Behavior Cloning is Not All You Need: The Optimality of On-Policy Distillation for Noisy Expert Feedback

Anonymous Author(s)
Affiliation
Address
email

Abstract

Imitation Learning (IL) is a natural framework for learning in sequential decision-making systems and has emerged as the dominant paradigm through which we understand language model training. A central puzzle is that, while in theory offline IL can be horizon-free and optimal, in practice online methods such as on-policy distillation (OPD) often outperform offline methods such as supervised fine-tuning (SFT). We propose a noisy expert model to explain this gap, in which the learner only has access to a noisy version of the expert's policy, but wishes to compete against the reward achieved by a clean expert, motivated by the fact that in many applications, e.g. training language models to perform long chains of thought, the expert is often imperfect. In this setting, we show a sharp separation between offline and online IL. Offline learning from noisy trajectories is fundamentally hard: to compete with the clean expert, the sample complexity must grow exponentially, in contradistinction to the clean expert setting where no explicit horizon dependence exists. In contrast, we prove that online interaction with the noisy expert via a novel variant of OPD enables horizon-free guarantees in some settings and polynomial dependence on horizon in general. Our analysis leads to an alternative loss function form that is commonly considered empirically for LM training. We further provide algorithms and lower bounds, and extend our results to the more realistic setting of unknown corruption when the clean expert is deterministic, thereby providing a theoretical foundation for why on-policy distillation can outperform standard supervised fine-tuning when training language models from imperfect teachers. We complement our theoretical results with experiments on synthetic and natural-language tasks, showing that the OPD variant suggested by our theory outperforms both offline BC and existing OPD objectives under noisy expert feedback.

1 Introduction

Imitation learning (IL) is a powerful framework for training agents to perform complex tasks in challenging environments by learning from expert demonstrations [7, 44, 47, 49]. In recent years, IL has gained significant attention due to its success in a wide range of applications, including robotics [5, 13], autonomous driving [11], and natural language processing [2, 7, 9, 41]. In particular, IL has been proposed as a natural framework through which to understand Language Models (LMs), where the training corpus is thought to consist of 'expert demonstrations' of human behavior. Due to the increasing importance and cost of LMs, this raises the question of how to best leverage expert access to train LMs, and more generally, how to best leverage expert access in IL.

Prior work has distinguished between two paradigms for IL: *offline* IL, where the learner only has access to a fixed dataset of expert demonstrations, and *online* IL, where the learner can interact with the expert and query for demonstrations in states visited by the learner [17, 45, 47, 49]. The canonical algorithm in offline IL is Behavioral Cloning (BC), where the learner simply performs supervised

Submitted to 40th Conference on Neural Information Processing Systems (NeurIPS 2026). Do not distribute.

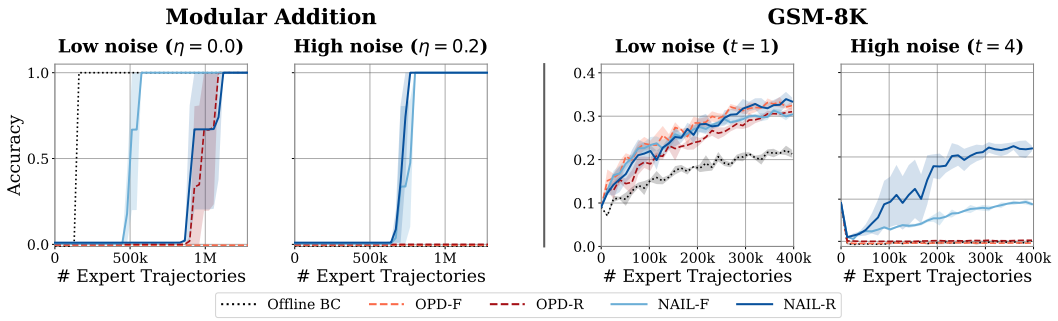

Figure 1: Comparison of offline BC, standard OPD from [2, 41], and our proposed NAIL for forward (F) and reverse (R) KL losses with clean and noisy experts. **Left:** Modular addition task: adding 31 numbers mod 7, where the expert is corrupted with probability $\eta \in \{0, 0.2\}$. **Right:** GSM-8K: math reasoning where the expert generates with temperature $t \in \{1, 4\}$. In both cases, NAIL is competitive with offline methods and standard OPD in the low noise regime and strongly outperforms these baselines in the high noise regime.

learning on the expert demonstrations in an effort to predict the action from the observation. While it was classically thought that BC suffers from compounding errors and thus has poor performance in long-horizon tasks, with the suboptimality of the learned policy growing quadratically with the horizon [47, 49], more recent work has shown that BC, when instantiated with the correct loss function, is both horizon-free and optimal in a minimax sense, even when comparing to online IL algorithms [17]. While this theoretical result may seem to suggest that BC is the best way to leverage expert access, in practice, many works have found that online IL algorithms, such as DAgger [49] and On-Policy Distillation (OPD) [2, 41], can significantly outperform offline IL in a wide range of settings. Recent work has suggested that this discrepancy between theory and practice may be due to the fact that the expert is often misspecified, meaning that the expert's policy is not realizable by the learner's policy class [45]; such misspecification can occur in robotics and autonomous driving problems when the expert and learner have different observation spaces (e.g. a human demonstrator has more sensory input than a robot learning) [5] and in LMs when we are trying to distill a large teacher model into a smaller student model [2, 24]. While misspecified experts are a natural model in many settings, they do not apply to one of the chief successes of IL for LMs, which is to use OPD to fine-tune a student LM from a teacher LM that has been trained with reinforcement learning (RL) but where both student and teacher share the same architecture [2, 41]. In this setting, the expert is not misspecified, but online IL (OPD) still outperforms offline IL (SFT) empirically.

In this work, we propose a *noisy expert* model to explain the discrepancy between theory and practice in IL. In particular, we suppose that there is a true expert policy $\pi^\star$ that achieves good expected reward when rolled out in an environment, but we only have access to a noisy version of this expert, $\pi^\star_\eta = (1-\eta) \cdot \pi^\star + \eta \cdot \nu$, where $0 \le \eta < 1$ is a noise level and ν is some noise distribution. This model captures the fact that in many real-world settings, such as math, code, and reasoning, the true expert against which we wish to compete could be approximately deterministic, but for computational reasons we are training a stochastic expert. Moreover, in the LM setting, the teacher model that we are distilling from is often a large model that is not fully converged and thus can be thought of as a noisy expert that occasionally makes mistakes [29, 61]. Such mistakes can also arise from human text that introduces minor errors, typos, and inconsistencies into datasets, which is known to be harmful for training LMs [35, 42, 60]. We thus ask: *Given access to a noisy expert, how should we leverage this expert to train a learner policy that performs well when rolled out in the environment?*

We provide an answer to this question in both the offline and online settings. For the sake of simplicity, in the introduction we discuss a special case of the noisy expert model (Definition 4), which is always satisfied when $\pi^\star$ is deterministic. We now summarize our main theoretical contributions. We first consider the setting where both η and ν are known to the learner; while unrealistic in practice, this setting allows us to cleanly characterize the fundamental limits of learning from a noisy expert without adding identifiability concerns (cf. Section 5). Our first result is that in offline settings, unlike

the clean expert setting where horizon-free guarantees are possible, we must pay *exponentially* in horizon H in order to guarantee learning whenever $\eta \gtrsim 1/H$.

Theorem 1 (Informal version of Theorem 6 and Proposition 1)**.** *Suppose an expert $\pi^\star$ is contained in a policy class Π and a learner has access to n trajectories rolled out from a noisy expert $\pi^\star_\eta$. In order to learn a policy $\hat{\pi}$ whose expected reward when rolled out is within ε of that of $\pi^\star$, BC requires $n \gtrsim (1-\eta)^{-(H+2)} \cdot \log(|\Pi|)/\varepsilon$ and this is optimal.*

This result stands in marked contrast to the clean expert setting, where BC can achieve horizon-free guarantees, and suggests that offline IL is fundamentally intractable when the expert is noisy. In the *online* setting, however, we show that we can circumvent this exponential dependence on the horizon via *On-Policy Distillation*. Our second main result shows that a natural analogue of OPD, where the learner queries the noisy expert for demonstrations in states visited by the learner, can achieve horizon-free guarantees even when the expert is noisy.

Theorem 2 (Informal version of Theorem 7)**.** *Let $\mathbb{P}^{\pi,\pi'}$ denote the law of an augmented trajectory of states and actions generated by rolling out π in an environment, but at each step querying π' for an auxiliary action. Then, the suboptimality in expected reward of a policy $\hat{\pi}$ can be bounded by $(1-\eta)^{-2} \cdot (\mathrm{D}_{\mathrm{KL}}(\mathbb{P}^{\hat{\pi},\pi^\star_\eta} \| \mathbb{P}^{\hat{\pi},\hat{\pi}_\eta}) \wedge \mathrm{D}_{\mathrm{KL}}(\mathbb{P}^{\hat{\pi},\hat{\pi}_\eta} \| \mathbb{P}^{\hat{\pi},\pi^\star_\eta}))$.*

Note that $\mathbb{P}^{\hat{\pi},\pi^\star_\eta}$ is only available through online feedback: we roll out policy $\hat{\pi}$ and, at each step, ask for what the (noisy) expert $\pi^\star_\eta$ would do in that state. *This result suggests that rolling out the student as will be done at deployment (e.g. greedily) and then using the noisy expert to score the actions taken by the student is the right way to leverage online access to a noisy expert, providing a principled justification for OPD.* We emphasize that the suggested loss function is *different* from that considered in Agarwal et al. [2], Lu and Lab [41], which do not distinguish between rollout distribution (e.g. greedy) and student policy (e.g. temperature 1) during training. Our final main theoretical result concerns our ability to control this forward KL divergence.

Theorem 3 (Informal version of Theorems 8 and 9)**.** *Suppose we have access to a policy class Π that contains $\pi^\star$. If η and ν are known, there is an OPD-like algorithm that can find a policy $\hat{\pi}$ with suboptimality in expected reward at most ε with $n \gtrsim H^2 \cdot \log(|\Pi|)/\varepsilon^2(1-\eta)^2$ rounds of interaction with the noisy expert $\pi^\star_\eta$ and this is optimal up to a factor of H. Moreover, if η and ν are unknown and $\pi^\star$ is deterministic, then as long as η is not too large and ν puts mass at most ρ on any action, $n \gtrsim \log(|\Pi|)/\varepsilon(1-\eta(1+\rho))^2$ rounds of interaction suffice and this is optimal.*

In essence, we show that horizon-free guarantees on suboptimality with respect to the *clean* expert are possible when given online access to a noisy expert, when the corruption is unknown but the expert is deterministic, while we pay only polynomially in the horizon in general when given online access to a noisy expert.

We empirically validate our theory in Figure 1 on both a synthetic task of modular addition [38] and a natural language task of math reasoning (GSM-8K) [14] by minimizing a loss function motivated by Theorem 7. Across both settings, online distillation substantially outperforms LogLossBC under noisy expert feedback, while remaining competitive under clean expert feedback, in line with our theoretical findings. Moreover, our NAIL variants consistently improve over standard OPD in the noisy expert setting, suggesting that the loss function suggested by our theory is indeed more effective for leveraging online access to a noisy expert. We defer further experimental details to Section 6.

In Section 2 we formally introduce the problem, as well as some prerequisite notions. In Section 3, we analyze the *offline* setting before discussing the *online* setting in Section 4. In Section 5, we extend to the setting where η and ν are unknown and the expert is deterministic. Finally, in Section 6 we present some preliminary empirical results validating our theoretical findings. We conclude with a discussion of related work and future directions in Appendix A and Table 1 summarizes our results.

2 Formal Problem Setup and Preliminaries

We consider Imitation Learning in the standard episodic Markov Decision Process (MDP) setting [17, 47, 49]. In particular, we have an MDP M with horizon H consisting of a state space $\mathcal{S}$, an action space $\mathcal{A}$, transition kernels $P_h : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ for each step $h \in [H]$, and a reward function $r : \mathcal{S} \times \mathcal{A} \to [0,1]$. We assume that the initial state distribution is given by some $\rho \in \Delta(\mathcal{S})$. We will consider learning *policies* $\pi : \mathcal{S} \times [H] \to \Delta(\mathcal{A})$ that can be non-stationary across the horizon,

mapping states to distributions over actions. Given a policy π we denote by $\mathbb{P}^\pi$ the distribution over trajectories induced by rolling out π in the MDP M, i.e., $\tau = (s_1, a_1, \ldots, s_H, a_H)$ with $s_1 \sim \rho$, $a_h \sim \pi_h(\cdot|s_h)$ and $s_{h+1} \sim P_h(\cdot|s_h, a_h)$. We will also use $\mathbb{E}^\pi$ to denote the expectation over trajectories induced by π. We are primarily concerned with finding policies π that have small *regret* with respect to the expert policy $\pi^\star$: $\mathrm{Reg}(\pi) = J(\pi^\star) - J(\pi)$, where $J(\pi) = \mathbb{E}^\pi\left[\sum_{h=1}^H r(s_h, a_h)\right]$. As in Foster et al. [17], we will decouple the reward range from the horizon by assuming that $0 \le \sum_{h=1}^H r(s_h, a_h) \le R$ for some R, for any trajectory τ. While naïvely, R can scale with the horizon H, in many settings of interest we can have R be a constant independent of H, which allows for horizon-free regret bounds, e.g. for LMs learning to solve math problems where the trajectory reward is binary [23, 54].

Unlike in reinforcement learning, imitation learning assumes access only to trajectories τ without any reward signal. While in classical IL, one assumes access to expert actions, in this work we consider instead a *noisy expert*, where there exists a corruption policy $\nu : \mathcal{S} \times [H] \to \Delta(\mathcal{A})$ and a corruption level $\eta \in [0, 1)$ such that the learner has access to $\pi^\star_\eta$ defined as

$$\pi^\star_{\eta,h}(a \mid s) = (1 - \eta) \cdot \pi^\star_h(a \mid s) + \eta \cdot \nu_h(a \mid s). \tag{1}$$

As in more traditional IL, we distinguish between two different ways in which to interact with the (noisy) expert. In the *offline* setting [17, 44, 47], we assume access only to expert trajectories and attempt to learn without additional interaction, formalized as follows.

Definition 1 (Offline Imitation Learning)**.** In the *offline* setting, a learner is given n trajectories $\tau^1, \ldots, \tau^n \sim \mathbb{P}^{\pi^\star_\eta}$ with $\pi^\star_\eta$ as in (1) and must output a policy $\hat{\pi}$ without further interaction.

Meanwhile, in *online* imitation learning [49], the learner is instead allowed to query the expert on trajectories rolled out by the learner itself, formalized as follows.

Definition 2. In the *online IL* setting, a learner interacts with the environment in rounds. In each round t, the learner deploys a policy π_t and observes a trajectory $\tau^{(t)}$ drawn from $\mathbb{P}^{\pi_t}$. The learner can then query the (noisy) expert $\pi^\star_\eta$ at each of the states visited in $\tau^{(t)}$ to get the expert's action at those states, forming $\tau^{(t)'} = (s_1, a'_1, \ldots, s_H, a'_H)$, where $a'_h \sim \pi^\star_{\eta,h}(\cdot|s_h)$.

When $\eta = 0$, this recovers the online IL setting studied in Ross et al. [49]. A priori, the online setting is more powerful than the offline setting and classically, when $\eta = 0$ (i.e., we are in the clean expert setting) this additional power was reflected in an improved sample complexity of online algorithms like DAgger [49] compared to offline algorithms like BC. More recently, however, Foster et al. [17] showed that in the realizable setting, BC is actually optimal and achieves horizon-free regret bounds. In order to state these bounds, we recall the definition of the Hellinger distance between two distributions P and Q over the same space $\mathcal{X}$:

Definition 3 (Hellinger Distance)**.** The Hellinger distance between two distributions P and Q over the same space $\mathcal{X}$ is defined as $\mathrm{D}_{\mathrm{H}^2}(P, Q) = 1/2 \cdot \int_{\mathcal{X}} (\sqrt{dP/d\mu(x)} - \sqrt{dQ/d\mu(x)})^2 d\,\mu(x)$, where μ is any distribution that dominates both P and Q.

While we defer to Polyanskiy and Wu [43] for a detailed discussion of the properties of the Hellinger distance, we provide a brief exposition in Appendix C.1. In particular, the Hellinger distance is intimately related to the more commonly used total variation distance up to a quadratic factor, and it is upper bounded by the KL divergence via Pinsker's inequality. The following result shows that regret is intimately related to the Hellinger distance between the trajectory distributions of the expert and learned policies.

Theorem 4 (Theorem 2.1 & 3.1 from [17])**.** *Let $\pi^\star$ and $\hat{\pi}$ be any two policies in an MDP with reward range R. Then it holds that*

$$J(\pi^\star) - J(\hat{\pi}) \lesssim R \cdot \sqrt{\mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\pi^\star}, \mathbb{P}^{\hat{\pi}}\right)} + R \cdot \mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\pi^\star}, \mathbb{P}^{\hat{\pi}}\right).$$

Moreover, if $\pi^\star$ is deterministic, then it holds that $J(\pi^\star) - J(\hat{\pi}) \lesssim R \cdot \mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\pi^\star}, \mathbb{P}^{\hat{\pi}}\right)$.

In addition, as we recall in Appendix C.3, Foster et al. [17] showed that the above reduction is tight, in the sense that for any two policies $\pi^\star$ and $\hat{\pi}$, there exists a reward function such that the Hellinger distance lower bounds the regret up to constant factors. Theorem 4 thus motivates us to focus on

controlling the Hellinger distance between the trajectory distributions of the expert and learned policies in order to get good regret guarantees; because of this, we will often abuse nomenclature and refer to $\mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\pi^\star}, \mathbb{P}^{\hat{\pi}}\right)$ as *regret*. Thus, the question we answer in this work is the following:

> *How much interaction with a noisy expert $\pi^\star_\eta$ is required in order to find a policy $\hat{\pi}$ such that $\mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\pi^\star}, \mathbb{P}^{\hat{\pi}}\right) \leq \varepsilon$ and how does online access to the expert affect this sample complexity?*

We will at times focus on the case where the corruption distribution ν satisfies a certain *domination* condition with respect to the policy class Π, defined as follows.

