Journal of Machine Learning Research 27 (2026) 1-43 Submitted 9/25; Revised 4/26; Published 6/26

Safe Learning Under Irreversible Dynamics via Asking for Help

Benjamin Plaut PLAUT@BERKELEY.EDU
Juan Liévano-Karim JP.LIEVANO10@UNIANDES.EDU.CO
Hanlin Zhu HANLINZHU@BERKELEY.EDU
Stuart Russell RUSSELL@BERKELEY.EDU

Division of Computer Science
University of California
Berkeley, CA 94720-1776, USA

Editor: Laurent Orseau

Abstract

Most learning algorithms with formal regret guarantees essentially rely on trying all possible behaviors, which is problematic when some errors cannot be recovered from. Instead, we allow the learning agent to ask for help from a mentor and to transfer knowledge between similar states. We show that this combination enables the agent to learn both safely and effectively. Under standard online learning assumptions, we provide an algorithm whose regret and number of mentor queries are both sublinear in the time horizon for Markov decision processes with irreversible dynamics and infinite state spaces. Our proof involves a sequence of three reductions, making our result more general than a single algorithm. Conceptually, our result may be the first formal proof that it is possible for an agent to obtain high reward while becoming self-sufficient in an unknown, unbounded, and high-stakes environment without resets.

Keywords: AI safety, safe RL, no-regret learning, asking for help, irreversibility

1. Introduction

Concern has been mounting over a variety of risks from AI (National Institute of Standards and Technology, 2024; Critch and Russell, 2023; Hendrycks et al., 2023; Slattery et al., 2024). Such risks include medical errors (Rajpurkar et al., 2022), self-driving car crashes (Kohli and Chadha, 2019), discriminatory sentencing (Villasenor and Foggo, 2020), autonomous weapon accidents (Abaimov and Martellini, 2020), bioterrorism (Mouton et al., 2024), high-stakes cyberattacks (Guembe et al., 2022), and loss of control (Bengio et al., 2024). These types of errors are particularly concerning because they are irreparable: a self-driving car cannot compensate for a crash by driving superbly later. We refer to irreparable errors as *catastrophes*.

Despite the gravity of these risks, there is little theoretical understanding of how to avoid them: nearly all of learning theory explicitly or implicitly assumes that any error can be recovered from. This assumption enables algorithms to cheerfully try all possible behaviors, since no matter how poorly the algorithm performs in the short term, it can always eventually make up for it. Although such approaches have produced positive results, we argue that they are fundamentally unsuitable for the high-stakes AI applications above: we do not want autonomous weapons or surgical robots to try all possible behaviors. One could train AI agents entirely

in controlled lab settings where all errors *are* recoverable, but we argue that sufficiently general agents will inevitably encounter novel scenarios when deployed in the real world. Models often behave unpredictably in unfamiliar situations (see, e.g., Quiñonero-Candela et al., 2022), and we do not want AI biologists or self-driving cars to behave unpredictably.

Instead, we suggest that agents should recognize when they are in unfamiliar situations and ask for help. When help is requested, a supervisor (here called a *mentor* to reflect their collaborative role) guides the agent to avoid problematic actions without the agent needing to try those actions firsthand. This approach is particularly suitable for high-stakes applications where AI systems are already configured to work alongside humans. For example, imagine a human doctor who oversees AI surgeons and is prepared to take over in tricky situations, or a backup human driver in an autonomous vehicle. In these scenarios, occasional requests for human guidance are both practical and valuable for ensuring safety.

In order to avoid excessive requests for help, the agent must be able to accurately identify unfamiliar situations, i.e., situations that are not similar to any of the agent's prior experiences. We model "similarity" as a distance metric on the state space and assume that the agent can transfer knowledge between similar states, but this transfer becomes less reliable as the states become less similar. We call this assumption *local generalization*.[1]

Guided by the principles above, we design an agent that (under standard online learning assumptions) performs nearly as well as the mentor while gradually becoming self-sufficient, even when some errors are irreparable.

1.1 Our Model

We study online learning, where the agent must perform well while learning: finding a good policy is futile if irreparable damage is caused along the way. We allow the MDP to be *non-communicating* (some states might not be reachable from other states) and do not allow the agent to be reset to the start state. This is the class of "multichain MDPs" (Puterman, 2014). Crucially, we allow an infinite state space: in the real world, an agent will likely never see the exact same state twice. Instead, the agent must use local generalization to judge how similar the present state is to previous states.

On each time step, the agent has the option to query a mentor. When queried, the mentor demonstrates the action they would take in the current state. We assume that the agent begins with no prior knowledge.[2] We want to minimize *regret*, defined as the difference between the expected sum of rewards obtained by the agent and the mentor when each is given the same starting state. (Regret typically refers to the performance gap between the agent and an optimal policy, but we allow the mentor to be either optimal or suboptimal, so our definition is arguably more general.) An algorithm for this problem should satisfy two criteria:

1. The agent's performance should approach that of the mentor. Formally, the average regret per time step should go to 0 as the time horizon T goes to infinity, or equivalently, the cumulative regret should be sublinear in T (denoted $o(T)$). An algorithm that guarantees $o(T)$ regret is called a *no-regret* algorithm.
2. The agent should become self-sufficient. Formally, the rate of querying the mentor should go to 0 as $T \to \infty$, or equivalently, the cumulative number of queries should be $o(T)$.

1. See Section 5.1 for the formal definition.
2. Future work could incorporate an initial offline data set. See Section 7 for further discussion.

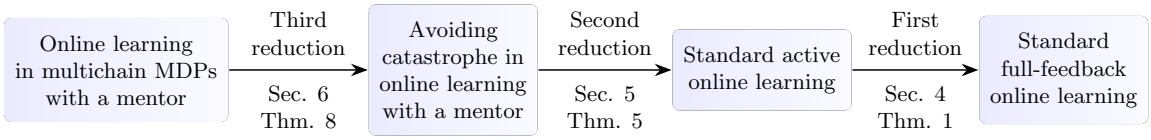

Figure 1: An illustration of our three-step reduction.

1.2 Our Contribution

We provide an algorithm which satisfies both criteria given local generalization and standard online learning assumptions. Our proof involves a three-step reduction across four models of online learning (Figure 1). Throughout the paper, we use "catastrophe" as a conceptual and informal term to refer to errors that immediately ruin the entire attempt; the precise reversibility assumptions (or lack thereof) are made clear when the models are formally defined later. Here we provide a brief conceptual overview of the models:

1. Standard full-feedback online learning, where catastrophe is impossible because the agent and the optimal policy are evaluated on the same states (and rewards are bounded).
2. Standard active online learning, which is the same as Model 1 but the agent only receives feedback when it queries.
3. Avoiding catastrophe in online learning, a new model where catastrophe *is* possible, but avoiding catastrophe is the agent's only goal (it does not also need to maximize reward).
4. Online learning in multichain MDPs, where the agent must maximize reward (which requires avoiding catastrophe).

Table 1 compares the models in more detail (note that Models 1 and 2 are combined under "standard adversarial active online learning").

First reduction. We first show that any algorithm for Model 1 can be transformed into an algorithm for Model 2. This reduction is based on standard ideas from Cesa-Bianchi et al. (2005), who proved a similar result but only for a single algorithm. However, to our knowledge, our other two reductions are technically novel.

Second reduction. The high-level idea is simple: mostly follow a "default" algorithm which works well in the absence of catastrophe, but ask for help in unfamiliar situations. However, bringing this to fruition requires carefully balancing the frequency of mentor queries, the frequency of errors, and the magnitude of errors. Altogether, we show that if the "default" algorithm is effective for Model 2, then the overall algorithm is effective for Model 3.

Third reduction. Finally, we reduce Model 4 to Model 3. This reduction requires no modifications of the algorithm: we show that any algorithm which guarantees avoidance of catastrophe (in the sense of Model 3) also ensures high reward in multichain MDPs. This is *not* saying that avoiding catastrophe in a particular MDP suffices for high reward in that MDP (which is generally false). This is saying that any algorithm guaranteed to avoid catastrophe in general must have certain properties which also guarantee high reward. The regret analysis involves a decomposition of regret into "state-based" and "action-based" regret which we believe is novel.

Chaining together all three reductions produces (to our knowledge) the first no-regret guarantee for multichain MDPs with an infinite state space.

Theorem 10 (Informal)**.** *Under standard online learning assumptions, there exists a no-regret algorithm for infinite-state multichain MDPs that makes $o(T)$ mentor queries.*

In other words, we prove that it is possible for an agent to obtain high reward while becoming self-sufficient in an unknown, unbounded, and high-stakes environment without resets. We are not aware of any prior work that can be said to imply this conclusion.

2. Prior Work

There is a vast body of work studying regret in MDPs, but all prior no-regret guarantees rely on restrictive assumptions. For context, the agent has no hope of success without further assumptions: any unexplored action could lead to paradise (so the agent cannot neglect it) but the action could also lead to disaster (so the agent cannot risk it). This intuition is formalized by the "Heaven or Hell" problem in Figure 2. Our work is not immune to this problem: rather, we argue that access to a mentor is a natural and practical complement to typical assumptions like reversibility or periodic resets that may not always hold in high-stakes AI applications.

Figure 3 provides a taxonomy of assumptions used to handle the Heaven or Hell problem, which will guide our discussion of related work. Unless otherwise specified, citations are representative, not exhaustive.

2.1 Assuming That All Errors Are Recoverable

The most common approach is to assume that any error can be offset by performing well enough in the future. This assumption can take various forms, including periodically resetting the agent (Azar et al., 2017) or assuming a communicating MDP (Jaksch et al., 2010). A less obvious form is defining regret relative to prior errors, i.e., allowing catastrophic errors as long as the agent does as well as possible going forward. This approach compares the performance of the agent and the reference policy *on the agent's state sequence* and does not consider whether the reference policy would have led to better states. For example, an agent which enters Hell in Figure 2 is considered successful, since all actions are equivalent thereafter and thus it obtains "optimal" reward on all future time steps. This comparison can be formulated as a type of regret (He et al., 2021; Liu and Su, 2021) or via the related notion of "asymptotic optimality" due to Lattimore and Hutter (2011).

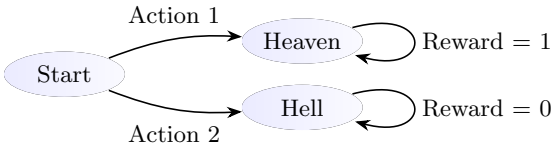

Figure 2: The "Heaven or Hell" problem shows why online learning in MDPs is doomed without further assumptions. Action 1 leads to high reward forever and Action 2 leads to low reward forever. Without further assumptions, the agent has no way to tell which action leads to Heaven and which leads to Hell, so it is impossible to guarantee high reward.

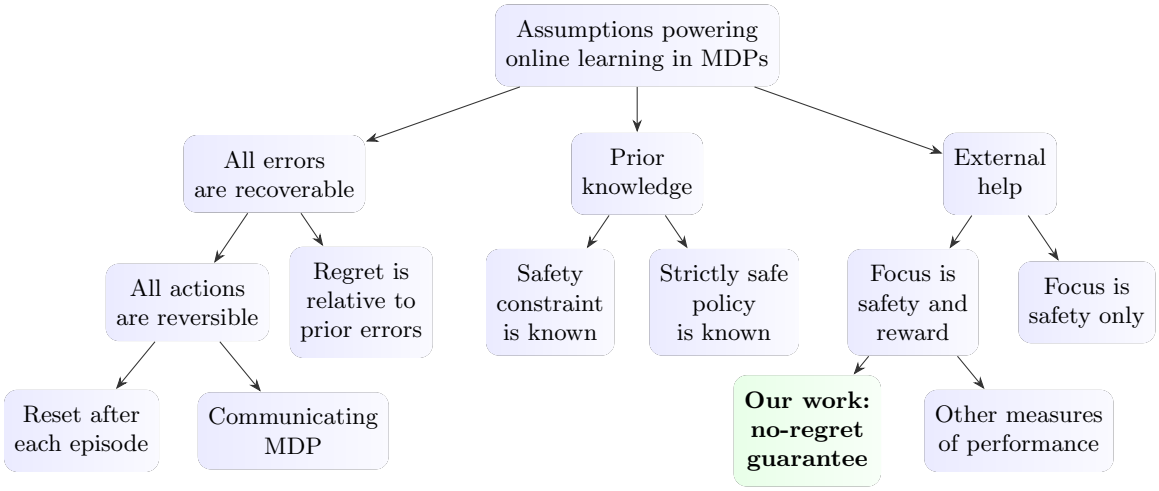

Figure 3: A taxonomy of assumptions which enable meaningful theoretical results for online learning in MDPs. Figure 2 shows why meaningful guarantees are impossible without further assumptions.

Regardless of the specific form of the assumption, this approach faces difficulty in high-stakes domains where some errors cannot be recovered from. For example, a robotic vehicle cannot make up for a crash by driving superbly afterward. Thus we argue that this approach is ill-suited for the types of AI risks we are concerned with.

One notable paper in this category is Ortner and Ryabko (2012), which studies infinite-state MDPs with the recoverability assumption and a Lipschitz assumption similar to local generalization. Their regret bound, like ours, has exponential dependence on the dimension of the metric space (in fact, with the exact same exponent of $\frac{2n+1}{2n+2}$).

2.2 Prior Knowledge

A smaller body of work allows for the possibility of catastrophe but provides the agent sufficient prior knowledge to avoid it, primarily within the formalism of constrained MDPs (CMDPs) (Altman, 1999). Some work assumes that the exact safety constraint is known, i.e., the agent knows upfront exactly which actions cause catastrophe (Zhao et al., 2022, 2023). Other work assumes the agent knows a fully safe policy and the agent is periodically reset (Liu et al., 2021; Stradi et al., 2024). In contrast, we assume no prior knowledge of the environment.

2.3 External Help

An even less common approach is to allow the agent to ask for help from a mentor. Crucially, this approach had not previously produced a no-regret guarantee for large or infinite multichain MDPs. Most prior work in this category focuses on safety alone, and if reward guarantees are provided, they fall into the "defining regret relative to prior errors" category (Cohen and Hutter, 2020; Cohen et al., 2021). The paper most relevant to our work is Plaut

et al. (2025), who proposed the "avoiding catastrophe in online learning with a mentor" model referred to in Figure 1. Theorem 6 recovers the primary result of that paper but with a more general proof that applies to any algorithm which meets the reduction criteria. In contrast, that paper's result only applies to a specific algorithm. More importantly, that paper only considers avoiding catastrophe and does not consider any effectiveness-related reward. In contrast, Theorem 6 is only a stepping stone to our main result: a guarantee of high reward in multichain MDPs (which requires both effectiveness and avoiding catastrophe).

We are aware of just two papers that use external help to obtain regret guarantees on reward in multichain MDPs: Maillard et al. (2019) and Kosoy (2019). First and foremost, both papers assume a finite state space, which means that with a large enough time horizon, the agent can eventually query every state. In contrast, in the real world, the exact same state never appears twice. Furthermore, both papers' regret bounds scale linearly with the number of states. Additional differences include:

1. The regret bound of Maillard et al. (2019) includes an instance-dependent γ^* parameter, whereas our bound is worst-case across all multichain MDPs that satisfy our assumptions.
2. The method of Kosoy (2019) requires computing a Bayesian posterior, which is intractable without strong assumptions. In their case, they assume a finite number of possible hypotheses.
3. Like us, Kosoy (2019) allows a suboptimal mentor. Unlike us, they show that the agent can still obtain asymptotically optimal reward as long as the mentor only takes safe actions and has a nonzero probability of taking the optimal action.
4. Kosoy (2019) studies the infinite-horizon setting where "no-regret" means that the average regret goes to 0 as the discount factor goes to 1.
5. By assuming a finite state space, both Maillard et al. (2019) and Kosoy (2019) avoid the need for a similarity metric between states. As a result, they avoid the exponential dimensionality dependence in the regret bound that is typical for online learning in metric spaces (e.g., our work and Ortner and Ryabko, 2012).

2.4 Other Related Work

There is a wide range of other work on designing agents to act cautiously when uncertain. Representative theoretical papers include Cortes et al. (2018); Hadfield-Menell et al. (2017); Li et al. (2008). Our model is also reminiscent of active learning (Hanneke, 2014) due to the agent proactively asking for help and of imitation learning (Osa et al., 2018) due to the agent attempting to follow the mentor policy. However, we are not aware of any direct technical implications. Finally, see García and Fernández (2015); Gu et al. (2024); Krasowski et al. (2023) for surveys of the safe reinforcement learning literature and Cesa-Bianchi and Lugosi (2006) for a survey of online learning in general.

3. Preliminaries

Here we describe the general online learning setup which is shared by all models we study. Each specific model is defined in its relevant section (Sections 4.1, 5.1, and 6.1). As such, the setup here is underspecified and also more general than each specific model. In particular, the adversary framework here generalizes MDPs, which are not defined until Section 6.1.

3.1 States, Actions, Queries, and Histories

Let $\Delta(X)$ denote the set of probability distributions on a set X, let $\mathbb{N}$ denote the strictly positive integers, and let $[k] = \{1, \ldots, k\}$ for $k \in \mathbb{N}$. Let $\mathcal{S}$ be a state space, $\mathcal{A}$ be a finite action space, and $T \in \mathbb{N}$ be a time horizon. We assume that $\mathcal{S}$ is countable for mathematical convenience, but there are no fundamental obstacles for continuous spaces. On each time step $t \in [T]$, the agent observes a state $s_t \in \mathcal{S}$ and selects an action $a_t \in \mathcal{A}$. Let $\boldsymbol{s} = (s_1, \ldots, s_T)$ and $\boldsymbol{a} = (a_1, \ldots, a_T)$.

Each time step t is also associated with a "correct" action a_t^m which can be interpreted as the mentor's action in state s_t (although we keep a_t^m more general for now). In addition to choosing an action a_t, the agent also makes a query decision $q_t \in \{0, 1\}$. The agent observes a_t^m if and only if $q_t = 1$. With slight abuse of notation, we can express this concisely by assuming that the agent observes $a_t^m q_t$: if the algorithm queries, then $q_t = 1$ so $a_t^m = a_t^m q_t$ is observed. If the algorithm does not query, then $0 = a_t^m q_t$ is observed. (We assume 0 is a "dummy" action not in $\mathcal{A}$.) In general we assume that the agent observes $a_t^m q_t$ before selecting a_t, but we also consider "query-agnostic" algorithms (see below).

For $t \in [T+1]$, let $h_t = (s_i, a_i, a_i^m q_i)_{i \in [t-1]}$ be the history at the end of time step $t-1$ (or equivalently, the start of time step t).[3] Let $\mathcal{H} = (\mathcal{S} \times \mathcal{A} \times (\mathcal{A} \cup \{0\}))^*$ be the set of possible histories. Note that $a_i^m q_i$ also tells us the value of q_i. We use $\frown$ to denote vector concatenation: for example, $h_{t+1} = h_t \frown (s_t, a_t, a_t^m q_t)$.

3.2 Adversaries and Algorithms

On each time step t, the state s_t and correct action a_t^m are determined by the *adversary*. Formally, an adversary V is a function $V : \mathcal{H} \to \Delta(\mathcal{S} \times \mathcal{A})$ which takes as input h_t and returns the distribution of s_t and a_t^m. In general, V can be any function, although for MDPs the adversary must use a Markov transition function.

The action a_t and query decision q_t are determined by the algorithm. Formally, an *active learning algorithm* (or just "algorithm") Γ consists of a query function $\Gamma^Q : \mathcal{H} \times \mathcal{S} \to \Delta(\{0, 1\})$ and an action function $\Gamma^{\mathcal{A}} : \mathcal{H} \times \mathcal{S} \times (\mathcal{A} \cup \{0\}) \to \Delta(\mathcal{A})$. The query function takes as input h_t and s_t and returns the distribution of q_t. The action function takes as input h_t, s_t, and the query result $a_t^m q_t$ and returns the distribution of a_t. Together, Γ and V determine the distributions of $\boldsymbol{s}, \boldsymbol{a}$, and $\boldsymbol{q}$. Unless otherwise stated, all expectations are over the randomness of $\boldsymbol{s}, \boldsymbol{a}$, and $\boldsymbol{q}$. We assume T is known, so V, Γ^Q, and $\Gamma^{\mathcal{A}}$ can all depend on it.

An algorithm Γ is *full-feedback* if it always queries, i.e., $\Gamma^Q(h, s) = 1$ for all $T \in \mathbb{N}, h \in \mathcal{H}, s \in \mathcal{S}$.[4] An algorithm Γ is *query-agnostic* if its action does not depend on the query result from that round, i.e., $\Gamma^{\mathcal{A}}(h, s, 0) = \Gamma^{\mathcal{A}}(h, s, a)$ for any $h \in \mathcal{H}, s \in \mathcal{S}, a \in \mathcal{A}$. Equivalently, query-agnostic algorithms only observe the query result $a_t^m q_t$ after choosing a_t. For query-agnostic algorithms, we write $\Gamma^{\mathcal{A}}(h_t, s_t)$ for simplicity. Full-feedback query-agnostic algorithms are the default in the online prediction literature and are typically defined without involving queries at all, but we treat this as a special case of our broader active learning model.

