol{\mu},\pi^m} \mathbb{E}[R_T^\times], \ \sup_{\boldsymbol{\mu},\pi^m} \mathbb{E}[R_T^+] \in \Omega\left(L\sqrt{\frac{T}{\sup_{\boldsymbol{\mu},\pi^m} \mathbb{E}[|Q_T|]+1}}\right)$$

Intuitively, the regret decreases as the number of queries increases. However, as long as the number of queries remains sublinear in T, the regret is unbounded as $T \to \infty$.

We also have the following corollary of Theorem 4.1:

Corollary 4.1.1. *Even when $\mu_t^m(x) = 1$ for all t and x, any algorithm with sublinear queries satisfies*

$$\lim_{T\to\infty} \sup_{\boldsymbol{\mu},\pi^m} \mathbb{E}\left[\prod_{t=1}^T \mu_t(x_t, y_t)\right] = 0$$

In other words, even if the mentor never causes catastrophe, any algorithm with sublinear queries causes catastrophe with probability 1 as $T \to \infty$ in the worst case.

4.1. Intuition

We partition $\mathcal{X}$ into equally-sized sections that are "independent" in the sense that querying an input in section i provides no information about section j. There will be $f(T)$ sections, where f is a function that we will choose. If $|Q_T| \in o(f(T))$, most of these sections will never contain a query. When the agent sees an input in a section not containing a query, it essentially must guess, meaning it will be wrong about half the time. We then choose a payoff function (which is the same for all time steps) which makes the wrong guesses as costly as possible, subject to the local generalization constraint. Figure 1 fleshes out this idea.

The choice of f is crucial. One idea is $f(T) = T$. If the agent is wrong about half the time, and the average payoff for wrong actions is $1 - \frac{L}{4T}$, we can estimate the regret as

$$\begin{aligned} R_T^\times &= \log \prod_{t=1}^T \mu_t^m(x_t) - \log \prod_{t=1}^T \mu_t(x_t, y_t) \\ &\approx \log 1 - \log \left(1 - \frac{L}{4T}\right)^{T/2} \\ &= -\frac{T}{2} \log \left(1 - \frac{L}{4T}\right) \\ &\approx \frac{T}{2} \cdot \frac{L}{4T} \\ &= \frac{L}{8} \end{aligned}$$

Thus $f(T) = T$ can at best give us a constant lower bound on regret. Instead, we choose f such that $|Q_T| \in o(f(T))$ and $f(T) \in o(T)$. Specifically, we choose $f(T) = \max(\sqrt{\sup_{\boldsymbol{\mu},\pi^m} \mathbb{E}[|Q_T|]T}, 1)$. Most sections still will not contain a query, so the agent is still wrong about half the time, but the payoff for wrong actions is worse. Then

$$\begin{aligned} R_T^\times &\approx \log 1 - \log \left(1 - \frac{L}{4f(T)}\right)^{T/2} \\ &\approx \frac{LT}{8f(T)} \\ &\in \Omega\left(L\sqrt{\frac{T}{\sup_{\boldsymbol{\mu},\pi^m} \mathbb{E}[|Q_T|] + 1}}\right) \end{aligned}$$

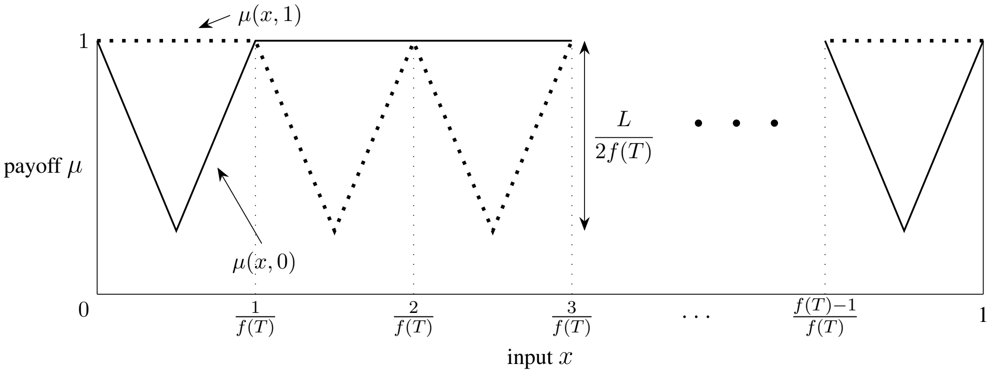

Figure 1: An illustration of the construction we use to prove Theorem 4.1 (not to scale). The horizontal axis indicates the input $x \in [0,1] = \mathcal{X}$ and the vertical axis indicates the payoff $\mu(x,y) \in [0,1]$. The solid line represents $\mu(x,0)$ and the dotted line represents $\mu(x,1)$. In each section, one of the actions has the optimal payoff of 1, and the other action has the worst possible payoff allowed by L, reaching a minimum of $1 - \frac{L}{2f(T)}$. Crucially, both actions result in a payoff of 1 at the boundaries between sections: this allows us to "reset" for the next section. As a result, we can freely toggle the optimal action for each section independently.

which produces the bound in Theorem 4.1.

VC dimension. The class of mentor policies in our construction has VC dimension $f(T)$; across all possible values of T, this implies infinite VC (and Littlestone) dimension. We know that this is necessary given our positive results.

4.2. Formal definition of construction

Let $\mathcal{X} = [0,1]$ and $\mathcal{D}_t = U(\mathcal{X})$ for each $t \in [T]$, where $U(\mathcal{X})$ is the uniform distribution on $\mathcal{X}$. Assume that $L \leq 1$; this will simplify the math and only makes the problem easier for the agent. We define a family of payoff functions parameterized by a function $f : \mathbb{N} \to \mathbb{N}$ and a bit string $\boldsymbol{a} = (a_1, a_2, \dots, a_{f(T)}) \in \{0,1\}^{f(T)}$. The bit a_j will denote the optimal action in section j. Note that $f(T) \geq 1$.

For each $j \in [f(T)]$, we refer to $X_j = \left[\frac{j-1}{f(T)}, \frac{j}{f(T)}\right]$ as the jth section. Let $m_j = \frac{j-0.5}{f(T)}$ be the midpoint of X_j. Assume that each x_t belongs to exactly one X_j (this happens with probability 1, so this assumption does not affect the expected regret). Let $j(x)$ denote the index of the section containing input x. Then $\mu_{f,\boldsymbol{a}}$ is defined by

$$\mu_{f,\boldsymbol{a}}(x,y) = \begin{cases} 1 & \text{if } y = a_{j(x)} \\ 1 - L\left(\dfrac{1}{2f(T)} - |m_{j(x)} - x|\right) & \text{if } y \neq a_{j(x)} \end{cases}$$

We use this payoff function for all time steps: $\mu_t = \mu_{f,\boldsymbol{a}}$ for all $t \in [T]$. Let π^m be any optimal policy for $\mu_{f,\boldsymbol{a}}$. Note that there is a unique optimal action for each x_t, since each x_t belongs to exactly one X_j; formally, $\pi^m(x_t) = a_{j(x_t)}$.

For any $\boldsymbol{a} \in \{0,1\}^{f(T)}$, $\mu_{f,\boldsymbol{a}}$ is piecewise linear (trivially) and continuous (because both actions have payoff 1 on the boundary between sections). Since the slope of each piece is in $\{-L, 0, L\}$, $\mu_{f,\boldsymbol{a}}$ is Lipschitz continuous. Thus by Proposition E.1, π^m satisfies local generalization.

5. Avoiding catastrophe given finite VC or Littlestone dimension

Theorem 4.1 shows that avoiding catastrophe is impossible in general. What if we restrict ourselves to settings where standard online learning is possible? Specifically, we assume that π^m belongs to a policy class Π where either (1) Π has finite VC dimension d and $\boldsymbol{x}$ is σ-smooth or (2) Π has finite Littlestone dimension d.[7] This section presents a simple algorithm which guarantees subconstant regret (both multiplicative and additive) and sublinear queries under either of those assumptions. Formal proofs are deferred to Appendix B but we provide intuition and a proof sketch here.

5.1. Intuition behind the algorithm

Algorithm 1 has two simple components: (1) run a modified version of the Hedge algorithm for online learning, but (2) ask for help for unfamiliar inputs (specifically, when the input is very different from any queried input with the same action under the proposed policy). Hedge ensures that the number of mistakes (i.e., the number of time steps where the agent's action doesn't match the mentor's) is small, and asking for help for unfamiliar inputs ensures that when we do make a mistake, the cost isn't too high. This algorithmic structure seems quite natural: mostly follow a baseline strategy, but ask for help when out-of-distribution.

Hedge. Hedge (Freund & Schapire, 1997) is a standard online learning algorithm which ensures sublinear regret when the number of hypotheses (in our case, the number of

[7] Recall from Section 1.3 that standard online learning becomes tractable under either of these assumptions.

policies in Π) is finite.[8] We would prefer not to assume that Π is finite. Luckily, any policy in Π can be approximated within ε when either (1) Π has finite VC dimension and $\boldsymbol{x}$ is σ-smooth or (2) Π has finite Littlestone dimension. Thus we can run Hedge on this approximative policy class instead.

One other modification is necessary. In standard online learning, losses are observed on every time step, but our agent only receives feedback in response to queries. To handle this, we modify Hedge to only perform updates on time steps with queries and to issue a query with probability p on each time step. Continuing our lucky streak, Russo et al. (2024) analyze exactly this modification of Hedge.

5.2. Local generalization

Local generalization is vital: this is what allows us to detect when an input is unfamiliar. Crucially, our algorithm does not need to know how inputs are encoded in $\mathbb{R}^n$ and does not need to know L: it only needs to be able to compute the nearest-neighbor distance $\min_{(x,y)\in S: y=\pi_t(x_t)} ||x_t - x||$. Thus we only need to assume that there exists *some* encoding satisfying local generalization.

To elaborate, recall the example that a 3 mm spot and a 3.1 mm spot on X-rays likely have similar risk levels (assuming similar density, location, etc.). If the risk level abruptly increases for any spot over 3 mm, then local generalization may not hold for a naive encoding which treats size as a single dimension. However, a more nuanced encoding would recognize that these two situations – a 3 mm vs 3.1 mm spot – are in fact *not* similar. Constructing a suitable encoding may be challenging, but we do *not* require the agent to have explicit access to such an encoding: the agent only needs a nearest-neighbor distance oracle.

Conceptually, the algorithm only needs to be able to detect when an input is unfamiliar. While this task remains far from trivial, we argue that it is more tractable than fully constructing a suitable encoding. See Section 6 for a discussion of potential future work on this topic.

We note that these encoding-related questions apply similarly to the more standard assumption of Lipschitz continuity. In fact, Lipschitz continuity implies local generalization when the mentor is optimal (Proposition E.1). We also mention that without local generalization, avoiding catastrophe is impossible even when the mentor policy class has finite VC dimension and $\boldsymbol{x}$ is σ-smooth (Theorem E.2).

Algorithm 1 successfully avoids catastrophe assuming finite VC or Littlestone dimension.

```
Inputs: $T \in \mathbb{N}$, $\varepsilon \in \mathbb{R}_{>0}$, $d \in \mathbb{N}$, policy class $\Pi$
if $\Pi$ has VC dimension $d$ then
    $\tilde{\Pi} \leftarrow$ any smooth $\varepsilon$-cover of $\Pi$ of size at most $(41/\varepsilon)^d$ (see Definition 5.3)
else
    $\tilde{\Pi} \leftarrow$ any adversarial cover of $\Pi$ of size at most $(eT/d)^d$ (see Definition 5.4)
$S \leftarrow \emptyset$
$w(\pi) \leftarrow 1$ for all $\pi \in \tilde{\Pi}$
$p \leftarrow 1/\sqrt{\varepsilon T}$
$\eta \leftarrow \max\left(\sqrt{\frac{p \log |\tilde{\Pi}|}{2T}}, \frac{p^2}{\sqrt{2}}\right)$
for $t$ from $1$ to $T$ do
    Run one step of Hedge, which selects policy $\pi_t$
    with probability $p$ : hedgeQuery ← true
    with probability $1-p$ : hedgeQuery ← false
    if hedgeQuery[9] then
        Query mentor and observe $\pi^m(x_t)$
        $\ell(t,\pi) \leftarrow \mathbf{1}(\pi(x_t) \neq \pi^m(x_t))$ for all $\pi \in \tilde{\Pi}$
        $\ell^* \leftarrow \min_{\pi\in\tilde{\Pi}} \ell(t,\pi)$
        $w(\pi) \leftarrow w(\pi) \cdot \exp(-\eta(\ell(t,\pi) - \ell^*))$ for all $\pi \in \tilde{\Pi}$
        $\pi_t \leftarrow \arg\min_{\pi\in\tilde{\Pi}} \ell(t,\pi)$
    else
        $P(\pi) \leftarrow w(\pi)/\sum_{\pi'\in\tilde{\Pi}} w(\pi')$ for all $\pi \in \tilde{\Pi}$
        Sample $\pi_t \sim P$
    if $\min_{(x,y)\in S: y=\pi_t(x_t)} ||x_t - x|| > \varepsilon^{1/n}$ then
        Ask for help if out-of-distribution
        Query mentor (if not already queried this round) and observe $\pi^m(x_t)$
        $S \leftarrow S \cup \{(x_t, \pi^m(x_t))\}$
    else
        Otherwise, follow Hedge's chosen policy
        Take action $\pi_t(x_t)$
```

5.3. Main result

For simplicity, here we only state our results for $\mathcal{Y} = \{0,1\}$; Appendix C extends our result to many actions using the standard "one versus rest" reduction. We first prove regret and query bounds parametrized by ε:

Theorem 5.1. *Let $\mathcal{Y} = \{0,1\}$. Assume $\pi^m \in \Pi$ where either (1) Π has finite VC dimension d and $\boldsymbol{x}$ is σ-smooth, or (2) Π has finite Littlestone dimension d. Then for any $T \in \mathbb{N}$ and $\varepsilon \in \left[\frac{1}{T}, \left(\frac{\mu_0^m}{2L}\right)^n\right]$, Algorithm 1 satisfies*

$$\mathbb{E}\left[R_T^\times\right] \in O\left(\frac{dL}{\sigma\mu_0^m} T\varepsilon^{1+1/n} \log(T + 1/\varepsilon)\right)$$

$$\mathbb{E}\left[R_T^+\right] \in O\left(\frac{dL}{\sigma} T\varepsilon^{1+1/n} \log(T + 1/\varepsilon)\right)$$

$$\mathbb{E}[|Q_T|] \in O\left(\sqrt{\frac{T}{\varepsilon}} + \frac{d}{\sigma} T\varepsilon \log(T+1/\varepsilon) + \frac{\mathbb{E}[\text{diam}(\boldsymbol{x})^n]}{\varepsilon}\right)$$

[8] Chapter 5 of Slivkins et al. (2019) and Chapter 21 of Shalev-Shwartz & Ben-David (2014) give modern introductions to Hedge.

[9] The reader may notice that we do not update S in this case. This is simply because those updates are not necessary for the desired bounds and omitting these updates simplifies the analysis.

In Case 1, the expectation is over the randomness of both $\boldsymbol{x}$ and the algorithm, while in Case 2, the expectation is over only the randomness of the algorithm. The bounds clearly have no dependence on σ in Case 2, but we include σ anyway to avoid writing two separate sets of bounds.

To obtain subconstant regret and sublinear queries, we can choose $\varepsilon = T^{\frac{-2n}{2n+1}}$. This also satisfies $2L\varepsilon^{1/n} \le \mu_0^m$ for large enough T.

Theorem 5.2. *Let $\mathcal{Y} = \{0,1\}$. Assume $\pi^m \in \Pi$ where either (1) Π has finite VC dimension d and $\boldsymbol{x}$ is σ-smooth or (2) Π has finite Littlestone dimension d. Then for any $T \in \mathbb{N}$, Algorithm 1 with $\varepsilon = T^{\frac{-2n}{2n+1}}$ satisfies*

$$\mathbb{E}\left[R_T^\times\right] \in O\left(\frac{dL}{\sigma\mu_0^m} T^{\frac{-1}{2n+1}} \log T\right)$$

$$\mathbb{E}\left[R_T^+\right] \in O\left(\frac{dL}{\sigma} T^{\frac{-1}{2n+1}} \log T\right)$$

$$\mathbb{E}[|Q_T|] \in O\left(T^{\frac{4n+1}{4n+2}}\left(\frac{d}{\sigma}\log T + \mathbb{E}[\operatorname{diam}(\boldsymbol{x})^n]\right)\right)$$

Before proceeding to the proof sketch, we highlight some advantages of our algorithm.

Limited knowledge required. Our algorithm needs to know Π, as is standard. However, the algorithm does not need to know σ (in the smooth case) or L. Although Algorithm 1 as written does require T as an input, it can be converted into an infinite horizon/anytime algorithm via the standard "doubling trick" (see, e.g., Slivkins et al., 2019).