Definition 4 (κ-Domination)**.** We say that a corruption distribution ν is κ-dominated by policies π, π' if for any time step $h \in [H]$, state $s \in \mathcal{S}$, and action $a \in \mathrm{supp}(\pi_h(\cdot \mid s)) \cup \mathrm{supp}(\pi'_h(\cdot \mid s))$, it holds that $\nu_h(a|s) \leq \kappa \cdot (\pi_h(a|s) + \pi'_h(a|s))$. We will say that ν is κ-dominated by a policy class Π if for fixed $\pi^\star \in \Pi$, ν is κ-dominated by $\pi^\star$ and π for any $\pi \in \Pi$.

While this condition may appear somewhat unmotivated,[1] we will show that it is necessary in order to provide horizon-free guarantees in the noisy expert setting. The condition is weaker than it may first appear, as the likelihood domination only needs to hold on the support of the policies π and π', which can be much smaller than the entire action space. Indeed, note that it is always satisfied with $\kappa = 1$ for deterministic policies π, which we will study in detail in Section 5.

3 Offline Imitation Learning with a Noisy Expert

We begin by considering the offline setting, where the learner only has access to a dataset of trajectories generated by the noisy expert $\pi^\star_\eta$, i.e. $\tau^1, \dots \tau^n \sim \mathbb{P}^{\pi^\star_\eta}$. In this and the next sections, we will focus on the *known* corruption setting, where both η and ν are known to the learner and generalize to *unknown* corruptions in Section 5. While the known corruption setting is often not realistic in practice, it allows the main ideas to be presented with significantly greater clarity, as well as providing a useful benchmark for the more difficult unknown corruption setting. In the classical regime, wherein the learner receives clean expert feedback, Foster et al. [17] showed:

Theorem 5 (Theorem 2.1 Foster et al. [17])**.** *Let $\pi^\star \in \Pi$ and suppose $\tau^1, \dots, \tau^n \sim \mathbb{P}^{\pi^\star}$. If*

$$\hat{\pi} \in \operatorname*{argmin}_{\pi \in \Pi} \sum_{i=1}^{n} \sum_{h=1}^{H} -\log \pi(a_h^{(i)} \mid s_h^{(i)}), \tag{2}$$

then with probability at least $1-\delta$, it holds that $\mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\hat{\pi}}, \mathbb{P}^{\pi^\star}\right) \lesssim \log(|\Pi|/\delta)/n$.

This analysis is based on the key observation that (2) is the *Maximum Likelihood Estimator* (MLE) of the trajectory distribution of the expert $\pi^\star$ based on the observed trajectories, and thus we can apply classical results on the convergence of MLEs in Hellinger distance [20, 64]. While this is a powerful result when given clean expert feedback, in the noisy expert setting considered here, we only have access to trajectories from $\pi^\star_\eta$ rather than $\pi^\star$, and thus, even when η and ν are known, we can guarantee closeness in Hellinger distance only to $\mathbb{P}^{\pi^\star_\eta}$ rather than $\mathbb{P}^{\pi^\star}$. Indeed, it would thus be natural to replace (2) with the following estimator:

$$\hat{\pi} \in \operatorname*{argmin}_{\pi \in \Pi} \sum_{i=1}^{n} \sum_{h=1}^{H} -\log \pi_\eta(a_h^{(i)} \mid s_h^{(i)}), \tag{3}$$

where $\pi_\eta = (1-\eta) \cdot \pi + \eta \cdot \nu$ is the noisy version of π. Critically, while the identical analysis as in Theorem 5 shows that $\mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\pi^\star_\eta}, \mathbb{P}^{\hat{\pi}_\eta}\right)$ is guaranteed to be small, our ultimate goal is to compete with the *clean* expert $\pi^\star$ rather than the noisy expert $\pi^\star_\eta$, and thus we need to understand how this noisy Hellinger distance controls the clean Hellinger distance $\mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\pi^\star}, \mathbb{P}^{\hat{\pi}}\right)$.

Theorem 6. *Let π, π' be policies that κ-dominate ν. Then, $\mathrm{D}_{\mathrm{H}^2}(\mathbb{P}^{\pi}, \mathbb{P}^{\pi'}) \lesssim (1+\eta\cdot\kappa)\,(1-\eta)^{-H-2} \cdot \mathrm{D}_{\mathrm{H}^2}(\mathbb{P}^{\pi_\eta}, \mathbb{P}^{\pi'_\eta})$. In particular, if $\hat{\pi}$ is as in (3) then with probability at*

[1] This assumption arises naturally due to the singularity of the map $\eta \mapsto \sqrt{\eta}$ near $\eta = 0$ (cf. Remark 1).

least $1-\delta$*, it holds that* $\mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\hat{\pi}}, \mathbb{P}^{\pi^\star}\right) \lesssim (1+\eta\cdot\kappa)\,(1-\eta)^{-H-2}\cdot {}^{\log(|\Pi|/\delta)}/_{n}$*. In general, absent* κ*-domination it holds that* $\mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\pi^\star}, \mathbb{P}^{\hat{\pi}}\right) \lesssim \sqrt{{}^{((1-\eta)^{-H}-1)}/_{\eta(1-\eta)}\cdot \mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\hat{\pi}_\eta}, \mathbb{P}^{\pi^\star_\eta}\right)}$.

The proof is deferred to Appendix D.1, where the key idea is to apply backward induction by peeling off the trajectory distributions one step at a time, and then to use the κ-domination condition to control the error at each step. Note that in the absence of κ-domination, as $\eta \downarrow 0$, we do not recover a tight dependence as a polynomial in H factor appears, which is tight.

While Theorem 6 shows that as long as η and ν are known, a natural analogue of BC is consistent, the sample complexity is exponential in the horizon H whenever $\eta \gg {}^{1}/_{H}$, even when the clean expert $\pi^\star$ is deterministic, in contradistinction to the clean setting where no explicit horizon dependence exists. Thus in the long horizon regime, e.g. LMs generating long chains of thought in order to solve a hard math problem, this upper bound becomes vacuous. Unfortunately, our next result shows that this exponential dependence is necessary.

Proposition 1. *For any* $H \geq 2$*,* $\kappa \geq 1$*, and* $\eta < 1$ *there exists a horizon* H *MDP with* 3 *actions, policies* $\pi^\star, \hat{\pi}$*, and* κ*-dominated corruption distribution* ν *such that* $\mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\hat{\pi}}, \mathbb{P}^{\pi^\star}\right) \gtrsim \eta\cdot\kappa\cdot(1-\eta)^{-H-1}\cdot \mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\hat{\pi}_\eta}, \mathbb{P}^{\pi^\star_\eta}\right)$*. Moreover, if* $\kappa = 1$*, we can take* $\pi^\star$ *to be deterministic. In the absence of* κ*-domination, we may even force* $\mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\pi^\star}, \mathbb{P}^{\hat{\pi}}\right) \gtrsim \sqrt{{}^{((1-\eta)^{-H}-1)}/_{1-\eta}\cdot \mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\hat{\pi}_\eta}, \mathbb{P}^{\pi^\star_\eta}\right)}$.

The constructions witnessing the above result can be found in Appendix D.2. This result demonstrates that even in the easiest possible setting of noisy IL, where the contamination is known and the expert is deterministic, any offline IL algorithm must have sample complexity that depends exponentially on the horizon H in the worst case, presenting a fundamental barrier in the noisy expert setting. Moreover, the result demonstrates the fundamental necessity of κ-domination. Even worse, as we show in Proposition 15, a corollary of this result demonstrates that *any* offline IL algorithm must suffer from this exponential dependence in sample complexity, making offline IL fundamentally intractable in the noisy expert setting. In the next section, we show that online IL can reduce this exponential dependence on the horizon to at most polynomial.

4 On-Policy Distillation Allows for Improved Guarantees

In the previous section, we showed that in the noisy expert setting, offline IL algorithms such as BC must suffer from an exponential dependence on the horizon H, which is unacceptable in the kinds of long-horizon tasks that are ubiquitous in modern AI applications [23, 42, 54]. In this section, we demonstrate that *online* access to the expert can circumvent this exponential dependence, in direct contradistinction to the classical setting where Foster et al. [17] showed that when given clean expert feedback, BC is optimal even in the presence of *online expert interaction*.

In order to motivate our main result of the section showing that online access can greatly improve the dependence on the horizon in the noisy expert setting, we first emphasize that the primary obstacle to offline IL in the noisy expert regime is precisely the tightness of Theorem 6: in order to recover the clean expert's trajectory distribution in Hellinger distance, we must have exponentially better control on the Hellinger distance between the trajectory distributions of the noisy expert and noisy learned policy. While this relationship is unfortunately tight, it is natural to wonder if a stronger guarantee on the closeness of the noisy policy trajectories can lead to one paying a smaller constant factor in the relationship with the clean policy trajectories. In order to show that this is indeed the case, we introduce the notation $\mathbb{P}^{\pi,\pi'}$ for augmented trajectory distributions. More precisely, for policies π, π' we say that $\tau' = (s_1, a'_1, \ldots, s_H, a'_H) \sim \mathbb{P}^{\pi,\pi'}$ if $\tau = (s_1, a_1, \ldots, s_H, a_H) \sim \mathbb{P}^{\pi}$ and $a'_h \sim \pi'(\cdot|s_h)$. In other words, $\mathbb{P}^{\pi,\pi'}$ is the distribution over trajectories obtained by rolling out π in the MDP but then taking actions according to π' instead of π; note that $\mathbb{P}^{\pi,\pi}$ is in general not the same as $\mathbb{P}^{\pi}$ if π is stochastic due to the resampling of actions, but they do have the same marginal distributions. The following result shows that if we change our notion of divergence from Hellinger to KL and we consider augmented trajectory distributions, then we can get a much tighter relationship.

Theorem 7. *Let* $\pi^\star, \hat{\pi}$ κ*-dominate the corruption* ν*. Then for any* $0 \leq \eta < 1$ *it holds that*

$$\mathrm{D}_{\mathrm{H}^2}(\mathbb{P}^{\pi^\star}, \mathbb{P}^{\hat{\pi}}) \lesssim {}^{(1+\eta\cdot\kappa)}/_{(1-\eta)^2}\cdot\left(\mathrm{D}_{\mathrm{KL}}(\mathbb{P}^{\hat{\pi},\pi^\star_\eta} \| \mathbb{P}^{\hat{\pi},\hat{\pi}_\eta}) \wedge \mathrm{D}_{\mathrm{KL}}(\mathbb{P}^{\hat{\pi},\hat{\pi}_\eta} \| \mathbb{P}^{\hat{\pi},\pi^\star_\eta})\right). \tag{4}$$

Algorithm 1 NAIL: Noise-robust Aggregation for Imitation Learning

Require: Number of rounds n, policy class Π, noisy expert $\pi^\star_\eta$, corruption level η, corruption distribution ν.

1: Initialize $w_1 = \mathrm{Unif}(\Pi)$.
2: **for** $t = 1$ to n **do**
3: Define $\mu_t = \sum_{\pi\in\Pi} w_t(\pi)\cdot\pi$ and deploy μ_t to get trajectory $\tau^{(t)} \sim \mathbb{P}^{\mu_t}$.
4: Query noisy expert $\pi^\star_\eta$ on $\tau^{(t)}$ to obtain augmented trajectory $\tau^{(t)'}$.
5: Update $w_{t+1}(\pi) \propto w_t(\pi)\cdot\left(\prod_{h=1}^{H}\pi_\eta(a_h^{(t)'} \mid s_h^{(t)})\right)^{1/H}$.
6: **end for**
7: **return** $\hat\pi = \mu_T$ where $T \sim \mathrm{Unif}([n])$.

Absent the κ-domination condition, it holds that

$$\mathrm{D}_{\mathrm{H}^2}(\mathbb{P}^{\pi^\star}, \mathbb{P}^{\hat\pi}) \lesssim (1-\eta)^{-1}\cdot\sqrt{H\cdot(\mathrm{D}_{\mathrm{KL}}(\mathbb{P}^{\hat\pi,\pi^\star_\eta}\|\mathbb{P}^{\hat\pi,\hat\pi_\eta}) \wedge \mathrm{D}_{\mathrm{KL}}(\mathbb{P}^{\hat\pi,\hat\pi_\eta}\|\mathbb{P}^{\hat\pi,\pi^\star_\eta}))}. \tag{5}$$

We defer a proof of this result to Appendix E.1. The key idea is to use the *subadditivity* of the Hellinger distance from Foster et al. [16, 18] to control the trajectory Hellinger distance by the sum of the per-step Hellinger distances, and then demonstrate that these per-step Hellinger distances can be controlled by the KL divergence of the noisy augmented trajectory distributions. Critically, we have to use KL divergence instead of Hellinger distance in the upper bound in order to stitch the per-step distances back into a divergence over the *trajectory distributions* because KL divergence satisfies the *chain rule* (cf. Proposition 4) while Hellinger distance does not. Moreover, in Proposition 17 we show that (5) is tight up to a polynomial dependence on $(1-\eta)$, and thus the improvement from Theorem 7 is indeed significant in the presence of κ-domination.

Note that, unlike the offline setting, Theorem 7 is fully *independent of horizon* in the presence of κ-domination and scales only polynomially with H in general. Moreover, the upper bounds in Theorem 7 are closely related to the popular *On-Policy Distillation* (OPD) algorithm [2, 41], which is used to finetune a student policy to match the behavior of a teacher policy by minimizing a loss closely related to the right hand side of (4). Indeed, a common approach is to roll out a policy π_t and then use a target $\pi^\star_\eta$ to score the actions taken by π_t, in particular attempting to minimize $\mathrm{D}_{\mathrm{KL}}(\mathbb{P}^{\pi_t}\|\mathbb{P}^{\pi_t,\pi^\star_\eta})$. While our analysis suggests instead minimizing divergence on the *augmented trajectory distributions*, this provides some explanation for why OPD can be so much more successful than the offline BC (i.e., SFT) in practice.

While variants of OPD remain the recommended empirical approach (Section 6) and Theorem 7 provides theoretical backing for the success thereof, we continue in this section by introducing a simple new algorithm, NAIL (Algorithm 1). At its core, much like the classical DAgger algorithm [49], NAIL uses online learning to aggregate policies that are trained on the noisy expert's behavior on trajectories rolled out by the learner. Unlike some instantiations of DAgger, however, NAIL is *always on-policy*, in the sense that it rolls out the current policy mixture μ_t at each round [49]. Moreover, NAIL learns the policy at a *trajectory level*, thereby circumventing the horizon dependence that DAgger necessarily incurs by learning a different policy at each time step h [17, 49]. More precisely, NAIL begins with a uniform distribution w_1 over policies in Π and at each round t rolls out the mixture policy $\mu_t = \sum_{\pi\in\Pi} w_t(\pi)\cdot\pi$ to get a trajectory $\tau^{(t)}$, queries the noisy expert $\pi^\star_\eta$ on $\tau^{(t)}$ to get an augmented trajectory $\tau'^{(t)}$, and then updates w_{t+1} using the standard exponential weights update [8] (cf. Appendix C.4) with the loss given by the negative log-likelihood of $\tau'^{(t)}$ under each policy π. Finally, NAIL returns a random policy across the time steps. We show that NAIL achieves the following regret guarantee in the noisy expert setting.

Theorem 8. *Let $\pi^\star \in \Pi$ be an arbitrary policy, ν be an arbitrary corruption distribution, and $0 \le \eta < 1$ be a corruption level. If $\hat\pi$ is the policy returned by NAIL (Algorithm 1) after n rounds with feedback from $\pi^\star_\eta$, then* $\mathbb{E}\left[\mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\pi^\star}, \mathbb{P}^{\hat\pi}\right)\right] \lesssim H/(1-\eta)\cdot\sqrt{\log(|\Pi|)/n}$.

We defer a proof of this result to Appendix E.3, which proceeds by using the *mixability* (cf. Appendix C.4) of the trajectory-level log-loss to control the sum of on-policy trajectory-level KL divergences across rounds $1 \le t \le n$, before applying Theorem 7 and the convexity of the Hellinger

distance to get the desired bound. We reiterate that NAIL should be thought of as a theoretical algorithm that is designed to be optimal in the worst case and that directly minimizing the right hand side of (4) is a more practical approach to OPD. We further remark that, while NAIL uses exponential weights, really any online learning algorithm with small regret for the trajectory-level log-loss could be used in its place, as Theorem 7 allows for a generic reduction to online learning in the spirit of Ross et al. [49]. We now show that NAIL is optimal up to factors in H and η, although we do not rule out a sharper bound under κ-domination: the correct rate under this condition remains one of the chief theoretical questions left open by our work.

Proposition 2. *For any $H \geq 2$, $0 < \eta < 1$, and small ε there exists a horizon H MDP with 3 actions, a policy class Π of size $|\Pi| = 2$ containing $\pi^\star$, and known corruption ν such that any algorithm with online access to the noisy expert $\pi^\star_\eta$ requires $n \gtrsim \eta \cdot H / (1-\eta)^2 \cdot \varepsilon^2$ in order to obtain regret ε.*

We defer a proof to Appendix E.4. To summarize: there is a marked distinction between the clean and noisy expert settings: in contradistinction to standard IL, when the expert is noisy, online interaction can exponentially improve over offline access to the expert.

5 Imitating Deterministic Experts with Unknown Corruptions

In the previous sections, we focused on the setting of noisy IL with a *known corruption*. While this setting is simple, it is somewhat unrealistic; we now relax this assumption. The first problem we run into is one of *identifiability*: without additional assumptions on the corruption, it may well be the case that it is impossible to identify the expert policy $\pi^\star$ from the noisy expert $\pi^\star_\eta$, even absent stochasticity in $\pi^\star$; a concrete example of this phenomenon is provided in Appendix F.1. While this does not present an issue in standard IL, where closeness to the observed actions is the key desideratum, in the noisy expert setting, we care about learning a policy that is close to the *clean* expert and thus identifiability is critical. This issue does not arise when ν and η are known, as clean experts can be recovered from their noisy analogues, but recovery is not possible absent known corruption.

While many identifiability assumptions are possible, we focus on a natural one, satisfied in many of our motivating applications. We first restrict our focus to *deterministic experts*, motivated by applications to reasoning, math, or coding in LMs, where we think of the true expert as producing a fixed 'correct' output [23, 54]. We further assume that the corruption distribution ν is 'smooth.'

Definition 5. We say that a corruption distribution ν is ρ-smooth if for all time steps $h \in [H]$, states $s \in \mathcal{S}$, and actions $a \in \mathcal{A}$, it holds that $\nu_h(a|s) \leq \rho$.