3. Since there is technically no time step 0 or $T+1$, h_1 and h_{T+1} must be interpreted as the history at the start of time step 1 and the end of time step T, respectively.
4. In typical online learning, observing a_t^m suffices to compute the loss of every action. See Section 4.1.

	Standard adversarial	**Avoiding catastrophe**	**Multichain MDPs**
Conceptual goal	High reward (assume no catastrophe)	Avoid catastrophe only	High reward (requires avoiding catastrophe)
Desired regret	Sublinear in T	Subconstant in T	Sublinear in T
Regret definition	Single state sequence $\boldsymbol{s}$	Single state sequence $\boldsymbol{s}$	Separate $\boldsymbol{s}, \boldsymbol{s^m}$ for agent and mentor
Source of each state	Adversary	Adversary	Transition function
Local generalization	No	Yes	Yes
When are query results observed?	After choosing action	Before choosing action	Before choosing action

Table 1: The differences between the online learning models we consider. There is no column for full-feedback online learning, since it is a special case of (standard adversarial) active online learning.

3.3 Overall Learning Protocol

To summarize, each time step $t \in [T]$ proceeds as follows:

1. The adversary selects a state and correct action $(s_t, a_t^m) \sim V(h_t)$.
2. The algorithm observes s_t and selects a query decision $q_t \sim \Gamma^Q(h_t, s_t)$.
3. The algorithm observes $a_t^m q_t$ and selects an action $a_t \sim \Gamma^{\mathcal{A}}(h_t, s_t, a_t^m q_t)$.

3.4 Objectives

The agent has two objectives: the number of queries and the regret. These are separate objectives and the agent must perform well according to both.[5] Let $Q(T, \Gamma, V) = \mathbb{E}[\sum_{t=1}^{T} q_t]$ be the expected number of queries. We want $Q(T, \Gamma, V)$ to be sublinear in T. The regret objective varies between the models, but all versions involve optimizing over a known *policy* class Π. Each (deterministic) policy $\pi \in \Pi$ is a function $\pi : \mathcal{S} \to \mathcal{A}$.

3.5 Further Technical Definitions

These definitions are not necessary to grasp the key ideas of the paper but are necessary to understand the precise technical results. For random variables X and Y, let $\mathcal{D}(X)$ and $\mathcal{D}(X \mid Y)$ respectively be the distribution of X and the conditional distribution of X with respect to Y. For a distribution p, let $p(S)$ be the probability that p assigns to a set S. For example, for a random variable X, $\mathcal{D}(X)(S) = \Pr[X \in S]$. When S contains a single element s, we write $\mathcal{D}(X)(\{s\}) = \mathcal{D}(X)(s)$ for simplicity. For $c \in [0, 1]$, let Bernoulli(c) be the random variable which is 1 with probability c and 0 otherwise.

We consider both full adversaries and "smooth" adversaries which are required to choose a not-too-concentrated distribution over the next state. This allows a smooth interpolation between the "easy" case of i.i.d. states and the "hard" case of fully adversarial states. Given

5. One could combine these into a single objective, but it is unclear how to choose the tradeoff parameter.

two distributions p, β on $\mathcal{S}$, we say that p is *σ-smooth* (with respect to β) if $p(X) \leq \frac{1}{\sigma}\beta(X)$ for every $X \subseteq \mathcal{S}$. The uniform distribution is a common choice for the "baseline" distribution β, but our results hold for any fixed β. Throughout the paper, we assume an arbitrary but fixed β. For convenience, we say that every distribution on $\mathcal{S}$ is 0-smooth; this corresponds to the fully adversarial case. For $\sigma \in [0, 1]$, an adversary V is σ-smooth if for any $h \in \mathcal{H}$, the marginal distribution of $V(h)$ on $\mathcal{S}$ is σ-smooth. Equivalently, $\Pr_{(s,a)\sim V(h)}[s \in X] \leq \frac{1}{\sigma}\beta(X)$ for any $X \subseteq \mathcal{S}$. Let $\mathcal{V}(\sigma)$ denote the set of σ-smooth adversaries. Since we will often handle $\sigma = 0$ separately, define the (scalar) Moore-Penrose pseudoinverse σ^+ by $\sigma^+ = \sigma^{-1}$ for $\sigma > 0$ and $0^+ = 0$. For convenience, we treat $\min_{x\in\emptyset} f(x)$ as ∞ for any function f.

4. First Reduction: Standard Adversarial Active Learning → Full-Feedback

We now present our first reduction (recall the roadmap in Figure 1). We show that in the standard adversarial online learning model, any full-feedback algorithm can be converted into an active learning algorithm, where the regret of the latter is bounded by the regret of the former and the expected number of queries.

4.1 The Standard Adversarial Online Learning Model

The agent's goal in this model is to predict the correct action (this is sometimes called the "online prediction" model). Formally, the objective in this model is to minimize a known loss function $\ell : \mathcal{A}\times\mathcal{A} \to [0, 1]$, where $\ell(a, a^m)$ is the loss from predicting a when the correct action is a^m. (In this model, all dependency on the states is captured implicitly through the actions.) One particular ℓ of interest is the binary loss $\ell_{\mathbf{1}}(a, a^m) = \mathbf{1}(a \neq a^m)$. Since ℓ is known, a full-feedback algorithm will observe a_t^m on every time step and thus can compute the loss of every action in $\mathcal{A}$. The standard adversarial regret of an algorithm Γ given an adversary V is

$$\mathrm{Reg}_{\mathrm{SA}}(T, \Gamma, V, \ell) = \sup_{\pi\in\Pi} \mathbb{E}\left[\sum_{t=1}^{T} \ell(a_t, a_t^m) - \sum_{t=1}^{T} \ell(\pi(s_t), a_t^m)\right].$$

The typical goal is for $\mathrm{Reg}_{\mathrm{SA}}$ to be sublinear in T, or equivalently, for the average regret per time step to go to 0. We assume throughout this section that all algorithms are query-agnostic, i.e., $a_t^m q_t$ cannot be used to choose a_t (equivalently, $a_t^m q_t$ is observed after a_t is chosen). This aligns our model with the standard adversarial online learning model in the literature.

4.2 The Reduction

To define our algorithm, some additional notation is needed. For a history h, define the length of h, denoted $\mathrm{len}(h)$, to be the total number of time steps covered by h. Note that $\mathrm{len}(h_t) = t - 1$. Given a history $h = (s_i', a_i', a_i^{m\prime} q_i')_{i\leq \mathrm{len}(h)}$ and vector $\boldsymbol{u} \in \{0, 1\}^T$, define the query-restricted history $h \cap \boldsymbol{u} = (s_i', a_i', a_i^{m\prime} q_i')_{i\leq \mathrm{len}(h):u_i=1}$. We will sometimes apply these definitions with $\boldsymbol{u} \neq \boldsymbol{q}$ and $(s_i', a_i', a_i^{m'}, q_i') \neq (s_i, a_i, a_i^m, q_i)$.

Given a full-feedback algorithm Γ_{FF} and expected query budget k (which need not be an integer), we define a new algorithm Γ_k by running Γ_{FF} on the query-restricted history

and querying independently with probability k/T on each time step. Formally,

$$\Gamma_k^Q(h, s) = \text{Bernoulli}(k/T) \text{ and}$$
$$\Gamma_k^{\mathcal{A}}(h, s) = \Gamma_{\text{FF}}^{\mathcal{A}}(h \cap (q_1, \ldots, q_{\text{len}(h)}), s).$$

We call attention to the following aspects of this definition:

1. If the algorithm is given a history h, then $\text{len}(h)$ time steps have already occurred and so the first $\text{len}(h)$ query decisions must be known to the algorithm. This enables the algorithm to compute $h \cap (q_1, \ldots, q_{\text{len}(h)})$. Furthermore, we have $h \cap (q_1, \ldots, q_{\text{len}(h)}) = h \cap \boldsymbol{q}$: by the definition of a query-restricted history, only time steps $t \leq \text{len}(h)$ can appear in $h \cap \boldsymbol{q}$, so the variables $(q_{\text{len}(h)+1}, \ldots, q_T)$ are ignored.
2. Since $\Gamma_k^{\mathcal{A}}$ does not depend on the query from the current time step, it is query-agnostic.
3. The algorithm as written needs to know T, but this requirement can be removed by standard doubling-trick arguments (Besson and Kaufmann, 2018; Chapter 2.3 in Cesa-Bianchi and Lugosi, 2006).

The algorithm Γ_k can equivalently be defined by the pseudocode in Algorithm 1.

Algorithm 1 The first reduction: given a full-feedback algorithm and an expected query budget, this algorithm obtains good regret while respecting the query budget.

Inputs: full-feedback algorithm Γ_{FF}, $T \in \mathbb{N}$, $k \in (0, T]$
for t **from** 1 **to** T **do**
  $\tilde{h}_t \leftarrow (s_i, a_i, a_i^m)_{i \in [t-1]: q_i = 1}$ ▷ Equal to $h_t \cap \boldsymbol{q}$
  Observe s_t from adversary
  Sample $a_t \sim \Gamma_{\text{FF}}^{\mathcal{A}}(\tilde{h}_t, s_t)$
  Sample $q_t \sim \text{Bernoulli}(k/T)$
  Take action a_t, make query decision q_t, and observe $a_t^m q_t$

Theorem 1 (First reduction) *Assume that $R : \mathbb{N} \times [0, 1] \to \mathbb{R}_{\geq 0}$ is concave in its first argument and that Γ_{FF} is a full-feedback algorithm which satisfies $\text{Reg}_{\text{SA}}(T, \Gamma_{\text{FF}}, V, \ell) \leq R(T, \sigma)$ for all $T \in \mathbb{N}, \sigma \in [0, 1]$, and $V \in \mathcal{V}(\sigma)$. Then for any $T \in \mathbb{N}, k \in (0, T], \sigma \in [0, 1]$, and $V \in \mathcal{V}(\sigma)$, Γ_k satisfies $Q(T, \Gamma_k, V) = k$ and*

$$\text{Reg}_{\text{SA}}(T, \Gamma_k, V, \ell) \leq \frac{T}{k} R(k, \sigma).$$

The proof of Theorem 1 can be found in Appendix A.

4.3 Applying the Reduction

To apply this reduction, we need a full-feedback algorithm. Many such algorithms exist, with typical regret bounds being roughly $O(\sqrt{T})$.[6] Theorem 1 can be applied to these results, but the resulting bounds are not good enough for our final reduction to produce sublinear regret in MDPs. Fortunately, we can obtain better bounds if we assume that Π contains a policy

6. Theorem 21.10 in Shalev-Shwartz and Ben-David (2014) is the canonical general bound. See Block et al. (2022); Haghtalab et al. (2022, 2024) for the σ-smooth case.

with zero loss. This assumption is called *realizability*. In our case, the loss function will be based on predicting the mentor's actions, so the mentor policy will have zero prediction loss, satisfying realizability. However, we do *not* assume that the mentor is optimal with respect to the MDP reward function that is our ultimate goal.

Lemma 2 is due to Littlestone (1988); see also Lemma 21.7 in Shalev-Shwartz and Ben-David (2014). Lemma 3 is folklore, but we provide a proof in Appendix D.

Lemma 2 *Assume* $\exists \pi^* \in \Pi$ *such that* $\pi^*(s_t) = a_t^m \ \forall t \in [T]$. *If* Π *has Littlestone dimension* d, *then there exists a full-feedback algorithm* Γ_{FF} *such that for any* $T \in \mathbb{N}$ *and* $V \in \mathcal{V}(0)$,

$$\text{Reg}_{\text{SA}}(T, \Gamma_{\text{FF}}, V, \ell_{\mathbf{1}}) \leq d.$$

Assumptions of finite Littlestone or VC dimension traditionally imply that $|\mathcal{A}| = 2$, so for brevity, we do not additionally state the $|\mathcal{A}| = 2$ assumption. While there exist multiclass generalizations of these dimensions (Daniely et al., 2015; Natarajan, 1989), they generally do not handle σ-smoothness. As such, we will use traditional Littlestone and VC dimension.

Lemma 3 *Assume* $\exists \pi^* \in \Pi$ *such that* $\pi^*(s_t) = a_t^m \ \forall t \in [T]$. *If* Π *has VC dimension* d, *there exists a full-feedback algorithm* Γ_{FF} *such that for any* $\sigma \in (0, 1]$ *and* $V \in \mathcal{V}(\sigma)$,

$$\text{Reg}_{\text{SA}}(T, \Gamma_{\text{FF}}, V, \ell_{\mathbf{1}}) \in O(d \log T + \sigma^+).$$

Letting $R(T, \sigma) = O(d \log T + \sigma^+)$, we obtain the following corollary of Theorem 1. (Note that $d \log T + \sigma^+$ is indeed concave in T.) Corollary 4 is the primary result we will use to instantiate our later reductions to obtain a complete no-regret guarantee for MDPs.

Corollary 4 (Main active learning bounds) *Assume* $\exists \pi^* \in \Pi$ *such that* $\pi^*(s_t) = a_t^m$ $\forall t \in [T]$ *and that either (1)* Π *has VC dimension* d *and* $\sigma \in (0, 1]$ *or (2)* Π *has Littlestone dimension* d *and* $\sigma = 0$. *Then for any* $k \in (0, T]$, *there exists an algorithm* Γ_k *such that for any* $V \in \mathcal{V}(\sigma)$ *we have* $Q(T, \Gamma_k, V) = k$ *and*

$$\text{Reg}_{\text{SA}}(T, \Gamma_k, V, \ell_{\mathbf{1}}) \in O\left(\frac{T}{k}(d \log k + \sigma^+)\right).$$

Corollary 4 assumes that $|\mathcal{A}| = 2$ (this is implied by finite VC or Littlestone dimension), but this assumption can easily be relaxed via the "one versus rest" reduction. Essentially, one runs a separate copy of the algorithm for each $a \in \mathcal{A}$ where each copy's goal is to predict whether $a_t^m = a$. If all copies are correct, then we can fully determine a_t^m. Since each copy is given a binary prediction task, Corollary 4 applies. Thus by the union bound, the total regret is at most the sum of regret bounds across the $|\mathcal{A}|$ copies, and similarly for queries. These bounds can likely be improved by, e.g., applying multiclass variants of VC or Littlestone dimension, but this is beyond the scope of the paper. See Appendix C of Plaut et al. (2025) or Chapter 29 of Shalev-Shwartz and Ben-David (2014) for more details.

5. Second Reduction: Avoiding Catastrophe → Standard Adversarial Active Learning

Our second reduction shows that any algorithm which performs well in the standard setting can be transformed into an algorithm which avoids catastrophe. The basic idea is to follow the original algorithm if the current state is "familiar" (defined by a small distance to previous relevant queries) and otherwise ask for help. To formalize this, we must define our model of avoiding catastrophe in online learning.

5.1 Our Model of Avoiding Catastrophe in Online Learning

In this model, the "correct" actions are determined by a mentor policy $\pi^m \in \Pi$ which is chosen upfront by the adversary. Formally, $a_t^m = \pi^m(s_t)$ for all $t \in [T]$. Conceptually, the adversary generates the environment, which includes the mentor policy. Once the mentor policy is chosen, the mentor simply reports $\pi^m(s_t)$ when queried.

Unlike before, the goal is not simply to predict the mentor's actions but to maximize a sequence of unknown reward functions $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_T) \in (\mathcal{S} \times \mathcal{A} \to [0,1])^T$. However, these are not "normal" reward functions: instead, $\mu_t(s_t, a_t)$ represents the probability of avoiding catastrophe at time t (conditioned on no prior catastrophe). Then the agent's overall chance of avoiding catastrophe is $\prod_{t=1}^{T} \mu_t(s_t, a_t)$. As before, we encourage the reader to think of "catastrophe" as an irreparable error; indeed, our final reduction will choose $\boldsymbol{\mu}$ to capture certain transition probabilities. However, the model is valid for any choice of $\boldsymbol{\mu}$. Importantly, we do not assume that π^m is optimal for $\boldsymbol{\mu}$.[7]

We wish to minimize the following multiplicative regret objective:

$$\mathrm{Reg}_{\mathrm{AC}}^{\times}(T, \Gamma, V, \boldsymbol{\mu}) = \mathbb{E}\left[\log \prod_{t=1}^{T} \mu_t(s_t, \pi^m(s_t)) - \log \prod_{t=1}^{T} \mu_t(s_t, a_t)\right].$$

In other words, the agent should avoid catastrophe nearly as well as the mentor. For brevity, let $\mu_t^m(s) = \mu_t(s, \pi^m(s))$ for $s \in \mathcal{S}$.

We assume that the agent never directly observes rewards: the only feedback it receives is from queries to the mentor. This is because in the real world, one never observes the true probability of catastrophe: only whether catastrophe occurred. (One may be able to detect "close calls" in some cases, but observing the precise probability seems unrealistic.)

The logarithms in the regret definition are included for consistency with the literature (e.g., Chapter 9 of Cesa-Bianchi and Lugosi, 2006) but do necessitate a special case for rewards of zero. One solution is to assume that rewards cannot be exactly zero. For an adversary V, let $\mu_0(V)$ be the infimum of possible rewards across all possible states, actions, and time steps given V. Typically V will be clear from context and we will just write μ_0. If $\mu_0 > 0$, then $\mathrm{Reg}_{\mathrm{AC}}^{\times}$ is always well-defined. We will explicitly note when we assume $\mu_0 > 0$, since this is slightly restrictive and our ultimate MDP results do not require this.

One can also consider a more standard additive regret objective:

$$\mathrm{Reg}_{\mathrm{AC}}^{+}(T, \Gamma, V, \boldsymbol{\mu}) = \mathbb{E}\left[\sum_{t=1}^{T} \mu_t(s_t, \pi^m(s_t)) - \sum_{t=1}^{T} \mu_t(s_t, a_t)\right].$$

7. Realizability only states that $\pi^m \in \Pi$; it says nothing about how much reward π^m will generate.

This objective can be interpreted as minimizing the "total amount of irreversible damage" rather than the probability of a single catastrophic event.

For both objectives, we want the *total* regret to go to 0 as $T \to \infty$, not just the average. This is because we want the total chance of catastrophe to go to 0, not just the average. Thus the goal in this model is *subconstant* regret instead of sublinear regret. This may be counterintuitive, since the regret is a sum of nonnegative quantities and thus is nondecreasing with t (at least for an optimal mentor). However, subconstant regret means that the total regret goes to 0 as a function of the time horizon T, not the time step t. Choosing algorithmic parameters as a function of T is the key to subconstant regret.

5.2 Local Generalization

We assume that $\boldsymbol{\mu}$ and π^m satisfy *L-local generalization*. Informally, if the mentor told us that action a is safe in state s', then a is probably also safe in a similar state s. Formally, let $\mathcal{S} \subseteq \mathbb{R}^n$ (one could also allow a generic metric space). We assume that there exists $L > 0$ such that for all $s, s' \in \mathcal{S}$ and $t \in [T]$, $|\mu_t^m(s) - \mu_t(s, \pi^m(s'))| \leq L||s - s'||$, where $||\cdot||$ denotes Euclidean distance. This represents the ability to transfer knowledge between similar states:

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

For $X \subseteq \mathcal{S}$, define $\operatorname{diam}(X) = \max_{s,s' \in X} ||s - s'||$. For brevity, we write $\operatorname{diam}(\boldsymbol{s}) = \operatorname{diam}(\{s_1, \ldots, s_T\})$. Let $\mathcal{U}(L)$ be the set of all $(\boldsymbol{\mu}, \pi^m)$ pairs satisfying L-local generalization.

The local generalization assumption is vital to our results: this is what allows us to detect when a state is unfamiliar. As such, we now take some time to justify this assumption in detail.

First, the ability to transfer knowledge between similar states seems fundamental to intelligence and is well-understood in psychology (e.g., Esser et al., 2023) and education (e.g., Hajian, 2019). Crucially, the state space $\mathcal{S} \subseteq \mathbb{R}^n$ can be any encoding of the agent's situation, not just its physical positioning. For example, a 3 mm spot and a 3.1 mm spot on X-rays likely have similar risk levels for cancer (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, nor does it need to know L: it only needs a nearest-neighbor distance oracle. Formally, the agent needs to compute $\min_{s \in X} ||s_t - s||$, where $X \subseteq \mathcal{S}$ is a particular subset of previously queried states. Informally, the agent only needs the ability to detect unfamiliar states. While this task remains far from trivial, we argue that it is more tractable than fully constructing a suitable encoding. See Section 7 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 in Plaut et al., 2025). We also mention that without local generalization, avoiding catastrophe is impossible even when the mentor policy class has finite VC dimension and the adversary is σ-smooth (Theorem E.2 in Plaut et al., 2025).