Unbounded environment. Our algorithm can handle an unbounded input space: the number of queries simply scales with the maximum distance between observed inputs in the form of $\mathbb{E}[\operatorname{diam}(\boldsymbol{x})^n]$.

Simultaneous bounds for all $\boldsymbol{\mu}$. Recall that the agent never observes payoffs and only learns from mentor queries. This means that the agent's behavior does not depend on $\boldsymbol{\mu}$ at all. In other words, the distribution of $(\boldsymbol{x}, \boldsymbol{y})$ depends on π^m but not $\boldsymbol{\mu}$. Consequently, a given π^m induces a *single* distribution $(\boldsymbol{x}, \boldsymbol{y})$ which satisfies the bounds in Theorem 5.2 *simultaneously* for all $\boldsymbol{\mu}$ satisfying local generalization.

5.4. Proof sketch

The formal proof of Theorem 5.1 can be found in Appendix B, but we outline the key elements here. The regret analysis consists of two ingredients: analyzing the Hedge component and analyzing the "ask for help when out-of-distribution" component. The former will bound the number of mistakes made by the algorithm (i.e., the number of time steps where the agent's action doesn't match the mentor's), and the latter will bound the cost of any single mistake. We must also show that the latter does not result in excessively many queries, which we do via a novel packing argument.

We begin by formalizing two notions of approximating a policy class:

Definition 5.3. Let U be the uniform distribution over $\mathcal{X}$. For $\varepsilon > 0$, a policy class $\tilde{\Pi}$ is a *smooth ε-cover* of a policy class Π if for every $\pi \in \Pi$, there exists $\tilde{\pi} \in \tilde{\Pi}$ such that $\Pr_{x\sim U}[\pi(x) \neq \tilde{\pi}(x)] \le \varepsilon$.

Definition 5.4. A policy class $\tilde{\Pi}$ is an *adversarial cover* of a policy class Π if for every $\boldsymbol{x} \in \mathcal{X}^T$ and $\pi \in \Pi$, there exists $\tilde{\pi} \in \tilde{\Pi}$ such that $\pi(x_t) = \tilde{\pi}(x_t)$ for all $t \in [T]$.

An adversarial cover is a perfect cover by definition. The idea of a smooth ε-cover is that if the probability of disagreement over the uniform distribution is small, then the probability of disagreement over a σ-smooth distribution cannot be too much larger.

Lemma 5.1. *Let $\tilde{\Pi}$ be a smooth ε-cover of Π and let $\mathcal{D}$ be a σ-smooth distribution. Then for any $\pi \in \Pi$, there exists $\tilde{\pi} \in \tilde{\Pi}$ such that $\Pr_{x\sim\mathcal{D}}[\pi(x) \neq \tilde{\pi}(x)] \le \varepsilon/\sigma$.*

Proof. Define $S(\tilde{\pi}) = \{x \in \mathcal{X} : \pi(x) \neq \tilde{\pi}(x)\}$. By the definition of a smooth ε-cover, there exists $\tilde{\pi} \in \tilde{\Pi}$ such that $\Pr_{x\sim U}[x \in S(\tilde{\pi})] \le \varepsilon$. Since $\mathcal{D}$ is σ-smooth, $\Pr_{x\sim\mathcal{D}}[\pi(x) \neq \tilde{\pi}(x)] = \Pr_{x\sim\mathcal{D}}[x \in S(\tilde{\pi})] \le \Pr_{x\sim U}[x \in S(\tilde{\pi})]/\sigma \le \varepsilon/\sigma$, as claimed. □

The existence of small covers is crucial:

Lemma 5.2 (Lemma 7.3.2 in Haghtalab (2018)[10]). *For all $\varepsilon > 0$, any policy class of VC dimension d admits a smooth ε-cover of size at most $(41/\varepsilon)^d$.*

Lemma 5.3 (Lemmas 21.13 and A.5 in Shalev-Shwartz & Ben-David (2014)). *Any policy class of Littlestone dimension d admits an adversarial cover of size at most $(eT/d)^d$.*

We will run a variant of Hedge on $\tilde{\Pi}$. The vanilla Hedge algorithm operates in the standard online learning model where on each time step, the agent selects a policy (or more generally, a hypothesis), and observes the *loss* of every policy. In general the loss function can depend arbitrarily on the time step, the policy, and prior events, but we will only use the indicator loss function $\ell(t,\pi) = \mathbf{1}(\pi(x_t) \neq \pi^m(x_t))$. Crucially, whenever we query and learn $\pi^m(x_t)$, we can compute $\ell(t,\pi)$ for every $\pi \in \tilde{\Pi}$.

We cannot afford to query on every time step, however. Recently, Russo et al. (2024) analyzed a variant of Hedge where losses are observed only in response to queries, which they call "label-efficient feedback". They proved a regret bound when a query is issued on each time step with fixed probability p. Lemma 5.4 restates their result in a form that is more convenient for us. See Appendices B.1 and B.3 for details on our usage of results from Russo et al. (2024). Full

[10] See also Haussler & Long (1995) or Lemma 13.6 in Boucheron et al. (2013) for variants of this lemma.

pseudocode for HEDGEWITHQUERIES can also be found in the appendix (Algorithm 2).

Lemma 5.4 (Lemma 3.5 in Russo et al., 2024)**.** *Assume $\tilde{\Pi}$ is finite. Then for any loss function $\ell : [T] \times \tilde{\Pi} \to [0, 1]$ and query probability $p > 0$,* HEDGEWITHQUERIES *enjoys the regret bound*

$$\sum_{t=1}^{T} \mathbb{E}[\ell(t, \pi_t)] - \min_{\tilde{\pi} \in \tilde{\Pi}} \sum_{t=1}^{T} \mathbb{E}[\ell(t, \pi)] \leq \frac{\log |\tilde{\Pi}|}{p^2}$$

where π_t is the policy chosen at time t and the expectation is over the randomness of the algorithm.

We apply Lemma 5.4 with $\ell(t, \pi) = \mathbf{1}(\pi(x_t) \neq \pi^m(x_t))$ and combine this with Lemmas 5.2 and 5.1 (in the σ-smooth case) and with Lemma 5.3 (in the adversarial case). This yields a $O\left(\frac{d}{\sigma} T\varepsilon \log(1/\varepsilon) \log T\right)$ bound on the number of mistakes made by Algorithm 1 (Lemma B.1).

The other key ingredient of the proof is analyzing the "ask for help when out-of-distribution" component. Combined with the local generalization assumption, this allows us to fairly easily bound the cost of a single mistake (Lemma B.2). The trickier part is bounding the number of resulting queries. It is tempting to claim that the inputs queried in the out-of-distribution case must all be separated by at least $\varepsilon^{1/n}$ and thus form an $\varepsilon^{1/n}$-packing, but this is actually false.

Instead, we bound the number of data points (i.e., queries) needed to cover a set *with respect to the realized actions of the algorithm* (Lemma B.6). This contrasts with vanilla packing arguments which consider all data points in aggregate. The key to our analysis is that the number of mistakes made by the algorithm – which we already bounded in Lemma B.1 – gives us crucial information about how data points are distributed with respect to the actions of the algorithm. Our technique might be useful in other contexts where a more refined packing argument is needed and a bound on the number of mistakes already exists.

6. Conclusion and future work

In this paper, we proposed a model of avoiding catastrophe in online learning. We showed that achieving subconstant regret in our problem (with the help of a mentor and local generalization) is no harder than achieving sublinear regret in standard online learning.

Remaining technical questions. First, we have not resolved whether our problem is tractable for finite VC dimension and fully adversarial inputs (although Appendix D shows that the problem is tractable for at least some classes with finite VC but infinite Littlestone dimension). Second, the time complexity of Algorithm 1 currently stands at a hefty $\Omega(|\tilde{\Pi}|)$ per time step plus the time to compute $\tilde{\Pi}$. In the standard online learning setting, Block et al. (2022) and Haghtalab et al. (2022) show how to replace discretization approaches like ours with *oracle-efficient* approaches, where a small number of calls to an optimization oracle are made per round. We are optimistic about leveraging such techniques to obtain efficient algorithms in our setting.

Local generalization. Our algorithm crucially relies on the ability to detect when an input is unfamiliar, i.e., differs significantly from prior observations in a metric space which satisfies local generalization. Without this ability, the practicality of our algorithm would be fundamentally limited. One option is to use out-of-distribution (OOD) detection, which is conceptually similar and well-studied (see Yang et al., 2024 for a survey). However, it is an open question whether standard OOD detection methods are measuring distance in a metric space which satisfies local generalization.

We are also interested in alternatives to local generalization. Theorem E.2 shows that our positive result breaks down if local generalization is removed, so some sort of assumption is necessary. One possible alternative is Bayesian inference. We intentionally avoided Bayesian approaches in this paper due to tractability concerns, but it seems premature to abandon those ideas entirely.

MDPs. Finally, we are excited to apply the ideas in this paper to Markov Decision Processes (MDPs): specifically, MDPs where some actions are irreversible ("non-communicating") and the agent only gets one attempt ("single-episode"). In such MDPs, the agent must not only avoid catastrophe but also obtain high reward. As discussed in Section 2, very little theory exists for RL in non-communicating single-episode MDPs. Can an agent learn near-optimal behavior in high-stakes environments while becoming self-sufficient over time? Formally, we pose the following open problem:

Is there an algorithm for non-communicating single-episode undiscounted MDPs which ensures that both the regret and the number of mentor queries are sublinear in T?

Impact statement

As AI systems become increasingly powerful, we believe that the safety guarantees of such systems should become commensurately robust. Irreversible costs are especially worrisome, and we hope that our work plays a small part in mitigating such risks. We do not believe that our work has any concrete potential risks that should be highlighted here.

Author contributions

B. Plaut conceived the project. B. Plaut designed the mathematical model with feedback from H. Zhu and S. Russell. B. Plaut proved all of the results. H. Zhu participated in proof brainstorming and designed counterexamples for several early conjectures. B. Plaut wrote the paper, with feedback from H. Zhu and S. Russell. S. Russell supervised the project and secured funding.

Acknowledgements

This work was supported in part by a gift from Open Philanthropy to the Center for Human-Compatible AI (CHAI) at UC Berkeley. This paper also benefited from discussions with many other researchers. We would like to especially thank (in alphabetical order) Aly Lidayan, Bhaskar Mishra, Cameron Allen, Juan Liévano-Karim, Matteo Russo, Michael Cohen, Nika Haghtalab, and Scott Emmons. We would also like to thank our anonymous reviewers for helpful feedback.

References

Abaimov, S. and Martellini, M. *Artificial Intelligence in Autonomous Weapon Systems*, pp. 141–177. Springer International Publishing, Cham, 2020.

Altman, E. *Constrained Markov Decision Processes*. Routledge, 2021.

Azar, M. G., Osband, I., and Munos, R. Minimax Regret Bounds for Reinforcement Learning. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 263–272. PMLR, July 2017. ISSN: 2640-3498.

Barman, S., Khan, A., Maiti, A., and Sawarni, A. Fairness and welfare quantification for regret in multi-armed bandits. In *Proceedings of the Thirty-Seventh Conference on Artificial Intelligence (AAAI 2023)*, 2023.

Bengio, Y., Hinton, G., Yao, A., Song, D., Abbeel, P., Darrell, T., Harari, Y. N., Zhang, Y.-Q., Xue, L., Shalev-Shwartz, S., et al. Managing extreme AI risks amid rapid progress. *Science*, 384(6698):842–845, 2024.

Betancourt, M. A Conceptual Introduction to Hamiltonian Monte Carlo, July 2018. arXiv:1701.02434 [stat].

Block, A., Dagan, Y., Golowich, N., and Rakhlin, A. Smoothed online learning is as easy as statistical learning. In *Conference on Learning Theory*, pp. 1716–1786. PMLR, 2022.

Boucheron, S., Lugosi, G., and Massart, P. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press, 02 2013.

Cesa-Bianchi, N. and Lugosi, G. *Prediction, learning, and games*. Cambridge university press, 2006.

Cohen, M. K. and Hutter, M. Pessimism About Unknown Unknowns Inspires Conservatism. In *Proceedings of Thirty Third Conference on Learning Theory*, pp. 1344–1373. PMLR, July 2020. ISSN: 2640-3498.

Cohen, M. K., Catt, E., and Hutter, M. Curiosity Killed or Incapacitated the Cat and the Asymptotically Optimal Agent. *IEEE Journal on Selected Areas in Information Theory*, 2(2):665–677, June 2021. Conference Name: IEEE Journal on Selected Areas in Information Theory.

Critch, A. and Russell, S. TASRA: a taxonomy and analysis of societal-scale risks from AI. *arXiv preprint arXiv:2306.06924*, 2023.

Esser, S., Haider, H., Lustig, C., Tanaka, T., and Tanaka, K. Action–effect knowledge transfers to similar effect stimuli. *Psychological Research*, 87(7):2249–2258, October 2023.

Freund, Y. and Schapire, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

Grinsztajn, N., Ferret, J., Pietquin, O., Preux, P., and Geist, M. There Is No Turning Back: A Self-Supervised Approach for Reversibility-Aware Reinforcement Learning. In *Advances in Neural Information Processing Systems*, volume 34, pp. 1898–1911. Curran Associates, Inc., 2021.

Gu, S., Yang, L., Du, Y., Chen, G., Walter, F., Wang, J., and Knoll, A. A review of safe reinforcement learning: Methods, theories, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12): 11216–11235, 2024.

Guembe, B., Azeta, A., Misra, S., Osamor, V. C., Fernandez-Sanz, L., and Pospelova, V. The Emerging Threat of AI-driven Cyber Attacks: A Review. *Applied Artificial Intelligence*, 36(1), December 2022.

Hadfield-Menell, D., Milli, S., Abbeel, P., Russell, S., and Dragan, A. D. Inverse reward design. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 6768–6777, Red Hook, NY, USA, December 2017. Curran Associates Inc.

Haghtalab, N. *Foundation of Machine Learning, by the People, for the People*. PhD thesis, Microsoft Research, 2018.

Haghtalab, N., Han, Y., Shetty, A., and Yang, K. Oracle-efficient online learning for smoothed adversaries. *Advances in Neural Information Processing Systems*, 35: 4072–4084, 2022.

Haghtalab, N., Roughgarden, T., and Shetty, A. Smoothed analysis with adaptive adversaries. *Journal of the ACM*, 71(3):1–34, 2024.

Hajian, S. Transfer of Learning and Teaching: A Review of Transfer Theories and Effective Instructional Practices. *IAFOR Journal of Education*, 7(1):93–111, 2019. Publisher: International Academic Forum ERIC Number: EJ1217940.

Hanneke, S. *Theory of Disagreement-Based Active Learning*, volume 7. Now Publishers Inc., Hanover, MA, USA, June 2014.

Haussler, D. and Long, P. M. A generalization of Sauer's lemma. *Journal of Combinatorial Theory, Series A*, 71 (2):219–240, August 1995.

Hendrycks, D., Mazeika, M., and Woodside, T. An overview of catastrophic AI risks. *arXiv preprint arXiv:2306.12001*, 2023.

Jaksch, T., Ortner, R., and Auer, P. Near-optimal Regret Bounds for Reinforcement Learning. *Journal of Machine Learning Research*, 11(51):1563–1600, 2010.

Jung, H. Ueber die kleinste kugel, die eine räumliche figur einschliesst. *Journal für die reine und angewandte Mathematik*, 123:241–257, 1901.

Kohli, P. and Chadha, A. Enabling pedestrian safety using computer vision techniques: A case study of the 2018 Uber Inc. self-driving car crash. In *Advances in Information and Communication: Proceedings of the 2019 Future of Information and Communication Conference (FICC), Volume 1*, pp. 261–279. Springer, 2020.

Kosoy, V. Delegative Reinforcement Learning: learning to avoid traps with a little help. SafeML ICLR 2019 Workshop, July 2019.

Littlestone, N. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine learning*, 2:285–318, 1988.

Liu, T., Zhou, R., Kalathil, D., Kumar, P., and Tian, C. Learning policies with zero or bounded constraint violation for constrained mdps. *Advances in Neural Information Processing Systems*, 34:17183–17193, 2021.

Maillard, O.-A., Mann, T., Ortner, R., and Mannor, S. Active Roll-outs in MDP with Irreversible Dynamics. July 2019.

Mindermann, S., Shah, R., Gleave, A., and Hadfield-Menell, D. Active Inverse Reward Design. In *Proceedings of the 1st Workshop on Goal Specifications for Reinforcement Learning*, 2018.

Moldovan, T. M. and Abbeel, P. Safe exploration in Markov decision processes. In *Proceedings of the 29th International Conference on Machine Learning*, ICML'12, pp. 1451–1458, Madison, WI, USA, June 2012. Omnipress.