Smoothness is a natural assumption when the action space is large, for example when we think of corruption as arising from typos or other minor errors in the output of an LM or human demonstrator. In particular, if ν is a uniform distribution over a large action space, then ν is ρ-smooth with $\rho = |\mathcal{A}|^{-1}$. Finally, we assume a bound on the noise level such that $\eta \leq \alpha < 1$ to ensure that a *margin* exists between the expert and the noise, which is necessary for identifiability.[2]

We propose an algorithm, NAILGUN (Algorithm 3), similar to NAIL, but that is capable of achieving *horizon-free* regret guarantees in the noisy expert setting with unknown corruption under the above assumptions. While we defer a detailed description of the algorithm to Appendix F.2, we note that it is very similar to NAIL, except that we replace the mixture policy μ_t by its greedy analogue $\overline{\mu}_t$, which plays the action with the highest probability under μ_t at each state and time step; moreover, the precise form of the update is different in order to account for the fact that we do not know η and ν.

Theorem 9. *Suppose that $\pi^\star \in \Pi$ is deterministic, ν is ρ-smooth, and $\eta \leq \alpha$ for some α satisfying $\alpha(1+\rho) < 1$. There exists an algorithm (Algorithm 3) without knowledge of η or ν that returns a policy $\hat{\pi}$ after n rounds of online interaction such that $\mathbb{E}\left[\mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\pi^\star}, \mathbb{P}^{\hat{\pi}}\right)\right] \lesssim \log(|\Pi|) / (1-\alpha(1+\rho))^2 \cdot n$.*

We defer a proof of Theorem 9 to Appendix F.3, which proceeds through an intricate analysis of the update rule and how the margin condition that $\alpha(1+\rho) < 1$ allows us to control the error without knowledge of η or ν. Note that unlike in the previous section, the particular form of the update rule is critical to achieving this guarantee, and thus we do not prove a generic reduction to online learning. We now conclude this section with a matching lower bound.

[2] We expect that we could replace determinism and smoothness with a more general margin condition and recover the same results, but we leave this for future work.

Proposition 3. *For any $0 < \alpha < 1$ and any ρ satisfying $\alpha(1+\rho) < 1$, there exists a horizon $H = 2$ MDP and a* deterministic *policy class Π of size $|\Pi| \lesssim 1/\rho$ such that in order to achieve expected regret at most ε, the learner requires $n \gtrsim \alpha\rho/\varepsilon(1-\alpha(1+\rho))^2$ rounds of interaction with an online noisy expert satisfying $\eta \leq \alpha$ and ν is ρ-smooth and* unknown.

The proof of this lower bound can be found in Appendix F.4. In particular, in the noisy expert regime where α and ρ are treated as constants, NAILGUN is optimal, even for horizon $H = 2$.

6 Experiments

We complement our theory with a small suite of experiments on synthetic and natural language tasks, showing that the OPD variant suggested by our theory (NAIL) can outperform both offline BC and existing OPD objectives under noisy expert feedback while remaining competitive under clean expert feedback. In particular, given an expert $\pi^\star_\eta$, we compare five algorithms: (i) LogLossBC (SFT), which minimizes the negative log-likelihood of the noisy expert trajectories; (ii) OPD-F, the standard OPD algorithm minimizing the forward KL divergence between student and teacher trajectory distributions [2]; (iii) OPD-R, the OPD algorithm that minimizes the reverse KL divergence between student and teacher trajectory distributions [2, 41]; (iv) NAIL-F, the variant of NAIL minimizing forward KL divergence between the augmented trajectory distributions; and (v) NAIL-R, the variant of NAIL minimizing reverse KL divergence. In particular, the primary difference between OPD and NAIL is that NAIL always generates *greedily* from the student policy, whereas OPD samples trajectories from the student policy; we emphasize that we do not instantiate the exponential weights update in Algorithm 1 but instead use gradient descent to minimize the upper bound in Theorem 7. We consider two tasks: (i) modular addition and (ii) GSM-8K [14]. Additional details are in Appendix B.1.

6.1 Modular Addition

We begin with a synthetic task from Li et al. [38], that of modular addition. Namely, for fixed p and m, given a sequence of m integers from $[p]$, the goal is to compute the sum of these integers modulo p. In the cited work, the authors demonstrated that low-depth transformers empirically benefit from a chain of thought (CoT) reasoning process in this task, making it a suitable testbed for studying the effect of horizon. All models are nanoGPT [30]. We train our expert $\pi^\star$ on a large corpus of data with $p = 7$ and $m = 31$ until its accuracy (with CoT) is approximately perfect. We then construct a noisy expert $\pi^\star_\eta$ by letting ν be uniform over tokens and corrupting the expert with probability η at each token if we have not yet seen an error, and corrupting all remaining tokens with probability 1 if we have already seen an error. We then train the student models using the five algorithms described above and evaluate the accuracy of the final learned policy $\hat{\pi}$ on a held-out test set of examples. We defer further details to Appendix B.1.3. Figure 1(left) summarizes the results of this experiment. For the clean expert ($\eta = 0$) regime, we see that most methods work, with offline BC (i.e., SFT) learning the fastest, which is consistent with the fact that offline BC is optimal in the clean expert setting [17]. For the noisy expert ($\eta = 0.2$) regime, we see a sharp separation between our proposed methods (NAIL-F, NAIL-R) and the baselines (LogLossBC, OPD-F, OPD-R), with the former achieving perfect accuracy and the latter failing to learn anything, even after over 1M expert interactions. We defer further comments and ablations to Appendix B.2 and Appendix B.3.

6.2 Mathematical Reasoning

We also consider a standard mathematical reasoning task, GSM-8K [14]. Here, we move beyond the realizable setting of our theory and use GEMMA3-1B-IT as the teacher and GEMMA3-270M-IT as the student [29]. To instantiate clean and noisy expert feedback with the same underlying model, we consider temperature 1 sampling as the 'clean' expert, which achieves 52.69% accuracy on GSM-8K test, and temperature 4 sampling as the 'noisy' expert, which achieves 0% accuracy. All training prompts are drawn from TinyGSM [40] and we evaluate zero-shot, exact-match accuracy on the GSM-8K test set with greedy decoding, reporting results over three random seeds. We defer further details to Appendix B.1.4. We report results in Figure 1(right), where we see that all online methods substantially outperform LogLossBC even in the clean expert setting, consistent with results from Agarwal et al. [2], Lu and Lab [41]. In the noisy expert setting, only NAIL-F and NAIL-R remain effective, achieving nontrivial performance while LogLossBC, OPD-F, and OPD-R all fail to learn anything, maintaining 0 accuracy, like the noisy expert, throughout training.

References

[1] Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J Hewett, Mojan Javaheripi, Piero Kauffmann, et al. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*, 2024.

[2] Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *The twelfth international conference on learning representations*, 2024.

[3] Awni Altabaa, Omar Montasser, and John Lafferty. Cot information: Improved sample complexity under chain-of-thought supervision. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026.

[4] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

[5] Jose Barreiros, Andrew Beaulieu, Aditya Bhat, Rick Cory, Eric Cousineau, Hongkai Dai, Ching-Hsin Fang, Kunimatsu Hashimoto, Muhammad Zubair Irshad, Masha Itkina, et al. A careful examination of large behavior models for multitask dexterous manipulation. *Science Robotics*, 11(113):eaea6201, 2026.

[6] Adam Block, Ali Jadbabaie, Daniel Pfrommer, Max Simchowitz, and Russ Tedrake. Provable guarantees for generative behavior cloning: Bridging low-level stability and high-level behavior. *Advances in Neural Information Processing Systems*, 36:48534–48547, 2023.

[7] Adam Block, Dylan J Foster, Akshay Krishnamurthy, Max Simchowitz, and Cyril Zhang. Butterfly effects of sgd noise: Error amplification in behavior cloning and autoregression. In *The Twelfth International Conference on Learning Representations*, 2024.

[8] Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.

[9] Jonathan Chang, Kianté Brantley, Rajkumar Ramamurthy, Dipendra Misra, and Wen Sun. Learning to generate better than your llm. In *NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following*, 2023.

[10] Fan Chen, Audrey Huang, Noah Golowich, Sadhika Malladi, Adam Block, Jordan T. Ash, Akshay Krishnamurthy, and Dylan J Foster. The coverage principle: How pre-training enables post-training. In *The Fourteenth International Conference on Learning Representations*, 2026.

[11] Jianyu Chen, Bodi Yuan, and Masayoshi Tomizuka. Deep imitation learning for autonomous driving in generic urban scenarios with enhanced safety. In *2019 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 2884–2890. IEEE, 2019.

[12] Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning converts weak language models to strong language models. In *International Conference on Machine Learning*, pages 6621–6642. PMLR, 2024.

[13] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.

[14] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[15] Dylan J Foster and Akshay Krishnamurthy. Efficient first-order contextual bandits: Prediction, allocation, and triangular discrimination. *Advances in Neural Information Processing Systems*, 34:18907–18919, 2021.

[16] Dylan J Foster, Sham M Kakade, Jian Qian, and Alexander Rakhlin. The statistical complexity of interactive decision making. *arXiv preprint arXiv:2112.13487*, 2021.

[17] Dylan J Foster, Adam Block, and Dipendra Misra. Is behavior cloning all you need? understanding horizon in imitation learning. *Advances in Neural Information Processing Systems*, 37: 120602–120666, 2024.

[18] Dylan J Foster, Yanjun Han, Jian Qian, and Alexander Rakhlin. Online estimation via offline estimation: An information-theoretic framework. *Advances in Neural Information Processing Systems*, 37:42840–42898, 2024.

[19] Yuqian Fu, Haohuan Huang, Kaiwen Jiang, Jiacai Liu, Zhuo Jiang, Yuanheng Zhu, and Dongbin Zhao. Revisiting on-policy distillation: Empirical failure modes and simple fixes. *arXiv preprint arXiv:2603.25562*, 2026.

[20] Sara A Geer. *Empirical Processes in M-estimation*, volume 6. Cambridge university press, 2000.

[21] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International journal of computer vision*, 129(6):1789–1819, 2021.

[22] Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge distillation of large language models. In *The Twelfth International Conference on Learning Representations*, 2024.

[23] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. DeepSeek-R1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[24] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[25] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. *Advances in neural information processing systems*, 29, 2016.

[26] Jonas Hübotter, Frederike Lübeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, et al. Reinforcement learning via self-distillation. *arXiv preprint arXiv:2601.20802*, 2026.

[27] Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. Tinybert: Distilling bert for natural language understanding. In *Findings of the association for computational linguistics: EMNLP 2020*, pages 4163–4174, 2020.

[28] Nirmit Joshi, Gal Vardi, Adam Block, Surbhi Goel, Zhiyuan Li, Theodor Misiakiewicz, and Nathan Srebro. A theory of learning with autoregressive chain of thought. In *The Thirty Eighth Annual Conference on Learning Theory*, pages 3161–3212. PMLR, 2025.

[29] Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

[30] Andrej Karpathy. NanoGPT. https://github.com/karpathy/nanoGPT, 2022.

[31] Yoon Kim and Alexander M Rush. Sequence-level knowledge distillation. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 1317–1327, 2016.

[32] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015*, 2015.

[33] Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. DISTILLM: towards streamlined distillation for large language models. In *Proceedings of the 41st International Conference on Machine Learning*, pages 24872–24895, 2024.

[34] Michael Laskey, Jonathan Lee, Roy Fox, Anca Dragan, and Ken Goldberg. Dart: Noise injection for robust imitation learning. In *Conference on robot learning*, pages 143–156. PMLR, 2017.

[35] Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Gadre, Hritik Bansal, Etash Guha, Sedrick Keh, Kushal Arora, et al. Datacomp-lm: In search of the next generation of training sets for language models. *Advances in Neural Information Processing Systems*, 37: 14200–14282, 2024.

[36] Yichen Li and Chicheng Zhang. Agnostic interactive imitation learning: New theory and practical algorithms. *arXiv preprint arXiv:2312.16860*, 2023.

[37] Yichen Li and Chicheng Zhang. Interactive and hybrid imitation learning: Provably beating behavior cloning. *arXiv preprint arXiv:2412.07057*, 2024.

[38] Zhiyuan Li, Hong Liu, Denny Zhou, and Tengyu Ma. Chain of thought empowers transformers to solve inherently serial problems. In *The Twelfth International Conference on Learning Representations*, 2024.

[39] Nick Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine learning*, 2(4):285–318, 1988.

[40] Bingbin Liu, Sebastien Bubeck, Ronen Eldan, Janardhan Kulkarni, Yuanzhi Li, Anh Nguyen, Rachel Ward, and Yi Zhang. TinyGSM: achieving 80% on GSM8k with one billion parameters. In *The 3rd Workshop on Mathematical Reasoning and AI at NeurIPS'23*, 2023.

[41] Kevin Lu and Thinking Machines Lab. On-policy distillation. *Thinking Machines Lab: Connectionism*, 2025. doi: 10.64434/tml.20251026. https://thinkingmachines.ai/blog/on-policy-distillation.

[42] Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, et al. 2 OLMo 2 furious. *arXiv preprint arXiv:2501.00656*, 2024.

[43] Yury Polyanskiy and Yihong Wu. *Information theory: From coding to learning*. Cambridge university press, 2025.

[44] Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, 1, 1988.

[45] Dhruv Rohatgi, Adam Block, Audrey Huang, Akshay Krishnamurthy, and Dylan J Foster. Computational-statistical tradeoffs at the next-token prediction barrier: Autoregressive and imitation learning under misspecification. In *The Thirty Eighth Annual Conference on Learning Theory*, pages 4831–4837. PMLR, 2025.

[46] Elvezio M Ronchetti and Peter J Huber. *Robust statistics*. John Wiley & Sons Hoboken, NJ, USA, 2009.

[47] Stéphane Ross and Drew Bagnell. Efficient reductions for imitation learning. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 661–668. JMLR Workshop and Conference Proceedings, 2010.

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

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

[50] Andrei A Rusu, Sergio Gomez Colmenarejo, Caglar Gulcehre, Guillaume Desjardins, James Kirkpatrick, Razvan Pascanu, Volodymyr Mnih, Koray Kavukcuoglu, and Raia Hadsell. Policy distillation. *arXiv preprint arXiv:1511.06295*, 2015.

[51] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[52] Ayush Sekhari, Karthik Sridharan, Wen Sun, and Runzhe Wu. Contextual bandits and imitation learning with preference-based active queries. *Advances in Neural Information Processing Systems*, 36:11261–11295, 2023.

[53] Ayush Sekhari, Karthik Sridharan, Wen Sun, and Runzhe Wu. Selective sampling and imitation learning via online regression. *Advances in Neural Information Processing Systems*, 36:67213–67268, 2023.

[54] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[55] Mingyang Song and Mao Zheng. A survey of on-policy distillation for large language models. *arXiv preprint arXiv:2604.00626*, 2026.

[56] Richard S Sutton, Andrew G Barto, et al. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

[57] Gokul Swamy, Sanjiban Choudhury, Drew Bagnell, and Steven Wu. Causal imitation learning under temporally correlated noise. In *International Conference on Machine Learning*, pages 20877–20890. PMLR, 2022.

[58] Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xiong-Hui Chen, Jianxin Yang, Zhenru Zhang, Yuqiong Liu, An Yang, Andrew Zhao, Yang Yue, Shiji Song, Bowen Yu, Gao Huang, and Junyang Lin. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for LLM reasoning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026.

[59] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in neural information processing systems*, 33:5776–5788, 2020.

[60] Maurice Weber, Daniel Y. Fu, Quentin Anthony, Yonatan Oren, Shane Adams, Anton Alexandrov, Xiaozhong Lyu, Huu Nguyen, Xiaozhe Yao, Virginia Adams, Ben Athiwaratkun, Rahul Chalamala, Kezhen Chen, Max Ryabinin, Tri Dao, Percy Liang, Christopher Ré, Irina Rish, and Ce Zhang. RedPajama: an open dataset for training large language models. *NeurIPS Datasets and Benchmarks Track*, 2024.

[61] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[62] Chenxu Yang, Chuanyu Qin, Qingyi Si, Minghui Chen, Naibin Gu, Dingyu Yao, Zheng Lin, Weiping Wang, Jiaqi Wang, and Nan Duan. Self-distilled rlvr. *arXiv preprint arXiv:2604.03128*, 2026.

[63] Ruixiang Zhang, Richard He Bai, Huangjie Zheng, Navdeep Jaitly, Ronan Collobert, and Yizhe Zhang. Embarrassingly simple self-distillation improves code generation. *arXiv preprint arXiv:2604.01193*, 2026.

[64] Tong Zhang. From ε-entropy to kl-entropy: Analysis of minimum information complexity density estimation. *The Annals of Statistics*, pages 2180–2210, 2006.

[65] Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. *arXiv preprint arXiv:2601.18734*, 2026.

[66] Binbin Zheng, Xing Ma, Yiheng Liang, Jingqing Ruan, Xiaoliang Fu, Kepeng Lin, Benchang Zhu, Ke Zeng, and Xunliang Cai. SCOPE: Signal-calibrated on-policy distillation enhancement with dual-path adaptive weighting. *arXiv preprint arXiv:2604.10688*, 2026.

Contents

A Discussion

In this work we investigated the problem of imitation learning with noisy experts, where the learner only has access to a corrupted version of the expert's policy. We showed theoretically that while offline imitation learning can be consistent in this setting, it suffers from an exponential dependence on the horizon, making it fundamentally intractable in the long horizon regime. On the other hand, we showed that online imitation learning can circumvent this exponential dependence and achieve horizon-free guarantees in some cases and polynomial in horizon guarantees in general, in direct contradistinction to the more classical IL setting with clean experts. Moreover, we provided formal theoretical justification for the benefits of the commonly used *On-Policy Distillation* (OPD) algorithm and validated our theory with a small suite of experiments on synthetic and natural language tasks. A summary of our theoretical results is in Table 1. In this section, we discuss related work and future directions for research.