5.3 The Reduction

We can now define our reduction. Given access to a "default" algorithm Γ which performs well in the absence of catastrophe, Algorithm 2 first computes what Γ would do in the current state (based on a simulated history of what Γ would have done in previous states). If the current state is "familiar" (i.e., has a small nearest-neighbor distance $\min_{(s,a)\in X:a=\tilde{a}_t} ||s_t - s||$), we follow Γ. Otherwise, we ask for help. The "familiarity" threshold ε must be carefully chosen to balance safety with self-sufficiency. Ultimately we choose $\varepsilon = T^{\frac{-1}{n+1}}$ (recall that n is the dimension of $\mathcal{S}$). However, we first prove bounds for general ε. Similar to the first reduction, Algorithm 2 requires knowledge of T, but this can be removed with standard doubling tricks.

Algorithm 2 The second reduction: given an algorithm which performs well in the absence of catastrophe, this algorithm avoids catastrophe using a limited number of additional queries.

Inputs: Algorithm Γ, $\varepsilon \in \mathbb{R}_{>0}$, $T \in \mathbb{N}$
$X \leftarrow \emptyset$
for t **from** 1 **to** T **do**
    Observe s_t from adversary
    $\tilde{h}_t \leftarrow \left(s_i, \tilde{a}_i, \pi^m(s_i)\tilde{q}_i\right)_{i\in[t-1]}$ ▷ Simulated history for Γ
    Sample $\tilde{q}_t \sim \Gamma^Q(\tilde{h}_t, s_t)$
    Sample $\tilde{a}_t \sim \Gamma^{\mathcal{A}}(\tilde{h}_t, s_t, 0)$
    if $\min_{(s,a)\in X:a=\tilde{a}_t} ||s_t - s|| > \varepsilon$ **then** ▷ Out-of-distribution: ask for help
        $q_t \leftarrow 1$, observe $\pi^m(s_t)$
        $a_t \leftarrow \pi^m(s_t)$
        $X \leftarrow X \cup \{(s_t, \pi^m(s_t))\}$
    else ▷ In-distribution: follow Γ
        $q_t \leftarrow \tilde{q}_t$, observe $\pi^m(s_t)q_t$
        $a_t \leftarrow \tilde{a}_t$

Theorem 5 (Second reduction) *Let $k \in \mathbb{R}_{\geq 0}$ and $R : \mathbb{R}_{\geq 0} \times \mathbb{N} \times [0,1] \to \mathbb{R}_{\geq 0}$. Let Γ be a query-agnostic algorithm satisfying $Q(T, \Gamma, V) \leq k$ and $\mathrm{Reg}_{\mathrm{SA}}(T, \Gamma, V, \ell_{\mathbf{1}}) \leq R(k, T, \sigma)$ for all $T \geq k, \sigma \in [0,1]$, and $V \in \mathcal{V}(\sigma)$. Let Γ_{AC} denote Algorithm 2 with inputs Γ, $T \geq k$, and $\varepsilon > 0$. Then for any $\sigma \in [0,1], V \in \mathcal{V}(\sigma)$, and $(\boldsymbol{\mu}, \pi^m) \in \mathcal{U}(L)$,*

$$\begin{aligned}
\mathrm{Reg}^{\times}_{\mathrm{AC}}(T, \Gamma_{\mathrm{AC}}, V, \boldsymbol{\mu}) &\leq \frac{L\varepsilon R(k,T,\sigma)}{\mu_0} \quad \text{if } \mu_0 > 0, \\
\mathrm{Reg}^{+}_{\mathrm{AC}}(T, \Gamma_{\mathrm{AC}}, V, \boldsymbol{\mu}) &\leq L\varepsilon R(k,T,\sigma), \\
Q(T, \Gamma_{\mathrm{AC}}, V) &\in O\left(k + R(k,T,\sigma) + \frac{|\mathcal{A}|\,\mathbb{E}[\mathrm{diam}(\boldsymbol{s})^n]}{\varepsilon^n}\right).
\end{aligned}$$

The proof of Theorem 5 appears in Appendix B. To obtain subconstant regret and sublinear queries, we apply Theorem 5 to Corollary 4 with $k = T^{\frac{2n+1}{2n+2}}$ and $\varepsilon = T^{\frac{-1}{n+1}}$. For readability, we have simplified the bounds at the cost of slightly looser bounds.

Theorem 6 (Final avoiding catastrophe bounds) *Assume that either (1) Π has VC dimension d and $\sigma \in (0,1]$ or (2) Π has Littlestone dimension d and $\sigma = 0$. Let $k = T^{\frac{2n+1}{2n+2}}$ and let Γ_k be the corresponding algorithm from Corollary 4. If Γ_{AC} denotes Algorithm 2 with inputs Γ_k, T, and $\varepsilon = T^{\frac{-1}{n+1}}$, then*

$$\begin{aligned}
\mathrm{Reg}^{\times}_{\mathrm{AC}}(T, \Gamma_{\mathrm{AC}}, V, \boldsymbol{\mu}) &\in O\left(\frac{L}{\mu_0} T^{\frac{-1}{2n+2}} (d \log T + \sigma^+)\right) \quad \text{if } \mu_0 > 0, \\
\mathrm{Reg}^{+}_{\mathrm{AC}}(T, \Gamma_{\mathrm{AC}}, V, \boldsymbol{\mu}) &\in O\left(L T^{\frac{-1}{2n+2}} (d \log T + \sigma^+)\right), \\
Q(T, \Gamma_{\mathrm{AC}}, V) &\in O\left(T^{\frac{2n+1}{2n+2}} \left(d + \sigma^+ + \mathbb{E}[\mathrm{diam}(\boldsymbol{s})^n]\right)\right),
\end{aligned}$$

for any $(\boldsymbol{\mu}, \pi^m) \in \mathcal{U}(L)$ and $V \in \mathcal{V}(\sigma)$.

Remark 7 Because the agent never observes rewards, the distribution of $(\boldsymbol{s}, \boldsymbol{a})$ depends only on the adversary V (which determines π^m) and the algorithm Γ_{AC}, and not on $\boldsymbol{\mu}$. Hence this *single* distribution of $(\boldsymbol{s}, \boldsymbol{a})$ satisfies the bounds in Theorem 6 *simultaneously* for all admissible choices of $\boldsymbol{\mu}$. In other words, run the online learning protocol for T time steps without choosing $\boldsymbol{\mu}$. Then, at the end, evaluate $\sup_{\boldsymbol{\mu}:(\boldsymbol{\mu},\pi^m)\in\mathcal{U}(L)} \mathrm{Reg}^{\times}_{\mathrm{AC}}(T, \Gamma_{\mathrm{AC}}, V, \boldsymbol{\mu})$, and similar for $\mathrm{Reg}^{+}_{\mathrm{AC}}$. This property will be important for the third reduction.

6. Third Reduction: Sublinear Regret in MDPs → Avoiding Catastrophe

Our third reduction shows that any algorithm which avoids catastrophe (i.e., has subconstant regret in the model from Section 5.1) guarantees sublinear regret in multichain MDPs. This leads to our ultimate goal: a no-regret guarantee for multichain MDPs using sublinear mentor queries.

6.1 Online Learning in MDPs with a Mentor

MDPs involve a type of constrained adversary who only chooses a transition function $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$, a mentor policy $\pi^m : \mathcal{S} \to \mathcal{A}$, and an initial state distribution

$\mathcal{D}_1 \in \Delta(\mathcal{S})$. Then $s_1 \sim \mathcal{D}_1$ and for all $t > 1$, s_t is sampled from $P(s_{t-1}, a_{t-1})$. As in Section 5.1, $a_t^m = \pi^m(s_t)$ for all $t \in [T]$. Sometimes it will be useful to denote the adversary as $V = (P, \mathcal{D}_1, \pi^m)$, in which case we can also write $(P, \mathcal{D}_1, \pi^m) \in \mathcal{V}(\sigma)$.

The agent's objective is to maximize a reward function $r : \mathcal{S} \times \mathcal{A} \to [0, 1]$. As in Section 5.1, we do not assume that the mentor is optimal. Also like Section 5.1, the agent here never directly observes rewards and only receives feedback from queries.[8]

Unlike the other models, the transition function allows us to define a distribution of *mentor states* $\boldsymbol{s^m} = (s_1^m, \ldots, s_T^m)$, where $s_1^m \sim \mathcal{D}_1$ and for $t > 1$, $s_t^m \sim P(s_{t-1}^m, \pi^m(s_{t-1}^m))$. Thus we can define MDP regret by comparing the expected reward of the agent and mentor on their *respective* sequences of states:

$$\mathrm{Reg}_{\mathrm{MDP}}(T, \Gamma, (P, \mathcal{D}_1, \pi^m), r) = \mathbb{E}\left[\sum_{t=1}^{T} r(s_t^m, \pi^m(s_t^m)) - \sum_{t=1}^{T} r(s_t, a_t)\right].$$

This corresponds to independently running the mentor and the agent each for T steps from the same initial state and comparing their expected total reward. For example, recall Figure 2 and suppose the mentor goes to Heaven but the agent goes to Hell. Our definition of regret appropriately penalizes such an agent: for all $t > 1$, we have $s_t^m = \text{Heaven}$ while $s_t = \text{Hell}$, resulting in a very poor regret of $T - 1$.

We assume that P and π^m satisfy L-local generalization with respect to total variation distance: $||P(s, \pi^m(s)) - P(s, \pi^m(s'))||_{TV} \leq L||s - s'||$ for any $s, s' \in \mathcal{S}$. (We assume $\mathcal{S} \subseteq \mathbb{R}^n$ and define $\mathrm{diam}(X)$ as in Section 5.1.) The most general version of the third reduction will also require L-local generalization for r: $|r(s, \pi^m(s)) - r(s, \pi^m(s'))| \leq L||s - s'||$ for any $s, s' \in \mathcal{S}$. We assume that L is the same for both P and r (if not, simply use the maximum). Let $\mathcal{P}(L)$ and $\mathcal{R}(L)$ denote the sets of (P, π^m) and (r, π^m) pairs satisfying L-local generalization, respectively.

6.2 The Reduction

This reduction requires no modification of the algorithm: any algorithm which has subconstant regret for $\mathrm{Reg}_{\mathrm{AC}}^+$ must have certain properties which ensure that it will also perform well in multichain MDPs.[9] Specifically, such an algorithm will have MDP regret at most $(T + 1) \cdot \mathrm{Reg}_{\mathrm{AC}}^+$, as stated by Theorem 8. If $\mathrm{Reg}_{\mathrm{AC}}^+$ is subconstant, then the resulting MDP regret is sublinear.

Theorem 8 (Third reduction) *Fix some $R : \mathbb{N} \times [0, 1] \to \mathbb{R}_{\geq 0}$ and let Γ be an algorithm which satisfies $\mathrm{Reg}_{\mathrm{AC}}^+(T, \Gamma, V, \boldsymbol{\mu}) \leq R(T, \sigma)$ for all $T \in \mathbb{N}, \sigma \in [0, 1], V \in \mathcal{V}(\sigma)$, and $(\boldsymbol{\mu}, \pi^m) \in \mathcal{U}(L)$. Then Γ satisfies*

$$\mathrm{Reg}_{\mathrm{MDP}}(T, \Gamma, (P, \mathcal{D}_1, \pi^m), r) \leq (T + 1)R(T, \sigma)$$

for all $T \in \mathbb{N}, \sigma \in [0, 1]$, and $(P, \mathcal{D}_1, \pi^m) \in \mathcal{V}(\sigma)$ with $(P, \pi^m) \in \mathcal{P}(L)$ and $(r, \pi^m) \in \mathcal{R}(L)$.

If we can also reduce to the standard adversarial setting—that is, we have access to a bound on $\mathrm{Reg}_{\mathrm{SA}}$—then we do not need local generalization for r.

8. In the typical MDP framework, the agent does observe rewards directly. However, our algorithms do not use these observations, so we might as well strengthen our results by removing such observations.
9. We reduce to $\mathrm{Reg}_{\mathrm{AC}}^+$ instead of $\mathrm{Reg}_{\mathrm{AC}}^\times$ to avoid dealing with μ_0.

Theorem 9 (Third reduction: no local generalization for r) *For functions $R_1, R_2 : \mathbb{N} \times [0,1] \to \mathbb{R}_{\geq 0}$, let Γ be an algorithm which satisfies $\mathrm{Reg}_{\mathrm{SA}}(T, \Gamma, V, \ell_{\mathbf{1}}) \leq R_1(T, \sigma)$ and $\mathrm{Reg}^{+}_{\mathrm{AC}}(T, \Gamma, V, \boldsymbol{\mu}) \leq R_2(T, \sigma)$ for any $T \in \mathbb{N}, \sigma \in [0,1], V \in \mathcal{V}(\sigma)$, and $(\boldsymbol{\mu}, \pi^m) \in \mathcal{U}(L)$. Then Γ satisfies*

$$\mathrm{Reg}_{\mathrm{MDP}}(T, \Gamma, (P, \mathcal{D}_1, \pi^m), r) \leq R_1(T, \sigma) + T R_2(T, \sigma)$$

for any $T \in \mathbb{N}, \sigma \in [0,1]$, and $(P, \mathcal{D}_1, \pi^m) \in \mathcal{V}(\sigma)$ with $(P, \pi^m) \in \mathcal{P}(L)$.

Theorem 9 requires a single algorithm with bounds on both $\mathrm{Reg}_{\mathrm{SA}}$ and $\mathrm{Reg}^{+}_{\mathrm{AC}}$. Although $\mathrm{Reg}_{\mathrm{SA}}$ and $\mathrm{Reg}^{+}_{\mathrm{AC}}$ originate from different models, one advantage of our unifying online learning framework is that these metrics are well-defined for any algorithm in our framework. Furthermore, in the course of proving Theorem 5, we actually already proved that Γ_{AC} (Algorithm 2) has bounds on both $\mathrm{Reg}_{\mathrm{SA}}$ and $\mathrm{Reg}^{+}_{\mathrm{AC}}$. Specifically, this is stated by Lemma 24, which we can directly reuse.

The culmination of our work is Theorem 10 below, which establishes the existence of a no-regret algorithm for multichain MDPs with sublinear queries. We use the exact same algorithm as Theorem 6. The bound on $Q(T, \Gamma, V) = Q(T, \Gamma, (P, \mathcal{D}_1, \pi^m))$ is automatically inherited and the regret bound only requires a short proof. Essentially, we apply Theorem 9 with $R_1(T, \sigma)$ as the bound on $\mathrm{Reg}_{\mathrm{SA}}$ from Corollary 4 and $R_2(T, \sigma)$ as the bound on $\mathrm{Reg}^{+}_{\mathrm{AC}}$ from Theorem 6.

Theorem 10 (Final no-regret guarantee for multichain MDPs) *Assume that either (1) Π has VC dimension d and $\sigma \in (0,1]$ or (2) Π has Littlestone dimension d and $\sigma = 0$. Let $k = T^{\frac{2n+1}{2n+2}}$, let Γ_k be the corresponding algorithm from Corollary 4, and let Γ_{AC} denote Algorithm 2 with inputs Γ_k, T, and $\varepsilon = T^{\frac{-1}{n+1}}$. Then Γ_{AC} satisfies*

$$\mathrm{Reg}_{\mathrm{MDP}}(T, \Gamma_{\mathrm{AC}}, (P, \mathcal{D}_1, \pi^m), r) \in O\left(L T^{\frac{2n+1}{2n+2}} (d \log T + \sigma^{+})\right),$$
$$Q(T, \Gamma_{\mathrm{AC}}, (P, \mathcal{D}_1, \pi^m)) \in O\left(T^{\frac{2n+1}{2n+2}} \left(d + \sigma^{+} + \mathbb{E}[\mathrm{diam}(\boldsymbol{s})^n]\right)\right),$$

for any $(P, \mathcal{D}_1, \pi^m) \in \mathcal{V}(\sigma)$ with $(P, \pi^m) \in \mathcal{P}(L)$.

For concreteness, we present a consolidated complete algorithm for the σ-smooth case below. Since our reduction is quite general, this is only one possible algorithm, but it is sufficient for illustrative purposes. For the full-feedback algorithm Γ_{FF}, we will use the algorithm from Lemma 3. This algorithm consists of forming a $\frac{1}{T}$-cover (see Appendix D for the definition) and running the "exponentially weighted average forecaster" from Corollary 2.3 of Cesa-Bianchi and Lugosi (2006) on this cover. The algorithm assumes that $A = \{0, 1\}$.

Corollary 11 *Assume that Π has VC dimension d and that $\sigma \in (0,1]$. Let Γ_{final} denote Algorithm 3 with inputs $k = T^{\frac{2n+1}{2n+2}}$ and $\varepsilon = T^{\frac{-1}{n+1}}$. Then Γ_{final} satisfies*

$$\mathrm{Reg}_{\mathrm{MDP}}(T, \Gamma_{\mathrm{final}}, (P, \mathcal{D}_1, \pi^m), r) \in O\left(L T^{\frac{2n+1}{2n+2}} (d \log T + \sigma^{-1})\right),$$
$$Q(T, \Gamma_{\mathrm{final}}, (P, \mathcal{D}_1, \pi^m)) \in O\left(T^{\frac{2n+1}{2n+2}} \left(d + \sigma^{-1} + \mathbb{E}[\mathrm{diam}(\boldsymbol{s})^n]\right)\right),$$

for any $(P, \mathcal{D}_1, \pi^m) \in \mathcal{V}(\sigma)$ with $(P, \pi^m) \in \mathcal{P}(L)$.

Algorithm 3 A final concrete algorithm obtained by applying our reduction to the exponentially-weighted average predictor from Cesa-Bianchi and Lugosi (2006).

Inputs: $T \in \mathbb{N}$, $k \in (0, T]$, $\varepsilon \in \mathbb{R}_{>0}$
$\tilde{\Pi} \leftarrow$ any $\frac{1}{T}$-cover of Π
$X \leftarrow \emptyset$
for $\pi \in \tilde{\Pi}$ **do**
    $w_\pi \leftarrow 1/|\tilde{\Pi}|$
for t **from** 1 **to** T **do**
    Observe s_t
    $p_t \leftarrow \sum_{\pi \in \tilde{\Pi}} w_\pi \pi(s_t)$
    Sample $\tilde{a}_t \sim \text{Bernoulli}(p_t)$
    `is_ood` $\leftarrow \left(\min_{(s,a) \in X : a = \tilde{a}_t} ||s_t - s|| > \varepsilon \right)$
    if `is_ood` **then** ▷ Out-of-distribution: ask for help
        $q_t \leftarrow 1$, observe $\pi^m(s_t)$
        $a_t \leftarrow \pi^m(s_t)$
        $X \leftarrow X \cup \{(s_t, \pi^m(s_t))\}$
    else ▷ In-distribution: follow exponentially-weighted average
        $a_t \leftarrow \tilde{a}_t$
    Sample $\tilde{q}_t \sim \text{Bernoulli}(k/T)$ ▷ Update weights w.p. k/T, even on OOD rounds
    if $\tilde{q}_t = 1$ **then**
        $q_t \leftarrow 1$, observe $\pi^m(s_t)$
        $W \leftarrow \sum_{\pi \in \tilde{\Pi}} w_\pi \exp(-\mathbf{1}(\pi^m(s_t) \neq \pi(s_t)))$
        for $\pi \in \tilde{\Pi}$ **do**
            $w_\pi \leftarrow \dfrac{w_\pi \exp(-\mathbf{1}(\pi^m(s_t) \neq \pi(s_t)))}{W}$
    if $\tilde{q}_t = 0$ and not `is_ood` **then**
        $q_t \leftarrow 0$

6.3 Key Proof Idea

The key idea is to decompose regret into *state-based regret* and *action-based regret* by simply adding and subtracting $\mathbb{E}[\sum_{t=1}^{T} r(s_t, \pi^m(s_t))]$:

$$\mathbb{E}\left[\sum_{t=1}^{T} r(s_t^m, \pi^m(s_t^m)) - \sum_{t=1}^{T} r(s_t, a_t)\right] =$$

$$\underbrace{\mathbb{E}\left[\sum_{t=1}^{T} r(s_t^m, \pi^m(s_t^m)) - \sum_{t=1}^{T} r(s_t, \pi^m(s_t))\right]}_{\text{State-based regret}} + \underbrace{\mathbb{E}\left[\sum_{t=1}^{T} r(s_t, \pi^m(s_t)) - \sum_{t=1}^{T} r(s_t, a_t)\right]}_{\text{Action-based regret}}.$$

State-based regret measures how bad the agent's states $\boldsymbol{s}$ are compared to the mentor's states $\boldsymbol{s^m}$. Here "bad" is evaluated based on the actions the mentor would take in each state. To bound the state-based regret, we use the local generalization of P to bound the deviation between the distributions of $\boldsymbol{s}$ and $\boldsymbol{s^m}$ as measured by $\sup_{X \subseteq \mathcal{S}}(\Pr[s_t^m \in X] - \Pr[s_t \in X])$. Most of the proof is focused on showing that the state-based regret is at most $T\mathrm{Reg}_{\mathrm{AC}}^{+}$.