Mouton, C., Lucas, C., and Guest, E. The operational risks of AI in large-scale biological attacks. Technical report, RAND Corporation, Santa Monica, 2024.

Osa, T., Pajarinen, J., Neumann, G., Bagnell, J., Abbeel, P., and Peters, J. *An Algorithmic Perspective on Imitation Learning*. Foundations and trends in robotics. Now Publishers, 2018.

Quiñonero-Candela, J., Sugiyama, M., Schwaighofer, A., and Lawrence, N. D. *Dataset shift in machine learning*. MIT Press, 2022.

Rajpurkar, P., Chen, E., Banerjee, O., and Topol, E. J. AI in health and medicine. *Nature medicine*, 28(1):31–38, 2022.

Russo, M., Celli, A., Colini Baldeschi, R., Fusco, F., Haimovich, D., Karamshuk, D., Leonardi, S., and Tax, N. Online learning with sublinear best-action queries. *Advances in Neural Information Processing Systems*, 37: 40407–40433, 2024.

Shalev-Shwartz, S. and Ben-David, S. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 1 edition, May 2014.

Slivkins, A. Contextual Bandits with Similarity Information. In *Proceedings of the 24th Annual Conference on Learning Theory (COLT)*, pp. 679–702, December 2011. ISSN: 1938-7228.

Slivkins, A. et al. Introduction to multi-armed bandits. *Foundations and Trends® in Machine Learning*, 12(1-2):1–286, 2019.

Spielman, D. A. and Teng, S.-H. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. *Journal of the ACM (JACM)*, 51(3): 385–463, 2004.

Stradi, F. E., Castiglioni, M., Marchesi, A., and Gatti, N. Learning adversarial MDPs with stochastic hard constraints. *arXiv preprint arXiv:2403.03672*, 2024.

Turchetta, M., Berkenkamp, F., and Krause, A. Safe Exploration in Finite Markov Decision Processes with Gaussian Processes. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

Vapnik, V. N. and Chervonenkis, A. Y. On the Uniform Convergence of Relative Frequencies of Events to Their Probabilities. *Theory of Probability & Its Applications*, 16(2):264–280, January 1971. Publisher: Society for Industrial and Applied Mathematics.

Villasenor, J. and Foggo, V. Artificial intelligence, due process and criminal sentencing. *Michigan State Law Review*, pp. 295–354, 2020.

Wachi, A., Shen, X., and Sui, Y. A Survey of Constraint Formulations in Safe Reinforcement Learning. volume 9, pp. 8262–8271, August 2024. ISSN: 1045-0823.

Wu, Y. *Lecture notes on: Information-theoretic methods for high-dimensional statistics*. 2020.

Yang, J., Zhou, K., Li, Y., and Liu, Z. Generalized out-of-distribution detection: A survey. *International Journal of Computer Vision*, pp. 1–28, 2024.

Zhao, W., He, T., Chen, R., Wei, T., and Liu, C. State-wise Safe Reinforcement Learning: A Survey. volume 6, pp. 6814–6822, August 2023. ISSN: 1045-0823.

A. Proof of Theorem 4.1

A.1. Proof notation

1. Let M_j be the set of time steps $t \le T$ where $|m_j - x_t| \le \frac{1}{4f(T)}$. In words, x_t is relatively close to the midpoint of X_j. This will imply that the suboptimal action is in fact quite suboptimal. This also implies that x_t is in X_j, since each X_j has length $1/f(T)$.
2. Let $J_{\neg Q} = \{j \in [f(T)] : x_t \notin X_j \ \forall t \in Q_T\}$ be the set of sections that are never queried. Since each query appears in exactly one section (because each input appears in exactly one section), $|J_{\neg Q}| \ge f(T) - |Q_T|$.
3. For each $j \in J_{\neg Q}$, let y^j be the most frequent action among time steps in M_j: $y^j = \arg\max_{y \in \{0,1\}} |\{t \in M_j : y = y_t\}|$.
4. Let $J'_{\neg Q} = \{j \in J_{\neg Q} : a_j \ne y^j\}$ be the set of sections where the more frequent action is wrong according to $\mu_{f,\boldsymbol{a}}$.
5. Let $M'_j = \{t \in M_j : y_t \ne a_j\}$ be the set of time steps where the agent chooses the wrong action according to $\mu_{f,\boldsymbol{a}}$, and x_t is close to the midpoint of section j.

Since $\boldsymbol{x}, \boldsymbol{y}$, and $\boldsymbol{a}$ are random variables, all variables defined on top of them (such as M_j) are also random variables. In contrast, the partition $\mathcal{X} = \{X_1, \dots, X_{f(T)}\}$ and properties thereof (like the midpoints m_j) are not random variables.

A.2. Proof roadmap

The proof considers an arbitrary algorithm with sublinear queries, and proceeds via the following steps:

1. Show that multiplicative regret and additive regret are tightly related (Lemma A.1). We will also use this lemma for our positive results.
2. Prove an asymptotic density lemma which we will use to show that $f(T) = \sqrt{|Q_T| T}$ is asymptotically between $|Q_T|$ and T (Lemma A.2).
3. Prove a simple variant of the Chernoff bound which we will apply multiple times (Lemma A.3).
4. Show that with high probability, $\sum_{j \in S} |M_j|$ is large for any subset of sections S (Lemma A.4).
5. Prove that $|J'_{\neg Q}|$ is large with high probability (Lemma A.5).
6. The key lemma is Lemma A.6, which shows that a randomly sampled $\boldsymbol{a}$ produces poor agent performance with high probability. The central idea is that at least $f(T) - |Q_T|$ sections are never queried (which is large, by Lemma A.2), so the agent has no way of knowing the optimal action in those sections. As a result, the agent picks the wrong answer at least half the time on average (and at least a quarter of the time with high probability). Lemma A.4 implies that a constant fraction of those time steps will have significantly suboptimal payoffs, again with high probability.
7. Apply $\sup\limits_{\boldsymbol{\mu}, \pi^m} \mathop{\mathbb{E}}\limits_{\boldsymbol{x},\boldsymbol{y}} R_T^\times(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\mu}, \pi^m) \ge \mathop{\mathbb{E}}\limits_{\pi^m, \boldsymbol{a} \sim U(\{0,1\}^{f(T)})} \mathop{\mathbb{E}}\limits_{\boldsymbol{x},\boldsymbol{y}} R_T^\times(\boldsymbol{x}, \boldsymbol{y}, \mu_{f,\boldsymbol{a}}, \pi^m)$. Here $U(\{0,1\}^{f(T)})$ is the uniform distribution over bit strings of length $f(T)$ and we write $\pi^m, \boldsymbol{a} \sim U(\{0,1\}^{f(T)})$ with slight abuse of notation, since π^m is not drawn from $U(\{0,1\}^{f(T)})$ but rather is determined by $\boldsymbol{a}$ which is drawn from $U(\{0,1\}^{f(T)})$.
8. The analysis above results in a lower bound on R_T^+. The last step is to use Lemma A.1 to obtain a lower bound on $R_T^\times$.

Step 7 is essentially an application of the probabilistic method: if a randomly chosen $\mu_{f,\boldsymbol{a}}$ has high expected regret, then the worst-case $\boldsymbol{\mu}$ also has high expected regret. We have included subscripts in the expectations above to distinguish between the randomness over $\boldsymbol{a}$ and $\boldsymbol{x}, \boldsymbol{y}$. When subscripts are omitted, the expected value is over all randomness, i.e., $\boldsymbol{a}, \boldsymbol{x}$, and $\boldsymbol{y}$.

A.3. Proof

Lemma A.1. *If $\mu_t^m(x_t) \ge \mu_t(x_t, y_t)$ for all t, then $R_T^+ \le R_T^\times$. If $\mu_t(x_t, y_t) > 0$ for all t, then $R_T^\times \le \frac{R_T^+}{\min_{t \in [T]} \mu_t(x_t, y_t)}$.*

Proof. Recall the standard inequalities $1 - \frac{1}{a} \le \log a \le a - 1$ for any $a > 0$.

Part 1: $R_T^+ \le R_T^\times$. If $\mu_t(x_t, y_t) = 0$ for any $t \in [T]$, then $R_T^\times = \infty$ and the claim is trivially satisfied. Thus assume $\mu_t(x_t, y_t) > 0$ for all $t \in [T]$.

$$R_T^+ = \sum_{t=1}^T \mu_t^m(x_t) - \sum_{t=1}^T \mu_t(x_t, y_t) \qquad \text{(Definition of } R_T^+)$$

$$
\begin{aligned}
&= \sum_{t=1}^{T} \frac{\mu_t^m(x_t) - \mu_t(x_t, y_t)}{\mu_t^m(x_t)} && (\mu_t(x_t,y_t) \leq \mu_t^m(x_t) \text{ and } 0 \leq \mu_t^m(x_t) \leq 1)\\
&\leq \sum_{t=1}^{T} \log\left(\frac{\mu_t^m(x_t)}{\mu_t(x_t,y_t)}\right) && \left(1 - \frac{1}{a} \leq \log a \text{ for any } a > 0\right)\\
&= \log \prod_{t=1}^{T} \mu_t^m(x_t) - \log \prod_{t=1}^{T} \mu_t(x_t,y_t) && \text{(Properties of logarithms)}\\
&= R_T^\times && \text{(Definition of } R_T^\times\text{)}
\end{aligned}
$$

Part 2: $R_T^\times \leq R_T^+ / \min_{t\in[T]} \mu_t(x_t,y_t)$. We have

$$
\begin{aligned}
R_T^\times &= \log \prod_{t=1}^{T} \mu_t^m(x_t) - \log \prod_{t=1}^{T} \mu_t(x_t,y_t) && \text{(Definition of } R_T^\times \text{ given } \mu_t(x_t,y_t) > 0\ \forall t \in [T])\\
&= \sum_{t=1}^{T} \log\left(\frac{\mu_t^m(x_t)}{\mu_t(x_t,y_t)}\right) && \text{(Properties of logarithms)}\\
&\leq \sum_{t=1}^{T} \left(\frac{\mu_t^m(x_t) - \mu_t(x_t,y_t)}{\mu_t(x_t,y_t)}\right) && (\log a \leq a - 1 \text{ for any } a > 0)\\
&\leq \sum_{t=1}^{T} \frac{\mu_t(x_t,y_t)}{\min_{i\in[T]} \mu_i(x_i,y_i)} \left(\frac{\mu_t^m(x_t) - \mu_t(x_t,y_t)}{\mu_t(x_t,y_t)}\right) && (\mu_t(x_t,y_t) \geq \min_{i\in[T]} \mu_i(x_i,y_i) > 0\ \forall t \in [T])\\
&= \frac{1}{\min_{t\in[T]} \mu_t(x_t,y_t)} \sum_{t=1}^{T} (\mu_t^m(x_t) - \mu_t(x_t,y_t)) && \text{(Arithmetic)}\\
&= \frac{R_T^+}{\min_{t\in[T]} \mu_t(x_t,y_t)} && \text{(Definition of } R_T^+\text{)}
\end{aligned}
$$

as claimed. □

Lemma A.2. *Let $a, b : \mathbb{R}_{>0} \to \mathbb{R}_{>0}$ be functions such that $a(x) \in o(b(x))$. Then $c(x) = \sqrt{a(x)b(x)}$ satisfies $a(x) \in o(c(x))$ and $c(x) \in o(b(x))$.*

Proof. Since a and b are strictly positive (and thus c is as well), we have

$$
\frac{a(x)}{c(x)} = \frac{a(x)}{\sqrt{a(x)b(x)}} = \sqrt{\frac{a(x)}{b(x)}} = \frac{\sqrt{a(x)b(x)}}{b(x)} = \frac{c(x)}{b(x)}
$$

Then $a(x) \in o(b(x))$ implies

$$
\lim_{x\to\infty} \frac{a(x)}{c(x)} = \lim_{x\to\infty} \frac{c(x)}{b(x)} = \lim_{x\to\infty} \sqrt{\frac{a(x)}{b(x)}} = 0
$$

as required. □

Lemma A.3. *Let $z_1, \dots, z_n$ be i.i.d. variables in $\{0,1\}$ and let $Z = \sum_{i=1}^n z_i$. If $\mathbb{E}[Z] \geq W$, then $\Pr\left[Z \leq W/2\right] \leq \exp(-W/8)$.*

Proof. By the Chernoff bound for i.i.d. binary variables, we have $\Pr[Z \leq \mathbb{E}[Z]/2] \leq \exp(-\mathbb{E}[Z]/8)$. Since $-\mathbb{E}[Z] \leq -W$ and exp is an increasing function, we have $\exp(-\mathbb{E}[Z]/8) \leq \exp(-W/8)$. Also, $W/2 \leq E[Z]/2$ implies $\Pr[Z \leq W/2] \leq \Pr[Z \leq \mathbb{E}[Z]/2]$. Combining these inequalities proves the lemma. □

Lemma A.4. *Let $S \subseteq [f(T)]$ be any nonempty subset of sections. Then*

$$\Pr\left[\sum_{j\in S} |M_j| \le \frac{T|S|}{4f(T)}\right] \le \exp\left(\frac{-T}{16f(T)}\right)$$

Proof. Fix any $j \in [f(T)]$. For each $t \in [T]$, define the random variable z_t by $z_t = 1$ if $t \in M_j$ for some $j \in S$ and 0 otherwise. We have $t \in M_j$ iff x_t falls within a particular interval of length $\frac{1}{2f(T)}$. Since these intervals are disjoint for different j's, we have $z_t = 1$ iff x_t falls within a portion of the input space with total measure $\frac{|S|}{2f(T)}$. Since x_t is uniformly random across $[0, 1]$, we have $\mathbb{E}[z_t] = \frac{|S|}{2f(T)}$. Then $\mathbb{E}[\sum_{t=1}^T z_t] = \mathbb{E}[\sum_{j\in S} |M_j|] = \frac{T|S|}{2f(T)}$. Furthermore, since $x_1, \dots, x_T$ are i.i.d., so are $z_1, \dots, z_T$. Then by Lemma A.3,

$$\Pr\left[\sum_{j\in S} |M_j| \le \frac{T|S|}{4f(T)}\right] \le \exp\left(\frac{-T|S|}{16f(T)}\right) \le \exp\left(\frac{-T}{16f(T)}\right)$$

with the last step due to $|S| \ge 1$. □

Lemma A.5. *We have*

$$\Pr\left[|J'_{\neg Q}| \le \frac{f(T) - \mathbb{E}[|Q_T|]}{4}\right] \le \exp\left(-\frac{f(T) - \mathbb{E}[|Q_T|]}{16}\right)$$

Proof. Define a random variable $z_j = \mathbf{1}_{j\in J'_{\neg Q}}$ for each $j \in J_{\neg Q}$. By definition, if $j \in J_{\neg Q}$, no input in X_j is queried. Since queries outside of X_j provide no information about a_j, the agent's actions must be independent of a_j. In particular, the random variables a_j and y^j are independent. Combining that independence with $\Pr[a_j = 0] = \Pr[a_j = 1] = 0.5$ yields $\Pr[z_j = 1] = 0.5$ for all $j \in J_{\neg Q}$. Then

$$\begin{aligned}
\mathbb{E}\left[|J'_{\neg Q}|\right] &= \mathbb{E}\left[\sum_{j\in J_{\neg Q}} z_j\right] \\
&= |J_{\neg Q}|/2 \\
&\ge \frac{f(T) - \mathbb{E}[|Q_T|]}{2}
\end{aligned}$$

Furthermore, since $a_1, \dots, a_{f(T)}$ are independent, the random variables $\{z_j : j \in J_{\neg Q}\}$ are also independent. Applying Lemma A.3 yields the desired bound. □

Lemma A.6. *Suppose $f : \mathbb{N} \to \mathbb{N}$ and independently sample $\boldsymbol{a} \sim U(\{0,1\}^{f(T)})$ and $\boldsymbol{x} \sim U(\mathcal{X})^T$.*[11] *Then with probability at least* $1 - \exp\left(\frac{-T}{16f(T)}\right) - \exp\left(-\frac{f(T)-\mathbb{E}[|Q_T|]}{16}\right)$,

$$R_T^+ \ge \frac{LT(f(T) - \mathbb{E}[|Q_T|])}{2^7 f(T)^2}$$

Proof. Consider any $j \in J'_{\neg Q}$ and $t \in M'_j \subseteq M_j$. By definition of M_j, we have $|m_j - x_t| \le \frac{1}{4f(T)}$. Then by the definition of $\mu_{f,\boldsymbol{a}}$,

$$\begin{aligned}
\mu_{f,\boldsymbol{a}}(x_t, y_t) &= 1 - L\left(\frac{1}{2f(T)} - |x_t - m_j|\right) \\
&\le 1 - L\left(\frac{1}{2f(T)} - \frac{1}{4f(T)}\right) \\
&= 1 - \frac{L}{4f(T)}
\end{aligned}$$

[11] That is, the entire set $\{a_1, \dots, a_{f(T)}, x_1, \dots, x_T\}$ is mutually independent.