A.1 Limitations

While our work presents a clean theoretical picture, there are several limitations in immediately translating our results to practice. First, consistent with the literature in RL and IL, our results are all in the finite Π setting [17, 45, 47, 49]; standard arguments can be used to extend our results to infinite Π under covering conditions. Second, we adopt an idealized corruption model, which may not capture all the nuances of real-world expert noise; we defer to future work the task of developing more sophisticated corruption models that may better capture the types of noise observed in practice. Third, many of our results are in the known corruption setting, and we only provide results in the unknown corruption setting when the expert is deterministic; while this setting is relevant for many of our motivating applications, e.g. LMs learning to perform long chains of thought, it would be interesting to understand the unknown corruption setting more generally. Fourth, there remains a linear in horizon gap between our upper bound for arbitrary experts and our lower bound; closing this gap would be an interesting direction for future work; similarly, whether or not fully horizon-free rates are possible for online IL in the presence of κ-domination is an interesting question. Finally, due to resource constraints, our experiments are limited in scope and scale; it would be interesting to conduct a more comprehensive empirical investigation of the phenomena we identify here, especially in the context of training larger language models for longer.

A.2 Related Work

Imitation Learning. Behavior cloning was first introduced empirically in Pomerleau [44] and has since engendered the field of Imitation Learning, with a large body of work considering both theoretical and empirical aspects of the problem [5, 6, 13, 25, 47]. Of particular note is Ross et al. [49], which introduced the online IL algorithm DAgger; while NAIL is conceptually similar (asking for expert feedback to correct potential student mistakes), the analysis and guarantees are quite different, with DAgger attempting to reduce horizon dependence in a clean expert setting.

Table 1: Summary of the main theoretical results in the general noisy-expert setting. Here $\mathsf{KL}_{\mathrm{aug}}(\pi)$ denotes the smaller of the two augmented-trajectory KL divergences appearing in Theorem 7. Bounds control $\mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\pi^\star}, \mathbb{P}^{\widehat{\pi}}\right)$ and are stated up to universal constants and logarithmic factors when indicated.

Regime	Setting	Bound	Reference
Offline			
	No κ-domination.	$\sqrt{\frac{(1-\eta)^{-H}-1}{\eta(1-\eta)} \cdot \frac{\log(\lvert\Pi\rvert/\delta)}{n}}$.	Theorem 6 and Proposition 1; this is tight.
	κ-domination.	$(1+\eta\cdot\kappa)(1-\eta)^{-H-2}\cdot\frac{\log(\lvert\Pi\rvert/\delta)}{n}$.	Proposition 13; this is tight.
Online			
	Augmented-trajectory KL reduction without κ-domination.	$\frac{1}{1-\eta}\sqrt{H \cdot \mathsf{KL}_{\mathrm{aug}}(\widehat{\pi})}$.	Theorem 7; this is tight up to factors in H and η.
	Augmented-trajectory KL reduction with κ-domination.	$\frac{1+\eta\cdot\kappa}{(1-\eta)^2} \cdot \mathsf{KL}_{\mathrm{aug}}(\widehat{\pi})$.	Theorem 7
	NAIL / trajectory-level online aggregation	$\frac{H}{1-\eta}\sqrt{\frac{\log\lvert\Pi\rvert}{n}}$.	Theorem 8. The lower bound in Proposition 2 is a factor of H and η away from this upper bound.
	Deterministic $\pi^\star$, ρ-smooth ν, and $\eta \le \alpha$ with $\alpha(1+\rho)<1$	$\frac{\log\lvert\Pi\rvert}{(1-\alpha(1+\rho))^2 n}$.	Theorem 9 and Proposition 3; this is tight up to a log factor.

Other IL algorithms study different feedback and data-collection models. SMILe [47] works in a similar fashion as DAgger to collect learner-induced states under clean expert feedback, but instead learns a sequence of policies and mixes them to control distribution shift. AggreVate [48] strengthens the feedback model by using expert cost-to-go or advantage information, and GAIL [25] instead casts IL as adversarial occupancy-measure matching from fixed expert demonstrations. DART [34] collects off-policy demonstrations from a noise-injected expert so that the demonstrations include recovery behavior.

More recently, Foster et al. [17] provided a comprehensive analysis of behavior cloning in the realizable, clean expert setting and Rohatgi et al. [45] extended this to the agnostic setting. In the online setting, Li and Zhang [37] considered a variant of DAgger where the learner pays per-state instead of per-trajectory in expert feedback, while Li and Zhang [36] also considers the agnostic setting but with strong assumptions on the MDP structure and a horizon dependence in their guarantees. In contrast to these works, we consider a more challenging setting with noisy experts and show that online IL can achieve horizon-free guarantees in some settings and polynomial dependence on horizon in general even in this more difficult setting, while offline IL suffers from an exponential dependence on the horizon. On the empirical side, OPD has seen recent success in training large language models (LLMs) to perform long chains of thought [2, 4, 41], and our theoretical results provide formal justification for the benefits of OPD in this setting.

The connections between IL and LMs have been explored in a number of recent works [7, 9, 17, 45], with the deterministic expert setting being particularly relevant to LMs learning long chains of thought [3, 28]. Our results generalize beyond this setting, allowing for stochastic experts and corrupted feedback.

Noisy Expert Feedback. Most prior work in IL assumes access to a clean expert, with the notable exceptions of Sekhari et al. [52, 53]. While a promising start, and the former work provides an exponential in horizon lower bound for offline learning in a particular setting, these works make strong assumptions on the noise and MDP structure and only apply to specific policy classes; moreover, they only consider a very specific type of corruption arising from preference-based feedback, in contradistinction to our bounds which apply to arbitrary policy classes and MDPs. Another related work that concerns a noisy-expert model is that of Swamy et al. [57], which focuses (under very strong assumptions) on causal IL under temporally correlated noise, where action noise changes future states and creates spurious state-action correlations that BC can learn. Our corruption model has the same rolled-in character, but our focus is different: rather than debiasing offline demonstrations via causal assumptions, we show that online local queries can avoid learning from fully contaminated trajectories.

Distillation. Knowledge Distillation is typically the process of training a student model to match a stronger teacher, emphasizing compression and efficient deployment. Early work emphasized transferring softened teacher predictions from cumbersome ensembles to smaller students [24], and the same idea was later extended to reinforcement-learning policies [50] and to autoregressive sequence models through sequence-level distillation [31]. Critically, this continued as an off-policy paradigm: the student is trained on a fixed transfer set, teacher-generated sequences, or teacher logits, as in [27, 51, 59]; see also the general survey of [21]. While highly effective for compression, these methods do not train on prefixes induced by the student's own generations, and are therefore susceptible to the train–test mismatch inherent in autoregressive generation, a well known problem in imitation learning [44, 47].

On-policy distillation [2] addresses this mismatch by training the student on self-generated sequences while receiving teacher feedback on those sequences using varying discrepancy measures between teacher and student. One objective in distillation is the forward KL from the teacher to the student, which corresponds in our terminology to OPD-F. Prior work of Agarwal et al. [2] has noted that this objective can be problematic when the student is not expressive enough to fit the teacher distribution: because forward KL is mode-covering, the resulting student may place probability on generations that are unlikely under the teacher. Subsequent work [22, 41] has therefore explored alternative on-policy objectives, advocating for sequence/trajectory-level reverse KL to avoid the mode-covering behavior of forward KL in generative settings, while [33] smooths the loss and reduces rollout cost with an adaptive off-policy reuse scheme.

Our results offer a more unifying perspective. In the noisy-expert setting, the difficulty is not only which KL direction is used, but also which prefix distribution the teacher is queried on and which feedback is informative. Sampling from the student can itself induce noisy or off-distribution prefixes, reducing the usefulness of local teacher feedback. This helps explain empirical findings that sampled-token OPD can be brittle, and that stability can improve by restricting both the rollout distribution and the teacher signal used for local updates [19]. It is also consistent with recent work that makes OPD updates more selective, for example by down-weighting teacher KL on failed trajectories where the teacher itself is uncertain [66]. More generally, work on pivotal or "forking" tokens suggests that a small number of CoT tokens can determine which reasoning branch the model follows, and hence whether downstream feedback remains informative [1, 58]. Our theory gives a principled prescription: matching teacher feedback along the learner's deployed prefix distribution; in our experiments with deterministic experts, using greedy student rollouts together with augmented trajectory-level KL objectives helps suppress sampling-induced noise and learn more effectively from imperfect teachers. We leave a more detailed study of when forward versus reverse KL is preferable to future work. More recently, self-play and self-distillation methods have shown that on-policy data generation can itself be a source of improvement even without a separate external teacher [12, 26, 62, 63, 65]. For a more detailed overview, we refer to the recent survey of Song and Zheng [55].

A.3 Future Directions

While we provided a near-tight theoretical analysis of the problems studied, there remain many interesting open questions for future work. First, in the unknown corruption setting, we focused on the case of deterministic experts and smooth corruptions with margin, but it would be interesting to understand the extent to which these assumptions can be relaxed while still achieving horizon-free guarantees, or even consistency. Second, while we considered the natural linear corruption setting,

motivated by classical statistical theory [46] and applications to LMs, alternative corruption models are also interesting to study, for example trajectory-level, multiplicative, or token-adaptive corruptions that target semantically pivotal "fork" tokens in long CoT reasoning [1, 10, 58, 66]. Third, our online IL guarantees are *in expectation* due to the inherited guarantees from online learning with log loss, whereas the offline guarantees are *with high probability* and resolving this discrepancy could provide additional insight. Finally, scaling up the experimental results to more complex environments and real-world tasks remains an important direction for future work.

A.4 Broader Impacts

This work is primarily theoretical and aims to improve our understanding of imitation learning from imperfect expert feedback. We hope that showing fundamental separation between offline and online methods (such as on-policy distillation) will lead to more reliable and sample-efficient post-training methods, especially in long-horizon settings.

While our methods could be seen as enabling effective learning from imperfect teachers, they could also make it easier to train models for harmful applications if deployed without appropriate safeguards. Our experiments are limited to synthetic tasks and standard mathematical reasoning benchmarks, and we do not release deployed systems or high-risk models. We therefore view the main impact of this work as methodological, with downstream risks depending on how future systems built using these ideas are trained, evaluated, and released.

B Further Empirical Results

This section provides additional details and discussion for the experiments in Section 6. Appendix B.1 describes the implementation of the methods in the autoregressive LM setting, as well as the synthetic and reasoning tasks setup. Appendix B.2 gives a more detailed interpretation of the main empirical results, including the differences between clean and noisy feedback and between greedy and sampled rollouts. Finally, Appendix B.3 reports additional experiments and ablations.

B.1 Further Experimental Details

In this section, we provide implementation details for the empirical results in Section 6. Appendix B.1.1 defines the offline and online learning objectives that we compare, including behavior cloning, standard OPD, and our NAIL-inspired variants; in Appendix B.1.2 we spell out their gradient estimators. Appendix B.1.3 describes the setup of the synthetic CoT tasks, including the data-generation procedure, corruption model, architectures, and training details. Finally, Appendix B.1.4 gives the corresponding details for the GSM8K experiments, including the teacher and student models, sampling procedure, training setup, and evaluation protocol.

We used NVIDIA RTX 6000 Pro GPUs for the modular addition experiments, with each run taking 1 hour. In total, the modular addition experiments used 50 GPU hours. For the GSM8K experiments, we used NVIDIA A100 40GB GPUs. Each online-method run took approximately 60 hours, while each offline-method run took approximately 3 hours; this disparity in runtime is an artifact of our highly suboptimal hardware setup and a naïve implementation and we do not believe such a discrepancy would exist in a more optimal setup. In total, the GSM8K experiments used roughly 800 GPU hours.

B.1.1 Methods and Implementations

We compare offline behavior cloning against several on-policy distillation objectives. The key distinction among the online methods is that there are two separate choices: the policy used to generate prefixes, and the divergence used to compare the student and teacher distributions on those prefixes. This distinction is motivated by Theorem 7, which controls the clean-expert trajectory error through KL divergences between *augmented* trajectory laws. In particular, the rollout distribution determines which prefixes are visited, while the per-prefix student distribution determines the next-token distribution being matched to the noisy expert.

Let π_θ denote the trainable student policy and let $\bar{\pi}_\theta$ denote the greedy rollout policy induced by π_θ. Thus $y_{1:T} \sim \bar{\pi}_\theta(\cdot \mid x)$ denotes the trajectory obtained by greedy decoding from the student, while $\pi_\theta(\cdot \mid x, y_{<t})$ denotes the student's full next-token distribution on the resulting prefix. We use $\pi^\star_\eta$ to

denote the noisy teacher. In the synthetic experiments, $\pi_\eta^\star$ is obtained by corrupting the clean expert token with the specified corruption law. In the language-model experiments, it is implemented by sampling from the teacher at the specified temperature.

NAIL-F. The forward-KL version of our method (which minimizes the first term in (4)) uses greedy student prefixes and trains the student to match noisy-teacher next-token feedback on those prefixes. Ideally, this corresponds to minimizing

$$\mathcal{L}_{\text{NAIL-F}}(\theta) = \mathbb{E}_{x\sim\mathcal{X}}\mathbb{E}_{y_{1:T}\sim\bar{\pi}_\theta(\cdot|x)}\left[\sum_{t=1}^{T} \mathrm{D}_{\mathrm{KL}}(\pi_\eta^\star(\cdot \mid x, y_{<t})\|\pi_\theta(\cdot \mid x, y_{<t}))\right],$$

Equivalently, up to the entropy of the noisy teacher, this is a soft-target cross-entropy objective:

$$-\mathbb{E}_{x\sim\mathcal{X}}\mathbb{E}_{y_{1:T}\sim\bar{\pi}_\theta(\cdot|x)}\left[\sum_{t=1}^{T}\sum_{a\in\Sigma} \pi_\eta^\star(a \mid x, y_{<t}) \log \pi_\theta(a \mid x, y_{<t})\right].$$

Since our implementation queries the teacher by sampling, we estimate this loss by drawing an independent teacher token $\tilde{y}_t \sim \pi_\eta^\star(\cdot \mid x, y_{<t})$ at each greedy student prefix and minimizing

$$\widehat{\mathcal{L}}_{\text{NAIL-F}}(\theta) = \mathbb{E}_{x\sim\mathcal{X}}\mathbb{E}_{y_{1:T}\sim\bar{\pi}_\theta(\cdot|x)}\mathbb{E}_{\tilde{y}_t\sim\pi_\eta^\star(\cdot|x,y_{<t})}\left[\sum_{t=1}^{T} -\log \pi_\theta(\tilde{y}_t \mid x, y_{<t})\right].$$

Thus, NAIL-F can be thought of as "local behavior cloning" on the learner's own greedy prefixes: rather than imitating complete noisy teacher trajectories, it repeatedly asks what the noisy teacher would do at states actually reached by greedily rolling out the current student, and then trains the student to predict the noisy teacher's sample via cross-entropy. This is the empirical analogue of the forward KL augmented-trajectory objective suggested by our theory. When η and ν are known, the literal theoretical objective replaces π_θ above by the noisy student policy $\pi_{\theta,\eta} = (1-\eta)\pi_\theta + \eta\nu$; in our experiments we use the simpler student-matching surrogate above.

NAIL-R. We also consider the reverse KL analogue suggested by minimizing the second term in (4). As in NAIL-F, prefixes are generated by the greedy student rollout, but the KL direction is reversed:

$$\mathcal{L}_{\text{NAIL-R}}(\theta) = \mathbb{E}_{x\sim\mathcal{X}}\mathbb{E}_{y_{1:T}\sim\bar{\pi}_\theta(\cdot|x)}\left[\sum_{t=1}^{T} \mathrm{D}_{\mathrm{KL}}(\pi_\theta(\cdot \mid x, y_{<t})\|\pi_\eta^\star(\cdot \mid x, y_{<t}))\right].$$

We estimate the student expectation in the reverse KL by drawing an auxiliary token $\hat{y}_t \sim \pi_\theta(\cdot \mid x, y_{<t})$ at each greedy prefix:

$$\widehat{\mathcal{L}}_{\text{NAIL-R}}(\theta) = \mathbb{E}_{x\sim\mathcal{X}}\mathbb{E}_{y_{1:T}\sim\bar{\pi}_\theta(\cdot|x)}\mathbb{E}_{\hat{y}_t\sim\pi_\theta(\cdot|x,y_{<t})}\left[\sum_{t=1}^{T} \log \pi_\theta(\hat{y}_t \mid x, y_{<t}) - \log \pi_\eta^\star(\hat{y}_t \mid x, y_{<t})\right].$$

We note that the auxiliary token $\hat{y}_t$ is not necessarily the greedy rollout token y_t. The greedy rollout tokens determine the prefix distribution, while the auxiliary samples estimate the student expectation in the reverse KL. This separation is needed to match the augmented-trajectory viewpoint: the rollout policy and the next-token distribution being compared need not be the same object.

OPD-F. As an ablation, we also consider the forward-KL objective with sampled student rollouts rather than greedy rollouts:

$$\mathcal{L}_{\text{OPD-F}}(\theta) = \mathbb{E}_{x\sim\mathcal{X}}\mathbb{E}_{y_{1:T}\sim\pi_\theta(\cdot|x)}\left[\sum_{t=1}^{T} \mathrm{D}_{\mathrm{KL}}(\pi_\eta^\star(\cdot \mid x, y_{<t})\|\pi_\theta(\cdot \mid x, y_{<t}))\right],$$

This isolates the effect of the rollout distribution: comparing OPD-F to NAIL-F tests whether greedy learner prefixes are important, while comparing OPD-F to OPD-R tests the effect of KL direction under the same sampled-prefix distribution.

OPD-R. Finally, we include the standard on-policy distillation baseline using reverse KL [41]. This method samples trajectories from the student, queries the teacher on those same student-generated prefixes, and updates the student using the teacher log-probability of the sampled student tokens:

$$\mathcal{L}_{\text{OPD-R}}(\theta) := \mathbb{E}_{x\sim\mathcal{X}}\mathbb{E}_{y\sim\pi_\theta(\cdot|x)}\left[\sum_{t=1}^{T} \mathrm{D}_{\mathrm{KL}}(\pi_\theta(\cdot \mid x, y_{<t})\|\pi_\eta^\star(\cdot \mid x, y_{<t}))\right].$$

Using the sampled rollout token $y_t \sim \pi_\theta$, this becomes:

$$\widehat{\mathcal{L}}_{\text{OPD-R}}(\theta) = \mathbb{E}_{x\sim\mathcal{X}}\mathbb{E}_{y_{1:T}\sim\pi_\theta(\cdot|x)}\left[\sum_{t=1}^{T}\Big(\log \pi_\theta(y_t \mid x, y_{<t}) - \log \pi^\star_\eta(y_t \mid x, y_{<t})\Big)\right].$$

LogLossBC. The offline baseline is log-loss behavior cloning on a fixed dataset of noisy expert rollouts [17]. Let $\mathcal{D}_\eta$ denote trajectories generated by rolling out the noisy expert $\pi^\star_\eta$. We train

$$\mathcal{L}_{\text{LogLossBC}}(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}_\eta}\left[-\sum_{t=1}^{|y|}\log \pi_\theta(y_t \mid x, y_{<t})\right].$$

Unlike the OPD variants, behavior cloning never queries the teacher on prefixes induced by the current student. Consequently, if an early token in an offline teacher trajectory is corrupted, the learner is also trained on all downstream prefixes induced by that corruption.