Action-based regret measures how bad the agent's actions are compared to the mentor's, evaluated on the agent's states $\boldsymbol{s}$. Local generalization of r will imply that the action-based regret is at most $\mathrm{Reg}_{\mathrm{AC}}^{+}$. Combining this with our bound on state-based regret yields $\mathrm{Reg}_{\mathrm{MDP}} \leq (T+1)\mathrm{Reg}_{\mathrm{AC}}^{+}$, as desired.

To our knowledge, this decomposition is novel and could be of independent interest. Superficially, our regret decomposition might resemble existing techniques like the reference-advantage decomposition (Zhang et al., 2020). The key difference is that each term in our decomposition includes only short-term rewards, and long-term effects remain implicit. In contrast, prior decompositions (including reference-advantage) typically use Q-functions and/or value functions which explicitly capture the long-term value of policies/states/actions. Neither approach is necessarily "better", but they may be suitable for different contexts.

7. Conclusion

In this paper, we provide a sequence of three reductions which produce (to our knowledge) the first no-regret guarantee for multichain MDPs. Conceptually, we prove that this algorithm obtains high reward while becoming self-sufficient, even when errors may be catastrophic.

Although we think these insights are broadly applicable, our algorithm is not ready for practical usage. For one, computing the $\frac{1}{T}$-cover is computationally expensive. There exist techniques for avoiding computing such covers explicitly (see, e.g., Block et al., 2022), and this could be a topic for future work.

Also, our final query and regret bounds of $\tilde{O}(T^{\frac{2n+1}{2n+2}})$ are only barely sublinear in T when n is large. The exponent arises from the number of queries needed to cover an n-dimensional space; essentially, the curse of dimensionality. Another factor is that our agent only learns from querying. Providing an initial offline data set would remove the need to cover the entire space with mentor queries and could significantly improve our bounds.

Another limitation is that the algorithm relies on perfectly computing distances between states in order to exploit local generalization. This is analogous to requiring a perfect

out-of-distribution (OOD) detector, which remains a major open problem (Yang et al., 2024). Future work might consider a more realistic model of OOD detection and/or a less strict version of the local generalization assumption. Our work also makes the standard yet limiting assumptions of total observability and knowledge of the policy class, which could be relaxed in future work.

We are also interested in cautious learning without a mentor, since a mentor may not always be available. One possibility is learning from bad-but-not-catastrophic experiences: for example, someone who gets serious (but not fatal) food poisoning may be more cautious with food safety in the future. This phenomenon is not captured in our current model.

More broadly, we think that the principle of acting cautiously when uncertain may be more powerful than previously suspected. We are hopeful that this idea can help make AI systems safer and more beneficial for all of society.

Acknowledgments and Disclosure of Funding

This work was supported by a gift from Open Philanthropy to the Center for Human-Compatible AI at UC Berkeley. We have no competing interests to declare.

We would like to thank (in alphabetical order) Aly Lidayan, Bhaskar Mishra, Cameron Allen, Daniel Jarne Ornia, Karim Abdel Sadek, Matteo Russo, Michael Cohen, Nika Haghtalab, Ondrej Bajgar, Scott Emmons, and Tianyi Alex Qiu for feedback and discussion which significantly improved the paper.

Appendix A. First Reduction Proofs

This section presents the proofs for our first reduction (Theorem 1).

A.1 Proof Notation

In the proof, we fix an arbitrary $T \in \mathbb{N}, \sigma \in [0,1], V \in \mathcal{V}(\sigma)$ and use the following notation:

1. Let $K = \sum_{t=1}^{T} q_t$ denote the realized number of queries. Note that $\mathbb{E}[K] = k$.
2. Let $t_1 < t_2 < \cdots < t_K$ be the time steps $t \in [T]$ where $q_t = 1$.
3. For each $t \in [T]$, define $\tilde{\ell}_t(a, a^m) = q_t \frac{T}{k} \ell(a, a^m)$.
4. Let $\Delta_T(\Gamma_k, \pi) = \sum_{t=1}^{T} \ell(a_t, a_t^m) - \sum_{t=1}^{T} \ell(\pi(s_t), a_t^m)$ for each $\pi \in \Pi$. Note that $\mathrm{Reg}_{\mathrm{SA}}(T, \Gamma_k, V, \ell) = \sup_{\pi \in \Pi} \mathbb{E}[\Delta_T(\Gamma_k, \pi)]$.
5. Let $\tilde{\Delta}_T(\Gamma_k, \pi) = \sum_{t=1}^{T} \tilde{\ell}_t(a_t, a_t^m) - \sum_{t=1}^{T} \tilde{\ell}_t(\pi(s_t), a_t^m)$.

The trickiest part of the proof is proving the equivalence between running Γ_k on the true instance $\mathcal{I} = (T, V, \Pi, \ell)$ and running Γ_{FF} on a different instance $\mathcal{I}^{\boldsymbol{u}}$, where $\boldsymbol{u} \in \{0,1\}^T$ is a possible realization of $\boldsymbol{q}$. Formally, for each $\sigma \in [0,1]$, $V \in \mathcal{V}(\sigma)$, and $\boldsymbol{u} \in \{0,1\}^T$, define $\mathcal{I}^{\boldsymbol{u}} = (T^{\boldsymbol{u}}, V^{\boldsymbol{u}}, \Pi, \ell, \Gamma_{\mathrm{FF}})$ as follows.

1. Let $T^{\boldsymbol{u}} = \sum_{t=1}^{T} u_t$.
2. For a history h of length $j-1$, define $V^{\boldsymbol{u}}(h) = \mathcal{D}(s_{t_j}, a_{t_j}^m \mid h_{t_j} \cap \boldsymbol{u} = h, \boldsymbol{q} = \boldsymbol{u})$ if $\Pr[h_{t_j} \cap \boldsymbol{u} = h, \boldsymbol{q} = \boldsymbol{u}] > 0$. Otherwise, let $V^{\boldsymbol{u}}(h)$ be any distribution whose marginal distribution on $\mathcal{S}$ is σ-smooth.

3. For clarity, let s_t, a_t^m, a_t, q_t, and h_t respectively denote the state, correct action, algorithm's action, query decision, and history at time t in $\mathcal{I}$, i.e., as induced by V and Γ_k. Let $s_t^{\boldsymbol{u}}, a_t^{m,\boldsymbol{u}}, a_t^{\boldsymbol{u}}, q_t^{\boldsymbol{u}}$, and $h_t^{\boldsymbol{u}}$ denote the analogous random variables in $\mathcal{I}^{\boldsymbol{u}}$, i.e., as induced by $V^{\boldsymbol{u}}$ and Γ_{FF}.[10]

A.2 Proof

Lemma 12 shows that $\tilde{\Delta}_T$ is an unbiased estimator of Δ_T. Lemma 14 ensures that $V^{\boldsymbol{u}}$ is σ-smooth, which is necessary in order to apply the assumed regret bound for Γ_{FF}. Lemma 15 is the key to the proof: it states that the distribution of histories is the same for $\mathcal{I}$ and $\mathcal{I}^{\boldsymbol{u}}$ (under a particular transformation). To complete the proof of Theorem 1, we apply the regret bound for Γ_{FF} to $\mathcal{I}^{\boldsymbol{u}}$, which we can then lift to a regret bound for Γ_k on the original instance $\mathcal{I}$ using Lemmas 12 and 15.

Lemma 12 *For all $\pi \in \Pi$, $\mathbb{E}[\tilde{\Delta}_T(\Gamma_k, \pi)] = \mathbb{E}[\Delta_T(\Gamma_k, \pi)]$.*

Proof Fix a $t \in [T]$ and let a be any random variable taking values in $\mathcal{A}$ such that q_t is independent of the pair (a, a_t^m). Then q_t is also independent of $\tilde{\ell}_t(a, a_t^m)$ so

$$\mathbb{E}[\tilde{\ell}_t(a, a_t^m)] = \mathbb{E}[q_t]\,\mathbb{E}\left[\frac{T}{k}\ell(a, a_t^m)\right] = \frac{k}{T}\frac{T}{k}\,\mathbb{E}[\ell(a, a_t^m)] = \mathbb{E}[\ell(a, a_t^m)].$$

Since $q_t \sim \text{Bernoulli}(k/T)$, we know that q_t is independent of both (a_t, a_t^m) and $(\pi(s_t), a_t^m)$. Therefore we can apply $\mathbb{E}[\tilde{\ell}_t(a, a_t^m)] = \mathbb{E}[\ell(a, a_t^m)]$ with both $a = a_t$ and $a = \pi(s_t)$, so

$$\begin{aligned}
\mathbb{E}[\Delta_T(\Gamma_k, \pi)] &= \sum_{t=1}^{T} \mathbb{E}[\ell(a_t, a_t^m)] - \sum_{t=1}^{T} \mathbb{E}[\ell(\pi(s_t), a_t^m)] \\
&= \sum_{t=1}^{T} \mathbb{E}[\tilde{\ell}_t(a_t, a_t^m)] - \sum_{t=1}^{T} \mathbb{E}[\tilde{\ell}_t(\pi(s_t), a_t^m)] \\
&= \mathbb{E}[\tilde{\Delta}_T(\Gamma_k, \pi)]
\end{aligned}$$

as required. ■

Lemma 13 (Tower property) *For random variables X, Y and an event E, if $\mathbb{E}_X[X \mid E]$ and $\mathbb{E}_Y[\mathbb{E}_X[X \mid Y, E] \mid E]$ both exist, then $\mathbb{E}_X[X \mid E] = \mathbb{E}_Y[\mathbb{E}_X[X \mid Y, E] \mid E]$.*

Although the subscripts of the expectations in Lemma 13 are not technically necessary, we include them to improve readability.

Lemma 14 *For each $\boldsymbol{u} \in \{0,1\}^T$, the adversary $V^{\boldsymbol{u}}$ is σ-smooth.*

Proof Fix some $j \in [T^{\boldsymbol{u}}]$, $h \in \mathcal{H}$, and $X \subseteq \mathcal{S}$. If $\sigma = 0$, the claim is trivial. If $\Pr[h_{t_j} \cap \boldsymbol{u} = h, \boldsymbol{q} = \boldsymbol{u}] = 0$, then the claim holds by definition of $V^{\boldsymbol{u}}$. Thus assume $\sigma > 0$ and

10. Since Γ_{FF} is a full-feedback algorithm, $q_t^{\boldsymbol{u}}$ is always 1, but we include it for completeness.

$V^{\boldsymbol{u}}(h) = \mathcal{D}(s_{t_j}, a_{t_j}^m \mid h_{t_j} \cap \boldsymbol{u} = h, \boldsymbol{q} = \boldsymbol{u})$. Let $E = \{h_{t_j} \cap \boldsymbol{u} = h, \boldsymbol{q} = \boldsymbol{u}\}$. Applying Lemma 13 with $X = \mathbf{1}(s_{t_j} \in X)$ and $Y = h_{t_j}$ gives us

$$\Pr_{(s,a)\sim V^{\boldsymbol{u}}(h)}[s \in X] = \Pr[s_{t_j} \in X \mid h_{t_j} \cap \boldsymbol{u} = h, \boldsymbol{q} = \boldsymbol{u}] = \mathop{\mathbb{E}}_{h_{t_j}}\big[\Pr[s_{t_j} \in X \mid E, h_{t_j}] \;\big|\; E\big].$$

We claim that $\Pr[s_{t_j} \in X \mid E, h_{t_j}] = \Pr[s_{t_j} \in X \mid h_{t_j}]$. Define events $E_1 = \{h_{t_j} \cap \boldsymbol{u} = h\}, E_2 = \{q_i = u_i \; \forall i < t_j\}$, and $E_3 = \{q_i = u_i \; \forall i \geq t_j\}$. Notice that all probability information in E_1 and E_2 is already contained in h_{t_j}, and that h_{t_j} is independent of E_3.[11] Thus

$$\begin{aligned}
\mathop{\mathbb{E}}_{h_{t_j}}\big[\Pr[s_{t_j} \in X \mid E, h_{t_j}] \mid E\big] &= \mathop{\mathbb{E}}_{h_{t_j}}\big[\Pr[s_{t_j} \in X \mid E_3, h_{t_j}] \mid E\big] && (E_1, E_2 \text{ contained within } h_{t_j}) \\
&= \mathop{\mathbb{E}}_{h_{t_j}}\big[\Pr[s_{t_j} \in X \mid h_{t_j}] \mid E\big] && (\text{Independence of } E_3, h_{t_j}) \\
&= \mathop{\mathbb{E}}_{h_{t_j}}\left[\Pr_{(s,a)\sim V(h_{t_j})}[s \in X] \;\middle|\; E\right] && (\text{Definition of } s_{t_j} \sim V(h_{t_j})) \\
&\leq \mathop{\mathbb{E}}_{h_{t_j}}\left[\frac{1}{\sigma}\beta(X) \;\middle|\; E\right] && (V \text{ is } \sigma\text{-smooth}) \\
&= \frac{1}{\sigma}\beta(X). && (\text{Expectation of a constant})
\end{aligned}$$

Therefore $\Pr_{(s,a)\sim V^{\boldsymbol{u}}(h)}[s \in X] \leq \frac{1}{\sigma}\beta(X)$, as required. ■

Lemma 15 *For any $g : \mathcal{S} \times \mathcal{A} \times \mathcal{A} \to \mathbb{R}$, $\mathbb{E}[g(s_j^{\boldsymbol{u}}, a_j^{m,\boldsymbol{u}}, a_j^{\boldsymbol{u}})] = \mathbb{E}[g(s_{t_j}, a_{t_j}^m, a_{t_j}) \mid \boldsymbol{q} = \boldsymbol{u}]$.*

Proof We prove by induction that for any $h \in \mathcal{H}$ and $j \in [T^{\boldsymbol{u}}+1]$, $\Pr[h_j^{\boldsymbol{u}} = h] = \Pr[h_{t_j} \cap \boldsymbol{u} = h \mid \boldsymbol{q} = \boldsymbol{u}]$. We trivially have $h_1^{\boldsymbol{u}} = \emptyset = h_{t_1} \cap \boldsymbol{u}$ always, which satisfies the base case. Now assume that the claim holds for some $j < T^{\boldsymbol{u}} + 1$. Fix some $h \in \mathcal{H}$ of length $j+1$ where all query decisions are 1. Then we can write $h = h' \frown (s, a, a^m)$ for some $h' \in \mathcal{H}$ of length j and $s \in \mathcal{S}, a \in \mathcal{A}, a^m \in \mathcal{A}$. We proceed by case analysis.

Case 1: $\Pr[h_{t_j} \cap \boldsymbol{u} = h', \boldsymbol{q} = \boldsymbol{u}] = 0$. Then

$$\begin{aligned}
\Pr[h_{t_j} \cap \boldsymbol{u} = h', \boldsymbol{q} = \boldsymbol{u}] &= \Pr[h_{t_j} \cap \boldsymbol{u} = h' \mid \boldsymbol{q} = \boldsymbol{u}] \Pr[\boldsymbol{q} = \boldsymbol{u}] && (\text{Chain rule of probability}) \\
&= \Pr[h_j^{\boldsymbol{u}} = h'] \Pr[\boldsymbol{q} = \boldsymbol{u}]. && (\text{Inductive hypothesis})
\end{aligned}$$

Since $\Pr[\boldsymbol{q} = \boldsymbol{u}] > 0$ for all $\boldsymbol{u} \in \{0,1\}^T$ and $\Pr[h_{t_j} \cap \boldsymbol{u} = h', \boldsymbol{q} = \boldsymbol{u}] = 0$, we must have $\Pr[h_j^{\boldsymbol{u}} = h'] = 0 = \Pr[h_{t_j} \cap \boldsymbol{u} = h' \mid \boldsymbol{q} = \boldsymbol{u}]$. Since $h = h' \frown (s, a, a^m)$, this implies that $\Pr[h_{j+1}^{\boldsymbol{u}} = h] = 0 = \Pr[h_{t_{j+1}} \cap \boldsymbol{q} = h \mid \boldsymbol{q} = \boldsymbol{u}]$, which satisfies the induction.

Case 2: $\Pr[h_{t_j} \cap \boldsymbol{u} = h', \boldsymbol{q} = \boldsymbol{u}] > 0$. Using definitions and the chain rule of probability,

$$\begin{aligned}
&\Pr[h_{j+1}^{\boldsymbol{u}} = h] \\
&= \Pr\big[h_j^{\boldsymbol{u}} \frown (s_j^{\boldsymbol{u}}, a_j^{\boldsymbol{u}}, a_j^{m,\boldsymbol{u}}) = h\big] \\
&= \Pr[h_j^{\boldsymbol{u}} = h', s_j^{\boldsymbol{u}} = s, a_j^{\boldsymbol{u}} = a, a_j^{m,\boldsymbol{u}} = a^m]
\end{aligned}$$

11. Technically E_1 depends on $\boldsymbol{u}$, and $\boldsymbol{u}$ is not determined by h_{t_j}. However, $\boldsymbol{u}$ is a constant in our setup.

$$= \Pr[a_j^{\boldsymbol{u}} = a \mid h_j^{\boldsymbol{u}} = h', s_j^{\boldsymbol{u}} = s, a_j^{m,\boldsymbol{u}} = a^m] \ \Pr[s_j^{\boldsymbol{u}} = s, a_j^{m,\boldsymbol{u}} = a^m \mid h_j^{\boldsymbol{u}} = h'] \ \Pr[h_j^{\boldsymbol{u}} = h'].$$

Since $\Pr[h_{t_j} \cap \boldsymbol{u} = h', \boldsymbol{q} = \boldsymbol{u}] > 0$, the second term $\Pr[s_j^{\boldsymbol{u}} = s, a_j^{m,\boldsymbol{u}} = a^m \mid h_j^{\boldsymbol{u}} = h']$ is exactly $V^{\boldsymbol{u}}(h')(s, a^m)$. The first term $\Pr[a_j^{\boldsymbol{u}} = a \mid h_j^{\boldsymbol{u}} = h', s_j^{\boldsymbol{u}} = s, a_j^{m,\boldsymbol{u}} = a^m]$ does not quite match the definition of $\Gamma_{\mathrm{FF}}^{\mathcal{A}}(h', s)(a) = \Pr[a_j^{\boldsymbol{u}} = a \mid h_j^{\boldsymbol{u}} = h', s_j^{\boldsymbol{u}} = s]$. However, since Γ_{FF} is query-agnostic (recall from Section 4.1 that we only deal with query-agnostic algorithms in this model), $a_j^{\boldsymbol{u}}$ and $a_j^{m,\boldsymbol{u}}$ are conditionally independent given any history and state. In this case, we are interested in independence conditional on $h_j^{\boldsymbol{u}} = h'$ and $s_j^{\boldsymbol{u}} = s$. Hence

$$\begin{aligned}
&\Pr[h_{j+1}^{\boldsymbol{u}} = h] \\
&= \Pr[a_j^{\boldsymbol{u}} = a \mid h_j^{\boldsymbol{u}} = h', s_j^{\boldsymbol{u}} = s, a_j^{m,\boldsymbol{u}} = a^m] \ \Pr[s_j^{\boldsymbol{u}} = s, a_j^{m,\boldsymbol{u}} = a^m \mid h_j^{\boldsymbol{u}} = h'] \ \Pr[h_j^{\boldsymbol{u}} = h'] \\
&= \Pr[a_j^{\boldsymbol{u}} = a \mid h_j^{\boldsymbol{u}} = h', s_j^{\boldsymbol{u}} = s] \ \Pr[s_j^{\boldsymbol{u}} = s, a_j^{m,\boldsymbol{u}} = a^m \mid h_j^{\boldsymbol{u}} = h'] \ \Pr[h_j^{\boldsymbol{u}} = h'] \\
&= \Gamma_{\mathrm{FF}}^{\mathcal{A}}(h', s)(a) \cdot V^{\boldsymbol{u}}(h')(s, a^m) \cdot \Pr[h_j^{\boldsymbol{u}} = h']. \qquad (1)
\end{aligned}$$