Since $\mu^m_{f,\boldsymbol{a}}(x_t) \geq \mu_{f,\boldsymbol{a}}(x_t, y_t)$ always, we can safely restrict ourselves to time steps $t \in M'_j$ for some $j \in J'_{\neg Q}$ and still obtain a lower bound:

$$\begin{aligned}
R^+_T &= \sum_{t=1}^{T} (\mu^m_{f,\boldsymbol{a}}(x_t) - \mu_{f,\boldsymbol{a}}(x_t, y_t)) && \text{(Definition of } R^+_T\text{)}\\
&\geq \sum_{j \in J'_{\neg Q}} \sum_{t \in M'_j} \big(\mu^m_{f,\boldsymbol{a}}(x_t) - \mu_{f,\boldsymbol{a}}(x_t, y_t)\big) && (\mu^m_{f,\boldsymbol{a}}(x_t) \geq \mu_{f,\boldsymbol{a}}(x_t, y_t))\\
&\geq \sum_{j \in J'_{\neg Q}} \sum_{t \in M'_j} (1 - \mu_{f,\boldsymbol{a}}(x_t, y_t)) && (\mu^m_{f,\boldsymbol{a}}(x_t) = 1 \text{ always})\\
&\geq \sum_{j \in J'_{\neg Q}} \sum_{t \in M'_j} \left(1 - 1 + \frac{L}{4f(T)}\right) && \text{(bound on } \mu_{f,\boldsymbol{a}}(x_t, y_t) \text{ for } t \in M'_j\text{)}\\
&= \sum_{j \in J'_{\neg Q}} \frac{L|M'_j|}{4f(T)} && \text{(Simplifying inner sum)}
\end{aligned}$$

Since $j \in J_{\neg Q}$, the mentor is not queried on any time step $t \in M_j$, so $y_t \in \{0, 1\}$ for all $t \in M_j$. Since the agent chooses one of two actions for each $t \in M_j$, the more frequent action must be chosen at least half of the time: $y_t = y^j$ for at least half of the time steps in M_j. Since $a_j \neq y^j$ for $j \in J'_{\neg Q}$, we have $y_t = y^j \neq a_j$ for those time steps, so $|M'_j| \geq |M_j|/2$. Thus

$$R^+_T \geq \sum_{j \in J'_{\neg Q}} \frac{L|M_j|}{8f(T)}$$

By Lemma A.4, Lemma A.5, and the union bound, with probability at least $1 - \exp\left(\frac{-T}{16f(T)}\right) - \exp\left(-\frac{f(T)-\mathbb{E}[|Q_T|]}{16}\right)$ we have $\sum_{j \in J'_{\neg Q}} |M_j| \geq \frac{T|J'_{\neg Q}|}{4f(T)}$ for all $j \in [f(T)]$ and $|J'_{\neg Q}| \geq \frac{f(T) - \mathbb{E}[|Q_T|]}{4}$. Assuming those inequalities hold, we have

$$\begin{aligned}
R^+_T &\geq \sum_{j \in J'_{\neg Q}} \frac{L|M_j|}{8f(T)}\\
&\geq \frac{L}{8f(T)} \cdot \frac{T|J'_{\neg Q}|}{4f(T)}\\
&\geq \frac{L}{8f(T)} \cdot \frac{T}{4f(T)} \cdot \frac{f(T) - \mathbb{E}[|Q_T|]}{4}\\
&= \frac{LT(f(T) - \mathbb{E}[|Q_T|])}{2^7 f(T)^2}
\end{aligned}$$

as required. □

For a given $f : \mathbb{N} \to \mathbb{N}$, define $\alpha_f(T) = \exp\left(\frac{-T}{16f(T)}\right) + \exp\left(-\frac{f(T)-\mathbb{E}[|Q_T|]}{16}\right)$ for brevity.

Theorem 4.1. *Any algorithm with sublinear queries has unbounded worst-case regret (both multiplicative and additive) as $T \to \infty$. Specifically,*

$$\sup_{\boldsymbol{\mu}, \pi^m} \mathbb{E}[R^\times_T], \ \sup_{\boldsymbol{\mu}, \pi^m} \mathbb{E}[R^+_T] \in \Omega\left(L\sqrt{\frac{T}{\sup_{\boldsymbol{\mu}, \pi^m} \mathbb{E}[|Q_T|]+1}}\right)$$

Proof. If the algorithm has sublinear queries, then there exists $g(T) \in o(T)$ such that $\sup_{\boldsymbol{\mu}, \pi^m} \mathbb{E}_{\boldsymbol{x}, \boldsymbol{y}}[|Q_T|] \leq g(T)$. Consider any such $g(T)$ satisfying $g(T) > 0$. Since this holds for every $\boldsymbol{\mu}$, it also holds in expectation over $\boldsymbol{a} \sim U(\{0,1\})^{f(T)}$, so $\mathbb{E}_{\boldsymbol{a}, \boldsymbol{x}, \boldsymbol{y}}[|Q_T|] = \mathbb{E}[|Q_T|] \leq g(T)$.

Next, Lemma A.2 gives us $g(T) \in o(\sqrt{g(T)T})$ and $\sqrt{g(T)T} \in o(T)$. Let $f(T) = \lceil \sqrt{g(T)T} \rceil$: then $f(T) \in \Theta(\sqrt{g(T)T})$, so $g(T) \in o(f(T))$ and $f(T) \in o(T)$. First, this implies that $\lim_{T \to \infty} \alpha_f(T) = 0$. Second, $g(T) \in o(f(T))$ implies that exists T_0 such that $g(T) \leq f(T)/2$ for all $T \geq T_0$. We also have $R_T^+ \geq 0$ always since $\mu_{f,\boldsymbol{a}}^m(x_t) \geq \mu_{f,\boldsymbol{a}}(x_t, y_t)$ always. Then for all $T \geq T_0$ we have

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\pi^m, \boldsymbol{a} \sim U(\{0,1\}^{f(T)})} \mathop{\mathbb{E}}_{\boldsymbol{x},\boldsymbol{y}} \left[R_T^+(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\mu}, \pi^m)\right] \\
&\geq \alpha_f(T) \cdot 0 + \left(1 - \alpha_f(T)\right) \left(\frac{LT(f(T) - \mathbb{E}[|Q_T|]}{2^7 f(T)^2}\right) && \text{(Lemma A.6 and } R_T^+ \geq 0) \\
&\geq \left(1 - \alpha_f(T)\right) \left(\frac{LT(f(T) - g(T))}{2^7 f(T)^2}\right) && (\mathbb{E}[Q_T] \leq g(T)) \\
&\geq \left(1 - \alpha_f(T)\right) \left(\frac{LT}{2^8 f(T)}\right) && (g(T) \leq f(T)/2)
\end{aligned}$$

Since $\lim_{T\to\infty} \alpha_f(T) = 0$,

$$\begin{aligned}
\sup_{\boldsymbol{\mu}, \pi^m} \mathop{\mathbb{E}}_{\boldsymbol{x},\boldsymbol{y}} \left[R_T^+(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\mu}, \pi^m)\right] &\geq \mathop{\mathbb{E}}_{\pi^m, \boldsymbol{a} \sim U(\{0,1\}^{f(T)})} \mathop{\mathbb{E}}_{\boldsymbol{x},\boldsymbol{y}} \left[R_T^+(\boldsymbol{x}, \boldsymbol{y}, (\mu_{f,\boldsymbol{a}}, \dots, \mu_{f,\boldsymbol{a}}), \pi^m)\right] \\
&\geq \left(1 - \alpha_f(T)\right) \left(\frac{LT}{2^8 f(T)}\right) \\
&\in \Omega\left(\frac{LT}{f(T)}\right) \\
&= \Omega\left(L\sqrt{\frac{T}{g(T)}}\right)
\end{aligned}$$

This holds for any $g(T) \in o(T)$ such that $\sup_{\boldsymbol{\mu}} \mathbb{E}[|Q_T|] \leq g(T)$ and $g(T) > 0$. Thus we can simply set $g(T) = \sup_{\boldsymbol{\mu}, \pi^m} \mathbb{E}[|Q_T|] + 1$, since $\sup_{\boldsymbol{\mu}, \pi^m} \mathbb{E}[|Q_T|]$ is indeed a function of only T.

Since $\mu_{f,\boldsymbol{a}}^m(x_t) \geq \mu_{f,\boldsymbol{a}}(x_t, y_t)$ for all $t \in [T]$, Lemma A.1 implies that

$$\begin{aligned}
\sup_{\boldsymbol{\mu}, \pi^m} \mathop{\mathbb{E}}_{\boldsymbol{x},\boldsymbol{y}} \left[R_T^\times(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\mu}, \pi^m)\right] &\geq \mathop{\mathbb{E}}_{\pi^m, \boldsymbol{a} \sim U(\{0,1\}^{f(T)})} \mathop{\mathbb{E}}_{\boldsymbol{x},\boldsymbol{y}} \left[R_T^\times(\boldsymbol{x}, \boldsymbol{y}, (\mu_{f,\boldsymbol{a}}, \dots, \mu_{f,\boldsymbol{a}}), \pi^m)\right] \\
&\in \Omega\left(L\sqrt{\frac{T}{\sup_{\boldsymbol{\mu}, \pi^m} \mathbb{E}[|Q_T|] + 1}}\right)
\end{aligned}$$

completing the proof. □

Corollary 4.1.1. *Even when $\mu_t^m(x) = 1$ for all t and x, any algorithm with sublinear queries satisfies*

$$\lim_{T\to\infty} \sup_{\boldsymbol{\mu}, \pi^m} \mathbb{E}\left[\prod_{t=1}^T \mu_t(x_t, y_t)\right] = 0$$

Proof. We have $\prod_{t=1}^T \mu_{f,\boldsymbol{a}}^m(x_t) = 1$ from our construction. Then Lemma A.1 implies that $R_T^\times \geq R_T^+$, so

$$\begin{aligned}
\exp(-R_T^+) &\geq \exp(-R_T^\times) \\
&= \exp\left(\log \prod_{t=1}^T \mu_{f,\boldsymbol{a}}(x_t, y_t) - \log 1\right) \\
&= \prod_{t=1}^T \mu_{f,\boldsymbol{a}}(x_t, y_t)
\end{aligned}$$

Then by Lemma A.6, with probability $1 - \alpha_f(T)$,

$$\prod_{t=1}^T \mu_{f,\boldsymbol{a}}(x_t, y_t) \leq \exp\left(-\frac{LT(f(T) - \mathbb{E}[|Q_T|])}{2^7 f(T)^2}\right) \leq \exp\left(-\frac{LT}{2^8 f(T)}\right)$$

Since $\prod_{t=1}^T \mu_{f,\boldsymbol{a}}(x_t, y_t) \le 1$ always,

$$\begin{aligned}
\lim_{T\to\infty} \mathop{\mathbb{E}}_{\pi^m, \boldsymbol{a}\sim U(\{0,1\}^{f(T)})} \mathop{\mathbb{E}}_{\boldsymbol{x},\boldsymbol{y}} \left[\prod_{t=1}^T \mu_{f,\boldsymbol{a}}(x_t, y_t)\right] &\le \lim_{T\to\infty} \mathop{\mathbb{E}}_{\pi^m, \boldsymbol{a}\sim U(\{0,1\}^{f(T)})} \mathop{\mathbb{E}}_{\boldsymbol{x},\boldsymbol{y}} \left[\prod_{t=1}^T \mu_{f,\boldsymbol{a}}(x_t, y_t)\right] \\
&\le \lim_{T\to\infty} \left(1 \cdot \alpha_f(T) + (1-\alpha_f(T)) \cdot \exp\left(-\frac{LT}{2^8 f(T)}\right)\right) \\
&\le \lim_{T\to\infty} (1-\alpha_f(T)) \cdot \lim_{T\to\infty} \exp\left(-\frac{LT}{2^8 f(T)}\right) \\
&= 1 \cdot \exp(-\infty) \\
&= 0
\end{aligned}$$

Since this upper bound holds for a randomly chosen $\mu_{f,\boldsymbol{a}}, \pi^m$, the same upper bound holds for a worst-case choice of $\boldsymbol{\mu}, \pi^m$ among $\boldsymbol{\mu}, \pi^m$ which satisfy $\mu_t^m(x) = 1$ for all $t \in [T], x \in \mathcal{X}$. Formally,

$$\lim_{T\to\infty} \sup_{\boldsymbol{\mu}, \pi^m:\, \mu_t(x)=1\, \forall t,x} \mathbb{E}\left[\prod_{t=1}^T \mu_t(x_t, y_t)\right] \le 0$$

Since $\prod_{t=1}^T \mu_t(x_t, y_t) \ge 0$ always, the inequality above holds with equality. □

B. Proof of Theorem 5.2

B.1. Context on Lemma 5.4

Before diving into the main proof, we provide some context on Lemma 5.4 from Section 5:

Lemma 5.4 (Lemma 3.5 in Russo et al., 2024)**.** *Assume $\tilde{\Pi}$ is finite. Then for any loss function $\ell : [T] \times \tilde{\Pi} \to [0,1]$ and query probability $p > 0$,* HEDGEWITHQUERIES *enjoys the regret bound*

$$\sum_{t=1}^T \mathbb{E}[\ell(t, \pi_t)] - \min_{\tilde{\pi}\in\tilde{\Pi}} \sum_{t=1}^T \mathbb{E}[\ell(t, \pi)] \le \frac{\log|\tilde{\Pi}|}{p^2}$$

where π_t is the policy chosen at time t and the expectation is over the randomness of the algorithm.

Lemma 5.4 is a restatement and simplification of Lemma 3.5 in Russo et al. (2024) with the following differences:

1. They parametrize their algorithm by the expected number of queries $\hat{k}$ instead of the query probability $p = \hat{k}/T$.
2. They include a second parameter k, which is the eventual target number of queries for their unconditional query bound. In our case, an expected query bound is sufficient, so we simply set $k = \hat{k}$.
3. They provide a second bound which is tighter for small k; that bound is less useful for us so we omit it.
4. Their "actions" correspond to policies in our setting, not actions in $\mathcal{Y}$. Their number of actions n corresponds to $|\tilde{\Pi}|$.
5. We include an expectation over both loss terms, while they only include an expectation over the agent's loss. This is because an adaptive adversary may choose the loss function for time t in a randomized manner. Since we eventually set $\ell(t, \pi) = \mathbf{1}(\pi(x_t) \ne \pi^m(x_t))$, the randomization in ℓ corresponds to the randomization in x_t.

Altogether, since Russo et al. (2024) set $\eta = \max\left(\frac{1}{T}\sqrt{\frac{\hat{k}\log n}{2}}, \frac{k\hat{k}}{\sqrt{2}T^2}\right)$, we end up with $\eta = \max\left(\sqrt{\frac{p\log|\tilde{\Pi}|}{2T}}, \frac{p^2}{\sqrt{2}}\right)$. Algorithm 2 provides precise pseudocode for the HEDGEWITHQUERIES algorithm to which Lemma 5.4 refers.

B.2. Main proof

We use the following notation throughout the proof:

1. For each $t \in [T]$, let S_t refer to the value of S at the start of time step t.
2. Let $M_T = \{t \in [T] : \pi_t(x_t) \ne \pi^m(x_t)\}$ be the set of time steps where Hedge's proposed action doesn't match the mentor's. Note that $|M_T|$ upper bounds the number of mistakes the algorithm makes (the number of mistakes could be smaller, since the algorithm sometimes queries instead of taking action $\pi_t(x_t)$).

Algorithm 2 A variant of the Hedge algorithm which only observes losses in response to queries.

```
function HedgeWithQueries($p \in (0,1]$, finite policy class $\tilde{\Pi}$, unknown $\ell : [T] \times \tilde{\Pi} \to [0,1]$)
  $w(\pi) \leftarrow 1$ for all $\pi \in \tilde{\Pi}$
  $\eta \leftarrow \max\left(\sqrt{\frac{p \log |\tilde{\Pi}|}{2T}},\ \frac{p^2}{\sqrt{2}}\right)$
  for $t$ from $1$ to $T$ do
    with probability $p$ : hedgeQuery ← true
    with probability $1-p$ : hedgeQuery ← false
    if hedgeQuery then
      Query and observe $\ell(t,\pi)$ for all $\pi \in \tilde{\Pi}$
      $\ell^* \leftarrow \min_{\pi \in \tilde{\Pi}} \ell(t,\pi)$
      $w(\pi) \leftarrow w(\pi) \cdot \exp(-\eta(\ell(t,\pi) - \ell^*))$ for all $\pi \in \tilde{\Pi}$
      Select policy $\arg\min_{\pi \in \tilde{\Pi}} \ell(t,\pi)$
    else
      $P(\pi) \leftarrow w(\pi) / \sum_{\pi' \in \tilde{\Pi}} w(\pi')$ for all $\pi \in \tilde{\Pi}$
      Sample $\pi_t \sim P$
      Select policy $\pi_t$
```

3. For $S \subseteq \mathcal{X}$, let $\text{vol}(S)$ denote the n-dimensional Lebesgue measure of S.
4. With slight abuse of notation, we will use inequalities of the form $f(T) \le g(T) + O(h(T))$ to mean that there exists a constant C such that $f(T) \le g(T) + Ch(T)$.
5. We will use "Case 1" to refer to finite VC dimension and σ-smooth $\boldsymbol{x}$ and "Case 2" to refer to finite Littlestone dimension.