B.1.2 Gradient estimators.

We now spell out the stochastic gradient estimators used to optimize the empirical losses defined above. Let $s_t = (x, y_{<t})$ denote a visited prefix, and write $p_{\theta,t}(\cdot) = \pi_\theta(\cdot \mid s_t)$ and $q_t(\cdot) = \pi^\star_\eta(\cdot \mid s_t)$. Throughout, $p_{\theta,t}$ denotes the temperature-one student distribution whose KL is optimized. Rollout-temperature ablations only change the distribution used to collect prefixes. In particular, for rollout temperature τ, let $\rho_{\theta,\tau}$ denote the stopped prefix-collection policy, with

$$\rho_{\theta,0} = \bar{\pi}_\theta, \qquad \rho_{\theta,\tau}(\cdot \mid s_t) = \mathsf{softmax}(z_\theta(s_t)/\tau) \quad \text{for } \tau > 0.$$

All gradients below treat prefixes sampled from $\rho_{\theta,\tau}$ as fixed; the gradient update is taken only through the next-token distribution $p_{\theta,t}$ at those visited prefixes. The default settings are $\tau = 0$ for NAIL-F and NAIL-R and $\tau = 1$ for OPD-F and OPD-R, while the rollout-temperature ablations vary this prefix-collection temperature.

For $\widehat{\mathcal{L}}_{\text{NAIL-F}}(\theta)$, after drawing prefixes $y_{1:T} \sim \rho_{\theta,\tau}(\cdot \mid x)$ and independent teacher tokens $\tilde{y}_t \sim q_t$, we use

$$\widehat{g}^{\mathrm{F}}_\tau = -\sum_{t=1}^{T}\nabla_\theta \log p_{\theta,t}(\tilde{y}_t).$$

This is an unbiased stochastic gradient estimator for the stopped-prefix forward-KL objective because the entropy term of q_t is independent of θ. The methods differ only in how the prefixes are collected: $\tau = 0$ gives the default NAIL-F estimator with greedy prefixes, while $\tau = 1$ gives the default sampled-prefix OPD-F estimator. Other rollout temperatures use the same next-token estimator on prefixes sampled from $\rho_{\theta,\tau}$. The offline LogLossBC baseline is ordinary cross-entropy on realized noisy-expert trajectories; it has the same token-level gradient form, but its prefixes and labels come from an offline dataset rather than from student rollouts.

For $\widehat{\mathcal{L}}_{\text{NAIL-R}}(\theta)$, prefix collection and reverse-KL token sampling are separate. After drawing prefixes $y_{1:T} \sim \rho_{\theta,\tau}(\cdot \mid x)$, the implementation draws an auxiliary token from a stopped copy of the current student distribution at the visited prefix: $\hat{y}_t \sim \mathsf{sg}(p_{\theta,t})$, where $\mathsf{sg}(\cdot)$ denotes stop-gradient. Thus, the rollout temperature τ affects which prefixes are visited, but the auxiliary token for the reverse-KL estimator is sampled from $p_{\theta,t}$, not from the rollout distribution. The stopped-prefix score-function estimator is

$$\widehat{g}^{\mathrm{R}}_\tau = \sum_{t=1}^{T}\left(\mathsf{sg}(\log p_{\theta,t}(\hat{y}_t)) - \log q_t(\hat{y}_t)\right)\nabla_\theta \log p_{\theta,t}(\hat{y}_t), \qquad \hat{y}_t \sim \mathsf{sg}(p_{\theta,t}).$$

Equivalently, this is implemented by backpropagating through the surrogate

$$\widehat{\ell}^{\mathrm{R}}_t(\theta) = -\exp(\log p_{\theta,t}(\hat{y}_t) - \mathsf{sg}(\log p_{\theta,t}(\hat{y}_t)))\left(\log q_t(\hat{y}_t) - \mathsf{sg}(\log p_{\theta,t}(\hat{y}_t))\right).$$

The exponential factor is numerically equal to one at the sampled parameter value, but its numerator still carries gradient, so differentiating gives exactly the estimator above.

For $\widehat{\mathcal{L}}_{\text{OPD-R}}(\theta)$, as in standard OPD, the sampled rollout token itself is used as the reverse-KL sample. Therefore, if the rollout token is sampled from the current temperature-one student distribution, this coincides with the on-policy reverse-KL estimator above. If one instead reuses rollout tokens sampled at a different temperature, then the token distribution no longer matches $p_{\theta,t}$, and the resulting update should be viewed as a temperature-mismatched surrogate. Note that this mirrors the recipe described by Lu and Lab [41].

For the interpolated objective in Appendix B.3, where

$$\widehat{\mathcal{L}}_{\beta}(\theta) = (1-\beta)\widehat{\mathcal{L}}_{\text{NAIL-F}}(\theta) + \beta\widehat{\mathcal{L}}_{\text{NAIL-R}}(\theta),$$

we use the corresponding convex combination of the two stopped-prefix estimators,

$$\widehat{g}_{\beta} = (1-\beta)\widehat{g}_0^{\mathrm{F}} + \beta\widehat{g}_0^{\mathrm{R}}.$$

B.1.3 Synthetic Task: Modular Addition

We use the modular-addition task of Li et al. [38], denoted C_p. Given $p \in \mathbb{N}$, the vocabulary is $\{0, \ldots, p-1, =\}$. An input has the form $x = (x_1, \ldots, x_m, =)$, where each x_i is sampled independently from $\{0, \ldots, p-1\}$. The final answer is $f^*(x) = \sum_{i=1}^{m} x_i \bmod p$ and the chain of thought consists of the running partial sums $(\sum_{i=1}^{t} x_i \bmod p)_{t=1}^{m}$. Modular addition is not intended to be an inherently serial hardness instance; indeed, Li et al. [38] note that it is parallelizable and, in principle, can be solved by constant-depth transformers with sufficient precision. Nevertheless, their experiments show that low-depth transformers without CoT can perform near chance on C_7, while CoT substantially improves performance, especially at longer sequence lengths. We include it as a simple controlled setting in which the target CoT has a transparent step-by-step structure, allowing us to study how noisy intermediate feedback affects offline and online imitation-learning methods.

Training and Evaluation Details. We use the nanoGPT codebase [30] for all modular-addition experiments. For the tokenizer, we use task-specific integer token IDs from the aforementioned symbolic vocabulary. Unless otherwise stated, all runs use a depth-one transformer with 8 attention heads, embedding dimension 512, dropout 0, no bias terms, and block size set to the CoT sequence length for the corresponding m. We train with batch size 64 using Adam [32] with learning rate 10^{-5}, warmup for 2000 iterations, weight decay 0, $\beta_1 = 0.9$, $\beta_2 = 0.95$, gradient clipping at 1.0, and evaluation every 500 iterations.

We fix $p = 7$ and $m = 31$. All methods use the same prompt bank, containing 15 million training prompts and 5000 validation prompts. We train on a fixed 3 million-prompt subset in a fixed order. The clean expert is fixed across all runs and is synthetically modified using the corruption law described below. It was trained for 10,000 optimizer steps with batch size 64. The model reached perfect validation accuracy after roughly 2500 optimizer steps and stayed at this level, but we use the final checkpoint for all experiments.

We compare the five methods for two noise levels, $\eta \in \{0, 0.2\}$, and across three seeds. For each run seed, we use the same seed for student optimization and for the stochastic components of the noisy teacher law. For LogLossBC, the seed also determines the independently rendered noisy dataset. For the online methods, there is no pre-rendered trajectory; the same seed controls the online teacher sampling.

The noisy teacher is an absorbing instantiation of the state-dependent noise model in (1). The state is augmented with a binary flag indicating whether the prefix has already made a semantic error. In the unpoisoned state, at each target token the corruption distribution is

$$\nu(\cdot \mid x, y_{<t}) = \mathrm{Unif}(\{0, \ldots, p-1\}),$$

so that

$$\pi_{\eta}^{\star}(\cdot \mid x, y_{<t}) = (1-\eta)\pi^{\star}(\cdot \mid x, y_{<t}) + \eta\nu(\cdot \mid x, y_{<t}).$$

Here the uniform distribution includes the clean token. Since every modular-addition target token is a digit representing a running sum, every target token is treated as semantic. Once the sampled token differs from the clean running-sum token, the trajectory enters the poisoned state. In the poisoned state, both $\pi^{\star}$ and ν are defined to be uniform over $\{0, \ldots, p-1\}$, so all subsequent semantic feedback is independent of the clean computation while remaining syntactically valid.

This absorbing corruption law models a worst-case form of semantic error propagation in long CoT reasoning: after an early pivotal mistake, the suffix may remain syntactically plausible while providing minimal information about the clean continuation. This is motivated by recent work on pivotal or forking tokens in reasoning models, where a small number of tokens can strongly affect the success of the final completion [1, 58]. In our synthetic task, the running-sum tokens play this role, so corrupting one makes downstream semantic feedback uninformative about the correct computation. This pessimistic noise model provides a starting point for a controlled setting in which offline training on corrupted rollouts can suffer from the trajectory-level contamination predicted by our theory, and we leave the consideration of more nuanced noise models to future work.

For LogLossBC, training trajectories are rendered autoregressively from this noisy teacher law. Thus, in the offline setting, an example is fully informative only if no visible semantic corruption occurs. Under the idealized independent-trigger calculation, the probability of this event is $(1 - \eta + \eta/p)^m \approx 2.9 \times 10^{-3}$ in our setting.

For the online methods, the poisoned flag is inferred from the student-generated prefix: if a previous running-sum token in the student prefix differs from the clean target, then later teacher feedback is uniform over residues. Consequently, $\eta = 0$ has different semantics for offline and online runs under this law: offline rendering is clean at $\eta = 0$, while online feedback can still become uninformative after a student prefix error.

Online methods are trained for one pass over the fixed 3 million-prompt subset, giving $3{,}000{,}000/64 = 46{,}875$ iterations. LogLossBC is also trained for one pass over the corresponding rendered 3 million-example dataset. All methods are evaluated on the same clean validation prompt bank.

B.1.4 Mathematical Reasoning: GSM-8K

We evaluate on the GSM-8K test set [14] using zero-shot greedy decoding with max generation length 512 and seed 42. Each problem is placed in the Gemma instruction-tuned chat template with the following prefix.

```
Please reason step by step, and put your final answer within \boxed{}.
```

For example, an evaluation prompt takes the form

```
<bos><start_of_turn>user
Please reason step by step, and put your final answer within \boxed{}.

Janet's ducks lay 16 eggs per day. She eats three for breakfast every morning
and bakes muffins for her friends every day with four. She sells the remainder
at the farmers' market daily for $2 per fresh duck egg. How much in dollars
does she make every day at the farmers' market?<end_of_turn>
<start_of_turn>model
```

We extract the final answer in the `\boxed{}` and compute exact-match accuracy after standard answer normalization.

Training data. We construct the training prompt set from TinyGSM [40]. Specifically, we first filter out examples whose code answers contain more than 1024 characters, and then remove examples whose code answers are not executable. This filtering yields approximately 11M examples. We then sample a 400K-example subset that is used for all training runs.

Training details. For the LogLossBC baseline, we generate fixed teacher rollouts on the 400K TinyGSM prompts using GEMMA3-1B-IT̀We generate clean rollouts with temperature 1 in the low-noise setting and noisy rollouts with temperature 4 in the high-noise setting, both using random seed 42 and a maximum generation length of 512 new tokens. We measure the accuracy of these clean and noisy teacher rollouts on the 400K TinyGSM subset by executing the reference code answers and treating the execution outputs as ground truth. The clean teacher obtains 50.1% accuracy on the training subset, while the noisy teacher obtains 0% accuracy. We then supervised fine-tune the GEMMA3-270M-IT student separately on the clean and noisy teacher rollouts.

For the online reverse-KL variants, OPD-R and NAIL-R, we follow the recipe of Lu and Lab [41]. We generate students' next token using temperature-1 sampling for OPD-R and greedy decoding for NAIL-R. Along these student-generated trajectories, we compute the student and teacher log probabilities conditioned on the same prefixes, using either the clean temperature-1 teacher or the noisy temperature-4 teacher. The student is then updated using the corresponding importance-sampling loss.

For the forward-KL variants, OPD-F and NAIL-F, we instead query the teacher locally on learner-visited prefixes. At each such prefix, we sample the teacher's next token, using the temperature-1 teacher in the low-noise setting and the temperature-4 teacher in the high-noise setting, and treat the sampled token as a hard next-token label. The student is then updated using the standard next-token log-loss. Subsequent learner prefixes are generated with temperature-1 sampling for OPD-F and greedy decoding for NAIL-F.

All methods are trained on the same 400K TinyGSM prompts with a maximum generation length of 512 new tokens. For the two online methods, we follow the recommended recipe from Lu and Lab [41]. For the offline method, we tune the learning rate slightly in preliminary experiments, while keeping all other hyperparameters fixed. Specifically, unless otherwise stated, all methods are trained for one epoch using AdamW with default parameters, learning rate 1e-4, linear warmup followed by cosine decay, batch size 64, and bf16 precision. We use LoRA with rank 128 and $\alpha = 256$, applied to all modules. For each method, we run three random seeds, 42, 43, and 44.

B.2 Further Discussion of Empirical Results in Section 6

All configurations are run with three random seeds. In the plots, each curve shows the mean across seeds, and the shaded region corresponds to one standard deviation. We note that some shaded regions are not visible as the runs are nearly identical across seeds. In particular, for Modular Addition, the maximum standard deviation is less than 10^{-2} for LogLossBC and OPD-F in the low-noise setting, and for OPD-F, OPD-R, and LogLossBC in the high-noise setting (as the latter all fail to learn).

For clarity of presentation, the main text shows accuracy only over the first 1M expert trajectories, where the relevant separations between methods are most visible. In Figure 2, we provide the corresponding validation-loss curves over the full 3M-trajectory training horizon. These longer-horizon curves confirm the same qualitative picture: in the low noise setting, the NAIL variants rapidly drive validation loss close to zero, while OPD-F plateaus early; in the high noise setting, the gap becomes more pronounced, with NAIL-F and NAIL-R remaining substantially more stable than the offline and OPD baselines.

To expand on the discussion in Section 6, Figure 2 shows a sharp contrast between the clean and noisy regimes for the modular addition task. When $\eta = 0.2$, LogLossBC fails because an early corrupted CoT token makes the remaining suffix random, so most offline trajectories contain little usable signal. In contrast, our NAIL-F and NAIL-R reach perfect accuracy, consistent with the idea that querying on student-induced prefixes avoids imitating fully corrupted trajectories. The failure of sampled-rollout OPD further suggests that controlling the rollout distribution, here via greedy prefixes, is important for keeping teacher feedback informative.

When $\eta = 0.0$, since offline traces are clean, LogLossBC learns fastest, as predicted by Foster et al. [17]. Among online methods, NAIL-F learns much faster than OPD-F, suggesting that greedy rollouts help suppress sampling-induced noise that hinders access to uncorrupted teacher feedback. The gap between NAIL-F and the reverse-KL methods here points to the fact that teacher-sampled cross-entropy gives a direct positive signal for the correct next token, whereas reverse KL only scores student-sampled tokens and is therefore less directly corrective. Finally, when there is no expert noise, greedy rollouts offer little benefit for NAIL-R over OPD-R because both methods update using student-sampled tokens.

B.3 Further Experiments

We include two additional ablations on the modular addition task to better understand which parts of the online objective matter. In Appendix B.3.2, we fix greedy student rollouts and interpolate between the forward- and reverse-KL augmented-trajectory losses, testing whether performance depends on a single KL direction.

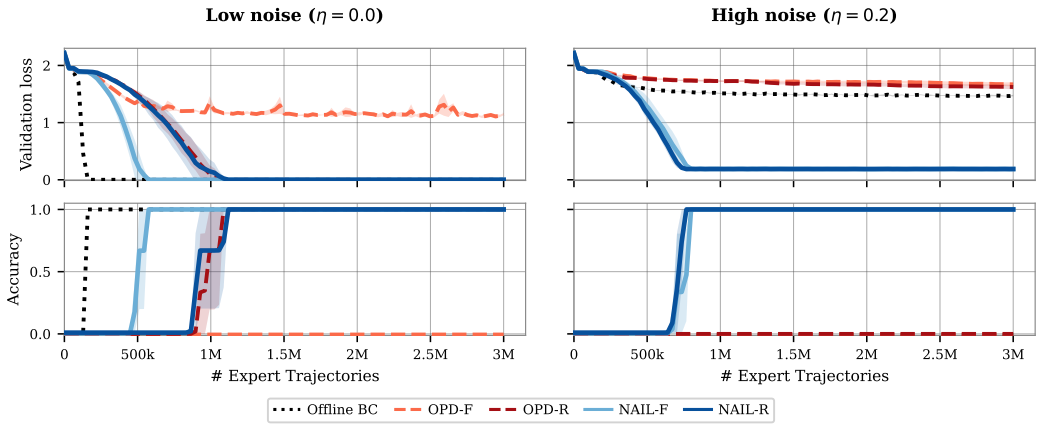

Figure 2: Full modular-addition results over 3M expert trajectories. **Top**: validation loss; **Bottom**: accuracy. Curves show the mean over three random seeds, with shaded regions indicating one standard deviation. In the low-noise setting ($\eta = 0$), the NAIL variants drive validation loss to zero and reach perfect accuracy, while OPD-F plateaus. In the high-noise setting ($\eta = 0.2$), the separation is more pronounced: NAIL-F and NAIL-R remain stable and reach perfect accuracy, whereas offline LogLossBC and the OPD baselines fail to solve the task.