By the definition of $V^{\boldsymbol{u}}$, we have

$$V^{\boldsymbol{u}}(h')(s, a^m) = \Pr[s_{t_j} = s, \, a_{t_j}^m = a^m \mid h_{t_j} \cap \boldsymbol{u} = h', \, \boldsymbol{q} = \boldsymbol{u}]. \qquad (2)$$

The definition of Γ_k implies that $a_{t_j} \sim \Gamma_k^{\mathcal{A}}(h_{t_j}, s_{t_j}) = \Gamma_{\mathrm{FF}}^{\mathcal{A}}(h_{t_j} \cap \boldsymbol{q}, s_{t_j})$. Using the same conditional independence argument as above, we get

$$\begin{aligned}
\Gamma_{\mathrm{FF}}^{\mathcal{A}}(h', s)(a) &= \Pr[a_{t_j} = a \mid h_{t_j} \cap \boldsymbol{u} = h', \, s_{t_j} = s, \, \boldsymbol{q} = \boldsymbol{u}] \\
&= \Pr[a_{t_j} = a \mid h_{t_j} \cap \boldsymbol{u} = h', \, s_{t_j} = s, \, a_{t_j}^m = a^m, \, \boldsymbol{q} = \boldsymbol{u}]. \qquad (3)
\end{aligned}$$

Furthermore, the inductive hypothesis implies that $\Pr[h_j^{\boldsymbol{u}} = h'] = \Pr[h_{t_j} \cap \boldsymbol{u} = h' \mid \boldsymbol{q} = \boldsymbol{u}]$. Combining this with Equations 1, 2, and 3 gives us

$$\begin{aligned}
&\Pr[h_{j+1}^{\boldsymbol{u}} = h] \\
&= \Pr[a_j^{\boldsymbol{u}} = a \mid h_j^{\boldsymbol{u}} = h', s_j^{\boldsymbol{u}} = s, a_j^{m,\boldsymbol{u}} = a^m] \ \Pr[s_j^{\boldsymbol{u}} = s, a_j^{m,\boldsymbol{u}} = a^m \mid h_j^{\boldsymbol{u}} = h'] \ \Pr[h_j^{\boldsymbol{u}} = h'] \\
&= \Gamma_{\mathrm{FF}}^{\mathcal{A}}(h', s)(a) \cdot V^{\boldsymbol{u}}(h')(s, a^m) \cdot \Pr[h_{t_j} = h' \cap \boldsymbol{u} \mid \boldsymbol{q} = \boldsymbol{u}] \\
&= \Pr[a_{t_j} = a \mid h_{t_j} \cap \boldsymbol{u} = h', \, s_{t_j} = s, \, a_{t_j}^m = a^m, \, \boldsymbol{q} = \boldsymbol{u}] \\
&\qquad \times \Pr[s_{t_j} = s, \, a_{t_j}^m = a^m \mid h_{t_j} \cap \boldsymbol{u} = h', \, \boldsymbol{q} = \boldsymbol{u}] \ \Pr[h_{t_j} \cap \boldsymbol{u} = h' \mid \boldsymbol{q} = \boldsymbol{u}] \\
&= \Pr[h_{t_j} \cap \boldsymbol{u} = h', \, s_{t_j} = s, \, a_{t_j} = a, \, a_{t_j}^m = a^m \mid \boldsymbol{q} = \boldsymbol{u}]. \qquad (4)
\end{aligned}$$

By the definition of $t_1, \ldots, t_K$, we have $q_{t_j} = 1$. Also, $q_t = 0$ when $t_j < t < t_{j+1}$. Thus

$$\begin{aligned}
h_{t_{j+1}} \cap \boldsymbol{q} &= (s_i, a_i, a_i^m q_i)_{i \in [t_{j+1}-1] : q_i = 1} \\
&= (s_i, a_i, a_i^m)_{i \in [t_{j+1}-1] : q_i = 1} \\
&= h_{t_j} \cap \boldsymbol{q} \frown (s_{t_j}, a_{t_j}, a_{t_j}^m). \qquad (5)
\end{aligned}$$

Therefore

$$\begin{aligned}
\Pr[h_{j+1}^{\boldsymbol{u}} = h] &= \Pr[h_{t_j} \cap \boldsymbol{u} = h', \, s_{t_j} = s, \, a_{t_j} = a, \, a_{t_j}^m = a^m \mid \boldsymbol{q} = \boldsymbol{u}] && \text{(Equation 4)} \\
&= \Pr[h_{t_j} \cap \boldsymbol{q} = h', \, s_{t_j} = s, \, a_{t_j} = a, \, a_{t_j}^m = a^m \mid \boldsymbol{q} = \boldsymbol{u}] && (\boldsymbol{q} = \boldsymbol{u})
\end{aligned}$$

$$
\begin{aligned}
&= \Pr[(h_{t_j} \cap \boldsymbol{q}) \frown (s_{t_j}, a_{t_j}, a_{t_j}^m) = h' \frown (s, a, a^m) \mid \boldsymbol{q} = \boldsymbol{u}] && \text{(Rearranging)} \\
&= \Pr\left[h_{t_{j+1}} \cap \boldsymbol{q} = h' \frown (s, a, a^m) \mid \boldsymbol{q} = \boldsymbol{u}\right] && \text{(Equation 5)} \\
&= \Pr[h_{t_{j+1}} \cap \boldsymbol{q} = h \mid \boldsymbol{q} = \boldsymbol{u}]. && \text{(Defn of } h\text{)}
\end{aligned}
$$

This completes the induction in Case 2 and thus completes the overall induction, which shows that $\Pr[h_j^{\boldsymbol{u}} = h] = \Pr[h_{t_j} \cap \boldsymbol{u} = h \mid \boldsymbol{q} = \boldsymbol{u}]$ for any $h \in \mathcal{H}$ and $j \in [T^{\boldsymbol{u}} + 1]$. Therefore $\mathcal{D}(s_j^{\boldsymbol{u}}, a_j^{m,\boldsymbol{u}}, a_j^{\boldsymbol{u}}) = \mathcal{D}(s_{t_j}, a_{t_j}^m, a_{t_j} \mid \boldsymbol{q} = \boldsymbol{u})$ and thus $\mathbb{E}[g(s_j^{\boldsymbol{u}}, a_j^{m,\boldsymbol{u}}, a_j^{\boldsymbol{u}})] = \mathbb{E}[g(s_{t_j}, a_{t_j}^m, a_{t_j}) \mid \boldsymbol{q} = \boldsymbol{u}]$ for any $j \in [T^{\boldsymbol{u}}]$. ■

Theorem 1 (First reduction) *Assume that $R : \mathbb{N} \times [0,1] \to \mathbb{R}_{\geq 0}$ is concave in its first argument and that Γ_{FF} is a full-feedback algorithm which satisfies $\mathrm{Reg}_{\mathrm{SA}}(T, \Gamma_{\mathrm{FF}}, V, \ell) \leq R(T, \sigma)$ for all $T \in \mathbb{N}, \sigma \in [0,1]$, and $V \in \mathcal{V}(\sigma)$. Then for any $T \in \mathbb{N}, k \in (0, T], \sigma \in [0,1]$, and $V \in \mathcal{V}(\sigma)$, Γ_k satisfies $Q(T, \Gamma_k, V) = k$ and*

$$\mathrm{Reg}_{\mathrm{SA}}(T, \Gamma_k, V, \ell) \leq \frac{T}{k} R(k, \sigma).$$

Proof We have $Q(T, \Gamma_k, V) = \mathbb{E}[\sum_{t=1}^T q_t] = k$ since each q_t is sampled from Bernoulli(k/T). Thus it remains to bound the regret. For any $\pi \in \Pi$ and $\boldsymbol{u} \in \{0,1\}^T$, we have

$$
\begin{aligned}
\mathbb{E}[\tilde{\Delta}_T(\Gamma_k, \pi) \,|\, \boldsymbol{q} = \boldsymbol{u}] &= \mathbb{E}\left[\sum_{t=1}^T \tilde{\ell}_t(a_t, a_t^m) - \sum_{t=1}^T \tilde{\ell}_t(\pi(s_t), a_t^m) \mid \boldsymbol{q} = \boldsymbol{u}\right] && \text{(Definition of } \tilde{\Delta}_T\text{)} \\
&= \frac{T}{k}\,\mathbb{E}\left[\sum_{t:q_t=1} \ell(a_t, a_t^m) - \sum_{t:q_t=1} \ell(\pi(s_t), a_t^m) \mid \boldsymbol{q} = \boldsymbol{u}\right] && \text{(Definition of } \tilde{\ell}_t\text{)} \\
&= \frac{T}{k}\,\mathbb{E}\left[\sum_{j=1}^K \ell(a_{t_j}, a_{t_j}^m) - \sum_{j=1}^K \ell(\pi(s_{t_j}), a_{t_j}^m) \mid \boldsymbol{q} = \boldsymbol{u}\right] && \text{(Change of index)} \\
&= \frac{T}{k}\,\mathbb{E}\left[\sum_{j=1}^{T^{\boldsymbol{u}}} \ell(a_{t_j}, a_{t_j}^m) - \sum_{j=1}^{T^{\boldsymbol{u}}} \ell(\pi(s_{t_j}), a_{t_j}^m) \mid \boldsymbol{q} = \boldsymbol{u}\right] && \left(K = \sum_{t=1}^T q_t = T^{\boldsymbol{u}}\right) \\
&= \frac{T}{k}\,\mathbb{E}\left[\sum_{j=1}^{T^{\boldsymbol{u}}} \ell(a_j^{\boldsymbol{u}}, a_j^{m,\boldsymbol{u}}) - \sum_{j=1}^{T^{\boldsymbol{u}}} \ell(\pi(s_j^{\boldsymbol{u}}), a_j^{m,\boldsymbol{u}})\right]. && \text{(Lemma 15)}
\end{aligned}
$$

Taking a supremum and applying the definition of $\mathrm{Reg}_{\mathrm{SA}}$,

$$
\begin{aligned}
\mathbb{E}[\tilde{\Delta}_T(\Gamma_k, \pi) \mid \boldsymbol{q} = \boldsymbol{u}] &\leq \sup_{\pi' \in \Pi} \mathbb{E}[\tilde{\Delta}_T(\Gamma_k, \pi') \mid \boldsymbol{q} = \boldsymbol{u}] \\
&\leq \frac{T}{k} \sup_{\pi' \in \Pi} \mathbb{E}\left[\sum_{j=1}^{T^{\boldsymbol{u}}} \ell(a_j^{\boldsymbol{u}}, a_j^{m,\boldsymbol{u}}) - \sum_{j=1}^{T^{\boldsymbol{u}}} \ell(\pi'(s_j^{\boldsymbol{u}}), a_j^{m,\boldsymbol{u}})\right] \\
&= \frac{T}{k} \mathrm{Reg}_{\mathrm{SA}}(T^{\boldsymbol{u}}, \Gamma_{\mathrm{FF}}, V^{\boldsymbol{u}}, \ell).
\end{aligned}
$$

Lemma 14 implies that $V^{\boldsymbol{u}}$ is σ-smooth, so the regret bound of Γ_{FF} gives us $\mathbb{E}[\tilde{\Delta}_T(\Gamma_k, \pi) \mid \boldsymbol{q} = \boldsymbol{u}] \leq \mathrm{Reg}_{\mathrm{SA}}(T^{\boldsymbol{u}}, \Gamma_{\mathrm{FF}}, V^{\boldsymbol{u}}, \ell) \leq R(T^{\boldsymbol{u}}, \sigma)$. By the law of total expectation,

$$\mathbb{E}[\tilde{\Delta}_T(\Gamma_k, \pi)] = \mathop{\mathbb{E}}_{\boldsymbol{u} \sim \mathcal{D}(\boldsymbol{q})} \left[\mathbb{E}\left[\tilde{\Delta}_T(\Gamma_k, \pi) \mid \boldsymbol{q} = \boldsymbol{u} \right] \right] \leq \mathop{\mathbb{E}}_{\boldsymbol{u} \sim \mathcal{D}(\boldsymbol{q})} \left[\frac{T}{k} R(T^{\boldsymbol{u}}, \sigma) \right] = \mathbb{E}\left[\frac{T}{k} R(T^{\boldsymbol{q}}, \sigma) \right].$$

By definitions, $T^{\boldsymbol{q}} = \sum_{t=1}^{T} q_t = K$. Therefore $\mathbb{E}[\tilde{\Delta}_T(\Gamma_k, \pi)] \leq \frac{T}{k} \mathbb{E}[R(K, \sigma)]$. Now let $R_\sigma(n) = R(n, \sigma)$. Then R_σ is concave, so Jensen's inequality implies that

$$\mathbb{E}[R(K, \sigma)] = \mathbb{E}[R_\sigma(K)] \leq R_\sigma(\mathbb{E}[K]) = R(\mathbb{E}[K], \sigma).$$

Thus $\mathbb{E}[\tilde{\Delta}_T(\Gamma_k, \pi)] \leq \frac{T}{k} R(\mathbb{E}[K], \sigma) = \frac{T}{k} R(k, \sigma)$. Since this holds for all $\pi \in \Pi$, we have $\sup_{\pi \in \Pi} \mathbb{E}[\tilde{\Delta}_T(\Gamma_k, \pi)] \leq \frac{T}{k} R(k, \sigma)$. Combining this with Lemma 12,

$$\mathrm{Reg}_{\mathrm{SA}}(T, \Gamma_k, V, \ell) = \sup_{\pi \in \Pi} \mathbb{E}[\Delta_T(\Gamma_k, \pi)] = \sup_{\pi \in \Pi} \mathbb{E}[\tilde{\Delta}_T(\Gamma_k, \pi)] \leq \frac{T}{k} R(k, \sigma)$$

as required. ■

Appendix B. Second Reduction Proofs

This section presents the proofs for our second reduction (Theorem 5).

B.1 Proof Notation

We use the following notation throughout the proof:

1. Fix some $T \in \mathbb{N}, \sigma \in [0, 1]$, and $V \in \mathcal{V}(\sigma)$.
2. For each $t \in [T]$, let X_t refer to the value of the variable X in Algorithm 2 at the start of time step t.
3. We use the variables $\tilde{q}_t$ and $\tilde{a}_t$ as defined in Algorithm 2.
4. Let $M_T = \{t \in [T] : \tilde{a}_t \neq \pi^m(s_t)\}$ be the set of time steps where the action from Γ doesn't match the mentor's.
5. For each $S \subseteq \mathcal{S}$, let $\mathrm{vol}(S)$ denote the n-dimensional Lebesgue measure of S.
6. With slight abuse of notation, we will use inequalities of the form $f(T) \leq g(T) + O(h(T))$ to mean that there exists a constant C such that $f(T) \leq g(T) + Ch(T)$.

There are two parts of the proof: the regret bounds ($\mathrm{Reg}^{\times}_{\mathrm{AC}}$ and $\mathrm{Reg}^{+}_{\mathrm{AC}}$) and the query bound. To bound the regret, we bound the number of time steps where the agent chooses the wrong action (Lemma 17) and bound the worst-case reward on the time steps when the agent does choose the wrong action (Lemma 18). Lemma 16 is an intermediate step which shows the existence of an appropriate σ-smooth adversary so that we can apply the assumed regret bound on Γ. Lemma 19 puts all of the above together to obtain the regret bounds. Lemma 23 proves the query bound; the main idea is to show that after sufficiently many queries, we will have fully covered the state space.

B.2 Proof

Lemma 16 *Under the conditions of Theorem 5, there exists $V' \in \mathcal{V}(\sigma)$ such that*

$$Q(T, \Gamma, V') = \mathbb{E}\left[\sum_{t=1}^{T} \tilde{q}_t\right],$$
$$\mathrm{Reg}_{\mathrm{SA}}(T, \Gamma, V', \ell_{\mathbf{1}}) = \sup_{\pi \in \Pi} \mathbb{E}\left[\sum_{t=1}^{T} \ell_{\mathbf{1}}(\tilde{a}_t, \pi^m(s_t)) - \sum_{t=1}^{T} \ell_{\mathbf{1}}(\pi(s_t), \pi^m(s_t))\right].$$

Proof The proof proceeds in three parts. Recall that Γ is defined in Theorem 5.

Part 1. We first claim that for any $t \in [T]$, the variables $(s_i, \tilde{q}_i, \tilde{a}_i)_{i\in[t]}$ and π^m uniquely determine the values of $(q_i, a_i, X_i)_{i\in[t]}$. First note that $X_1 = \emptyset$ and for $i \in [t-1]$, X_{i+1} is given by

$$X_{i+1} = \begin{cases} X_i \cup \{(s_i, \pi^m(s_i))\} & \text{if } X_i = \emptyset \text{ or } \min_{(s,a)\in X_i : a = \tilde{a}_i} ||s_i - s|| > \varepsilon \\ X_i & \text{otherwise.} \end{cases}$$

Thus we can determine the value of X_{i+1} using only $X_i, s_i, \tilde{a}_i$, and π^m. Since we know $(s_i, \tilde{q}_i, \tilde{a}_i)_{i\in[t]}, \pi^m$, and X_1, we can determine X_2, and use X_2 to determine X_3, and so on. Once we have determined X_i, we can determine q_i and a_i as follows:

$$(q_i, a_i) = \begin{cases} (1, \pi^m(s_i)) & \text{if } X_i = \emptyset \text{ or } \min_{(s,a)\in X_i : a = \tilde{a}_i} ||s_i - s|| > \varepsilon \\ (\tilde{q}_i, \tilde{a}_i) & \text{otherwise.} \end{cases}$$

Thus $(s_i, \tilde{q}_i, \tilde{a}_i)_{i\in[t]}$ and π^m uniquely determine the values of $(q_i, a_i, X_i)_{i\in[t]}$. Then there exists a deterministic function $f : \mathcal{H} \to \mathcal{H}$ such that $f\big((s_i, \tilde{a}_i, \pi^m(s_i)\tilde{q}_i)_{i\in[t]}\big) = (s_i, a_i, \pi^m(s_i)q_i)_{i\in[t]} = h_{t+1}$ for any $t \in [T]$. (Recall that h_{t+1} is the history at the start of time $t+1$ and thus includes s_t, a_t, q_t but not $s_{t+1}, a_{t+1}, q_{t+1}$.) For a history h, define $V'(h) = V(f(h))$. Note that the algorithm need not know f: since we are defining an adversary, all we need is for this function to exist. Since V is σ-smooth by assumption, V' is also σ-smooth.

Part 2. We need to show that V' induces the desired distribution of states and actions. For each $t \in [T]$, let s'_t, q'_t, a'_t be the random variables corresponding to the state, query decision, and action at time t as induced by Γ and V'. Also, for each $t \in [T]$ let $\tilde{h}_t = \big(s_i, \tilde{a}_i, \pi^m(s_i)\tilde{q}_i\big)_{i\in[t-1]}$ as defined in Algorithm 2 and let $h'_t = \big(s'_i, a'_i, \pi^m(s'_i)q'_i\big)_{i\in[t-1]}$. We will show by induction that $\mathcal{D}(h'_t) = \mathcal{D}(\tilde{h}_t)$ for each $t \in [T]$. We trivially have $h'_1 = \emptyset = \tilde{h}_1$, so assume the claim holds for some $t \in [T]$. Let h^+ be any history of nonzero length and write $h^+ = h \frown (s, a, \pi^m(s)q)$ for some $h \in \mathcal{H}, s \in \mathcal{S}, a \in \mathcal{A}, q \in \{0, 1\}$. Using definitions and the chain rule of probability,

$$\begin{aligned} &\Pr[h'_{t+1} = h^+] \\ =\ &\Pr[h'_{t+1} = h \frown (s, a, \pi^m(s)q)] \\ =\ &\Pr[h'_t = h, s'_t = s, a'_t = a, q'_t = q] \\ =\ &\Pr[h'_t = h] \Pr[s'_t = s \mid h'_t = h] \Pr[q'_t = q \mid s'_t = s, h'_t = h] \Pr[a'_t = a \mid q'_t = q, s'_t = s, h'_t = h]. \end{aligned}$$

By the inductive hypothesis, $\Pr[h'_t = h] = \Pr[\tilde{h}_t = h]$. Also, by definitions we have $\Pr[s'_t = s \mid h'_t = h] = V'(h)(s, \pi^m(s))$, $\Pr[q'_t = q \mid s'_t = s, h'_t = h] = \Gamma^Q(h, s)$, and

$\Pr[a'_t = a \mid q'_t = q, s'_t = s, h'_t = h] = \Gamma^{\mathcal{A}}(h, s, \pi^m(s)q)$. Since Γ is query-agnostic by assumption, we have $\Gamma^{\mathcal{A}}(h, s, \pi^m(s)q) = \Gamma^{\mathcal{A}}(h, s, 0)$. Therefore

$$\begin{aligned}\Pr[h'_{t+1} = h^+] &= \Pr[\tilde{h}_t = h] \cdot V'(h)(s, \pi^m(s)) \cdot \Gamma^Q(h, s)(q) \cdot \Gamma^{\mathcal{A}}(h, s, \pi^m(s)q)(a) \\ &= \Pr[\tilde{h}_t = h] \cdot V(f(h))(s, \pi^m(s)) \cdot \Gamma^Q(h, s)(q) \cdot \Gamma^{\mathcal{A}}(h, s, 0)(a).\end{aligned}$$

Since $s_t \sim V(h_t), q_t \sim \Gamma^Q(h_t, s_t)$, and $\tilde{a}_t \sim \Gamma^{\mathcal{A}}(h_t, s_t, 0)$, we have

$$\begin{aligned}\Pr[h'_{t+1} = h^+] &= \Pr[\tilde{h}_t = h, s_t = s, \tilde{q}_t = q, \tilde{a}_t = a] \\ &= \Pr[\tilde{h}_t = h] \cdot \Pr[s_t = s \mid \tilde{h}_t = h] \cdot \Pr[\tilde{q}_t = q \mid s_t = s, \tilde{h}_t = h] \\ &\quad \cdot \Pr[\tilde{a}_t = a \mid \tilde{q}_t = q, s_t = s, \tilde{h}_t = h] \\ &= \Pr[\tilde{h}_t = h, s_t = s, \tilde{a}_t = a, \tilde{q}_t = q] \\ &= \Pr[\tilde{h}_{t+1} = h \frown (s, a, \pi^m(s)q)] \\ &= \Pr[\tilde{h}_{t+1} = h^+].\end{aligned}$$

Hence $\mathcal{D}(h'_{t+1}) = \mathcal{D}(\tilde{h}_{t+1})$, which completes the induction. In particular, this shows that $\mathcal{D}(q'_t) = \mathcal{D}(\tilde{q}_t)$ and $\mathcal{D}(s'_t, a'_t) = \mathcal{D}(s_t, \tilde{a}_t)$ for all $t \in [T]$.