Lemma B.1. *Let $\mathcal{Y} = \{0,1\}$. Assume $\pi^m \in \Pi$ where either (1) Π has finite VC dimension d and $\boldsymbol{x}$ is σ-smooth, or (2) Π has finite Littlestone dimension d. Then for any $T \in \mathbb{N}$ and $\varepsilon \ge 1/T$,*[12] *Algorithm 1 satisfies*

$$\mathbb{E}[|M_T|] \in O\left(\frac{d}{\sigma} T\varepsilon \log(T + 1/\varepsilon)\right)$$

Proof. Define $\ell : [T] \times \tilde{\Pi} \to [0,1]$ by $\ell(t,\pi) = \mathbf{1}(\pi(x_t) \ne \pi^m(x_t))$, and let w^h and π_t^h denote the values of w and π_t respectively in HedgeWithQueries, while w and π_t refer to the variables in Algorithm 1. Then w and w^h evolve in the exact same way, so the distributions of π_t and π_t^h coincide. Also, $\varepsilon \ge 1/T$ implies that $p = 1/\sqrt{\varepsilon T} \in (0,1]$. Thus by Lemma 5.4,

$$\begin{aligned}\sum_{t=1}^T \mathbb{E}[\ell(t,\pi_t)] - \min_{\tilde{\pi} \in \tilde{\Pi}} \sum_{t=1}^T \mathbb{E}[\ell(t,\tilde{\pi})] &\le \frac{\log |\tilde{\Pi}|}{p^2} \\ &= T\varepsilon \log |\tilde{\Pi}|\end{aligned}$$

Since $|M_T| = \sum_{t=1}^T \mathbf{1}(\pi_t(x_t) \ne \pi^m(x_t)) = \sum_{t=1}^T \ell(t,\pi_t)$, we have

$$\mathbb{E}[|M_T|] \le T\varepsilon \log |\tilde{\Pi}| + \min_{\tilde{\pi} \in \tilde{\Pi}} \sum_{t=1}^T \mathbb{E}[\mathbf{1}(\tilde{\pi}(x_t) \ne \pi^m(x_t))]$$

Case 1: Since $\tilde{\Pi}$ is a smooth ε-cover and $\pi^m \in \Pi$, Lemma 5.1 implies that $\mathbb{E}[\mathbf{1}(\tilde{\pi}(x_t) \ne \pi^m(x_t))] \le \varepsilon/\sigma$ for any $\tilde{\pi} \in \tilde{\Pi}$. Since $|\tilde{\Pi}| \le (41/\varepsilon)^d$ by construction (and such a $\tilde{\Pi}$ is guaranteed to exist by Lemma 5.2), we get

$$\begin{aligned}\mathbb{E}[|M_T|] &\le T\varepsilon \log((41/\varepsilon)^d) + \min_{\tilde{\pi} \in \tilde{\Pi}} \sum_{t=1}^T \frac{\varepsilon}{\sigma} \\ &= dT\varepsilon \log(41/\varepsilon) + \frac{T\varepsilon}{\sigma}\end{aligned}$$

[12] Note that this lemma omits the assumption of $\varepsilon \le (\frac{\mu_0^m}{2L})^n$, since we do not need it for this lemma, and we would like to apply this lemma in the multi-action case without that assumption.

$$\in O\left(\frac{d}{\sigma} T\varepsilon \log(T + 1/\varepsilon)\right)$$

Case 2: Since $\tilde{\Pi}$ is an adversarial cover of Π and $\pi^m \in \Pi$, there exists $\tilde{\pi} \in \tilde{\Pi}$ such that $\sum_{t=1}^T \mathbf{1}(\tilde{\pi}(x_t) \neq \pi^m(x_t)) = 0$. Since $|\tilde{\Pi}| \le (eT/d)^d$ (with such a $\tilde{\Pi}$ guaranteed to exist by Lemma 5.3),

$$\begin{aligned}
\mathbb{E}[|M_T|] &\le T\varepsilon \log|\tilde{\Pi}| + \min_{\tilde{\pi}\in\tilde{\Pi}} \sum_{t=1}^T \mathbf{1}(\tilde{\pi}(x_t) \neq \pi^m(x_t)) \\
&\le T\varepsilon d \ln(eT/d) \\
&\in O\left(\frac{d}{\sigma} T\varepsilon \log(T + 1/\varepsilon)\right)
\end{aligned}$$

as required. □

Lemma B.2. *For all $t \in [T]$, $\mu_t(x_t, y_t) \ge \mu_t^m(x_t) - L\varepsilon^{1/n}$.*

Proof. Consider an arbitrary $t \in [T]$. If $t \in Q_T$, then $\mu_t(x_t, y_t) = \mu_t^m(x_t)$ trivially, so assume $t \notin Q_T$. Let $(x', y') = \arg\min_{(x,y)\in S_t : \pi_t(x_t)=y} ||x_t - x||$. Since $t \notin Q_T$, we must have $||x_t - x'|| \le \varepsilon^{1/n}$.

We have $y' = \pi^m(x')$ by construction of S_t and $\pi_t(x_t) = y'$ by construction of y'. Combining these with the local generalization assumption, we get

$$\begin{aligned}
\mu_t(x_t, y_t) &= \mu_t(x_t, \pi_t(x_t)) \\
&= \mu_t(x_t, \pi^m(x')) \\
&\ge \mu_t^m(x_t) - L||x_t - x'|| \\
&\ge \mu_t^m(x_t) - L\varepsilon^{1/n}
\end{aligned}$$

as required. □

Lemma B.3. *Under the conditions of Theorem 5.1, Algorithm 1 satisfies*

$$\begin{aligned}
\mathbb{E}\left[R_T^\times\right] &\in O\left(\frac{dL}{\sigma\mu_0^m} T\varepsilon^{1+1/n} \log(T + 1/\varepsilon)\right) \\
\mathbb{E}\left[R_T^+\right] &\in O\left(\frac{dL}{\sigma} T\varepsilon^{1+1/n} \log(T + 1/\varepsilon)\right)
\end{aligned}$$

Proof. We first claim that $y_t = \pi^m(x_t)$ for all $t \notin M_T$. If $t \in Q_T$, the claim is immediate. If not, we have $y_t = \pi_t(x_t)$ by the definition of the algorithm and $\pi_t(x_t) = \pi^m(x_t)$ by the definition of $t \notin M_T$. Thus $\mu_t(x_t, y_t) = \mu_t^m(x_t)$ for $t \notin M_T$. For $t \in M_T$, Lemma B.2 implies that $\mu_t^m(x_t) - \mu_t(x_t, y_t) \le L\varepsilon^{1/n}$, so

$$\begin{aligned}
R_T^+ &= \sum_{t\in M_T} (\mu_t^m(x_t) - \mu_t(x_t, y_t)) \\
&\le \sum_{t\in M_T} L\varepsilon^{1/n} \\
&= |M_T| L\varepsilon^{1/n} \qquad (1)
\end{aligned}$$

Since $\varepsilon \le \left(\frac{\mu_0^m}{2L}\right)^n$ by assumption, we have $L\varepsilon^{1/n} \le \mu_0^m/2$ and thus $\mu_t(x_t, y_t) \ge \mu_t^m(x_t) - L\varepsilon^{1/n} \ge \mu_0^m - \mu_0^m/2 = \mu_0^m/2 > 0$ for all $t \in [T]$. Then by Lemma A.1,

$$R_T^\times \le \frac{R_T^+}{\mu_0^m/2} \le \frac{2|M_T|L\varepsilon^{1/n}}{\mu_0^m} \qquad (2)$$

Taking the expectation and applying Lemma B.1 to Equations 1 and 2 completes the proof. □

Definition B.1. Let $(K, ||\cdot||)$ be a normed vector space and let $\delta > 0$. Then a multiset $S \subseteq K$ is a δ-packing of K if for all $a, b \in S$, $||a - b|| > \delta$. The δ-packing number of K, denoted $\mathcal{M}(K, ||\cdot||, \delta)$, is the maximum cardinality of any δ-packing of K.

We only consider the Euclidean distance norm, so we just write $M(K, ||\cdot||, \delta) = M(K, \delta)$.

Lemma B.4 (Theorem 14.2 in (Wu, 2020))**.** *If $K \subset \mathbb{R}^n$ is convex, bounded, and contains a ball with radius $\delta > 0$, then*

$$\mathcal{M}(K, \delta) \leq \frac{3^n \operatorname{vol}(K)}{\delta^n \operatorname{vol}(B)}$$

where B is a unit ball.

Lemma B.5 (Jung's Theorem (Jung, 1901))**.** *If $S \subset \mathbb{R}^n$ is compact, then there exists a closed ball with radius at most $\operatorname{diam}(S)\sqrt{\frac{n}{2(n+1)}}$ containing S.*

Lemma B.6. *Under the conditions of Theorem 5.1, Algorithm 1 satisfies*

$$\mathbb{E}[|Q_T|] \in O\left(\sqrt{\frac{T}{\varepsilon}} + \frac{d}{\sigma} T\varepsilon \log(T + 1/\varepsilon) + \frac{\mathbb{E}[\operatorname{diam}(\boldsymbol{x})^n]}{\varepsilon}\right)$$

Proof. If $t \in Q_T$, then either $\texttt{hedgeQuery} = \texttt{true}$ or $\min_{(x,y)\in S_t:\pi_t(x_t)=y} ||x_t - x|| > \varepsilon^{1/n}$ (or both). The expected number of time steps with $\texttt{hedgeQuery} = \texttt{true}$ is $pT = \sqrt{T/\varepsilon}$. Let $\hat{Q} = \{t \in Q_T : \min_{(x,y)\in S_t:\pi_t(x_t)=y} ||x_t - x|| > \varepsilon^{1/n}\}$. We further subdivide $\hat{Q}$ into $\hat{Q}_1 = \{t \in \hat{Q} : \pi_t(x_t) \neq \pi^m(x_t)\}$ and $\hat{Q}_2 = \{t \in \hat{Q} : \pi_t(x_t) = \pi^m(x_t)\}$. Since $\hat{Q}_1 \subseteq M_T$, Lemma B.1 implies that $\mathbb{E}[|\hat{Q}_1|] \in O\left(\frac{d}{\sigma} T\varepsilon \log(T + 1/\varepsilon)\right)$.

Next, fix a $y \in \mathcal{Y}$ and let $X_y = \{x \in \boldsymbol{x} : \pi^m(x) = y\}$ be the multiset of observed inputs whose mentor action is y. Also let $\hat{X}_2 = \{x_t : t \in \hat{Q}_2\}$ be the multiset of inputs associated with time steps in $\hat{Q}_2$. Note that $|\hat{X}_2| = |\hat{Q}_2|$, since $\hat{X}_2$ is a multiset. We claim that $\hat{X}_2 \cap X_y$ is an $\varepsilon^{1/n}$-packing of X_y. Suppose instead that there exists $x, x' \in \hat{X}_2 \cap X_y$, with $||x - x'|| \leq \varepsilon^{1/n}$. WLOG assume x was queried after x' and let t be the time step on which x was queried. Since $x' \in \hat{X}_2$, this implies $(x', \pi^m(x')) \in S_t$. Also, since $x, x' \in \hat{X}_2$ we have $\pi_t(x_t) = \pi^m(x_t) = y = \pi^m(x')$. Therefore

$$\min_{(x'',y'')\in S_t:y''=\pi_t(x_t)} ||x_t - x''|| \leq ||x_t - x'|| \leq \varepsilon^{1/n}$$

which contradicts $t \in \hat{Q}$. Thus $\hat{X}_2 \cap X_y$ is an $\varepsilon^{1/n}$-packing of X_y.

By Lemma B.5, there exists a ball B_1 of radius $R := \operatorname{diam}(\boldsymbol{x})\sqrt{\frac{n}{2(n+1)}}$ which contains $\boldsymbol{x}$. Let B_2 be the ball with the same center as B_1 but with radius $\max(R, \varepsilon^{1/n})$. Since $X_y \subset \boldsymbol{x} \subset B_1 \subset B_2$ and $\hat{X}_2 \cap X_y$ is an $\varepsilon^{1/n}$-packing of X_y, $\hat{X}_2 \cap X_y$ is also an $\varepsilon^{1/n}$-packing of B_2. Also, B_2 must contain a ball of radius $\varepsilon^{1/n}$, so Lemma B.4 implies that

$$\begin{aligned}
|\hat{X}_2 \cap X_y| &\leq \mathcal{M}(B_2, \varepsilon^{1/n}) \\
&\leq \frac{3^n \operatorname{vol}(B_2)}{\varepsilon \operatorname{vol}(B)} \\
&= \left(\max(R, \varepsilon^{1/n})\right)^n \frac{3^n \operatorname{vol}(B)}{\varepsilon \operatorname{vol}(B)} \\
&= \max\left(\operatorname{diam}(\boldsymbol{x})^n \left(\frac{n}{2(n+1)}\right)^{n/2}, \varepsilon\right) \frac{3^n}{\varepsilon} \\
&\leq O\left(\frac{\operatorname{diam}(\boldsymbol{x})^n}{\varepsilon} + 1\right)
\end{aligned}$$

(The $+1$ is necessary for now since $\operatorname{diam}(\boldsymbol{x})$ could theoretically be zero.) Since $\hat{X}_2 \subseteq \{x_1, \ldots, x_T\} \subseteq \cup_{y\in\mathcal{Y}} X_y$, we have $|\hat{X}_2| \leq \sum_{y\in\mathcal{Y}} |\hat{X}_2 \cap X_y|$ by the union bound. Therefore

$$\mathbb{E}[|Q_T|] \leq \sqrt{\frac{T}{\varepsilon}} + \mathbb{E}[|\hat{Q}|]$$

$$
\begin{aligned}
&= \sqrt{\frac{T}{\varepsilon}} + \mathbb{E}[|\hat{Q}_1|] + \mathbb{E}[|\hat{Q}_2|] \\
&= \sqrt{\frac{T}{\varepsilon}} + \mathbb{E}[|\hat{Q}_1|] + \mathbb{E}[|\hat{X}_2|] \\
&\leq \sqrt{\frac{T}{\varepsilon}} + \mathbb{E}[|\hat{Q}_1|] + \mathbb{E}\left[\sum_{y \in \mathcal{Y}} |\hat{X}_2 \cap X_y|\right] \\
&\leq \sqrt{\frac{T}{\varepsilon}} + O\left(\frac{d}{\sigma} T \varepsilon \log(T + 1/\varepsilon)\right) + \sum_{y \in \mathcal{Y}} O\left(\frac{\mathbb{E}[\mathrm{diam}(\boldsymbol{x})^n]}{\varepsilon} + 1\right) \\
&\leq \sqrt{\frac{T}{\varepsilon}} + O\left(\frac{d}{\sigma} T \varepsilon \log(T + 1/\varepsilon)\right) + |\mathcal{Y}| \cdot O\left(\frac{\mathbb{E}[\mathrm{diam}(\boldsymbol{x})^n]}{\varepsilon} + 1\right) \\
&\leq O\left(\sqrt{\frac{T}{\varepsilon}} + \frac{d}{\sigma} T \varepsilon \log(T + 1/\varepsilon) + \frac{\mathbb{E}[\mathrm{diam}(\boldsymbol{x})^n]}{\varepsilon}\right)
\end{aligned}
$$

as required. □

Theorem 5.1 follows from Lemmas B.3 and B.6:

Theorem 5.1. *Let $\mathcal{Y} = \{0, 1\}$. Assume $\pi^m \in \Pi$ where either (1) Π has finite VC dimension d and $\boldsymbol{x}$ is σ-smooth, or (2) Π has finite Littlestone dimension d. Then for any $T \in \mathbb{N}$ and $\varepsilon \in \left[\frac{1}{T}, \left(\frac{\mu_0^m}{2L}\right)^n\right]$, Algorithm 1 satisfies*

$$
\begin{aligned}
\mathbb{E}\left[R_T^\times\right] &\in O\left(\frac{dL}{\sigma \mu_0^m} T \varepsilon^{1+1/n} \log(T + 1/\varepsilon)\right) \\
\mathbb{E}\left[R_T^+\right] &\in O\left(\frac{dL}{\sigma} T \varepsilon^{1+1/n} \log(T + 1/\varepsilon)\right) \\
\mathbb{E}[|Q_T|] &\in O\left(\sqrt{\frac{T}{\varepsilon}} + \frac{d}{\sigma} T \varepsilon \log(T{+}1/\varepsilon) + \frac{\mathbb{E}[\mathrm{diam}(\boldsymbol{x})^n]}{\varepsilon}\right)
\end{aligned}
$$