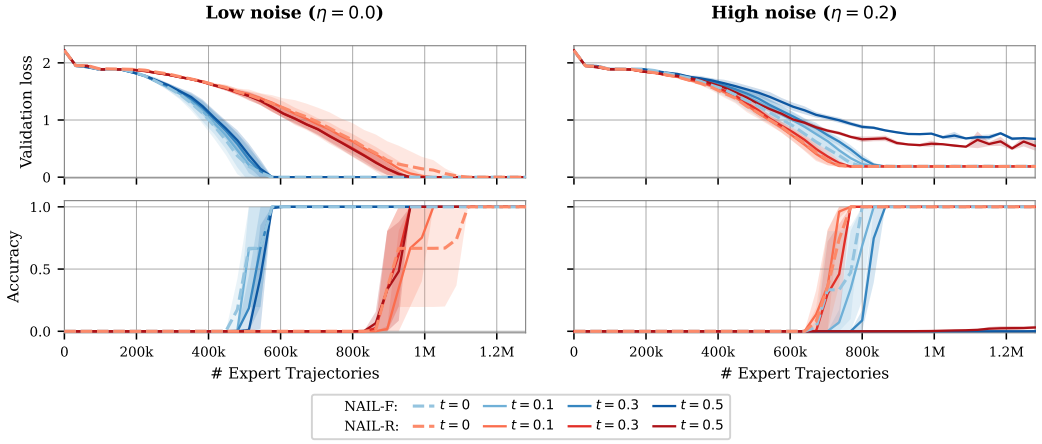

Figure 3: Ablation of student rollout temperature for NAIL-F and NAIL-R on Modular Addition. The parameter t controls the student sampling temperature used during training rollout. Curves show the mean over three random seeds, with shaded regions indicating one standard deviation. **Left:** low-noise setting, NAIL-F learns faster and is relatively insensitive to temperature, while NAIL-R is substantially slower. **Right:** high-noise setting, moderate student temperatures improve robustness for both objectives, with NAIL-F solving the task across temperatures and NAIL-R degrading at larger t.

B.3.1 Ablating student rollout temperature

We next ablate the temperature used when sampling student rollouts. In the main experiments, the student prefixes are generated *greedily*, i.e. temperature $t = 0$. Here, we instead sample from the student policy with various temperatures $t \in \{0.1, 0.3, 0.5\}$. We evaluate both NAIL-F and NAIL-R on Modular Addition under the same low- and high-noise settings as in the main body. All variants are trained using the same recipe as in Appendix B.1; for readability, Figure 3 shows only the first 1.25M expert-query trajectories.

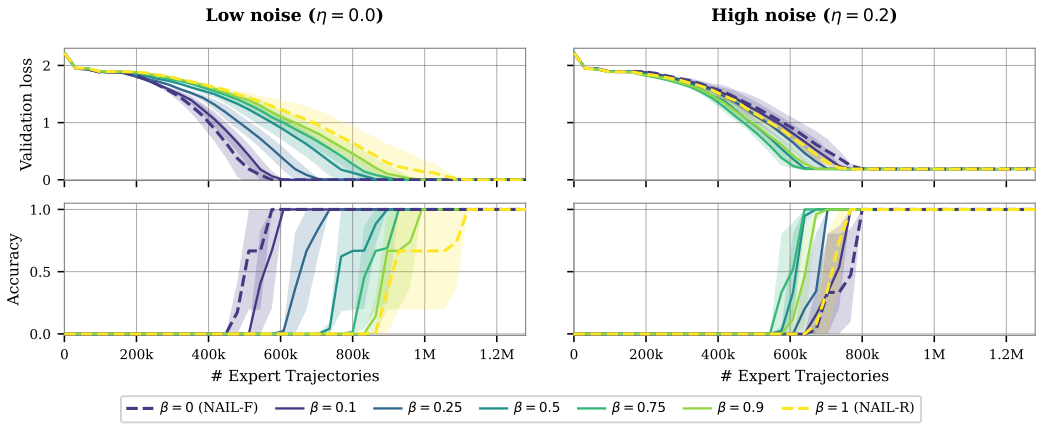

Figure 4: Interpolation between NAIL-F and NAIL-R on Modular Addition. The parameter β interpolates between the forward-KL ($\beta = 0$) and the reverse-KL ($\beta = 1$) losses. Curves show the mean over three random seeds, with shaded regions indicating one standard deviation. All interpolated variants eventually solve the task in both noise regimes, but the learning speed depends strongly on β. **Left:** in the low-noise setting, forward-KL-heavy objectives learn fastest. **Right:** in the high-noise setting, intermediate and reverse-KL-heavy objectives are competitive, suggesting that (i) the robustness of NAIL is primarily driven by querying the expert on learner-induced prefixes rather than by a single KL direction, and (ii) the best KL direction is task- and noise-dependent.

In the low-noise setting ($\eta = 0$), NAIL-F solves the task at roughly the same sample complexity across rollout temperatures, indicating that forward-KL training is fairly robust to this choice. In contrast, NAIL-R remains slower across temperatures, consistent with the main results in Figure 1. In the high-noise setting ($\eta = 0.2$), small to moderate rollout temperatures can still solve the task, but large temperature substantially hurts both NAIL-F and NAIL-R. This suggests that some stochasticity in student rollouts is tolerable, but excessive exploration can produce prefixes that are too noisy or off-distribution for effective expert querying.

B.3.2 Interpolating between NAIL-F and NAIL-R

To further probe the role of the KL direction, we fix greedy student rollouts and interpolate between the forward- and reverse-KL losses. For $\beta \in [0, 1]$, we minimize

$$J_\beta(\pi_\theta) = (1-\beta) \cdot \mathrm{D}_{\mathrm{KL}}\left(\mathbb{P}^{\bar{\pi}_\theta, \pi^\star_\eta} \middle\| \mathbb{P}^{\bar{\pi}_\theta, \pi_\theta}\right) + \beta \cdot \mathrm{D}_{\mathrm{KL}}\left(\mathbb{P}^{\bar{\pi}_\theta, \pi_\theta} \middle\| \mathbb{P}^{\bar{\pi}_\theta, \pi^\star_\eta}\right).$$

Thus $\beta = 0$ recovers NAIL-F, while $\beta = 1$ recovers NAIL-R. We train each variant for 3M expert-query trajectories using the same recipe as in Appendix B.1; for readability, Figure 4 shows only the first 1.25M trajectories. All interpolated variants eventually reach perfect accuracy in both the low-noise and high-noise regimes, but their learning dynamics differ.

When $\eta = 0$, performance changes smoothly from that of NAIL-F to NAIL-R as β increases, matching the behavior in Figure 1: forward-KL-heavy objectives learn fastest. When $\eta = 0.2$, however, intermediate and reverse-KL-heavy mixtures learn slightly faster than either endpoint, with larger values of β reaching perfect accuracy roughly 100K trajectories earlier. These results suggest that the optimal KL mixture is task- and noise-dependent. A more systematic study of when to prefer forward KL, reverse KL, or mixtures of the two is an interesting direction for future work.

C Technical Tools

In this section, we recall some technical tools that are used throughout the proofs. We begin in Appendix C.1, where we recall some basic definitions and properties of KL divergence and Hellinger distance. We proceed in Appendix C.2 by stating some classical results on the performance of maximum likelihood estimators. In Appendix C.3, we recall some key results from the theory of

imitation learning that relate regret to Hellinger distance between trajectory distributions. Finally, in Appendix C.4 we recall some standard results from online learning that are used in the analysis of our online algorithms.

C.1 Information Theory

In this section we recall some basic results from information theory that are used throughout the paper. For a more complete introduction to the topic, see Polyanskiy and Wu [43].

We first recall the definitions of KL divergence.

Definition 6. Let P, Q be two distributions over the same space $\mathcal{X}$. The KL divergence between P and Q is defined as

$$\mathrm{D}_{\mathrm{KL}}\left(P\|Q\right) = \mathbb{E}_{X\sim P}\left[\log \frac{dP}{dQ}(X)\right]$$

with $\mathrm{D}_{\mathrm{KL}}\left(P\|Q\right) = \infty$ if P is not absolutely continuous with respect to Q.

We use the following classical properties of KL divergence repeatedly throughout the paper; see Polyanskiy and Wu [43] for details.

Proposition 4. *Let P, Q be distributions over the same space $\mathcal{X}$. Then it holds that $\mathrm{D}_{\mathrm{KL}}\left(P\|Q\right) \geq 0$ with equality if and only if $P = Q$. Moreover, $(P, Q) \mapsto \mathrm{D}_{\mathrm{KL}}\left(P\|Q\right)$ is jointly convex in its arguments, i.e. for any $\lambda \in [0,1]$ and any distributions P_1, P_2, Q_1, Q_2,*

$$\mathrm{D}_{\mathrm{KL}}\left(\lambda P_1 + (1-\lambda)P_2 \| \lambda Q_1 + (1-\lambda)Q_2\right) \leq \lambda \cdot \mathrm{D}_{\mathrm{KL}}\left(P_1\|Q_1\right) + (1-\lambda)\cdot \mathrm{D}_{\mathrm{KL}}\left(P_2\|Q_2\right).$$

Furthermore, if $P, Q \in \Delta(\mathcal{X}_1 \times \cdots \mathcal{X}_n)$, then the KL divergence satisfies a chain rule: if $P_{X_1,\ldots,X_n}$ and $Q_{X_1,\ldots,X_n}$ are the distributions of $(X_1,\ldots,X_n)$ under P and Q respectively, then

$$\mathrm{D}_{\mathrm{KL}}\left(P_{X_1,\ldots,X_n}\|Q_{X_1,\ldots,X_n}\right) = \sum_{i=1}^{n} \mathbb{E}_{X_1,\ldots,X_{i-1}\sim P}\left[\mathrm{D}_{\mathrm{KL}}\left(P_{X_i|X_1,\ldots,X_{i-1}}\middle\|Q_{X_i|X_1,\ldots,X_{i-1}}\right)\right].$$

While KL divergence is a fundamental notion of distance between distributions, it is infinite when the two distributions are not absolutely continuous with respect to each other, which can be problematic in many settings in IL. We thus also consider the *Hellinger distance*, defined as follows.

Definition 7. Let P, Q be two distributions over the same space $\mathcal{X}$. The Hellinger distance between P and Q is defined as

$$\begin{aligned}
\mathrm{D}_{\mathrm{H}^2}\left(P, Q\right) &= 1 - \mathbb{E}_{X\sim P}\left[\sqrt{\frac{dQ}{dP}(X)}\right] \\
&= 1 - \mathbb{E}_{X\sim Q}\left[\sqrt{\frac{dP}{dQ}(X)}\right] \\
&= \frac{1}{2}\cdot\int_{\mathcal{X}}\left(\sqrt{\frac{dP}{d\mu}(x)} - \sqrt{\frac{dQ}{d\mu}(x)}\right)^2 d\mu(x),
\end{aligned}$$

where μ is any measure such that P and Q are absolutely continuous with respect to μ, e.g. $\mu = {}^{(P+Q)}/_{2}$.

While Hellinger distance is also nonnegative and equal to zero if and only if the two distributions are equal, it is a weaker notion of distance than KL divergence, as it is always bounded by 1. In addition, it satisfies a Pinsker-type inequality that relates it to KL divergence.

Proposition 5 (Pinsker's inequality)**.** *Let P, Q be two distributions over the same space $\mathcal{X}$. Then,*

$$\mathrm{D}_{\mathrm{H}^2}\left(P, Q\right) \leq \mathrm{D}_{\mathrm{KL}}\left(P\|Q\right).$$

The Hellinger distance is also intimately related to the total variation distance $\mathrm{TV}\left(P, Q\right) = \sup_{A\subseteq\mathcal{X}} |P(A) - Q(A)|$ up to a quadratic factor.

Proposition 6. *Let P, Q be two distributions over the same space $\mathcal{X}$. Then,*

$$\mathrm{D}_{\mathrm{H}^2}(P, Q) \leq \mathrm{TV}(P, Q) \leq \sqrt{2 \cdot \mathrm{D}_{\mathrm{H}^2}(P, Q)}.$$

Much like KL divergence, Hellinger is jointly convex in its arguments. In contradistinction to KL divergence, however, Hellinger distance does not satisfy a chain rule, but it does satisfy two weaker properties that will be sufficient for our purposes.

Proposition 7. *Let P, Q be two distributions over $\mathcal{X}_1 \times \cdots \times \mathcal{X}_n$. Let $B_{n+1}(x_{1:n}) = 1$ and for all $i \leq n$, let for some common dominating measure μ,*

$$B_i(x_{1:i-1}) = \int_{\mathcal{X}_i} \sqrt{\frac{dP_{X_i|X_{1:i-1}=x_{1:i-1}}}{d\mu}(x_i) \cdot \frac{dQ_{X_i|X_{1:i-1}=x_{1:i-1}}}{d\mu}(x_i)} \cdot B_{i+1}(x_{1:i}) d\mu(x_i).$$

Then,

$$\mathrm{D}_{\mathrm{H}^2}(P, Q) = 1 - B_1.$$

In particular, if P, Q are product distributions, i.e. $P = P_1 \times \cdots P_n$ and $Q = Q_1 \times \cdots Q_n$, then

$$\mathrm{D}_{\mathrm{H}^2}(P, Q) = 1 - \prod_{i=1}^{n} (1 - \mathrm{D}_{\mathrm{H}^2}(P_i, Q_i)) \leq \sum_{i=1}^{n} \mathrm{D}_{\mathrm{H}^2}(P_i, Q_i).$$

The second property is more similar in form to the chain rule for KL divergence, but it is an inequality rather than an equality.

Proposition 8 (Lemma D.2 Foster et al. [18])**.** *Let P, Q be two distributions over $\mathcal{X}_1 \times \cdots \times \mathcal{X}_n$. Then,*

$$\mathrm{D}_{\mathrm{H}^2}(P, Q) \leq \sum_{h=1}^{H} \mathbb{E}_P \left[\mathrm{D}_{\mathrm{H}^2}\left(P_{X_h|X_{1:h-1}}, Q_{X_h|X_{1:h-1}} \right) \right].$$

C.2 Maximum Likelihood Estimation

Maximum Likelihood Estimation (MLE) is a fundamental statistical estimation technique that returns the density in a given class that maximizes the likelihood of the observed data. More precisely if Π' is a class of conditional distributions $\mathcal{S} \mapsto \Delta(\mathcal{A})$ and $(s^{(i)}, a^{(i)})_{i=1}^n$ are samples, then the MLE $\hat{\pi}$ is defined as

$$\hat{\pi} \in \operatorname*{argmax}_{\pi \in \Pi'} \sum_{i=1}^{n} \log \pi(a^{(i)}|s^{(i)}) = \operatorname*{argmin}_{\pi \in \Pi'} \sum_{i=1}^{n} -\log \pi(a^{(i)}|s^{(i)}). \tag{6}$$

While the following result is due to Geer [20], Zhang [64], we state the version from Foster et al. [17] that is most relevant to our setting.

Theorem 10 (Proposition B.1 from [17])**.** *Let Π' be a finite class of conditional distributions $\mathcal{S} \mapsto \Delta(\mathcal{A})$ and let $\pi^\star \in \Pi'$. Let $(s^{(i)}, a^{(i)})_{i=1}^n$ be i.i.d. samples from $\mathbb{P}^{\pi^\star}$, where $\mathbb{P}^{\pi^\star}$ is the distribution over $\mathcal{S} \times \mathcal{A}$ induced by sampling $s \sim \rho$ and $a \sim \pi^\star(\cdot|s)$. If $\hat{\pi}$ is the MLE in (6), then with probability at least $1 - \delta$,*

$$\mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\pi^\star}, \mathbb{P}^{\hat{\pi}} \right) \leq 12 \cdot \frac{\log\left(|\Pi'|/\delta \right)}{n}.$$

An immediate consequence of Theorem 10 is that Behavior Cloning with the logarithmic loss achieves Hellinger distance that scales in a horizon-free manner with the number of samples.

Corollary 1 (Proposition 2.1 from [17])**.** *Let Π be a finite class of policies $\mathcal{S} \times [H] \to \Delta(\mathcal{A})$ and let $\pi^\star \in \Pi$. Let $\tau^{(i)} = (s_1^{(i)}, a_1^{(i)}, \ldots, s_H^{(i)}, a_H^{(i)})$ be i.i.d. trajectories from $\mathbb{P}^{\pi^\star}$ and let*

$$\hat{\pi} \in \operatorname*{argmin}_{\pi \in \Pi} \sum_{i=1}^{n} \sum_{h=1}^{H} -\log \pi(a_h^{(i)}|s_h^{(i)}).$$

Then with probability at least $1-\delta$,

$$\mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\hat{\pi}}, \mathbb{P}^{\pi^\star}\right) \lesssim \frac{\log\left(|\Pi|/\delta\right)}{n}.$$

In particular,

$$\mathbb{E}\left[\mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\hat{\pi}}, \mathbb{P}^{\pi^\star}\right)\right] \lesssim \frac{\log\left(n \cdot |\Pi|\right)}{n}.$$

Indeed, Corollary 1 follows from observing that the policy does not affect the transition densities and thus

$$\hat{\pi} = \operatorname*{argmin}_{\pi\in\Pi} \sum_{i=1}^{n}\sum_{h=1}^{H} -\log\left(\pi\left(a_h^{(i)} \mid s_h^{(i)}\right)\right) = \operatorname*{argmin}_{\pi\in\Pi} \sum_{i=1}^{n}\sum_{h=1}^{H} -\log\left(\pi\left(a_h^{(i)} \mid s_h^{(i)}\right) \cdot P_h\left(s_{h+1}^{(i)} \mid a_h^{(i)}, s_h^{(i)}\right)\right)$$
$$= \operatorname*{argmin}_{\pi\in\Pi} \sum_{i=1}^{n} -\log \mathbb{P}^{\pi}(\tau^{(i)}).$$

The second statement is immediate from the first and the fact that Hellinger distance is bounded by 1.

C.3 Imitation Learning

Recent work has revealed the fundamental importance of the Hellinger distance in the theory of interactive decision making [15, 16, 45]. In this section we recall the key fact that, at least in a minimax sense, Imitation Learning is essentially equivalent to learning the trajectory distribution of the expert in Hellinger distance. Following Foster et al. [17], we consider the deterministic and stochastic cases separately. In the deterministic case, we have the following result.

Theorem 11 (Theorem 2.1 from [17])**.** *Let* $\pi^\star$ *be a* deterministic *policy and let* $\hat{\pi}$ *be an arbitrary (possibly stochastic) policy. Then,*

$$J(\pi^\star) - J(\hat{\pi}) \leq 4R \cdot \mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\hat{\pi}}, \mathbb{P}^{\pi^\star}\right),$$

where R *is such that*

$$0 \leq \sum_{h=1}^{H} r(s_h, a_h) \leq R$$

for all trajectories $\tau = (s_1, a_1, \ldots, s_H, a_H)$.