Part 3. We have

$$\begin{aligned}Q(T, \Gamma, V') &= \mathbb{E}\left[\sum_{t=1}^{T} q'_t\right] && \text{(Definition of } Q(T, \Gamma, V') \text{ and } q'_1, \ldots, q'_T) \\ &= \mathbb{E}\left[\sum_{t=1}^{T} \tilde{q}_t\right]. && (\mathcal{D}(q'_t) = \mathcal{D}(\tilde{q}_t) \text{ for all } t \in [T])\end{aligned}$$

Similarly, using the definition of $\mathrm{Reg}_{\mathrm{SA}}(T, \Gamma, V', \ell_{\mathbf{1}})$ and $\mathcal{D}(s'_t, a'_t) = \mathcal{D}(s_t, \tilde{a}_t)$ for all $t \in [T]$,

$$\begin{aligned}\mathrm{Reg}_{\mathrm{SA}}(T, \Gamma, V', \ell_{\mathbf{1}}) &= \sup_{\pi \in \Pi} \mathbb{E}\left[\sum_{t=1}^{T} \ell_{\mathbf{1}}(a'_t, \pi^m(s'_t)) - \sum_{t=1}^{T} \ell_{\mathbf{1}}(\pi(s'_t), \pi^m(s'_t))\right] \\ &= \sup_{\pi \in \Pi} \mathbb{E}\left[\sum_{t=1}^{T} \ell_{\mathbf{1}}(\tilde{a}_t, \pi^m(s_t)) - \sum_{t=1}^{T} \ell_{\mathbf{1}}(\pi(s_t), \pi^m(s_t))\right],\end{aligned}$$

as required. ■

Lemma 17 *Under the conditions of Theorem 5, Algorithm 2 satisfies* $\mathbb{E}[|M_T|] \le R(k, T, \sigma)$.

Proof We have

$$\begin{aligned}\mathbb{E}[|M_T|] &= \mathbb{E}\left[\sum_{t=1}^{T} \mathbf{1}(\tilde{a}_t \neq \pi^m(s_t))\right] && \text{(Definition of } M_T) \\ &= \mathbb{E}\left[\sum_{t=1}^{T} \ell_{\mathbf{1}}(\tilde{a}_t, \pi^m(s_t))\right] && \text{(Definition of } \ell_{\mathbf{1}})\end{aligned}$$

$$
\begin{aligned}
&= \mathbb{E}\left[\sum_{t=1}^{T} \ell_{\mathbf{1}}(\tilde{a}_t, \pi^m(s_t)) - \sum_{t=1}^{T} \ell_{\mathbf{1}}(\pi^m(s_t), \pi^m(s_t))\right] && (\ell_{\mathbf{1}}(\pi^m(s_t), \pi^m(s_t)) = 0)\\
&\leq \mathrm{Reg}_{\mathrm{SA}}(T, \Gamma, V', \ell_{\mathbf{1}}) && (\text{Lemma 16 and } \pi^m \in \Pi)\\
&\leq R(k, T, \sigma) && (\text{Assumption of Theorem 5}),
\end{aligned}
$$

as required. ■

Lemma 18 *For all $t \in [T]$, $\mu_t(s_t, a_t) \geq \mu_t^m(s_t) - L\varepsilon$.*

Proof Consider an arbitrary $t \in [T]$ and let $\Delta_t = \min_{(s,a) \in X_t : a = \tilde{a}_t} ||s_t - s||$. If $\Delta_t > \varepsilon$, then we query the mentor and take action $a_t = \pi^m(s_t)$, so $\mu_t(s_t, a_t) = \mu_t(s_t, \pi^m(s_t)) = \mu_t^m(s_t)$. Now consider $\Delta_t \leq \varepsilon$. Let $(s', a') = \arg\min_{(s,a) \in X_t : \tilde{a}_t = a} ||s_t - s||$; then $||s_t - s'|| \leq \varepsilon$.

We have $a' = \pi^m(s')$ by construction of X_t and $a' = \tilde{a}_t$ by construction of a' via the arg min. Combining these with the local generalization assumption, we get

$$\mu_t(s_t, a_t) = \mu_t(s_t, \tilde{a}_t) = \mu_t(s_t, \pi^m(s')) \geq \mu_t^m(s_t) - L||s_t - s'|| \geq \mu_t^m(s_t) - L\varepsilon,$$

as required. ■

Lemma 19 *Under the conditions of Theorem 5, Algorithm 2 satisfies*

$$
\begin{aligned}
\mathrm{Reg}^{\times}_{\mathrm{AC}}(T, \Gamma_{\mathrm{AC}}, V, \boldsymbol{\mu}) &\leq \frac{L\varepsilon R(k, T, \sigma)}{\mu_0} \quad \text{if } \mu_0 > 0,\\
\mathrm{Reg}^{+}_{\mathrm{AC}}(T, \Gamma_{\mathrm{AC}}, V, \boldsymbol{\mu}) &\leq L\varepsilon R(k, T, \sigma).
\end{aligned}
$$

Proof First consider any $t \notin M_T$. Then $\tilde{a}_t = \pi^m(s_t)$, and since $a_t \in \{\tilde{a}_t, \pi^m(s_t)\}$ for all t, we have $a_t = \pi^m(s_t)$. Thus $\mu_t(s_t, a_t) = \mu_t^m(s_t)$ for all $t \notin M_T$. For $t \in M_T$, Lemma 18 implies that $\mu_t^m(s_t) - \mu_t(s_t, a_t) \leq L\varepsilon$, so

$$
\begin{aligned}
\mathrm{Reg}^{+}_{\mathrm{AC}}(T, \Gamma_{\mathrm{AC}}, V, \boldsymbol{\mu}) &= \mathbb{E}\left[\sum_{t=1}^{T} \mu_t(s_t, \pi^m(s_t)) - \sum_{t=1}^{T} \mu_t(s_t, a_t)\right]\\
&= \mathbb{E}\left[\sum_{t \in M_T} (\mu_t^m(s_t) - \mu_t(s_t, a_t))\right]\\
&\leq \mathbb{E}\left[\sum_{t \in M_T} L\varepsilon\right]\\
&= \mathbb{E}[|M_T|]L\varepsilon. \qquad (6)
\end{aligned}
$$

The analysis for $\mathrm{Reg}^{\times}_{\mathrm{AC}}$ is similar but requires a bit more work to deal with the product. First, we have

$$\mathrm{Reg}^{\times}_{\mathrm{AC}}(T, \Gamma_{\mathrm{AC}}, V, \boldsymbol{\mu}) = \mathbb{E}\left[\log \prod_{t=1}^{T} \mu_t(s_t, \pi^m(s_t)) - \log \prod_{t=1}^{T} \mu_t(s_t, a_t)\right]$$

$$= \mathbb{E}\left[\sum_{t=1}^{T} \log \frac{\mu_t(s_t, \pi^m(s_t))}{\mu_t(s_t, a_t)}\right]$$
$$= \mathbb{E}\left[\sum_{t \in M_T} \log \frac{\mu_t(s_t, \pi^m(s_t))}{\mu_t(s_t, a_t)}\right].$$

Note that when $\mu_0 > 0$, $\mu_t(s_t, a_t) \geq \mu_0 > 0$ so the ratio above is well-defined. By Lemma 18,

$$\mathbb{E}\left[\sum_{t \in M_T} \log \frac{\mu_t(s_t, \pi^m(s_t))}{\mu_t(s_t, a_t)}\right] \leq \mathbb{E}\left[\sum_{t \in M_T} \log \frac{\mu_t(s_t, a_t) + L\varepsilon}{\mu_t(s_t, a_t)}\right]$$
$$= \mathbb{E}\left[\sum_{t \in M_T} \log \left(1 + \frac{L\varepsilon}{\mu_t(s_t, a_t)}\right)\right]$$
$$\leq \mathbb{E}\left[\sum_{t \in M_T} \log \left(1 + \frac{L\varepsilon}{\mu_0}\right)\right].$$

Since $\log(1 + x) \leq x$ for all $x \geq 0$, we get

$$\mathrm{Reg}^{\times}_{\mathrm{AC}}(T, \Gamma_{\mathrm{AC}}, V, \boldsymbol{\mu}) \leq \mathbb{E}\left[\sum_{t \in M_T} \log \left(1 + \frac{L\varepsilon}{\mu_0}\right)\right]$$
$$\leq \mathbb{E}\left[\sum_{t \in M_T} \frac{L\varepsilon}{\mu_0}\right]$$
$$= \frac{L\varepsilon\, \mathbb{E}[|M_T|]}{\mu_0}. \tag{7}$$

Applying Lemma 17 to Equations 6 and 7 completes the proof. ■

Definition 20 (Packing numbers) *Let $(K, ||\cdot||)$ be a normed vector space and let $\delta > 0$. Then $S \subseteq K$ is a δ-packing of K if for all $a, b \in S$ with $a \neq b$, we have $||a - b|| > \delta$. 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 21 (Theorem 14.2 in Wu, 2020) *If $K \subset \mathbb{R}^n$ is convex, bounded, and contains a ball with radius $\delta > 0$, then*

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

where B is a unit ball.

Lemma 22 (Jung's Theorem, Jung, 1901; Theorem 3.3 in Gruber, 2007) *If $S \subset \mathbb{R}^n$ is bounded, then there exists a closed ball with radius at most $\operatorname{diam}(S)\sqrt{\frac{n}{2(n+1)}}$ containing S.*

Jung's Theorem also applies to multisets, since duplicates do not affect the diameter of a set.

Lemma 23 *Under the conditions of Theorem 5, Algorithm 2 satisfies*

$$Q(T, \Gamma_{\mathrm{AC}}, V) \in O\left(k + R(k, T, \sigma) + \frac{|\mathcal{A}|\,\mathbb{E}[\operatorname{diam}(\boldsymbol{s})^n]}{\varepsilon^n}\right).$$

Proof Let $\Delta_t = \min_{(s,a) \in X_t : a = \tilde{a}_t} ||s_t - s||$. For any $t \in [T]$ with $q_t = 1$, either $\tilde{q}_t = 1$ or $\Delta_t > \varepsilon$ (or both). Thus $\sum_{t=1}^T q_t \leq \sum_{t=1}^T \tilde{q}_t + \sum_{t=1}^T \mathbf{1}(\Delta_t > \varepsilon)$. We have $\mathbb{E}[\sum_{t=1}^T \tilde{q}_t] = Q(T, \Gamma, V')$ by Lemma 16 and $Q(T, \Gamma, V') \leq k$ by assumption, so $\mathbb{E}[\sum_{t=1}^T \tilde{q}_t] \leq k$. Thus it remains to bound $\mathbb{E}[\sum_{t=1}^T \mathbf{1}(\Delta_t > \varepsilon)]$. Let $\hat{Q} = \{t \in [T] : q_t = 1 \text{ and } \Delta_t > \varepsilon\}$. We further subdivide $\hat{Q}$ into $\hat{Q}_1 = \{t \in \hat{Q} : \tilde{a}_t \neq \pi^m(s_t)\}$ and $\hat{Q}_2 = \{t \in \hat{Q} : \tilde{a}_t = \pi^m(s_t)\}$. Since $\hat{Q}_1 \subseteq M_T$, Lemma 17 implies that $\mathbb{E}[|\hat{Q}_1|] \leq R(k, T, \sigma)$.

Next, fix an $a \in \mathcal{A}$ and let $S_a = \{s \in \boldsymbol{s} : \pi^m(s) = a\}$ be the multiset of observed inputs whose mentor action is a. Also let $\hat{S}_2 = \{s_t : t \in \hat{Q}_2\}$ be the multiset of inputs associated with time steps in $\hat{Q}_2$. Note that $|\hat{S}_2| = |\hat{Q}_2|$, since $\hat{S}_2$ is a multiset. We claim that $\hat{S}_2 \cap S_a$ is an ε-packing of S_a. Suppose instead that there exists $s, s' \in \hat{S}_2 \cap S_a$, with $||s - s'|| \leq \varepsilon$. WLOG assume s was queried after s' and let t be the time step on which s was queried. Since $s' \in \hat{S}_2$, this implies $(s', \pi^m(s')) \in X_t$. Also, since $s, s' \in \hat{S}_2$ we have $\tilde{a}_t = \pi^m(s_t) = a = \pi^m(s')$. Therefore

$$\Delta_t = \min_{(x,y) \in X_t : y = \tilde{a}_t} ||s_t - x|| \leq ||s_t - s'|| \leq \varepsilon$$

which contradicts $t \in \hat{Q}$. Thus $\hat{S}_2 \cap S_a$ is an ε-packing of S_a.

By Lemma 22, there exists a ball B_1 of radius $h := \operatorname{diam}(\boldsymbol{s})\sqrt{\frac{n}{2(n+1)}}$ which contains $\boldsymbol{s}$. Let B_2 be the ball with the same center as B_1 but with radius $\max(h, \varepsilon)$. Since $S_a \subset \boldsymbol{s} \subset B_1 \subset B_2$ and $\hat{S}_2 \cap S_a$ is an ε-packing of S_a, $\hat{S}_2 \cap S_a$ is also an ε-packing of B_2. Also, B_2 must contain a ball of radius ε, so Lemma 21 implies that

$$\begin{aligned}
|\hat{S}_2 \cap S_a| \leq \mathcal{M}(B_2, \varepsilon) &\leq \frac{3^n \operatorname{vol}(B_2)}{\varepsilon^n \operatorname{vol}(B)} = \big(\max(h, \varepsilon)\big)^n \frac{3^n \operatorname{vol}(B)}{\varepsilon^n \operatorname{vol}(B)} \\
&= \max\left(\operatorname{diam}(\boldsymbol{s})^n \left(\frac{n}{2(n+1)}\right)^{n/2}, \varepsilon^n\right) \frac{3^n}{\varepsilon^n} \;\in\; O\left(\frac{\operatorname{diam}(\boldsymbol{s})^n}{\varepsilon^n} + 1\right).
\end{aligned}$$

(The $+1$ is necessary for now since $\operatorname{diam}(\boldsymbol{s})$ could theoretically be zero.) Since $\hat{S}_2 \subseteq \{s_1, \ldots, s_T\} \subseteq \cup_{a \in \mathcal{A}} S_a$, we have $|\hat{S}_2| \leq \sum_{a \in \mathcal{A}} |\hat{S}_2 \cap S_a|$ by the union bound. Therefore

$$\begin{aligned}
Q(T, \Gamma_{\mathrm{AC}}, V) &\leq \mathbb{E}\left[\sum_{t=1}^T \tilde{q}_t\right] + \mathbb{E}\left[\sum_{t=1}^T \mathbf{1}(\Delta_t > \varepsilon)\right] \\
&\leq k + \mathbb{E}[|\hat{Q}|]
\end{aligned}$$

$$
\begin{aligned}
&= k + \mathbb{E}[|\hat{Q}_1|] + \mathbb{E}[|\hat{Q}_2|] \\
&\leq k + R(k,T,\sigma) + \mathbb{E}[|\hat{S}_2|] \\
&\leq k + R(k,T,\sigma) + \mathbb{E}\left[\sum_{a\in\mathcal{A}} |\hat{S}_2 \cap S_a|\right] \\
&\leq k + R(k,T,\sigma) + \sum_{a\in\mathcal{A}} O\left(\frac{\mathbb{E}[\mathrm{diam}(\boldsymbol{s})^n]}{\varepsilon^n} + 1\right) \\
&\in O\left(k + R(k,T,\sigma) + |\mathcal{A}|\frac{\mathbb{E}[\mathrm{diam}(\boldsymbol{s})^n]}{\varepsilon^n}\right),
\end{aligned}
$$

as required. ■

Theorem 5 follows from Lemmas 19 and 23:

Theorem 5 (Second reduction) *Let $k \in \mathbb{R}_{\geq 0}$ and $R : \mathbb{R}_{\geq 0} \times \mathbb{N} \times [0,1] \to \mathbb{R}_{\geq 0}$. Let Γ be a query-agnostic algorithm satisfying $Q(T,\Gamma,V) \leq k$ and $\mathrm{Reg}_{\mathrm{SA}}(T,\Gamma,V,\ell_{\mathbf{1}}) \leq R(k,T,\sigma)$ for all $T \geq k, \sigma \in [0,1]$, and $V \in \mathcal{V}(\sigma)$. Let Γ_{AC} denote Algorithm 2 with inputs Γ, $T \geq k$, and $\varepsilon > 0$. Then for any $\sigma \in [0,1], V \in \mathcal{V}(\sigma)$, and $(\boldsymbol{\mu}, \pi^m) \in \mathcal{U}(L)$,*

$$
\begin{aligned}
\mathrm{Reg}^{\times}_{\mathrm{AC}}(T,\Gamma_{\mathrm{AC}},V,\boldsymbol{\mu}) &\leq \frac{L\varepsilon R(k,T,\sigma)}{\mu_0} \quad \text{if } \mu_0 > 0, \\
\mathrm{Reg}^{+}_{\mathrm{AC}}(T,\Gamma_{\mathrm{AC}},V,\boldsymbol{\mu}) &\leq L\varepsilon R(k,T,\sigma), \\
Q(T,\Gamma_{\mathrm{AC}},V) &\in O\left(k + R(k,T,\sigma) + \frac{|\mathcal{A}|\,\mathbb{E}[\mathrm{diam}(\boldsymbol{s})^n]}{\varepsilon^n}\right).
\end{aligned}
$$

To obtain Theorem 6, we apply Theorem 5 to Corollary 4:

Theorem 6 (Final avoiding catastrophe bounds) *Assume that either (1) Π has VC dimension d and $\sigma \in (0,1]$ or (2) Π has Littlestone dimension d and $\sigma = 0$. Let $k = T^{\frac{2n+1}{2n+2}}$ and let Γ_k be the corresponding algorithm from Corollary 4. If Γ_{AC} denotes Algorithm 2 with inputs Γ_k, T, and $\varepsilon = T^{\frac{-1}{n+1}}$, then*

$$
\begin{aligned}
\mathrm{Reg}^{\times}_{\mathrm{AC}}(T,\Gamma_{\mathrm{AC}},V,\boldsymbol{\mu}) &\in O\left(\frac{L}{\mu_0} T^{\frac{-1}{2n+2}}(d\log T + \sigma^+)\right) \quad \text{if } \mu_0 > 0, \\
\mathrm{Reg}^{+}_{\mathrm{AC}}(T,\Gamma_{\mathrm{AC}},V,\boldsymbol{\mu}) &\in O\left(LT^{\frac{-1}{2n+2}}(d\log T + \sigma^+)\right), \\
Q(T,\Gamma_{\mathrm{AC}},V) &\in O\left(T^{\frac{2n+1}{2n+2}}\left(d + \sigma^+ + \mathbb{E}[\mathrm{diam}(\boldsymbol{s})^n]\right)\right),
\end{aligned}
$$

for any $(\boldsymbol{\mu}, \pi^m) \in \mathcal{U}(L)$ and $V \in \mathcal{V}(\sigma)$.