We then perform some arithmetic to obtain Theorem 5.2:

Theorem 5.2. *Let $\mathcal{Y} = \{0, 1\}$. Assume $\pi^m \in \Pi$ where either (1) Π has finite VC dimension d and $\boldsymbol{x}$ is σ-smooth or (2) Π has finite Littlestone dimension d. Then for any $T \in \mathbb{N}$, Algorithm 1 with $\varepsilon = T^{\frac{-2n}{2n+1}}$ satisfies*

$$
\begin{aligned}
\mathbb{E}\left[R_T^\times\right] &\in O\left(\frac{dL}{\sigma \mu_0^m} T^{\frac{-1}{2n+1}} \log T\right) \\
\mathbb{E}\left[R_T^+\right] &\in O\left(\frac{dL}{\sigma} T^{\frac{-1}{2n+1}} \log T\right) \\
\mathbb{E}[|Q_T|] &\in O\left(T^{\frac{4n+1}{4n+2}} \left(\frac{d}{\sigma} \log T + \mathbb{E}[\mathrm{diam}(\boldsymbol{x})^n]\right)\right)
\end{aligned}
$$

Proof. We have

$$
\begin{aligned}
\mathbb{E}\left[R_T^\times\right] &\in O\left(\frac{dL}{\sigma \mu_0^m} T^{1-\frac{2n}{2n+1}-\frac{2}{2n+1}} \left(\log T + \log(T^{\frac{2n}{2n+1}})\right)\right) \\
&= O\left(\frac{dL}{\sigma \mu_0^m} T^{\frac{-1}{2n+1}} \log T\right)
\end{aligned}
$$

and similarly for $\mathbb{E}[R_T^+]$. For $\mathbb{E}[|Q_T|]$,

$$
\mathbb{E}[|Q_T|] \in O\left(\sqrt{T^{1+\frac{2n}{2n+1}}} + \frac{d}{\sigma} T^{1-\frac{-2n}{2n+1}} \left(\log T + \log(T^{\frac{2n}{2n+1}})\right) + T^{\frac{2n}{2n+1}} \mathbb{E}[\mathrm{diam}(\boldsymbol{x})^n]\right)
$$

$$
\begin{aligned}
&= O\left(T^{\frac{2n+0.5}{2n+1}} + \frac{d}{\sigma} T^{\frac{1}{2n+1}} \log T + T^{\frac{2n}{2n+1}}\, \mathbb{E}[\operatorname{diam}(\boldsymbol{x})^n]\right)\\
&\leq O\left(T^{\frac{4n+1}{4n+2}} \left(\frac{d}{\sigma} \log T + \mathbb{E}[\operatorname{diam}(\boldsymbol{x})^n]\right)\right)
\end{aligned}
$$

□

B.3. Adaptive adversaries

If x_t is allowed to depend on the events of prior time steps, we say that the adversary is adaptive. In contrast, a non-adaptive or "oblivious" adversary must choose the entire input upfront. This distinction is not relevant for deterministic algorithms, since an adversary knows exactly how the algorithm will behave for any input. In other words, the adversary gains no new information during the execution of the algorithm. For randomized algorithms, an adaptive adversary can base the choice of x_t on the results of randomization on previous time steps (but not on the current time step), while an oblivious adversary cannot.

In the standard online learning model, Hedge guarantees sublinear regret against both oblivious and adaptive adversaries (Chapter 5 of Slivkins et al. (2019) or Chapter 21 of Shalev-Shwartz & Ben-David (2014)). However, Russo et al. (2024) state their result only for oblivious adversaries. In order for our overall proof of Theorem 5.1 to hold for adaptive adversaries, Lemma 5.4 (Lemma 3.5 in Russo et al., 2024) must also hold for adaptive adversaries. In this section, we argue why the proof of Lemma 5.4 (Lemma 3.5 in their paper) goes through for adaptive adversaries as well. For the rest of Appendix B.3, lemma numbers refer to the numbering in Russo et al. (2024).

The importance of independent queries. Recall from Appendix B.1 that Russo et al. (2024) allow two separate parameters k and $\hat{k}$, which we unify for simplicity. Recall also that Lemma 3.5 refers to the variant of Hedge which queries with probability $p = \hat{k}/T = k/T$ independently on each time step (Algorithm 2). More precisely, on each time step t, the algorithm samples $X_t \sim \text{Bernoulli}(p)$ and queries if $X_t = 1$. The key idea is that X_t is independent of events on previous time steps. Thus even conditioning on the history up to time t, for any random variable Y_t we can write

$$
\mathbb{E}[Y_t] = (1-p)\,\mathbb{E}[Y_t \mid X_t = 0] + p\,\mathbb{E}[Y_t \mid X_t = 1]
$$

This insight immediately extends Observation 3.3 to adaptive adversaries (with the minor modification that queries are now issued independently with probability p on each time step instead of issuing k uniformly distributed queries). Specifically, using the notation from Russo et al. (2024) where i_t is the action chosen at time t, i_t^0 is the action chosen at time t if a query is not issued, and i_t^* is the optimal action at time t, we have

$$
\begin{aligned}
\mathbb{E}[\ell_t(i_t)] &= (1-p)\,\mathbb{E}[\ell_t(i_t^0)] + p\,\mathbb{E}[\ell_t(i_t^*)]\\
&= \left(1 - \frac{k}{T}\right)\mathbb{E}[\ell_t(i_t^0)] + \frac{k}{T}\,\mathbb{E}[\ell_t(i_t^*)]
\end{aligned}
$$

The same logic applies to other statements like $\mathbb{E}[\hat{\ell}_t(i) \mid X_{\leq t-1}, I_{\leq t-1}] = \ell_t(i) - \ell_t(i_t^*)$ and immediately extends those statements to adaptive adversaries as well.

Applying Observation 3.3. The other tricky part of the proof is applying Observation 3.3 using a new loss function $\hat{\ell}$ defined by $\hat{\ell}_t(i) = \frac{T}{k}(\ell_t(i) - \ell_t(i_t^*))\mathbf{1}(X_t = 1)$. To do so, we must argue that standard Hedge run on $\hat{\ell}$ is the "counterpart without queries" of HedgeWithQueries. Specifically, both algorithms must have the same weight vectors on every time step, and the only difference should be that HedgeWithQueries takes the optimal action on each time step independently with probability p (and otherwise behaves the same as standard Hedge). On time steps with $X_t = 0$, standard Hedge observes $\hat{\ell}_t(i) = 0$ for all actions i and thus makes no updates, and HedgeWithQueries makes no updates by definition. On time steps with $X_t = 1$, both algorithms perform the typical updates $w_{t+1}(i) = w_t(i) \cdot \exp(-\eta(\hat{\ell}_t(i) - \hat{\ell}_t(i_t^*)))$. Thus the weight vectors are the same for both algorithms on every time step. Furthermore, HedgeWithQueries takes the optimal action at time t iff $X_t = 1$, which occurs independently with probability p on each time step. Thus standard Hedge run on $\hat{\ell}$ is the "counterpart without queries" of HedgeWithQueries. Note that since $\hat{\ell}$ is itself a random variable, the law of iterated expectation is necessary to formalize this.

Algorithm 3 extends Algorithm 1 to many actions.

```
Inputs: $T \in \mathbb{N},\ \varepsilon \in \mathbb{R}_{>0},\ d \in \mathbb{N},$ policy class $\Pi$
for $y \in \mathcal{Y}$ do
  if $\Pi_y$ has VC dimension $d$ then
    $\tilde{\Pi}_y \leftarrow$ any smooth $\varepsilon$-cover of $\Pi_y$ of size at most $(41/\varepsilon)^d$
  else if $\Pi_y$ has Littlestone dimension $d$ then
    $\tilde{\Pi}_y \leftarrow$ any adversarial cover of $\Pi_y$ of size at most $(eT/d)^d$
for $t$ from $1$ to $T$ do
  for $y \in \mathcal{Y}$ do
    $b_t^y \leftarrow$ action at time $t$ from the copy of Algorithm 1 running on $\Pi_y$ (with the same $T, \varepsilon, d$)
  if $b_t^y \neq \tilde{y}\ \forall y \in \mathcal{Y}$ and $\exists a \in \mathcal{Y} : b_t^y = 1$ then
    Take any action $y$ with $b_t^y = 1$
  else
    Take an arbitrary action in $\mathcal{Y}$
```

The rest of the proof. The other elements of the proof of Lemma 3.5 are as follows:

1. Lemma 3.1, which analyzes the standard version of Hedge (i.e., no queries and losses are observed on every time step).
2. Applying Lemma 3.1 to $\hat{\ell}$.
3. Arithmetic and rearranging terms.

The proof of Lemma 3.1 relies on simple arithmetic properties of the Hedge weights. Regardless of the adversary's behavior, $\hat{\ell}$ is a well-defined loss function, so Lemma 3.1 can be applied. Step 3 clearly has no dependence on the type of adversary. Thus we conclude that Lemma 3.5 extends to adaptive adversaries.

C. Generalizing Theorem 5.2 to many actions

We use the standard "one versus rest" reduction (see, e.g., Chapter 29 of Shalev-Shwartz & Ben-David, 2014). For each action y, we will learn a binary classifier which predicts whether action y is the mentor's action. Formally, for each $y \in \mathcal{Y}$, define the policy class $\Pi_y = \{\pi_y : \pi \in \Pi \text{ and } \pi_y(x) = \mathbf{1}(\pi(x) = y)\ \forall x \in \mathcal{X}\}$. In words, for each policy $\pi : \mathcal{X} \to \mathcal{Y}$ in Π, there exists a policy $\pi_y : \mathcal{X} \to \{0, 1\}$ in Π_y such that $\pi_y(x) = \mathbf{1}(\pi(x) = y)$ for all $x \in \mathcal{X}$.

Algorithm 3 runs one copy of our binary-action algorithm (Algorithm 1) for each action $y \in \mathcal{Y}$. At each time step t, the copy for action y returns an action b_t^y, with $b_t^y = 1$ indicating a belief that $y = \pi^m(x_t)$ and $b_t^y = 0$ indicating a belief that $y \neq \pi^m(x_t)$. (Note that $b_t^y = \tilde{y}$ is also possible, indicating that the mentor was queried.)

The key idea is that if b_t^y is correct for each action y, there will be exactly one y such that $b_t^y = 1$, and specifically it will be $y = \pi^m(x_t)$. Thus we are guaranteed to take the mentor's action on such time steps. The analysis for Theorem 5.2 (specifically, Lemma B.1) bounds the number of time steps when a given copy of Algorithm 1 is incorrect, so by the union bound, the number of time steps where *any* copy is incorrect is $|\mathcal{Y}|$ times that bound. That in turn bounds the number of time steps where Algorithm 3 takes an action other than the mentor's. Similarly, the number of queries made by Algorithm 3 is at most $|\mathcal{Y}|$ times the bound from Theorem 5.2. The result is the following theorem:

Theorem C.1. *Assume $\pi^m \in \Pi$ where either (1) Π_y has finite VC dimension d and $\boldsymbol{x}$ is σ-smooth or (2) Π_y has finite Littlestone dimension d for all $y \in \mathcal{Y}$. Then for any $T \in \mathbb{N}$, Algorithm 3 with $\varepsilon = T^{\frac{-2n}{2n+1}}$ satisfies*

$$\mathbb{E}\left[R_T^{\times}\right] \in O\left(\frac{|\mathcal{Y}| dL}{\sigma \mu_0^m} T^{\frac{-1}{2n+1}} \log T\right)$$

$$\mathbb{E}\left[R_T^{+}\right] \in O\left(\frac{|\mathcal{Y}| dL}{\sigma} T^{\frac{-1}{2n+1}} \log T\right)$$

$$\mathbb{E}[|Q_T|] \in O\left(|\mathcal{Y}| T^{\frac{4n+1}{4n+2}} \left(\frac{d}{\sigma} \log T + \mathbb{E}[\mathrm{diam}(\boldsymbol{x})^n]\right)\right)$$

We use the following terminology and notation in the proof of Theorem C.1:

1. We refer to the copy of Algorithm 1 running on Π_y as "copy y of Algorithm 1".
2. Recall that S_t refers to the value of S at the start of time step t in Algorithm 1. Let π_t^y and S_t^y refer to the values of π_t and S_t for copy y of Algorithm 1.
3. Let $\pi^{my} : \mathcal{X} \to \{0, 1\}$ be the policy defined by $\pi^{my}(x) = \mathbf{1}(\pi^m(x) = y)$. Note that querying the mentor tells the agent $\pi^m(x_t)$, which allows the agent to compute $\pi^{my}(x_t)$: this is necessary when Algorithm 1 queries while running on some Π_y.
4. Let $M_T^y = \{t \in [T] : b_t^y \neq \pi^{my}(x_t)\}$ be the set of time steps where π_t^y does not correctly determine whether the mentor would take action y and let $M_T = \{t \in [T] : y_t \neq \pi^m(x_t)\}$ be the set of time steps where the agent's action does not match the mentor's.

Lemma C.1. *We have* $|M_T| \leq \sum_{y \in \mathcal{Y}} |M_T^y|$.

Proof. We claim that $M_T \subseteq \cup_{y \in \mathcal{Y}} M_T^y$. Suppose the opposite: then there exists $t \in M_T$ such that $b_t^y = \pi^{my}(x_t)$ for all $y \in \mathcal{Y}$. Since $\pi^m(x_t) \in \mathcal{Y}$, there is exactly one $y \in \mathcal{Y}$ such that $\mathbf{1}(\pi^m(x_t) = y) = \pi^{my}(x_t) = b_t^y = 1$. Specifically, this holds for $y = \pi^m(x_t)$. But then Algorithm 3 takes action $\min\{y \in \mathcal{Y} : b_t^y = 1\} = \pi^m(x_t)$, which contradicts $t \in M_T$. Therefore $M_T \subseteq \cup_{y \in \mathcal{Y}} M_T^y$, and applying the union bound completes the proof. □

Lemma C.2. *For all* $t \in [T]$, $\mu_t^m(x_t) - \mu_t(x_t, y_t) \leq L\varepsilon^{1/n}$.

Proof. The argument is similar to the proof of Lemma B.2. If $\mu_t^m(x_t) \neq \mu_t(x_t, y_t)$, then $y_t = y$ for some $y \in \mathcal{Y}$ where $b_t^y = 1$. Therefore copy y of Algorithm 1 did not query at time t and $\pi_t^y(x_t) = 1$. Let $(x', y') = \arg\min_{(x,y) \in S_t^y : \pi_t^y(x_t) = y} ||x_t - x||$. Then $||x_t - x'|| \leq \varepsilon^{1/n}$ and $y' = \pi_t^y(x_t) = 1$.

By construction of S_t^y, $y' = \pi^{my}(x')$ so $\pi^{my}(x') = 1$ which implies $\pi^m(x') = y$. Then by the local generalization assumption,

$$\mu_t(x_t, y_t) = \mu_t(x_t, y) = \mu_t(x_t, \pi^m(x')) \geq \mu_t^m(x_t) - L||x_t - x'|| \geq \mu_t^m(x_t) - L\varepsilon^{1/n}$$

as required. □

We now proceed to the proof of Theorem C.1.