In the stochastic case, we recall the following result, which is a consequence of the more general results in Foster et al. [17].

Theorem 12 (Theorem 3.1 and Proposition 3.1 from Foster et al. [17])**.** *Let* $\pi^\star$ *be a (possibly stochastic) policy and let* $\hat{\pi}$ *be any policy. Then,*

$$J(\pi^\star) - J(\hat{\pi}) \lesssim \sqrt{R^2 \cdot \mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\pi^\star}, \mathbb{P}^{\hat{\pi}}\right)} + R\log\left(\frac{R}{\mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\hat{\pi}}, \mathbb{P}^{\pi^\star}\right)}\right) \cdot \mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\hat{\pi}}, \mathbb{P}^{\pi^\star}\right).$$

Note that Foster et al. [17] proves a tighter result, replacing the R^2 under the square root with

$$\sigma_{\pi^\star}^2 = \sum_{h=1}^{H} \mathbb{E}^{\pi^\star}\left[\left(\mathbb{E}\left[Q_h^{\pi^\star}(s_h, a_h) | s_h\right] - Q_h^{\pi^\star}(s_h, a_h)\right)^2\right],$$

where $Q^{\pi^\star}$ is the Q-function of $\pi^\star$ (cf. e.g. Sutton et al. [56]). The second cited result above, Foster et al. [17, Proposition 3.1], controls $\sigma_{\pi^\star}^2$ by R^2 leading to the version stated above. While $\sigma_{\pi^\star}^2$ is a significantly more refined quantity that can be much smaller than R^2 in many settings of interest, such as when $\pi^\star$ is near-deterministic, we use the version stated above for simplicity. We leave to future work the interesting problem of understanding the precise role of $\sigma_{\pi^\star}^2$ in the noisy expert setting and whether it can be used to obtain improved guarantees in certain regimes.

Note that Foster et al. [17, Theorem G.3] shows that the above reductions are tight up to constants in the sense that for any MDP and pair of policies, there exist reward functions that achieve the reverse inequalities up to constant factors. Thus, in a minimax sense, IL is essentially equivalent to learning the trajectory distribution of the expert in Hellinger distance. More formally, the result states the following.

Algorithm 2 Exponential Weights

Require: Number of rounds n, loss function $\ell : \Pi \times \mathcal{Y} \to \mathbb{R}$, learning rate $\lambda > 0$.

Set $w_1 = \mathrm{Unif}(\Pi)$.
for $t = 1$ to n **do**
  Observe y_t and suffer loss $\mathbb{E}_{\pi \sim w_t}[\ell(\pi, y_t)]$.
  Update $w_{t+1}(\pi) \propto w_t(\pi) \cdot e^{-\lambda \cdot \ell(\pi, y_t)}$.
end for

Theorem 13 (Theorem G.3 from Foster et al. [17]). *Let M be an MDP. Then for any pair of policies $\pi^\star, \hat{\pi}$ and any $\sigma > 0$ there exists a reward function r such that $\sigma^2_{\pi^\star} \leq \sigma^2$ and*

$$J(\pi^\star) - J(\hat{\pi}) \gtrsim \sqrt{\sigma^2 \cdot \mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\pi^\star}, \mathbb{P}^{\hat{\pi}}\right)}.$$

Moreover, there exists a reward function r such that

$$J(\pi^\star) - J(\hat{\pi}) \gtrsim R \cdot \mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\pi^\star}, \mathbb{P}^{\hat{\pi}}\right)$$

and the same conclusion applies even if we assume $\pi^\star$ is deterministic.

By Theorem 13, for any of our lower bounds on regret, it suffices to construct instances where learning the trajectory distribution of the expert in Hellinger distance is hard, which is what we do.

C.4 Online Learning

Several of our results rest on the use of online learning algorithms, and in particular the exponential weights algorithm, to learn policies in an online fashion. In this section we recall the key definitions and results from online learning that are used throughout the paper. For a more complete introduction to the topic, see Cesa-Bianchi and Lugosi [8].

We are only concerned with online learning over finite classes of experts in this work. For a finite class Π, the online learning problem proceeds in rounds $t \in [n]$ as follows. The learner at the beginning is informed of a loss function $\ell : \Pi \times \mathcal{Y} \to \mathbb{R}$ and must choose a distribution $w_t \in \Delta(\Pi)$ over the experts. Then, an outcome $y_t \in \mathcal{Y}$ is revealed and the learner suffers loss $\mathbb{E}_{\pi \sim w_t}[\ell(\pi, y_t)]$. The goal of the learner is to minimize regret, defined as

$$\mathrm{Reg}_n = \sum_{t=1}^{n} \mathbb{E}_{\pi \sim w_t}[\ell(\pi, y_t)] - \min_{\pi \in \Pi} \sum_{t=1}^{n} \ell(\pi, y_t).$$

The exponential weights algorithm is a simple and classical online learning algorithm that achieves optimal regret guarantees in the adversarial setting, given in Algorithm 2.

We make use of the following definition.

Definition 8. A loss $\ell : \Pi \times \mathcal{Y} \to \mathbb{R}$ is β-exp-concave if for all $y \in \mathcal{Y}$, the function $\pi \mapsto e^{-\beta \cdot \ell(\pi, y)}$ is concave.

We recall the following result about the regret of the exponential weights algorithm when the loss is exp-concave.

Proposition 9 (Proposition 3.1 from Cesa-Bianchi and Lugosi [8]). *If ℓ is β-exp-concave, then the exponential weights algorithm with learning rate $\lambda = \beta$ achieves regret*

$$\mathrm{Reg}_n \leq \frac{\log(|\Pi|)}{\beta}.$$

We emphasize that this regret is independent of the precise choice of $y_1, \ldots, y_n$, and thus holds even if the outcomes are chosen by an adversary that observes the learner's distribution w_t at each round. Finally, we will recall a more general definition that is similar to exp-concavity but allows for a better regret bound.

Definition 9. A loss $\ell : \Pi \times \mathcal{Y} \to \mathbb{R}$ is β-mixable if for all $y \in \mathcal{Y}$ and all distributions $w \in \Delta(\Pi)$, there exists $\pi_w \in \Pi$ such that

$$e^{-\beta \cdot \ell(\pi_w, y)} \geq \mathbb{E}_{\pi \sim w}\left[e^{-\beta \cdot \ell(\pi, y)}\right].$$

Mixability is a classical notion in online learning and a more complete discussion of it can be found in Cesa-Bianchi and Lugosi [8]. We recall the following guarantee for the exponential weights algorithm when the loss is mixable.

Proposition 10 (Proposition 3.2 from Cesa-Bianchi and Lugosi [8])**.** *If ℓ is β-mixable, then the exponential weights algorithm with learning rate $\lambda = \beta$ achieves regret*

$$\mathrm{Reg}_n \leq \frac{\log(|\Pi|)}{\beta}.$$

D Proofs from Section 3

In this appendix, we prove the two results in Section 3 involving offline imitation learning with a noisy expert. We begin by proving the upper bound that scales exponentially in horizon in Appendix D.1 before proving that this exponential dependence is necessary in the worst case in Appendix D.2. We conclude in Appendix D.3 by showing that any offline IL algorithm must suffer from this exponential dependence, making offline IL fundamentally intractable in the noisy expert setting.

D.1 Proof of Theorem 6

We first state a slightly tighter version of the main theorem, which recovers Theorem 6. We first conclude the proof under κ-domination and then show how to get a weaker guarantee in the absence of κ-domination.

Theorem 14. *Let π, π' be policies that κ-dominate the corruption ν. For any $0 \leq \eta < 1$, it holds that*

$$\mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\pi}, \mathbb{P}^{\pi'}\right) \leq \frac{2(1+\eta(2\kappa-1))}{(1-\eta)^{H+2}} \cdot \mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\pi_\eta}, \mathbb{P}^{\pi'_\eta}\right).$$

Noting that $\eta < 1$ immediately shows that the first statement of Theorem 6 follows from Theorem 14. To prove the latter, we fix policies π, π' and introduce some notation in order to help with the proof. First, for any state $s \in \mathcal{S}$, define

$$\mathbb{P}^{\pi}_h(s) = \mathbb{P}^{\pi}\left(a_h, \tau_{h+1:H} | s_h = s\right)$$

to be the conditional distribution of the action a_h and the future trajectory of states and actions conditioned on the event that $s_h = s$. We then define

$$D_h(s) = \mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\pi}_h(s), \mathbb{P}^{\pi'}_h(s)\right) \quad \text{and} \quad D^{\eta}_h(s) = \mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\pi_\eta}_h(s), \mathbb{P}^{\pi'_\eta}_h(s)\right).$$

We have the following recursion.

Lemma 1. *Let π, π' be any two policies. Then $D_{H+1}(s) = 0$ for all $s \in \mathcal{S}$ and for any $s \in \mathcal{S}$ and $h \leq H$, it holds that*

$$D_h(s) = \mathrm{D}_{\mathrm{H}^2}\left(\pi_h(\cdot|s), \pi'_h(\cdot|s)\right) + \sum_{a \in \mathcal{A}} \sqrt{\pi_h(a|s) \cdot \pi'_h(a|s)} \cdot \int D_{h+1}(s') dP_h(s'|s,a).$$

Proof. We use Proposition 7. Indeed, we compute

$$\begin{aligned} 1 - D_h(s) &= \mathbb{E}^{\pi}\left[\sqrt{\frac{\pi'_h(a_h|s)}{\pi_h(a_h|s)}}\left(1 - D_{h+1}(s_{h+1})\right)\right] \\ &= \sum_{a \in \mathcal{A}} \sqrt{\pi_h(a|s) \cdot \pi'_h(a|s)} \cdot \mathbb{E}_{s_{h+1} \sim P_h(\cdot|s,a)}\left[1 - D_{h+1}(s_{h+1})\right]. \end{aligned}$$

The result follows by rearranging and using the definition of the Hellinger squared distance. □

Lemma 1 reduces the problem of bounding the contraction in Hellinger distance of trajectory distributions to bounding the contraction on a per-step basis. We thus prove that in the case of κ-dominated corruption distributions ν, we can control this contraction at each time step h.

Lemma 2. *Let π, π' be arbitrary policies and suppose that ν is κ-dominated, i.e., for all $1 \le h \le H$, all $s \in \mathcal{S}$ and $a \in \operatorname{supp}(\pi_h(\cdot|s)) \cup \operatorname{supp}(\pi'_h(\cdot|s))$, it holds that*

$$\nu_h(a|s) \le \kappa \left(\pi_h(a|s) + \pi'_h(a|s)\right).$$

Then for any $0 \le \eta < 1$, any state $s \in \mathcal{S}$ and time $1 \le h \le H$, it holds that

$$\mathrm{D}_{\mathrm{H}^2}\left(\pi_h(\cdot|s), \pi'_h(\cdot|s)\right) \le \frac{2\left(1 + \eta\left(2\kappa - 1\right)\right)}{(1-\eta)^2} \cdot \mathrm{D}_{\mathrm{H}^2}\left(\pi_{\eta,h}(\cdot|s), \pi'_{\eta,h}(\cdot|s)\right).$$

Proof. For the sake of notational simplicity, we will suppress the h in this proof; in addition, because we have fixed a state s, we will write $\pi(a)$ for $\pi_h(a|s)$. We observe that for any $a \in \operatorname{supp}(\pi) \cup \operatorname{supp}(\pi')$, it holds that

$$\begin{aligned}
\pi_\eta(a) + \pi'_\eta(a) &= (1-\eta)\left(\pi(a) + \pi'(a)\right) + 2\eta \cdot \nu(a) \\
&\le (1-\eta)\left(\pi(a) + \pi'(a)\right) + 2\eta\kappa \cdot \left(\pi(a) + \pi'(a)\right) \\
&= \left(1 + \eta(2\kappa - 1)\right)\left(\pi(a) + \pi'(a)\right).
\end{aligned}$$

Thus,

$$\left(\sqrt{\pi_\eta(a)} + \sqrt{\pi'_\eta(a)}\right)^2 \le 2\left(\pi_\eta(a) + \pi'_\eta(a)\right) \le 2\left(1 + \eta(2\kappa-1)\right)\left(\sqrt{\pi(a)} + \sqrt{\pi'(a)}\right)^2,$$

where we used the fact that $a + b \le \left(\sqrt{a} + \sqrt{b}\right)^2 \le 2(a+b)$ for nonnegative a, b. Plugging into the definition of Hellinger distance, we see that

$$\begin{aligned}
\mathrm{D}_{\mathrm{H}^2}\left(\pi_\eta, \pi'_\eta\right) &= \frac{1}{2}\sum_{a \in \mathcal{A}} \left(\sqrt{\pi_\eta(a)} - \sqrt{\pi'_\eta}(a)\right)^2 \\
&= \frac{1}{2}\sum_{a \in \mathcal{A}} \frac{\left(\pi_\eta(a) - \pi'_\eta(a)\right)^2}{\left(\sqrt{\pi_\eta(a)} + \sqrt{\pi'_\eta(a)}\right)^2} \\
&\ge \frac{(1-\eta)^2}{2(1 + \eta(2\kappa - 1))} \cdot \frac{1}{2}\sum_{a \in \mathcal{A}} \frac{\left(\pi(a) - \pi'(a)\right)^2}{\left(\sqrt{\pi(a)} + \sqrt{\pi'(a)}\right)^2} \\
&= \frac{(1-\eta)^2}{2(1 + \eta(2\kappa - 1))} \cdot \mathrm{D}_{\mathrm{H}^2}\left(\pi, \pi'\right).
\end{aligned}$$

The result follows. □

Remark 1. Note that it is precisely in Lemma 2 that κ-domination is used and in the proof one can understand the necessity of such an assumption. Indeed, the problem is precisely that the map $\eta \mapsto \sqrt{\eta}$ is not Lipschitz near $\eta = 0$. For a simple example of what can go wrong, let

$$\pi = \text{Bernoulli}(0) = \delta_0, \quad \pi' = \text{Bernoulli}(\varepsilon), \quad \text{and} \quad \nu = \text{Bernoulli}\left(1/2\right).$$

An elementary computation then shows that

$$\mathrm{D}_{\mathrm{H}^2}\left(\pi, \pi'\right) \asymp \varepsilon \quad \text{but} \quad \mathrm{D}_{\mathrm{H}^2}\left(\pi_\eta, \pi'_\eta\right) \asymp \varepsilon^2,$$

for $\eta > 0$. Thus, in order to prevent such a quadratic blowup, we need to ensure that ν does not put too much mass on actions that receive small, but positive, probability under π or π', which is precisely what κ-domination ensures.

Remark 2. Note that even with κ-domination, the $(1-\eta)^2$ dependence in the comparison is real. Indeed, suppose that $\pi^\star = \delta_{a_1}$ and $\hat{\pi} = \delta_{a_2}$ for some $a_1 \ne a_2$, and $\nu = 1/2\left(\pi^\star + \hat{\pi}\right)$. Then we have κ-domination with $\kappa = 1$. On the other hand, we have $\mathrm{D}_{\mathrm{H}^2}\left(\pi^\star, \hat{\pi}\right) = 1$, whereas

$$\mathrm{D}_{\mathrm{H}^2}\left(\pi^\star_\eta, \hat{\pi}_\eta\right) = 1 - \sqrt{\eta(2-\eta)} \asymp \frac{(1-\eta)^2}{2}.$$

Thus the $(1-\eta)^2$ dependence is tight up to constant factors.

We are now ready to conclude the proof of the result.

Proof of Theorem 14. We prove the following claim by reverse induction from $H, \dots, 1$: it holds for any $s \in \mathcal{S}$ that

$$D_h^\eta(s) \geq (1-\eta)^{H+1-h} \cdot \frac{(1-\eta)^2}{2\left(1+\eta(2\kappa-1)\right)} \cdot D_h(s). \tag{7}$$

The case $h = H$ follows immediately from Lemma 2. We thus suppose that (7) holds for $h+1$. By Lemma 1, it holds that

$$\begin{aligned}
D_h^\eta(s) &= \mathrm{D}_{\mathrm{H}^2}\left(\pi_{\eta,h}(\cdot|s), \pi'_{\eta,h}(\cdot|s)\right) + \sum_{a\in\mathcal{A}} \sqrt{\pi_{\eta,h}(a|s)\cdot\pi'_{\eta,h}(a|s)} \cdot \int D_{h+1}^\eta(s')dP_h(s'|s,a) \\
&\geq \frac{(1-\eta)^2}{2\left(1+\eta(2\kappa-1)\right)} \cdot \mathrm{D}_{\mathrm{H}^2}\left(\pi_h(\cdot|s), \pi'_h(\cdot|s)\right) + \sum_{a\in\mathcal{A}} \sqrt{\pi_{\eta,h}(a|s)\cdot\pi'_{\eta,h}(a|s)} \cdot \int D_{h+1}^\eta(s')dP_h(s'|s,a) \\
&\geq \frac{(1-\eta)^2}{2\left(1+\eta(2\kappa-1)\right)} \cdot \mathrm{D}_{\mathrm{H}^2}\left(\pi_h(\cdot|s), \pi'_h(\cdot|s)\right) + (1-\eta)\cdot\sum_{a\in\mathcal{A}} \sqrt{\pi_h(a|s)\cdot\pi'_h(a|s)} \cdot \int D_{h+1}^\eta(s')dP_h(s'|s,a) \\
&\geq \frac{(1-\eta)^2}{2\left(1+\eta(2\kappa-1)\right)} \cdot \mathrm{D}_{\mathrm{H}^2}\left(\pi_h(\cdot|s), \pi'_h(\cdot|s)\right) \\
&\quad + (1-\eta)\cdot(1-\eta)^{H-h} \cdot \frac{(1-\eta)^2}{2\left(1+\eta(2\kappa-1)\right)} \cdot \sum_{a\in\mathcal{A}} \sqrt{\pi_h(a|s)\cdot\pi'_h(a|s)} \cdot \int D_{h+1}(s')dP_h(s'|s,a) \\
&\geq \frac{(1-\eta)^{H+3-h}}{2\left(1+\eta(2\kappa-1)\right)} \cdot \left(\mathrm{D}_{\mathrm{H}^2}\left(\pi_h(\cdot|s), \pi'_h(\cdot|s)\right) + \sum_{a\in\mathcal{A}} \sqrt{\pi_h(a|s)\cdot\pi'_h(a|s)} \cdot \int D_{h+1}(s')dP_h(s'|s,a)\right) \\
&= \frac{(1-\eta)^{H+3-h}}{2\left(1+\eta(2\kappa-1)\right)} \cdot D_h(s),
\end{aligned}$$

where the first inequality used Lemma 2, the second inequality used the fact that

$$\begin{aligned}
\sqrt{\pi_{\eta,h}(a|s)\cdot\pi'_{\eta,h}(a|s)} &= \sqrt{\left((1-\eta)\pi_h(a|s) + \eta\nu_h(a|s)\right)\left((1-\eta)\pi'_h(a|s) + \eta\nu_h(a|s)\right)} \\
&\geq (1-\eta)\cdot\sqrt{\pi_h(a|s)\cdot\pi'_h(a|s)},
\end{aligned}$$

the third inequality used the inductive hypothesis, and the final inequality again used Lemma 1. The result follows immediately. □

Note that in the case that π, π' are both *deterministic* policies, then any ν is κ-dominated with $\kappa = 1$ and thus

$$\mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^\pi, \mathbb{P}^{\pi'}\right) \lesssim (1-\eta)^{-H-2} \cdot \mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\pi_\eta}, \mathbb{P}^{\pi'_\eta}\right) \leq \frac{e^{\eta H/1-\eta}}{(1-\eta)^2} \cdot \mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\pi_\eta}, \mathbb{P}^{\pi'_\eta}\right).$$

We now show how to get a weaker guarantee in the absence of κ-domination. We restate the result with constants made explicit now.