Proof By Corollary 4, Γ_k satisfies $\mathrm{Reg}_{\mathrm{SA}}(T,\Gamma_k,V,\ell_{\mathbf{1}}) \in O(\frac{T}{k}(d\log k + \sigma^+))$ for any $\sigma \in [0,1]$ and $V \in \mathcal{V}(\sigma)$. Then the conditions of Theorem 5 hold for some $R(k,T,\sigma) \in O(\frac{T}{k}(d\log k + \sigma^+)) = O(T^{\frac{1}{2n+2}}(d\log T + \sigma^+))$. Thus for any $V \in \mathcal{V}(\sigma)$,

$$
\mathrm{Reg}^{+}_{\mathrm{AC}}(T,\Gamma_{\mathrm{AC}},V,\boldsymbol{\mu}) \in O\left(L\varepsilon T^{\frac{1}{2n+2}}(d\log T + \sigma^+)\right)
$$

$$= O\left(LT^{\frac{-1}{n+1}+\frac{1}{2n+2}}\left(d\log T+\sigma^+\right)\right)$$
$$= O\left(LT^{\frac{-1}{2n+2}}\left(d\log T+\sigma^+\right)\right).$$

The bound for $\mathrm{Reg}^{\times}_{\mathrm{AC}}$ follows from the same arithmetic with an added factor of $1/\mu_0$. Finally, Theorem 5 also gives us

$$Q(T,\Gamma_{\mathrm{AC}},V)\in O\left(T^{\frac{2n+1}{2n+2}}+T^{\frac{1}{2n+2}}\left(d\log T+\sigma^+\right)+\frac{\mathbb{E}[\mathrm{diam}(\boldsymbol{s})^n]}{T^{\frac{-n}{n+1}}}\right)$$
$$\subseteq O\left(T^{\frac{2n+1}{2n+2}}(d+\sigma^+\mathbb{E}[\mathrm{diam}(\boldsymbol{s})^n])\right),$$

as desired. ■

We can also apply the reduction to Lemma 17 directly to get the following result, which will be useful later:

Lemma 24 *Under the conditions of Theorem 6,* $\mathrm{Reg}_{\mathrm{SA}}(T,\Gamma_{\mathrm{AC}},V,\ell_{\mathbf{1}})\in O(T^{\frac{1}{2n+2}}(d\log T+\sigma^+))$ *for any* $\sigma\in[0,1]$ *and* $V\in\mathcal{V}(\sigma)$.

Proof Picking up from the proof of Theorem 6, Lemma 17 gives us $\mathbb{E}[|M_T|]\le R(k,T,\sigma)\in O(T^{\frac{1}{2n+2}}(d\log T+\sigma^+))$. Then

$$\begin{aligned}
&\mathrm{Reg}_{\mathrm{SA}}(T,\Gamma_{\mathrm{AC}},V,\ell_{\mathbf{1}})\\
=\ &\sup_{\pi\in\Pi}\mathbb{E}\left[\sum_{t=1}^{T}\ell_{\mathbf{1}}(a_t,a_t^m)-\sum_{t=1}^{T}\ell_{\mathbf{1}}(\pi(s_t),a_t^m)\right] && \text{(Definition of } \mathrm{Reg}_{\mathrm{SA}})\\
=\ &\sup_{\pi\in\Pi}\mathbb{E}\left[\sum_{t=1}^{T}\ell_{\mathbf{1}}(a_t,\pi^m(s_t))-\sum_{t=1}^{T}\ell_{\mathbf{1}}(\pi(s_t),\pi^m(s_t))\right] && \text{(Assumption of } a_t^m=\pi^m(s_t))\\
=\ &\mathbb{E}\left[\sum_{t=1}^{T}\ell_{\mathbf{1}}(a_t,\pi^m(s_t))\right] && \text{(Assumption of } \pi^m\in\Pi)\\
\le\ &\mathbb{E}\left[\sum_{t=1}^{T}\ell_{\mathbf{1}}(\tilde{a}_t,\pi^m(s_t))\right] && \text{(Either } a_t=\tilde{a}_t \text{ or } a_t=\pi^m(s_t))\\
=\ &\mathbb{E}[|M_T|]. && \text{(Definition of } M_T)
\end{aligned}$$

Thus $\mathrm{Reg}_{\mathrm{SA}}(T,\Gamma_{\mathrm{AC}},V,\ell_{\mathbf{1}})\in O(T^{\frac{1}{2n+2}}(d\log T+\sigma^+))$ as claimed. ■

Appendix C. Third Reduction Proofs

This section presents the proofs for our third reduction (Theorem 8 and Theorem 9).

C.1 Proof Notation

1. Throughout the proof, fix some $R : \mathbb{N} \times [0,1] \to \mathbb{R}_{\geq 0}, T \in \mathbb{N}, \sigma \in [0,1], (P, \mathcal{D}_1, \pi^m) \in \mathcal{V}(\sigma)$ with $(P, \pi^m) \in \mathcal{P}(L)$ and $(r, \pi^m) \in \mathcal{R}(L)$, and some algorithm Γ.
2. Let V denote the adversary defined by $P, \mathcal{D}_1, \pi^m$.
3. For each $X \subseteq \mathcal{S}$, let $P(s,a,X)$ denote the probability which $P(s,a)$ assigns to X, i.e., $P(s,a,X) = \Pr[s_{t+1} \in X \mid s_t = s, a_t = a]$ for any $t \in [T]$.
4. Let p_t and p_t^m denote the distributions of s_t and s_t^m respectively. That is, for any $X \subseteq \mathcal{S}$, $p_t(X) = \Pr[s_t \in X]$ and $p_t^m(X) = \Pr[s_t^m \in X]$.
5. Let $\Delta_t = \sup_{X \subseteq \mathcal{S}} (p_t^m(X) - p_t(X))$.
6. With "N" standing for "next", for each $X \subseteq \mathcal{S}$ let $N_t(s,X) = \Pr[s_{t+1} \in X \mid s_t = s]$ and $N_t^m(s,X) = \Pr[s_{t+1}^m \in X \mid s_t^m = s]$.
7. Let $\alpha_t(X) = \mathbb{E}[N_t^m(s_t, X) - N_t(s_t, X)]$.

C.2 Proof

Bounding the action-based regret is a direct application of local generalization of r (Lemma 25). It remains to bound the state-based regret. Rather than trying to analyze where the agent's states are better or worse than the mentor's, we simply bound how much $\boldsymbol{s}$ and $\boldsymbol{s^m}$ differ at all. The key quantity here is Δ_t. Lemma 26 shows that a wide range of functions can be bounded using Δ_t. Lemma 27 provides an initial bound on Δ_t in terms of a sum of α_i's. Lemma 28 uses this to show that $\Delta_t \leq R(T,\sigma)$. To complete the bound on state-based regret, Lemma 29 applies Lemma 26 to the $\Delta_t \leq R(T,\sigma)$ bound.

Lemma 25 (Action-based regret) *Under the conditions of Theorem 8,* $\mathbb{E}[\sum_{t=1}^T r(s_t, \pi^m(s_t)) - \sum_{t=1}^T r(s_t, a_t)] \leq R(T,\sigma)$.

Proof Define $\mu_t(s,a) = r(s,a)$ for all $t \in [T]$. Since $(r, \pi^m) \in \mathcal{R}(L)$ by assumption, we have $(\boldsymbol{\mu}, \pi^m) \in \mathcal{U}(L)$. Then by assumption of Theorem 8, $\mathbb{E}[\sum_{t=1}^T r(s_t, \pi^m(s_t)) - \sum_{t=1}^T r(s_t, a_t)] = \mathrm{Reg}^+_{\mathrm{AC}}(T, \Gamma, V, (r, \ldots, r)) \leq R(T,\sigma)$, as claimed. ■

Lemma 26 *For any $t \in [T]$ and any function $f : \mathcal{S} \to [0,1]$, we have $\mathbb{E}[f(s_t^m) - f(s_t)] \leq \Delta_t$.*

Proof By the tail sum formula for expected value (e.g., Lemma 4.4 in Kallenberg, 1997),

$$\mathbb{E}[f(s_t^m)] = \int_{h=0}^{\infty} \Pr[f(s_t^m) > h] \,\mathrm{d}h \qquad \text{and} \qquad \mathbb{E}[f(s_t)] = \int_{h=0}^{\infty} \Pr[f(s_t) > h] \,\mathrm{d}h.$$

Since $f(s) \in [0,1]$ for all $s \in \mathcal{S}$, we can restrict the integrals to $[0,1]$. Next, for each $h \in \mathbb{R}_{\geq 0}$, define $X_h = \{s \in \mathcal{S} : f(s) > h\}$. Then $\Pr[f(s_t^m) > h] = \Pr[s_t^m \in X_h] = p_t^m(X_h)$ and $\Pr[f(s_t) > h] = \Pr[s_t \in X_h] = p_t(X_h)$, and so

$$\begin{aligned}
\mathbb{E}[f(s_t^m) - f(s_t)] &= \int_{h=0}^{1} \big(\Pr[f(s_t^m) > h] - \Pr[f(s_t) > h]\big) \,\mathrm{d}h \\
&= \int_{h=0}^{1} \big(p_t^m(X_h) - p_t(X_h)\big) \,\mathrm{d}h \leq \int_{h=0}^{1} \Delta_t \,\mathrm{d}h = \Delta_t.
\end{aligned}$$

■

Lemma 27 *For any $t \in [T]$, $\Delta_t \leq \sum_{i=1}^{t-1} \sup_{X \subseteq \mathcal{S}} \alpha_i(X)$.*

Proof We proceed by induction. We have $p_1(X) = \mathcal{D}_1(X) = p_1^m(X)$ for all $X \subseteq \mathcal{S}$, so $\Delta_1 = 0 = \sum_{i=1}^{0} \sup_{X \subseteq \mathcal{S}} \alpha_i(X)$. Thus the lemma holds for $t = 1$.

Now assume the lemma holds for some $t \in [T]$ and fix any $X \subseteq \mathcal{S}$. The next step is to analyze $\mathbb{E}[N_t(s_t, X)]$, which requires a bit of care since $\mathbb{E}[N_t(s_t, X)]$ is an expectation over s_t but s_t also appears in the definition of $N_t(s, X)$. To prevent confusion, we can rewrite $\mathbb{E}[N_t(s_t, X)]$ as $\mathbb{E}_{s \sim \mathcal{D}(s_t)}[N_t(s, X)]$. Then

$$\begin{aligned}
\mathbb{E}[N_t(s_t, X)] &= \mathop{\mathbb{E}}_{s \sim \mathcal{D}(s_t)}[N_t(s, X)] \\
&= \mathop{\mathbb{E}}_{s \sim \mathcal{D}(s_t)}[\Pr[s_{t+1} \in X \mid s_t = s]] \\
&= \mathop{\mathbb{E}}_{s \sim \mathcal{D}(s_t)}[\mathbb{E}[\mathbf{1}(s_{t+1} \in X) \mid s_t = s]] .
\end{aligned}$$

By the law of total expectation,

$$\mathop{\mathbb{E}}_{s \sim \mathcal{D}(s_t)}[\mathbb{E}[\mathbf{1}(s_{t+1} \in X) \mid s_t = s]] = \mathbb{E}[\mathbf{1}(s_{t+1} \in X)] = \Pr[s_{t+1} \in X].$$

Therefore $\mathbb{E}[N_t(s_t, X)] = p_{t+1}(X)$. The same argument can be used to show that $\mathbb{E}[N_t^m(s_t^m, X)] = p_{t+1}^m(X)$ by simply replacing s_t and p_t with s_t^m and p_t^m respectively. Thus

$$\begin{aligned}
p_{t+1}^m(X) - p_{t+1}(X) &= \mathbb{E}[N_t^m(s_t^m, X)] - \mathbb{E}[N_t(s_t, X)] \\
&= \mathbb{E}[N_t^m(s_t^m, X) - N_t^m(s_t, X)] + \mathbb{E}[N_t^m(s_t, X) - N_t(s_t, X)] \\
&= \mathbb{E}[N_t^m(s_t^m, X) - N_t^m(s_t, X)] + \alpha_t(X).
\end{aligned}$$

Next, define $f : \mathcal{S} \to [0, 1]$ by $f(s) = N_t^m(s, X)$. Then Lemma 26 implies that $\mathbb{E}[N_t^m(s_t^m, X) - N_t^m(s_t, X)] \leq \Delta_t$, which gives us $p_{t+1}^m(X) - p_{t+1}(X) \leq \Delta_t + \alpha_t(X) \leq \Delta_t + \sup_{Y \subseteq \mathcal{S}} \alpha_t(Y)$. Since this holds for all $X \subseteq \mathcal{S}$, we have

$$\Delta_{t+1} = \sup_{X \subseteq \mathcal{S}} (p_{t+1}^m(X) - p_{t+1}(X)) \leq \Delta_t + \sup_{X \subseteq \mathcal{S}} \alpha_t(X).$$

Combining this with the inductive hypothesis of $\Delta_t \leq \sum_{i=1}^{t-1} \sup_{X \subseteq \mathcal{S}} \alpha_i(X)$ gives us

$$\Delta_{t+1} \leq \Delta_t + \sup_{X \subseteq \mathcal{S}} \alpha_t(X) \leq \sum_{i=1}^{t} \sup_{X \subseteq \mathcal{S}} \alpha_i(X)$$

which completes the induction. ■

Lemma 28 *If Γ satisfies the conditions of Theorem 8, then $\Delta_t \leq R(T, \sigma)$ for all $t \in [T]$.*

Proof Fix an arbitrary sequence of subsets $X_1, \ldots, X_{t-1} \subseteq \mathcal{S}$ and define $\boldsymbol{\mu}$ as follows:

$$\mu_i(s,a) = \begin{cases} P(s,a,X_i) & \text{if } i \in [t-1] \\ 1 & \text{otherwise.} \end{cases}$$

For $i > t-1$ we have $|\mu_i(s,\pi^m(s)) - \mu_i(s,\pi^m(s'))| = 0 \leq L||s-s'||$. For $i \in [t-1]$, $(P,\pi^m) \in \mathcal{P}(L)$ implies that

$$\begin{aligned} |\mu_i(s,\pi^m(s)) - \mu_i(s,\pi^m(s'))| &= \left|P\big(s,\pi^m(s),X_i\big) - P(s,\pi^m(s'),X_i)\big)\right| \\ &\leq \sup_{Y \subseteq \mathcal{S}} \left|P\big(s,\pi^m(s),Y\big) - P(s,\pi^m(s'),Y)\big)\right| \\ &= ||P(s,\pi^m(s)) - P(s,\pi^m(s'))||_{TV} \\ &\leq L||s-s'||. \end{aligned}$$

Therefore $\boldsymbol{\mu}$ satisfies local generalization. Next, we analyze $\mu_i(s_i,a_i)$. For each $i \in [t-1]$,

$$\begin{aligned} \mathbb{E}[\mu_i(s_i,a_i)] &= \mathop{\mathbb{E}}_{s\sim\mathcal{D}(s_i),\, a\sim\mathcal{D}(a_i)}[\mu_i(s,a)] \\ &= \mathop{\mathbb{E}}_{s\sim\mathcal{D}(s_i),\, a\sim\mathcal{D}(a_i)}[P(s,a,X_i)] && \text{(Defn of } \mu_i \text{ for } i \in [t-1]) \\ &= \mathop{\mathbb{E}}_{s\sim\mathcal{D}(s_i),\, a\sim\mathcal{D}(a_i)}[\Pr[s_{i+1} \in X_i \mid s_i = s, a_i = a]] && \text{(Defn of } P(s,a,X_i)) \\ &= \mathop{\mathbb{E}}_{s\sim\mathcal{D}(s_i),\, a\sim\mathcal{D}(a_i)}[\mathbb{E}[\mathbf{1}(s_{i+1} \in X_i) \mid s_i = s, a_i = a]] && \text{(Expectation of indicator)} \\ &= \mathbb{E}[\mathbf{1}(s_{i+1} \in X_i)] && \text{(Law of total expectation)} \\ &= \mathbb{E}[N_i(s_i,X_i)]. && \text{(Proof of Lemma 27)} \end{aligned}$$

Similarly but more easily,

$$N_i^m(s_i,X_i) = \Pr[s_{i+1}^m \in X_i \mid s_i^m = s_i] = P(s_i,\pi^m(s_i),X_i) = \mu_i(s_i,\pi^m(s_i)).$$

Therefore

$$\begin{aligned} \sum_{i=1}^{t-1} \alpha_i(X_i) &= \sum_{i=1}^{t-1} \mathbb{E}\left[N_i^m(s_i,X_i) - N_i(s_i,X_i)\right] \\ &= \sum_{i=1}^{t-1} \mathbb{E}[\mu_i(s_i,\pi^m(s_i)) - \mu_i(s_i,a_i)] \\ &= \sum_{i=1}^{T} \mathbb{E}[\mu_i(s_i,\pi^m(s_i)) - \mu_i(s_i,a_i)] \\ &= \mathrm{Reg}^+_{\mathrm{AC}}(T,\Gamma,V,\boldsymbol{\mu}) \\ &\leq R(T,\sigma). \end{aligned}$$

Since this holds for any sequence of subsets $X_1, \ldots, X_{t-1} \subseteq \mathcal{S}$, we have

$$\sum_{i=1}^{t-1} \sup_{X \subseteq \mathcal{S}} \alpha_i(X) = \sup_{X_1,\ldots,X_{t-1} \subseteq \mathcal{S}} \sum_{i=1}^{t-1} \alpha_i(X_i) \leq R(T,\sigma).$$

Applying Lemma 27 completes the proof. ■

Lemma 29 (State-based regret) *If Γ satisfies the conditions of Theorem 8, then*

$$\mathbb{E}\left[\sum_{t=1}^{T} r(s_t^m, \pi^m(s_t^m)) - \sum_{t=1}^{T} r(s_t, \pi^m(s_t))\right] \leq TR(T, \sigma).$$

Proof Let $r^m(s) = r(s, \pi^m(s))$ and fix $t \in [T]$. Then Lemma 26 implies that $\mathbb{E}[r^m(s_t^m) - r^m(s_t)] \leq \Delta_t$. Applying Lemma 28 gives us $\mathbb{E}[r^m(s_t^m) - r^m(s_t)] \leq R(T, \sigma)$, so

$$\begin{aligned}
\mathbb{E}\left[\sum_{t=1}^{T} r(s_t^m, \pi^m(s_t^m)) - \sum_{t=1}^{T} r(s_t, \pi^m(s_t))\right] &= \mathbb{E}\left[\sum_{t=1}^{T} \left(r^m(s_t^m) - r^m(s_t)\right)\right] \\
&\leq \sum_{t=1}^{T} R(T, \sigma) \\
&= TR(T, \sigma)
\end{aligned}$$

as required. ■

Theorem 8 follows from the bounds on state-based regret and action-based regret (Lemmas 25 and 29) and the decomposition shown in Section 6.3.

We now move on to Theorem 9, the version of the reduction which does not require local generalization for r. As mentioned above, local generalization of r was never used when bounding the state-based regret, so Lemma 29 still holds. Thus the only part of the proof that needs to change for Theorem 9 is how we bound the action-based regret.

Theorem 9 (Third reduction: no local generalization for r) *For functions $R_1, R_2 : \mathbb{N} \times [0,1] \to \mathbb{R}_{\geq 0}$, let Γ be an algorithm which satisfies $\mathrm{Reg}_{\mathrm{SA}}(T, \Gamma, V, \ell_{\mathbf{1}}) \leq R_1(T, \sigma)$ and $\mathrm{Reg}^{+}_{\mathrm{AC}}(T, \Gamma, V, \boldsymbol{\mu}) \leq R_2(T, \sigma)$ for any $T \in \mathbb{N}, \sigma \in [0,1], V \in \mathcal{V}(\sigma)$, and $(\boldsymbol{\mu}, \pi^m) \in \mathcal{U}(L)$. Then Γ satisfies*

$$\mathrm{Reg}_{\mathrm{MDP}}(T, \Gamma, (P, \mathcal{D}_1, \pi^m), r) \leq R_1(T, \sigma) + TR_2(T, \sigma)$$

for any $T \in \mathbb{N}, \sigma \in [0,1]$, and $(P, \mathcal{D}_1, \pi^m) \in \mathcal{V}(\sigma)$ with $(P, \pi^m) \in \mathcal{P}(L)$.