Proof of Theorem C.1. Theorem 5.2 implies that each copy of Algorithm 1 makes $O\big(T^{\frac{4n+1}{4n+2}}\big(\frac{d}{\sigma}\log T + \mathbb{E}[\operatorname{diam}(\boldsymbol{x})^n]\big)\big)$ queries in expectation. Thus by linearity of expectation the expected number of queries made by Algorithm 3 is $O\big(|\mathcal{Y}|T^{\frac{4n+1}{4n+2}}\big(\frac{d}{\sigma}\log T + \mathbb{E}[\operatorname{diam}(\boldsymbol{x})^n]\big)\big)$.[13] Similar to the proof of Lemma B.3, we have

$$\begin{aligned}
R_T^+ &= \sum_{t \in M_T} (\mu_t^m(x_t) - \mu_t(x_t, y_t)) && (\mu_t^m(x_t) = \mu_t(x_t, y_t) \text{ for all } t \notin M_T)\\
&\leq \sum_{t \in M_T} L\varepsilon^{1/n} && (\text{Lemma C.2})\\
&= |M_T| L\varepsilon^{1/n} && (\text{Simplifying sum})\\
&\leq L\varepsilon^{1/n} \sum_{y \in \mathcal{Y}} |M_T^y| && (\text{Lemma C.1})
\end{aligned}$$

Since each copy satisfies the conditions of Lemma B.1, we get

$$\mathbb{E}[R_T^+] \leq L\varepsilon^{1/n} \sum_{y \in \mathcal{Y}} O\left(\frac{d}{\sigma} T\varepsilon \log(1/\varepsilon) \log T\right) = O\left(|\mathcal{Y}| L\varepsilon^{1/n} \frac{d}{\sigma} T\varepsilon \log(1/\varepsilon) \log T\right)$$

Since $\lim_{T \to \infty} \varepsilon = 0$, there exists T_0 such that $L\varepsilon^{1/n} \leq \mu_0^m/2$ for all $T \geq T_0$. Then by Lemma A.1,

$$\mathbb{E}[R_T^\times] \leq \frac{\mathbb{E}[R_T^+]}{\mu_0^m/2} \in O\left(|\mathcal{Y}| L\varepsilon^{1/n} \frac{d}{\sigma\mu_0^m} T\varepsilon \log(1/\varepsilon) \log T\right) \tag{3}$$

Plugging $\varepsilon = T^{\frac{-2n}{2n+1}}$ to the bounds above on $\mathbb{E}[R_T^+]$ and $\mathbb{E}[R_T^\times]$ yields the desired bounds (see the arithmetic in the proof of Theorem 5.2 in Appendix B). □

[13] This is an overestimate because the agent makes at most one query per time step, even if multiple copies request a query.

D. There exist policy classes which are learnable in our setting but not in the standard online model

This section presents another algorithm with subconstant regret and sublinear queries, but under different assumptions. The primary takeaway here is that our algorithm can handle the class of thresholds on $[0, 1]$, which is known to have infinite Littlestone dimension and thus be hard in the standard online learning model. (Example 21.4 in Shalev-Shwartz & Ben-David, 2014).

Specifically, we assume a 1D input space and we allow the input sequence to be fully adversarial chosen. Instead of VC/Littlestone dimension, we consider the following notion of simplicity:

Definition D.1. Given a mentor policy π^m, partition the input space $\mathcal{X}$ into intervals such that all inputs within each interval share the same mentor action. Let $\{X_1, \dots, X_k\}$ be a partition that minimizes the number of intervals. We call each X_j a *segment*. Let $S(\pi^m)$ denote the number of segments in π^m.

Bounding the number of segments is similar conceptually to VC dimension in that it limits the ability of the policy class to realize arbitrary combinations of labels (i.e., mentor actions) on $\boldsymbol{x}$. For example, if Π is the class of thresholds on $[0, 1]$, every $\pi \in \Pi$ has at most two segments, and thus the positive result in this section will apply. This demonstrates the existence of policy classes which are learnable in our setting but not learnable in the standard online learning model, meaning that the two settings do not exactly coincide.

Unlike our primary algorithm (Algorithm 1), this algorithm does require direct access to the input encoding. However, the point of this section is not to present a practical algorithm: it is simply to demonstrate that our setting and the standard online setting do not exactly coincide.

We prove the following regret bound. Like our previous results, this bound applies to both multiplicative and additive regret.

Theorem D.2. *For any $\boldsymbol{x} \in \mathcal{X}^T$, any π^m with $S(\pi^m) \leq K$, and any function $g : \mathbb{N} \to \mathbb{N}$ satisfying $g(T) \geq 2L/\mu_0^m$, Algorithm 4 satisfies*

$$R_T^\times \leq \frac{4LKT}{g(T)^2 \mu_0^m}$$
$$R_T^+ \leq \frac{2LKT}{g(T)^2}$$
$$|Q_T| \leq (\mathrm{diam}(\boldsymbol{x}) + 4) g(T)$$

Choosing $g(T) = T^c$ for $c \in (1/2, 1)$ is sufficient for subconstant regret and sublinear queries:

Theorem D.3. *For any $c \in (1/2, 1)$, Algorithm 4 with $g(T) = T^c$ satisfies*

$$R_T^\times \in O\left(\frac{LKT^{1-2c}}{\mu_0^m}\right)$$
$$R_T^+ \in O\left(LKT^{1-2c}\right)$$
$$|Q_T| \in O(T^c(\mathrm{diam}(\boldsymbol{x}) + 1))$$

D.1. Intuition behind the algorithm

We call our algorithm “Dynamic Bucketing With Routine Querying”, or DBWRQ (pronounced “DBWRQ”). The algorithm maintains a set of buckets which partition the observed portion of the input space. Each bucket’s length determines the maximum loss in payoff we will allow from that subset of the input space. As long as the bucket contains a query from a prior time step, local generalization allows us to bound $\mu_t^m(x_t) - \mu_t(x_t, y_t)$ based on the length of the bucket containing x_t. We always query if the bucket does not contain a prior query; in this sense the querying is “routine”.

The granularity of the buckets is controlled by a function g, with the initial buckets having length $1/g(T)$. Since we can expect one query per bucket, we need $g(T) \in o(T)$ to ensure sublinear queries.

Regardless of the bucket length, the adversary can still place multiple segments in the same bucket B. A single query only tells us the optimal action for one of those segments, so we risk a payoff as bad as $\mu_t^m(x_t) - O(\mathrm{len}(B))$ whenever we

Algorithm 4 achieves subconstant regret when the mentor's policy has a bounded number of segments.

```
1: function DBWRQ($T \in \mathbb{N},\ g : \mathbb{N} \to \mathbb{N}$)
2:     $X_Q \gets \emptyset$ (previously queried inputs)
3:     $\pi \gets \emptyset$ (records $\pi^m(x)$ for each $x \in X_Q$)
4:     $\mathcal{B} \gets \emptyset$ (The set of active buckets)
5:     for $t$ from $1$ to $T$ do
6:         EvaluateInput($x_t$)
7:     end for
8: end function
9: function EvaluateInput($x \in \mathcal{X}$)
10:    if $x \notin B$ for all $B \in \mathcal{B}$ then
11:        $B \gets \left[\frac{j-1}{g(T)}, \frac{j}{g(T)}\right]$ for $j \in \mathbb{Z}$ such that $x \in B$
12:        $\mathcal{B} \gets \mathcal{B} \cup \{B\}$
13:        $n_B \gets 0$
14:        EvaluateInput($x$)
15:    else
16:        $B \gets$ any bucket containing $x$
17:        if $X_Q \cap B = \emptyset$ then
18:            Query mentor and observe $\pi^m(x)$
19:            $\pi(x) \gets \pi^m(x)$
20:            $X_Q \gets X_Q \cup \{x\}$
21:            $n_B \gets n_B + 1$
22:        else if $n_B < T/g(T)$ then
23:            Let $x' \in X_Q \cap B$
24:            Take action $\pi(x')$
25:            $n_B \gets n_B + 1$
26:        else
27:            $B = [a, b]$
28:            $(B_1, B_2) \gets \Big( \left[a, \frac{a+b}{2}\right], \left[\frac{a+b}{2}, b\right] \Big)$
29:            $(x_{B_1}, x_{B_2}) \gets (0, 0)$
30:            $\mathcal{B} \gets \mathcal{B} \cup \{B_1, B_2\} \setminus B$
31:            EvaluateInput($x$)
32:        end if
33:    end if
34: end function
```

choose not to query. We can endure a limited number of such payoffs, but if we never query again in that bucket, we may suffer $\Theta(T)$ such payoffs. Letting $\mu_t^m(x_t) = 1$ for simplicity, that would lead to $\prod_{t=1}^T \mu_t(x_t, y_t) \leq \left(1 - \frac{1}{O(g(T))}\right)^{\Theta(T)}$, which converges to 0 (i.e., guaranteed catastrophe) when $g(T) \in o(T)$.

This failure mode suggests a natural countermeasure: if we start to suffer significant (potential) losses in the same bucket, then we should probably query there again. One way to structure these supplementary queries is by splitting the bucket in half when enough time steps have involved that bucket. It turns out that splitting after $T/g(T)$ time steps is a sweet spot.

D.2. Proof notation

We will use the following notation throughout the proof of Theorem D.2:

1. Let $M_T = \{t \in [T] : \mu_t(x_t, y_t) < \mu_t^m(x_t)\}$ be the set of time steps with a suboptimal payoff.
2. Let B_t be the bucket that is used on time step t (as defined on line 16 of Algorithm 4).
3. Let $d(B)$ be the *depth* of bucket B.
    (a) Buckets created on line 11 are depth 0.

   (b) We refer to B_1, B_2 created on line 28 as the children of the bucket B defined on line 16.
   (c) If B' is the child of B, $d(B') = d(B) + 1$.
   (d) Note that $\text{len}(B) = \frac{1}{g(T)2^{d(B)}}$.

4. Viewing the set of buckets as a binary tree defined by the "child" relation, we use the terms "ancestor" and "descendant" in accordance with their standard tree definitions.
5. Let $\mathcal{B}_V = \{B : \exists t \in M_T \text{ s.t. } B_t = B\}$ be the set of buckets that ever produced a suboptimal payoff.
6. Let $\mathcal{B}'_V = \{B \in \mathcal{B}_V : \text{no descendant of } B \text{ is in } \mathcal{B}_V\}$.

D.3. Proof roadmap

The proof proceeds in the following steps:

1. Bound the total number of buckets and therefore the total number of queries (Lemma D.1).
2. Bound the suboptimality on a single time step based on the bucket length and L (Lemma D.2).
3. Bound the sum of bucket lengths on time steps where we make a mistake (Lemma D.4), with Lemma D.3 as an intermediate step. This captures the "total amount of suboptimality".
4. Lemma D.5 uses Lemma D.2 and Lemma D.4 to bound the regret.
5. Theorem D.2 directly follows from Lemmas D.1 and D.5.

D.4. Proof

Lemma D.1. *Algorithm 4 performs at most* $(\text{diam}(\boldsymbol{x}) + 4)g(T)$ *queries.*

Proof. Algorithm 4 performs at most one query per bucket, so the total number of queries is bounded by the total number of buckets. There are two ways to create a bucket: from scratch (line 11), or by splitting an existing bucket (line 28).

Since depth 0 buckets overlap only at their boundaries, and each depth 0 bucket has length $1/g(T)$, at most $g(T) \max_{t,t' \in [T]} |x_t - x_{t'}| = g(T) \,\text{diam}(\boldsymbol{x})$ depth 0 buckets are subsets of the interval $[\min_{t\in[T]} x_t, \max_{t\in[T]} x_t]$. At most two depth 0 buckets are not subsets of that interval (one at each end), so the total number of depth 0 buckets is at most $g(T)\,\text{diam}(\boldsymbol{x}) + 2$.

We split a bucket B when n_B reaches $T/g(T)$, which creates two new buckets. Since each time step increments n_B for a single bucket B, and there are a total of T time steps, the total number of buckets created via splitting is at most $\frac{2T}{T/g(T)} = 2g(T)$. Therefore the total number of buckets ever in existence is $(\text{diam}(\boldsymbol{x}) + 2)g(T) + 2 \leq (\text{diam}(\boldsymbol{x}) + 4)g(T)$, so Algorithm 4 performs at most $(\text{diam}(\boldsymbol{x}) + 4)g(T)$ queries. □

Lemma D.2. *For each* $t \in [T]$, $\mu_t(x_t, y_t) \geq \mu_t^m(x_t) - L\,\text{len}(B_t)$.

Proof. If we query at time t, then $\mu_t(x_t, y_t) = \mu_t^m(x_t)$. Thus assume we do not query at time t: then there exists $x' \in B_t$ (as defined on line 23 of Algorithm 4) such that $y_t = \pi(x') = \pi^m(x')$. Since x_t and x' are both in B_t, $|x_t - x'| \leq \text{len}(B_t)$. Then by local generalization, $\mu_t(x_t, y_t) = \mu_t(x_t, \pi^m(x')) \geq \mu_t^m(x_t) - L||x_t - x'|| \geq \mu_t^m(x_t) - L\,\text{len}(B_t)$. □

Lemma D.3. *If* π^m *has at most* K *segments,* $|\mathcal{B}'_V| \leq K$.

Proof. Now consider any $B \in \mathcal{B}'_V$. By definition of $\mathcal{B}'_V$, there exists $t \in M_T$ such that $x_t \in B$. Then there exists $x' \in B$ (as defined in Algorithm 4) such that $y_t = \pi(x') = \pi^m(x')$. Since $t \in M_T$, we have $\pi^m(x_t) \neq y_t = \pi^m(x')$. Thus x_t and x' are in different segments, but are both in B. Therefore any $B \in \mathcal{B}'_V$ must intersect at least two segments. Since B is an interval, if it intersects two segments, it must intersect two adjacent segments X_j and X_{j+1}. Furthermore, B must contain an open neighborhood centered on the boundary between X_j and X_{j+1}.

Now consider some $B' \in \mathcal{B}'_V$ with $B \neq B'$. We have $|B \cap B'| \leq 1$: otherwise one must be the descendant of the other, which contradicts the definition of $\mathcal{B}'_V$. Suppose B' also intersects both X_j and X_{j+1}: since B' is also an interval, B' must also contain an open neighborhood centered on the boundary between those two segments. But then $|B \cap B'| > 1$, which is a contradiction.

Therefore for any pair of adjacent segments X_j and X_{j+1}, there is at most one bucket in $\mathcal{B}'_V$ which contains an open neighborhood around their boundary. Since there are at most $K-1$ pairs of adjacent segments, we have $|\mathcal{B}'_V| \le K-1 \le K$. □

Lemma D.4. *We have* $\sum_{t\in M_T} \operatorname{len}(B_t) \le \frac{2KT}{g(T)^2}$.

Proof. For every $t \in M_T$, we have $B_t = B$ for some $B \in \mathcal{B}_V$, so

$$\sum_{t\in M_T} \operatorname{len}(B_t) = \sum_{B\in\mathcal{B}_V} \sum_{t\in M_T : B=B_t} \operatorname{len}(B_t)$$

Next, observe that every $B \in \mathcal{B}_V \setminus \mathcal{B}'_V$ must have a descendant in $\mathcal{B}'_V$: otherwise we would have $B \in \mathcal{B}'_V$. Let $\mathcal{A}(B)$ denote the set of ancestors of B, plus B itself. Then we can write

$$\begin{aligned}\sum_{t\in M_T} \operatorname{len}(B_t) &\le \sum_{B'\in\mathcal{B}'_V} \sum_{B\in\mathcal{A}(B')} \sum_{t\in M_T : B=B_t} \operatorname{len}(B_t)\\ &= \sum_{B'\in\mathcal{B}'_V} \sum_{B\in\mathcal{A}(B')} |\{t \in M_T : B = B_t\}| \cdot \operatorname{len}(B_t)\end{aligned}$$

For any bucket B, the number of time steps t with $B = B_t$ is at most $T/g(T)$. Also recall that $\operatorname{len}(B) = \frac{1}{g(T)2^{d(B)}}$, so

$$\begin{aligned}\sum_{B\in\mathcal{A}(B')} \frac{|\{t \in M_T : B = B_t\}|}{g(T)2^{d(B)}} &\le \frac{T}{g(T)^2} \sum_{B\in\mathcal{A}(B')} \frac{1}{2^{d(B)}}\\ &= \frac{T}{g(T)^2} \sum_{d=0}^{d(B')} \frac{1}{2^d}\\ &\le \frac{T}{g(T)^2} \sum_{d=0}^{\infty} \frac{1}{2^d}\\ &= \frac{2T}{g(T)^2}\end{aligned}$$

Then by Lemma D.3,

$$\sum_{t\in M_T} \operatorname{len}(B_t) \le \sum_{B'\in\mathcal{B}'_V} \frac{2T}{g(T)^2} = \frac{2T|\mathcal{B}'_V|}{g(T)^2} \le \frac{2KT}{g(T)^2}$$

as claimed. □

Lemma D.5. *Under the conditions of Theorem D.2, Algorithm 4 satisfies*

$$\begin{aligned}R_T^\times &\le \frac{2LKT}{\mu_0^m g(T)^2}\\ R_T^+ &\le \frac{2LKT}{g(T)^2}\end{aligned}$$

Proof. We have

$$\begin{aligned}R_T^+ &\le \sum_{t\in M_T} (\mu_t^m(x_t) - \mu_t(x_t, y_t)) && (\mu_t(x_t, y_t) \ge \mu_t^m(x_t) \text{ for } t \notin M_T)\\ &\le \sum_{t\in M_T} L \operatorname{len}(B_t) && (\text{Lemma D.2})\\ &\le \frac{2LKT}{g(T)^2} && (\text{Lemma D.4})\end{aligned}$$

Since $g(T) \geq 2L/\mu_0^m$ and every bucket length is at most $\frac{1}{g(T)}$,

$$\begin{aligned}\mu_t(x_t, y_t) &\geq \mu_t^m(x_t) - L\,\mathrm{len}(B_t)\\ &\geq \mu_0^m - \frac{L}{g(T)}\\ &\geq \mu_0^m - \mu_0^m/2\\ &\geq \mu_0^m/2\\ &> 0\end{aligned}$$

Invoking Lemma A.1 completes the proof:

$$R_T^\times \leq \frac{2R_T^+}{\mu_0^m} \leq \frac{4LKT}{g(T)^2\mu_0^m}$$

□

Theorem D.2 follows from Lemma D.1 and Lemma D.5.

E. Properties of local generalization

Proposition E.1 states that Lipschitz continuity implies local generalization when the mentor is optimal.

Proposition E.1. *Assume that μ satisfies Lipschitz continuity: for all $x, x' \in \mathcal{X}$ and $y \in \mathcal{Y}$, $|\mu(x,y) - \mu(x',y)| \leq L||x - x'||$. Also assume that $\mu(x, \pi^m(x)) = \max_{y \in \mathcal{Y}} \mu(x,y)$ for all $x \in \mathcal{X}$. Then μ satisfies local generalization with constant $2L$.*

Proof. For any $x, x' \in \mathcal{X}$, we have

$$\begin{aligned}\mu(x, \pi^m(x')) &\geq \mu(x', \pi^m(x')) - L||x - x'|| && \text{(Lipschitz continuity of } \mu)\\ &\geq \mu(x', \pi^m(x)) - L||x - x'|| && (\pi^m \text{ is optimal for } x')\\ &\geq \mu(x, \pi^m(x)) - 2L||x - x'|| && \text{(Lipschitz continuity of } \mu \text{ again)}\\ &= \mu^m(x) - 2L||x - x'|| && \text{(Definition of } \mu^m(x))\end{aligned}$$

Since π^m is optimal for x, we have

$$\mu^m(x) + 2L||x - x'|| \geq \mu^m(x) \geq \mu(x, \pi^m(x'))$$

So $-2L||x - x'|| \leq \mu(x, \pi^m(x')) - \mu^m(x) \leq 2L||x - x'||$ and therefore $|\mu(x, \pi^m(x')) - \mu^m(x)| \leq 2L||x - x'||$. □

Theorem E.2 shows that avoiding catastrophe is impossible without local generalization, even when $\boldsymbol{x}$ is σ-smooth and Π has finite VC dimension. The first insight is that without local generalization, we can define $\mu(x, y) = \mathbf{1}(y = \pi^m(x))$ so that a single mistake causes $\prod_{t=1}^T \mu(x_t, y_t) = 0$. To lower bound $\Pr\left[\prod_{t=1}^T \mu(x_t, y_t) = 0\right]$, we use a similar approach to the proof of Theorem 4.1: divide $\mathcal{X} = [0, 1]$ into $f(T)$ independent sections with $|Q_T| << f(T) << T$, so that the agent can only query a small fraction of these sections. However, the proof of Theorem E.2 is a bit easier, since we only need the agent to make a single mistake.

Note that Theorem E.2 as stated only provides a bound on $R_T^\times$. A similar bound can be obtained for R_T^+, but it is more tedious and we do not believe it would add much to the paper.

Theorem E.2. *Let $\mathcal{X} = [0, 1]$ and $\mathcal{Y} = \{0, 1\}$. Let each input be sampled i.i.d. from the uniform distribution on $\mathcal{X}$ and define the mentor policy class as the set of intervals within $\mathcal{X}$, i.e., $\Pi = \{\pi : \exists a, b \in [0, 1] \text{ s.t } \pi(x) = \mathbf{1}(x \in [a, b])\ \forall x \in \mathcal{X}\}$. Then without the local generalization assumption, any algorithm with sublinear queries satisfies $\lim_{T\to\infty} \sup_{\boldsymbol{\mu}, \pi^m} \mathbb{E}[R_T^\times] = \infty$.*

Proof. **Part 1: Setup.** Consider any algorithm which makes sublinear worst-case queries: then there exists $g : \mathbb{N} \to \mathbb{N}$ where $\sup_{\boldsymbol{\mu}, \pi^m} \mathbb{E}[|Q_T|] \leq g(T)$ and $g(T) \in o(T)$. WLOG assume $g(T) \geq 0$ for all T; if not, redefine $g(T)$ to be $\max(g(T), 1)$.

Define $f(T) := \lceil \sqrt{g(T)T} \rceil$; by Lemma A.2, $g(T) \in o(f(T))$ and $f(T) \in o(T)$. Divide $\mathcal{X}$ into $f(T)$ equally sized sections $X_1, \dots, X_{f(T)}$ in exactly the same way as in Section 4.2; see also Figure 1. Assume that each x_t is in exactly one section: this assumption holds with probability 1, so it does not affect the regret.

We use the probabilistic method: sample a segment $j^m \in [f(T)]$ uniformly at random, define π^m by $\pi^m(x) = \mathbf{1}(x \in X_{j^m})$, and define μ by $\mu(x,y) = \mathbf{1}(y = \pi^m(x))$. In words, the mentor takes action 1 iff the input is in section j^m, and the agent receives payoff 1 if its action matches the mentor's and zero otherwise. Since any choice of j^m defines a valid μ and π^m,

$$\sup_{\boldsymbol{\mu},\pi^m} \mathop{\mathbb{E}}_{\boldsymbol{x},\boldsymbol{y}} \left[R_T^+(\boldsymbol{x},\boldsymbol{y},\boldsymbol{\mu},\pi^m)\right] \geq \mathop{\mathbb{E}}_{j^m} \mathop{\mathbb{E}}_{\boldsymbol{x},\boldsymbol{y}} \left[R_T^+(\boldsymbol{x},\boldsymbol{y},(\mu,\dots,\mu),\pi^m)\right]$$

Let $J_{\neg Q} = \{j \in [f(T)] : x_t \notin X_j \ \forall t \in Q_T\}$ be the set of sections which are never queried. Let $j_1,\dots,j_k$ be the sequence of sections queried by the agent: then $k = |Q_T| \leq g(T)$.

Part 2: The agent is unlikely to determine j^m. By the chain rule of probability,

$$\Pr[j^m \in J_{\neg Q}] = \Pr\left[j_i \neq j^m \ \forall i\right] = \prod_{i=1}^k \Pr\left[j_i \neq j^m \mid j_r \neq j^m \ \forall r < i\right]$$

Now fix i and assume $j_r \neq j^m \ \forall r < i$. Queries in sections other than j^m provide no information about the value of j^m, so j^m is uniformly distributed across the set of sections not yet queried, i.e., $\{j \in [f(T)] : j_r \neq j \ \forall r < i\}$. There are at least $f(T) - i + 1$ such sections, since there are $i-1$ prior queries at this point. Thus $\Pr[j_i \neq j^m \mid j_r \neq j^m \ \forall r < i] \geq \frac{f(T)-i}{f(T)-i+1}$ (the inequality is because this probability could also be 1 if $j_i = j_r$ for some $i < r$). Therefore

$$\begin{aligned}\Pr\left[j^m \in J_{\neg Q}\right] &\geq \prod_{i=1}^k \frac{f(T)-i}{f(T)-i+1}\\ &= \frac{f(T)-1}{f(T)} \cdot \frac{f(T)-2}{f(T)-1} \cdots \frac{f(T)-k+1}{f(T)-k+2} \cdot \frac{f(T)-k}{f(T)-k+1}\\ &= \frac{f(T)-k}{f(T)}\\ &\geq 1 - \frac{g(T)}{f(T)}\end{aligned}$$

Part 3: If the agent fails to determine j^m, it is likely to make at least one mistake. For each $j \in J_{\neg Q}$, let $V_j = \{t \in [T] : x_t \in X_j\}$ be the set of time steps with inputs in section j. By Lemma A.4, $\Pr[|V_{j^m}| = 0] \leq \exp\left(\frac{T}{16f(T)}\right)$. Then by the union bound, $\Pr[j^m \in J_{\neg Q} \text{ and } |V_{j^m}| > 0] \geq 1 - \frac{g(T)}{f(T)} - \exp\left(\frac{-T}{16f(T)}\right)$. For the rest of Part 3, assume $j^m \in J_{\neg Q}$ and $|V_{j^m}| > 0$.

Case 1: For all $j \in J_{\neg Q}$ and $t \in V_j$, we have $y_t = 0$. In particular, this holds for $j = j^m$, and we know there exists at least one $t \in V_{j^m}$ since $|V_{j^m}| > 0$. Then $y_t \neq \pi^m(x_t)$, so $\mu(x_t, y_t) = 0$ and thus $\Pr\left[\prod_{r=1}^T \mu(x_r,y_r) = 0 \,\middle|\, j^m \in J_{\neg Q} \text{ and } |V_{j^m}| > 0 \text{ and } y_t = 0 \ \forall j \in J_{\neg Q}, t \in V_j\right] = 1$.

Case 2: There exists $j \in J_{\neg Q}$ and $t \in V_j$ with $y_t = 1$. Then $\mu(x_t,y_t) = 0$ unless $j = j^m$, so

$$\begin{aligned}&\Pr\left[\prod_{r=1}^T \mu(x_r,y_r) = 0 \,\middle|\, j^m \in J_{\neg Q} \text{ and } |V_{j^m}| > 0 \text{ and } \exists j \in J_{\neg Q}, t \in V_j \text{ s.t. } y_t = 1\right]\\ \geq\ &\Pr\left[\mu(x_t,y_t) = 0 \,\middle|\, j^m \in J_{\neg Q} \text{ and } |V_{j^m}| > 0 \text{ and } \exists j \in J_{\neg Q}, t \in V_j \text{ s.t. } y_t = 1\right]\\ =\ &\Pr\left[j \neq j^m \,\middle|\, j^m \in J_{\neg Q} \text{ and } |V_{j^m}| > 0 \text{ and } \exists j \in J_{\neg Q}, t \in V_j \text{ s.t. } y_t = 1\right]\end{aligned}$$

Since $j^m \in J_{\neg Q}$, the agent has no information about j^m other than that it is in $J_{\neg Q}$. This means that j^m is uniformly distributed across $J_{\neg Q}$, so

$$\Pr\left[\prod_{r=1}^T \mu(x_r,y_r) = 0 \,\middle|\, j^m \in J_{\neg Q} \text{ and } |V_{j^m}| > 0 \text{ and } \exists j \in J_{\neg Q}, t \in V_j \text{ s.t. } y_t = 1\right] \geq 1 - \frac{1}{|J_{\neg Q}|} \geq 1 - \frac{1}{f(T) - g(T)}$$

Combining Case 1 and Case 2, we get the overall bound of

$$\Pr\left[\prod_{t=1}^T \mu(x_t,y_t) = 0 \,\middle|\, j^m \in J_{\neg Q} \text{ and } |V_{j^m}| > 0\right] \geq 1 - \frac{1}{f(T) - g(T)}$$

and thus

$$\begin{aligned}
\Pr\left[\prod_{t=1}^T \mu(x_t, y_t) = 0\right] &\geq \Pr\left[\prod_{t=1}^T \mu(x_t, y_t) = 0 \text{ and } j^m \in J_{\neg Q} \text{ and } |V_{j^m}| > 0\right]\\
&= \Pr\left[\prod_{t=1}^T \mu(x_t, y_t) = 0 \,\middle|\, j^m \in J_{\neg Q} \text{ and } |V_{j^m}| > 0\right] \cdot \Pr\left[j^m \in J_{\neg Q} \text{ and } |V_{j^m}| > 0\right]\\
&\geq \left(1 - \frac{1}{f(T) - g(T)}\right)\left(1 - \frac{g(T)}{f(T)} - \exp\left(\frac{-T}{16 f(T)}\right)\right)
\end{aligned}$$

For brevity, let $\alpha(T)$ denote this final bound. Since $g(T) \in o(f(T))$ and $f(T) \in o(T)$, we have

$$\lim_{T\to\infty} \alpha(T) = \lim_{T\to\infty} \left(1 - \frac{1}{f(T) - g(T)}\right)\left(1 - \frac{g(T)}{f(T)} - \exp\left(\frac{-T}{16 f(T)}\right)\right) = (1-0)(1-0-0) = 1$$

Part 4: Putting it all together. Consider any $\varepsilon \in (0, 1]$; to avoid dealing with infinite expectations, we will deal with $\Pr[\prod_{t=1}^T \mu(x_t, y_t) \leq \varepsilon]$ instead of $\Pr[\prod_{t=1}^T \mu(x_t, y_t) = 0]$. Since $\prod_{t=1}^T \mu(x_t, y_t) \leq 1$ always, we have

$$\begin{aligned}
\mathop{\mathbb{E}}_{j^m} \mathop{\mathbb{E}}_{\boldsymbol{x},\boldsymbol{y}}\left[\log \prod_{t=1}^T \mu(x_t, y_t)\right] &= \mathop{\mathbb{E}}_{j^m} \mathop{\mathbb{E}}_{\boldsymbol{x},\boldsymbol{y}}\left[\log \prod_{t=1}^T \mu(x_t, y_t) \,\middle|\, \prod_{t=1}^T \mu(x_t, y_t) \leq \varepsilon\right] \cdot \Pr\left[\prod_{t=1}^T \mu(x_t, y_t) \leq \varepsilon\right]\\
&+ \mathop{\mathbb{E}}_{j^m} \mathop{\mathbb{E}}_{\boldsymbol{x},\boldsymbol{y}}\left[\log \prod_{t=1}^T \mu(x_t, y_t) \,\middle|\, \prod_{t=1}^T \mu(x_t, y_t) > \varepsilon\right] \cdot \Pr\left[\prod_{t=1}^T \mu(x_t, y_t) > \varepsilon\right]\\
&\leq \log \varepsilon \cdot \Pr\left[\prod_{t=1}^T \mu(x_t, y_t) \leq \varepsilon\right] + 1 \cdot \left(1 - \Pr\left[\prod_{t=1}^T \mu(x_t, y_t) \leq \varepsilon\right]\right)\\
&\leq 1 - (1 - \log \varepsilon) \Pr\left[\prod_{t=1}^T \mu(x_t, y_t) \leq \varepsilon\right]
\end{aligned}$$

Since $\varepsilon \in (0, 1]$, we have $1 - \log \varepsilon > 0$. Also, $\Pr[\prod_{t=1}^T \mu(x_t, y_t) \leq \varepsilon] \geq \Pr[\prod_{t=1}^T \mu(x_t, y_t) = 0]$, so

$$\begin{aligned}
\mathop{\mathbb{E}}_{j^m} \mathop{\mathbb{E}}_{\boldsymbol{x},\boldsymbol{y}}\left[\log \prod_{t=1}^T \mu(x_t, y_t)\right] &\leq 1 - (1 - \log \varepsilon) \Pr\left[\prod_{t=1}^T \mu(x_t, y_t) \leq \varepsilon\right]\\
&\leq 1 - (1 - \log \varepsilon) \Pr\left[\prod_{t=1}^T \mu(x_t, y_t) = 0\right]\\
&\leq 1 - (1 - \log \varepsilon)\alpha(T)
\end{aligned}$$

Since $\prod_{t=1}^T \mu^m(x_t) = 1$ always, we have

$$\begin{aligned}
\sup_{\boldsymbol{\mu}, \pi^m} \mathop{\mathbb{E}}_{\boldsymbol{x},\boldsymbol{y}}\left[R_T^+(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\mu}, \pi^m)\right] &\geq \mathop{\mathbb{E}}_{j^m} \mathop{\mathbb{E}}_{\boldsymbol{x},\boldsymbol{y}}\left[R_T^+(\boldsymbol{x}, \boldsymbol{y}, (\mu, \dots, \mu), \pi^m)\right]\\
&= \log 1 - \mathop{\mathbb{E}}_{j^m} \mathop{\mathbb{E}}_{\boldsymbol{x},\boldsymbol{y}}\left[\log \prod_{t=1}^T \mu(x_t, y_t)\right]\\
&\geq -1 + (1 - \log \varepsilon)\alpha(T)
\end{aligned}$$

Therefore

$$\begin{aligned}
\lim_{T\to\infty} \sup_{\boldsymbol{\mu}, \pi^m} \mathop{\mathbb{E}}_{\boldsymbol{x},\boldsymbol{y}}\left[R_T^+(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\mu}, \pi^m)\right] &\geq -1 + (1 - \log \varepsilon) \lim_{T\to\infty} \alpha(T)\\
&\geq -1 + (1 - \log \varepsilon)\\
&\geq -\log \varepsilon
\end{aligned}$$

This holds for every $\varepsilon \in (0, 1]$, which is only possible if $\lim_{T\to\infty} \sup_{\boldsymbol{\mu}, \pi^m} \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}}\left[R_T^+(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\mu}, \pi^m)\right] = \infty$, as desired. □