Proof Let $M_T = \{t \in [T] : a_t \neq \pi^m(s_t)\}$. Then

$$\begin{aligned}
\mathbb{E}[|M_T|] &= \mathbb{E}\left[\sum_{t=1}^{T} \mathbf{1}(a_t \neq \pi^m(s_t))\right] && \text{(Definition of } M_T\text{)} \\
&= \mathbb{E}\left[\sum_{t=1}^{T} \ell_{\mathbf{1}}(a_t, \pi^m(s_t)) - \sum_{t=1}^{T} \ell_{\mathbf{1}}(\pi^m(s_t), \pi^m(s_t))\right] && \text{(Definition of } \ell_{\mathbf{1}}\text{)} \\
&\leq \sup_{\pi \in \Pi} \mathbb{E}\left[\sum_{t=1}^{T} \ell_{\mathbf{1}}(a_t, \pi^m(s_t)) - \sum_{t=1}^{T} \ell_{\mathbf{1}}(\pi(s_t), \pi^m(s_t))\right] && (\pi^m \in \Pi)
\end{aligned}$$

$$
\begin{aligned}
&= \mathrm{Reg}_{\mathrm{SA}}(T, \Gamma, V, \ell_{\mathbf{1}}) && \text{(Definition of } \mathrm{Reg}_{\mathrm{SA}}\text{)} \\
&\le R_1(T, \sigma). && \text{(Theorem assumption)}
\end{aligned}
$$

Since $r(s_t, \pi^m(s_t)) = r(s_t, a_t)$ for all $t \notin M_T$ and $r(s, a) \in [0, 1]$ for all $s \in \mathcal{S}, a \in \mathcal{A}$, we have

$$
\begin{aligned}
\mathbb{E}\left[\sum_{t=1}^{T} r(s_t, \pi^m(s_t)) - \sum_{t=1}^{T} r(s_t, a_t)\right] &= \mathbb{E}\left[\sum_{t \in M_T} \big(r(s_t, \pi^m(s_t)) - r(s_t, a_t)\big)\right] \\
&\le \mathbb{E}\left[\sum_{t \in M_T} 1\right] = \mathbb{E}[|M_T|] \le R_1(T, \sigma).
\end{aligned}
$$

Thus the action-based regret is at most $R_1(T, \sigma)$. Lemma 29 tells us that the state-based regret is at most $TR_2(T, \sigma)$. Combining these proves the theorem. ■

Lastly, we apply Theorem 9 to Corollary 4 to obtain our final result.

Theorem 10 (Final no-regret guarantee for multichain MDPs) *Assume that either (1) Π has VC dimension d and $\sigma \in (0, 1]$ or (2) Π has Littlestone dimension d and $\sigma = 0$. Let $k = T^{\frac{2n+1}{2n+2}}$, let Γ_k be the corresponding algorithm from Corollary 4, and let Γ_{AC} denote Algorithm 2 with inputs $\Gamma_k, T,$ and $\varepsilon = T^{\frac{-1}{n+1}}$. Then Γ_{AC} satisfies*

$$
\begin{aligned}
\mathrm{Reg}_{\mathrm{MDP}}(T, \Gamma_{\mathrm{AC}}, (P, \mathcal{D}_1, \pi^m), r) &\in O\left(LT^{\frac{2n+1}{2n+2}}(d \log T + \sigma^+)\right), \\
Q(T, \Gamma_{\mathrm{AC}}, (P, \mathcal{D}_1, \pi^m)) &\in O\left(T^{\frac{2n+1}{2n+2}}\left(d + \sigma^+ + \mathbb{E}[\mathrm{diam}(\boldsymbol{s})^n]\right)\right),
\end{aligned}
$$

for any $(P, \mathcal{D}_1, \pi^m) \in \mathcal{V}(\sigma)$ with $(P, \pi^m) \in \mathcal{P}(L)$.

Proof Theorem 6 gives us the required bounds on $\mathrm{Reg}^+_{\mathrm{AC}}(T, \Gamma_{\mathrm{AC}}, V, \boldsymbol{\mu})$ and $Q(T, \Gamma_{\mathrm{AC}}, V) = Q(T, \Gamma_{\mathrm{AC}}, (P, \mathcal{D}_1, \pi^m))$. Lemma 24 gives us the required bound on $\mathrm{Reg}_{\mathrm{SA}}(T, \Gamma_{\mathrm{AC}}, V, \ell_{\mathbf{1}})$. Thus we can apply Theorem 9 with some $R_1(T, \sigma) \in O(T^{\frac{1}{2n+2}}(d \log T + \sigma^+))$ and $R_2(T, \sigma) \in O(LT^{\frac{-1}{2n+2}}(d \log T + \sigma^+))$ to obtain

$$
\begin{aligned}
\mathrm{Reg}_{\mathrm{MDP}}(T, \Gamma, (P, \mathcal{D}_1, \pi^m), r) &\in O\left(T^{\frac{1}{2n+2}}(d \log T + \sigma^+) + T \cdot LT^{\frac{-1}{2n+2}}(d \log T + \sigma^+)\right) \\
&= O\left(LT^{\frac{2n+1}{2n+2}}(d \log T + \sigma^+)\right)
\end{aligned}
$$

as desired. ■

Appendix D. Logarithmic Regret in Full-Feedback with Realizability

Theorem 1 reduces active online learning to full-feedback online learning. However, to obtain our desired subconstant regret, we will need a full-feedback algorithm with a better regret bound than the typical $O(\sqrt{T})$ bound. Lemma 2 provides such a bound under

the assumptions of $\pi^m \in \Pi$ (also known as *realizability*) and finite Littlestone dimension. However, finite Littlestone dimension is quite a strong assumption: even some simple classes have infinite Littlestone dimension (for example, the class of thresholds on $[0, 1]$: see Example 21.4 in Shalev-Shwartz and Ben-David, 2014). In this section, we provide an algorithm with $O(\log T)$ regret for σ-smooth adversaries under the weaker assumption of finite VC dimension (also assuming realizability).

Our algorithm involves approximating Π by a finite class $\tilde{\Pi}$. Formally: for $\varepsilon > 0$, a policy class $\tilde{\Pi}$ is an *ε-cover* of a policy class Π if for every $\pi \in \Pi$, there exists $\tilde{\pi} \in \tilde{\Pi}$ such that $\Pr_{s\sim\beta}[\pi(s) \neq \tilde{\pi}(s)] \leq \varepsilon$. The existence of small ε-covers is crucial:

Lemma 30 *For all $\varepsilon > 0$, any policy class of VC dimension d admits an ε-cover of size at most $(41/\varepsilon)^d$.*

There are many references for bounds of the form in Lemma 30. Lemma 7.3.2 in Haghtalab (2018) is one (although it nominally assumes that β is the uniform distribution, the exact same proof holds for an arbitrary β). See also Haussler and Long (1995) or Lemma 13.6 in Boucheron et al. (2013) for similar bounds.

We next show that an ε-cover is a good approximation for any σ-smooth distribution:

Lemma 31 *Let $\tilde{\Pi}$ be an ε-cover of Π and let $\sigma \in (0, 1]$. Then for any $\pi \in \Pi$, there exists $\tilde{\pi} \in \tilde{\Pi}$ such that for any σ-smooth distribution $\mathcal{D}$, $\Pr_{s\sim\mathcal{D}}[\pi(s) \neq \tilde{\pi}(s)] \leq \varepsilon/\sigma$.*

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

Our last ingredient is Lemma 32, which provides a particular type of loss bound:

Lemma 32 (Corollary 2.3 in Cesa-Bianchi and Lugosi, 2006) *There exists an algorithm Γ such that for any adversary, when Γ is run on a finite policy class Π, we have*

$$\mathbb{E}\left[\sum_{t=1}^{T} \ell_{\mathbf{1}}(a_t, a_t^m)\right] \leq \frac{e}{e-1}\left(\min_{\pi\in\Pi} \mathbb{E}\left[\sum_{t=1}^{T} \ell_{\mathbf{1}}(\pi(s_t), a_t^m)\right] + \ln|\Pi|\right).$$

We can now prove the main result of this section. The algorithm needs to know T (in order to construct an appropriate ε-cover) but not σ.

Lemma 3 *Assume $\exists \pi^* \in \Pi$ such that $\pi^*(s_t) = a_t^m$ $\forall t \in [T]$. If Π has VC dimension d, there exists a full-feedback algorithm Γ_{FF} such that for any $\sigma \in (0, 1]$ and $V \in \mathcal{V}(\sigma)$,*

$$\mathrm{Reg}_{\mathrm{SA}}(T, \Gamma_{\mathrm{FF}}, V, \ell_{\mathbf{1}}) \in O(d \log T + \sigma^+).$$

Proof By Lemma 30, Π admits a $\frac{1}{T}$-cover of size at most $(41T)^d$; let $\tilde{\Pi}$ be any such $\frac{1}{T}$-cover. Let Γ be the algorithm from Lemma 32 and define Γ_{FF} by running Γ on $\tilde{\Pi}$. Letting $c = \frac{e}{e-1}$,

$$\mathrm{Reg}_{\mathrm{SA}}(T, \Gamma_{\mathrm{FF}}, V, \ell_{\mathbf{1}}) = \sup_{\pi\in\Pi} \mathbb{E}\left[\sum_{t=1}^{T} \ell_{\mathbf{1}}(a_t, a_t^m) - \sum_{t=1}^{T} \ell_{\mathbf{1}}(\pi(s_t), a_t^m)\right]$$

$$
\begin{aligned}
&= \mathbb{E}\left[\sum_{t=1}^{T} \ell_{\mathbf{1}}(a_t, \pi^m(s_t))\right] \\
&\leq c\left(\min_{\tilde{\pi}\in\tilde{\Pi}} \mathbb{E}\left[\sum_{t=1}^{T} \ell_{\mathbf{1}}(\tilde{\pi}(s_t), \pi^m(s_t))\right] + \ln|\tilde{\Pi}|\right) \\
&\leq c\left(\min_{\tilde{\pi}\in\tilde{\Pi}} \sum_{t=1}^{T} \Pr[\tilde{\pi}(s_t) \neq \pi^m(s_t)] + \ln|\tilde{\Pi}|\right).
\end{aligned}
$$

Since V is σ-smooth, we have $\Pr_{(s,a)\sim V(h)}[s \in X] \leq \beta(X)/\sigma$ for any $X \subseteq \mathcal{S}$ and $h \in \mathcal{H}$. Then by the law of total expectation, for any $X \subseteq \mathcal{S}$ we have

$$
\Pr[s_t \in X] = \mathop{\mathbb{E}}_{h_t}[\Pr[s_t \in X \mid h_t]] = \mathop{\mathbb{E}}_{h_t}\left[\Pr_{(s,a)\sim V(h_t)}[s \in X]\right] \leq \mathop{\mathbb{E}}_{h_t}\left[\frac{\beta(X)}{\sigma}\right] \leq \frac{\beta(X)}{\sigma}.
$$

Thus the distribution of s_t is also σ-smooth, so by Lemma 31,

$$
\begin{aligned}
&\mathrm{Reg}_{\mathrm{SA}}(T, \Gamma_{\mathrm{FF}}, V, \ell_{\mathbf{1}}) \leq c\left(\min_{\tilde{\pi}\in\tilde{\Pi}} \sum_{t=1}^{T} \Pr[\tilde{\pi}(s_t) \neq \pi^m(s_t)] + \ln|\tilde{\Pi}|\right) \\
&\leq c\left(\sum_{t=1}^{T} \frac{1}{T\sigma} + \ln|\tilde{\Pi}|\right) = c\left(\sigma^{-1} + \ln|\tilde{\Pi}|\right) = c\left(\sigma^{-1} + \ln(41T)^d\right) \in O(d\log T + \sigma^{+})
\end{aligned}
$$

as required. ■

References

Stanislav Abaimov and Maurizio Martellini. Artificial intelligence in autonomous weapon systems. In Maurizio Martellini and Ralf Trapp, editors, *21st Century Prometheus: Managing CBRN Safety and Security Affected by Cutting-Edge Technologies*, pages 141–177. Springer International Publishing, Cham, 2020.

Eitan Altman. *Constrained Markov Decision Processes*, volume 7. CRC Press, 1999.

Mohammad Gheshlaghi Azar, Ian Osband, and Rémi Munos. Minimax regret bounds for reinforcement learning. In *International Conference on Machine Learning*, pages 263–272, 2017.

Yoshua Bengio, Geoffrey Hinton, Andrew Yao, Dawn Song, Pieter Abbeel, Trevor Darrell, Yuval Noah Harari, Ya-Qin Zhang, Lan Xue, Shai Shalev-Shwartz, et al. Managing extreme AI risks amid rapid progress. *Science*, 384(6698):842–845, 2024.

Lilian Besson and Emilie Kaufmann. What doubling tricks can and can't do for multi-armed bandits. *arXiv preprint 1803.06971*, 2018.

Adam Block, Yuval Dagan, Noah Golowich, and Alexander Rakhlin. Smoothed online learning is as easy as statistical learning. In *Conference on Learning Theory*, pages 1716–1786, 2022.

Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press, 2013.

Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

Nicolo Cesa-Bianchi, Gábor Lugosi, and Gilles Stoltz. Minimizing regret with label efficient prediction. *IEEE Transactions on Information Theory*, 51(6):2152–2162, 2005.

Michael K Cohen and Marcus Hutter. Pessimism about unknown unknowns inspires conservatism. In *Conference on Learning Theory*, pages 1344–1373, 2020.

Michael K Cohen, Elliot Catt, and Marcus Hutter. Curiosity killed or incapacitated the cat and the asymptotically optimal agent. *IEEE Journal on Selected Areas in Information Theory*, 2(2):665–677, 2021.

Corinna Cortes, Giulia DeSalvo, Claudio Gentile, Mehryar Mohri, and Scott Yang. Online learning with abstention. In *International Conference on Machine Learning*, pages 1059–1067, 2018.

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

Amit Daniely, Sivan Sabato, Shai Ben-David, and Shai Shalev-Shwartz. Multiclass learnability and the ERM principle. *Journal of Machine Learning Research*, 16(1):2377–2404, 2015.

Sarah Esser, Hilde Haider, Clarissa Lustig, Takumi Tanaka, and Kanji Tanaka. Action–effect knowledge transfers to similar effect stimuli. *Psychological Research*, 87(7):2249–2258, 2023.

Javier García and Fernando Fernández. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(42):1437–1480, 2015.

Peter M Gruber. *Convex and Discrete Geometry*. Springer, 2007.

Shangding Gu, Long Yang, Yali Du, Guang Chen, Florian Walter, Jun Wang, and Alois Knoll. A review of safe reinforcement learning: Methods, theories, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):11216–11235, 2024.

Blessing Guembe, Ambrose Azeta, Sanjay Misra, Victor Chukwudi Osamor, Luis Fernandez-Sanz, and Vera Pospelova. The emerging threat of AI-driven cyber attacks: A review. *Applied Artificial Intelligence*, 36(1), 2022.

Dylan Hadfield-Menell, Smitha Milli, Pieter Abbeel, Stuart Russell, and Anca D. Dragan. Inverse reward design. In *Conference on Neural Information Processing Systems*, pages 6768–6777, 2017.

Nika Haghtalab. *Foundation of Machine Learning, by the People, for the People*. PhD thesis, Carnegie Mellon University, 2018.

Nika Haghtalab, Yanjun Han, Abhishek Shetty, and Kunhe Yang. Oracle-efficient online learning for smoothed adversaries. In *Conference on Neural Information Processing Systems*, volume 35, pages 4072–4084. Curran Associates, Inc., 2022.

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

Shiva Hajian. Transfer of learning and teaching: A review of transfer theories and effective instructional practices. *IAFOR Journal of Education*, 7(1):93–111, 2019.

Steve Hanneke. Theory of disagreement-based active learning. *Foundations and Trends® in Machine Learning*, 7(2-3):131–309, 2014.

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

Jiafan He, Dongruo Zhou, and Quanquan Gu. Nearly minimax optimal reinforcement learning for discounted MDPs. In *Conference on Neural Information Processing Systems*, volume 34, pages 22288–22300, 2021.

Dan Hendrycks, Mantas Mazeika, and Thomas Woodside. An overview of catastrophic AI risks. *arXiv preprint 2306.12001*, 2023.

Thomas Jaksch, Ronald Ortner, and Peter Auer. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11(51):1563–1600, 2010.

Heinrich Jung. Ueber die kleinste Kugel, die eine räumliche Figur einschliesst. *Journal für die reine und angewandte Mathematik*, 123:241–257, 1901.

Olav Kallenberg. *Foundations of Modern Probability*. Springer, 1997.

Puneet Kohli and Anjali Chadha. Enabling pedestrian safety using computer vision techniques: A case study of the 2018 Uber Inc. self-driving car crash. In *Future of Information and Communication Conference*, pages 261–279. Springer, 2019.

Vanessa Kosoy. Delegative reinforcement learning: Learning to avoid traps with a little help. *arXiv preprint 1907.08461*, 2019.

Hanna Krasowski, Jakob Thumm, Marlon Müller, Lukas Schäfer, Xiao Wang, and Matthias Althoff. Provably safe reinforcement learning: Conceptual analysis, survey, and benchmarking. *Transactions on Machine Learning Research*, 2023.

Tor Lattimore and Marcus Hutter. Asymptotically optimal agents. In *International Conference on Algorithmic Learning Theory*, pages 368–382, Berlin, Heidelberg, 2011. Springer-Verlag.

Lihong Li, Michael L Littman, and Thomas J Walsh. Knows what it knows: a framework for self-aware learning. In *International Conference on Machine Learning*, pages 568–575, 2008.

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

Shuang Liu and Hao Su. Regret bounds for discounted MDPs. *arXiv preprint 2002.05138*, 2021.

Tao Liu, Ruida Zhou, Dileep Kalathil, Panganamala Kumar, and Chao Tian. Learning policies with zero or bounded constraint violation for constrained MDPs. In *Conference on Neural Information Processing Systems*, volume 34, pages 17183–17193, 2021.

Odalric-Ambrym Maillard, Timothy Mann, Ronald Ortner, and Shie Mannor. Active roll-outs in MDPs with irreversible dynamics. HAL Open Science, 2019.

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

Balas Kausik Natarajan. On learning sets and functions. *Machine Learning*, 4(1):67–97, 1989.

National Institute of Standards and Technology. Artificial intelligence risk management framework: Generative artificial intelligence profile. Technical report, National Institute of Standards and Technology, Gaithersburg, MD, 2024.

Ronald Ortner and Daniil Ryabko. Online regret bounds for undiscounted continuous reinforcement learning. In *Conference on Neural Information Processing Systems*, volume 25, 2012.

Takayuki Osa, Joni Pajarinen, Gerhard Neumann, J Andrew Bagnell, Pieter Abbeel, and Jan Peters. An algorithmic perspective on imitation learning. *Foundations and Trends® in Robotics*, 7(1-2):1–179, 2018.

Benjamin Plaut, Hanlin Zhu, and Stuart Russell. Avoiding catastrophe in online learning by asking for help. In *International Conference on Machine Learning*, volume 267, pages 49537–49568, 2025.

Martin L Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 2014.

Joaquin Quiñonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset Shift in Machine Learning*. MIT Press, 2022.

Pranav Rajpurkar, Emma Chen, Oishi Banerjee, and Eric J Topol. AI in health and medicine. *Nature Medicine*, 28(1):31–38, 2022.

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

Peter Slattery, Alexander K Saeri, Emily AC Grundy, Jess Graham, Michael Noetel, Risto Uuk, James Dao, Soroush Pour, Stephen Casper, and Neil Thompson. The AI risk repository: A comprehensive meta-review, database, and taxonomy of risks from artificial intelligence. *arXiv preprint 2408.12622*, 2024.

Francesco Emanuele Stradi, Matteo Castiglioni, Alberto Marchesi, and Nicola Gatti. Learning adversarial MDPs with stochastic hard constraints. *arXiv preprint 2403.03672*, 2024.

John Villasenor and Virginia Foggo. Artificial intelligence, due process and criminal sentencing. *Michigan State Law Review*, pages 295–354, 2020.

Yihong Wu. Lecture notes on: Information-theoretic methods for high-dimensional statistics. Technical report, Yale University, 2020.

Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. Generalized out-of-distribution detection: a survey. *International Journal of Computer Vision*, 132(12):5635–5662, 2024.

Zihan Zhang, Yuan Zhou, and Xiangyang Ji. Almost optimal model-free reinforcement learning via reference-advantage decomposition. In *Conference on Neural Information Processing Systems*, volume 33, pages 15198–15207, 2020.

Weiye Zhao, Tairan He, and Changliu Liu. Model-free safe control for zero-violation reinforcement learning. In Aleksandra Faust, David Hsu, and Gerhard Neumann, editors, *Conference on Robot Learning*, volume 164, pages 784–793, 2022.

Weiye Zhao, Tairan He, Rui Chen, Tianhao Wei, and Changliu Liu. State-wise safe reinforcement learning: a survey. In *International Joint Conference on Artificial Intelligence*, volume 6, pages 6814–6822, 2023.