Proposition 11. *Let M be a horizon H MDP and let π, π' be two policies. For any choice of ν and any $0 < \eta < 1$, it holds that*

$$\mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^\pi, \mathbb{P}^{\pi'}\right) \leq \sqrt{2\cdot\frac{(1-\eta)^{-H}-1}{\eta(1-\eta)} \cdot \mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\pi_\eta}, \mathbb{P}^{\pi'_\eta}\right)}.$$

We first prove the following general result on the contraction of Hellinger distance under arbitrary corruptions.

Lemma 3. *Let P, Q be two distributions over a common space $\mathcal{X}$ and let ν be an arbitrary distribution over $\mathcal{X}$. For any $0 < \eta < 1$, letting $P_\eta = (1-\eta)P + \eta\nu$ and $Q_\eta = (1-\eta)Q + \eta\nu$, it holds that*

$$\mathrm{D}_{\mathrm{H}^2}\left(P, Q\right) \leq \frac{\sqrt{2}}{1-\eta} \cdot \sqrt{\mathrm{D}_{\mathrm{H}^2}\left(P_\eta, Q_\eta\right)}.$$

Proof. By Proposition 6, it holds that

$$\begin{aligned}\sqrt{2\mathrm{D}_{\mathrm{H}^2}\left(P_\eta, Q_\eta\right)} &\geq \mathrm{TV}\left(P_\eta, Q_\eta\right) \\ &= \sup_{0\leq f\leq 1} \left|\mathbb{E}_{P_\eta}[f] - \mathbb{E}_{Q_\eta}[f]\right| \\ &= (1-\eta)\cdot \sup_{0\leq f\leq 1} \left|\mathbb{E}_{P}[f] - \mathbb{E}_{Q}[f]\right| \\ &= (1-\eta)\cdot \mathrm{TV}\left(P, Q\right) \\ &\geq (1-\eta)\cdot \mathrm{D}_{\mathrm{H}^2}\left(P, Q\right),\end{aligned}$$

where the second equality comes from the linearity of expectation and the final inequality again comes from Proposition 6. □

We can now prove the proposition.

Proof of Proposition 11. We will apply backward induction and Lemma 1. Indeed, we use the identical notation as that used in the proof thereof. We suppose that there are constants C_h for $h = H, \ldots, 1$ satisfying $C_H = 2^{-1}\cdot(1-\eta)^2$ and

$$\frac{1}{C_h} = \frac{2}{(1-\eta)^2} + \frac{1}{(1-\eta)\cdot C_{h+1}}$$

such that

$$D_h^\eta(s) \geq C_h \cdot D_h(s)^2 \quad \text{for all } s\in\mathcal{S}.$$

By Lemma 3, the statement holds for $h = H$. Now, by Lemma 1, it holds that

$$\begin{aligned}D_h^\eta(s) &= \mathrm{D}_{\mathrm{H}^2}\left(\pi_{\eta,h}(\cdot|s), \pi'_{\eta,h}(\cdot|s)\right) + \sum_{a\in\mathcal{A}} \sqrt{\pi_{\eta,h}(a|s)\cdot\pi'_{\eta,h}(a|s)}\cdot\int D_{h+1}^\eta(s')dP_h(s'|s,a) \\ &\geq 2^{-1}\cdot(1-\eta)^2\cdot\mathrm{D}_{\mathrm{H}^2}\left(\pi_h(\cdot|s), \pi'_h(\cdot|s)\right)^2 \\ &\quad + (1-\eta)\cdot\sum_{a\in\mathcal{A}}\sqrt{\pi_h(a|s)\cdot\pi'_h(a|s)}\cdot\int D_{h+1}^\eta(s')dP_h(s'|s,a) \\ &\geq 2^{-1}\cdot(1-\eta)^2\cdot\mathrm{D}_{\mathrm{H}^2}\left(\pi_h(\cdot|s), \pi'_h(\cdot|s)\right)^2 \\ &\quad + (1-\eta)\cdot\sum_{a\in\mathcal{A}}\sqrt{\pi_h(a|s)\cdot\pi'_h(a|s)}\cdot\int C_{h+1}D_{h+1}(s')^2dP_h(s'|s,a) \\ &\geq 2^{-1}\cdot(1-\eta)^2\cdot\mathrm{D}_{\mathrm{H}^2}\left(\pi_h(\cdot|s), \pi'_h(\cdot|s)\right)^2 \\ &\quad + C_{h+1}(1-\eta)\cdot\left(\sum_{a\in\mathcal{A}}\sqrt{\pi_h(a|s)\cdot\pi'_h(a|s)}\cdot\int D_{h+1}(s')dP_h(s'|s,a)\right)^2,\end{aligned}$$

where the second inequality follows from the inductive hypothesis and the final inequality follows from Jensen's inequality and the fact that $\sum_a \sqrt{\pi_h(a|s)\cdot\pi'_h(a|s)} \leq 1$. By AM-GM, it holds that for $a, b, x, y \geq 0$,

$$a\cdot x^2 + b\cdot y^2 \geq \frac{1}{\frac{1}{a}+\frac{1}{b}}\cdot(x+y)^2.$$

Thus, applying this to the final expression in the above display, we see that

$$\begin{aligned}D_h^\eta(s) &\geq C_h\cdot\left(\mathrm{D}_{\mathrm{H}^2}\left(\pi_h(\cdot|s), \pi'_h(\cdot|s)\right) + \sum_{a\in\mathcal{A}}\sqrt{\pi_h(a|s)\cdot\pi'_h(a|s)}\cdot\int D_{h+1}(s')dP_h(s'|s,a)\right)^2 \\ &= C_h\cdot D_h(s)^2,\end{aligned}$$

where the equality follows from Lemma 1. Thus, the inductive step is complete.

It remains to bound C_h itself. Letting $c_h = C_h^{-1}$, we see that

$$c_H = 2(1-\eta)^{-2}, \qquad c_h = 2(1-\eta)^{-2} + (1-\eta)^{-1} \cdot c_{h+1}$$

for $h < H$. Thus by induction, it holds that

$$c_1 = 2 \cdot \frac{(1-\eta)^{-H} - 1}{\eta(1-\eta)}.$$

The result follows. □

Finally, we note that Theorem 6 follows immediately from Theorem 14 and Proposition 11, concluding the proof.

D.2 Proof of Proposition 1

We will prove three lower bounds: one for arbitrary κ, one for $\kappa = 1$ with a deterministic expert, and one that holds absent κ-domination. We begin by stating the common construction for the first two, before proving these first two results separately.

We will take an MDP that has $H+1$ states $s_1, \dots, s_H$ and an absorbing state $\perp$. We will suppose the action space $\mathcal{A} = \{a_1, a_2, a_3\}$. Let the transition functions for $h < H$ be

$$P_h(s'|s,a) = \begin{cases} \delta_{s_{h+1}} & a = a_1 \text{ and } s = s_h \\ \delta_\perp & a \in \{a_2, a_3\} \text{ or } s = \perp \end{cases}.$$

In other words, $\perp$ is an absorbing state that is reached by taking a 'wrong' action at any state, otherwise we deterministically transition to the next state.

We now state the formal proposition for the case of $\kappa = 1$ and deterministic experts.

Proposition 12. *For any $H \geq 1$ and $0 \leq \eta < 1$, there exists an MDP, a policy class Π of size 2, and a corruption distribution ν such that both policies in Π are deterministic, but*

$$\mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\pi^\star}, \mathbb{P}^{\hat{\pi}}\right) = 1 \quad \textit{but} \quad \mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\pi^\star_\eta}, \mathbb{P}^{\hat{\pi}_\eta}\right) \leq (1-\eta)^{H+1}.$$

Proof. We will consider the MDP described above with policy class $\Pi = \{\pi^\star, \hat{\pi}\}$ of size 2 such that $\hat{\pi}(\cdot|s_h) = \pi^\star(\cdot|s_h) = \delta_{a_1}$ (an atom on a_1) for $h < H$, $\pi^\star(s_H) = \delta_{a_1}$, $\hat{\pi}(s_H) = \delta_{a_2}$, and both policies take the same action at $\perp$. Finally, suppose that

$$\nu(\cdot|s_H) = \frac{\delta_{a_1} + \delta_{a_2}}{2} \quad \text{and} \quad \nu(\cdot|s_h) = \delta_{a_3} \text{ for } h < H.$$

We claim that for any $0 \leq \eta < 1$, the result of the proposition holds for this construction. Note that the $\eta = 0$ case is trivial, so suppose that $\eta > 0$. We now observe that $\mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\pi^\star}, \mathbb{P}^{\hat{\pi}}\right) = 1$ because the policies deterministically differ in the final time step, which is reached with probability 1. Thus we focus on upper bounding $\mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\pi^\star_\eta}, \mathbb{P}^{\hat{\pi}_\eta}\right)$.

Note first that for $\pi \in \{\pi^\star, \hat{\pi}\}$, with probability at least $1 - (1-\eta)^{H-1}$, it holds that the trajectory under π will transition to $\perp$ before the final time step; moreover, conditional on this event, both corrupted policies induce the same distribution. On the complementary event of no contamination up to step H, it holds that

$$\pi^\star_\eta = \left(1 - \frac{\eta}{2}\right) \cdot \delta_{a_1} + \frac{\eta}{2} \cdot \delta_{a_2} \quad \text{and} \quad \hat{\pi}_\eta = \frac{\eta}{2} \cdot \delta_{a_1} + \left(1 - \frac{\eta}{2}\right) \cdot \delta_{a_2}.$$

Thus,

$$\mathrm{D}_{\mathrm{H}^2}\left(\pi^\star_\eta(\cdot|s_H), \hat{\pi}_\eta(\cdot|s_H)\right) = 1 - \sqrt{\eta(2-\eta)} = \frac{(1-\eta)^2}{1 + \sqrt{\eta(2-\eta)}} \leq (1-\eta)^2.$$

Combining these observations, we see that

$$\mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\pi^\star_\eta}, \mathbb{P}^{\hat{\pi}_\eta}\right) \leq (1-\eta)^{H-1} \cdot (1-\eta)^2 = (1-\eta)^{H+1}.$$

The result follows. □

We now state the formal proposition for the case of arbitrary κ.

Proposition 13. *For any $H \geq 1$, $\kappa \geq 1$, and $0 \leq \eta < 1$, there exists an MDP, a policy class Π of size 2, and a corruption distribution ν such that both policies in Π κ-dominate ν, but*

$$\mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\pi}, \mathbb{P}^{\pi'}\right) \geq 4\eta \cdot \kappa \cdot (1-\eta)^{-H-1} \cdot \mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\pi_\eta}, \mathbb{P}^{\pi'_\eta}\right).$$

Proof. We will consider the MDP described above but now let for $h < H$,

$$\pi_h(\cdot|s_h) = \pi'_h(\cdot|s_h) = (1-\varepsilon) \cdot \delta_{a_1} + \varepsilon \cdot \delta_{a_3}$$

and

$$\pi_H(\cdot|s_H) = (1-\varepsilon) \cdot \delta_{a_1} + \varepsilon \cdot \delta_{a_2} \quad \text{and} \quad \pi'_H(\cdot|s_H) = (1-\varepsilon) \cdot \delta_{a_1} + \varepsilon \cdot \delta_{a_3}.$$

Let both policies take action a_2 on $\perp$. Finally, let

$$\nu_h(\cdot|s_h) = \delta_{a_3} \text{ for } h < H \quad \text{and} \quad \nu_H(\cdot|s_H) = \frac{\delta_{a_2} + \delta_{a_3}}{2},$$

and let $\nu(\cdot|\perp) = \delta_{a_2}$. Suppose that $\varepsilon = 1/2\kappa$. We first note that ν is κ-dominated. Indeed, for $h < H$ this is immediate for action a_1, and for a_3,

$$\nu_h(a_3|s_h) = 1 = \kappa \cdot 2\varepsilon = \kappa \cdot \left(\pi_h(a_3|s_h) + \pi'_h(a_3|s_h)\right).$$

At s_H, for a_2 we have

$$\nu_H(a_2|s_H) = \frac{1}{2} = \kappa \cdot \varepsilon = \kappa \cdot \left(\pi_H(a_2|s_H) + \pi'_H(a_2|s_H)\right),$$

and similarly for a_3. Thus ν is κ-dominated.

We now compute the Hellinger distance between the *clean* trajectory distributions. Let $\mathcal{E}$ denote the event that the trajectory reaches s_H. Note that the two laws are identical on $\mathcal{E}^c$. On the other hand,

$$\mathbb{P}^{\pi}(\mathcal{E}) = \mathbb{P}^{\pi'}(\mathcal{E}) = (1-\varepsilon)^{H-1} \quad \text{and} \quad \mathrm{D}_{\mathrm{H}^2}\left(\pi_H(\cdot|s_H), \pi'_H(\cdot|s_H)\right) = \varepsilon$$

Thus,

$$\mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\pi}, \mathbb{P}^{\pi'}\right) = (1-\varepsilon)^{H-1} \cdot \varepsilon.$$

We now compute the Hellinger distance between the corrupted trajectory distributions. For $h < H$ the policies coincide and the probability of reaching s_H is $(1-\eta)^{H-1} \cdot (1-\varepsilon)^{H-1}$. At the final time step,

$$\mathrm{D}_{\mathrm{H}^2}\left(\pi_{\eta,H}(\cdot|s_H), \pi'_{\eta,H}(\cdot|s_H)\right) = \left(\sqrt{\eta/2 + (1-\eta)\varepsilon} - \sqrt{\eta/2}\right)^2 \leq \frac{(1-\eta)^2\varepsilon^2}{2\eta}.$$

Thus,

$$\begin{aligned}
\mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\pi_\eta}, \mathbb{P}^{\pi'_\eta}\right) &\leq \left((1-\eta)(1-\varepsilon)\right)^{H-1} \cdot \frac{(1-\eta)^2 \cdot \varepsilon^2}{2\eta} \\
&\leq \frac{(1-\eta)^{H+1}\varepsilon}{2\eta} \cdot \mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\pi}, \mathbb{P}^{\pi'}\right) \\
&= \frac{(1-\eta)^{H+1}}{4\eta \cdot \kappa} \cdot \mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\pi}, \mathbb{P}^{\pi'}\right).
\end{aligned}$$

The result follows. □

We now provide a lower bound that holds absent κ-domination.

Proposition 14. *For any $H \geq 1$ and $0 < \eta < 1$, there exists a horizon H MDP with 3 actions, policies $\pi^\star, \hat{\pi}$, and a corruption ν such that*

$$\mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\pi^\star}, \mathbb{P}^{\hat{\pi}}\right) \gtrsim \sqrt{\frac{(1-\eta)^{-H} - 1}{1-\eta} \cdot \mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\pi^\star_\eta}, \mathbb{P}^{\hat{\pi}_\eta}\right)}.$$

Proof. Let M have $H+1$ states $s_1, \dots, s_H$ and $\perp$, where

$$P_h(\cdot|s_h, a_2) = \delta_{s_{h+1}} \quad \text{and} \quad P_h(\cdot|s_h, a_1) = P_h(\cdot|s_h, a_3) = P_h(\cdot|\perp, a) = \delta_\perp.$$

Let $t < 1/4$, let

$$S = \sum_{h=0}^{H-1} (1-\eta)^{-h} = \frac{(1-\eta)\left((1-\eta)^{-H} - 1\right)}{\eta} \quad \text{and} \quad \lambda = t/S.$$

Let

$$u_h = \lambda(1-\eta)^{-h+1}.$$

Note that $u_h \leq 1/4$ for every h. Now, let

$$\pi^\star_h(\cdot|s_h) = u_h \cdot \delta_{a_1} + (1-u_h) \cdot \delta_{a_2} \quad \text{and} \quad \hat{\pi}_h(\cdot|s_h) = u_h \cdot \delta_{a_3} + (1-u_h) \cdot \delta_{a_2};$$

let $\pi^\star, \hat{\pi}$ agree on $\perp$ and define

$$\nu_h(\cdot|s_h) = \frac{1}{2} \cdot \delta_{a_1} + \frac{1}{2} \cdot \delta_{a_3} \quad \text{and} \quad \nu_h(\cdot|\perp) = \delta_{a_1}.$$

We will let

$$D_h = \mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}_h^{\pi^\star}(s_h), \mathbb{P}_h^{\hat{\pi}}(s_h)\right) \quad \text{and} \quad D_h^\eta = \mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}_h^{\pi^\star_\eta}(s_h), \mathbb{P}_h^{\hat{\pi}_\eta}(s_h)\right).$$

Applying Lemma 1, we see that

$$D_h = u_h + (1-u_h) \cdot D_{h+1}.$$

Thus, by induction, we have that

$$1 - e^{-t} \leq D_1 = 1 - \prod_{h=1}^{H}(1-u_h) \leq t.$$

Because $t \leq 1/4$, it thus holds that $t/2 \leq D_1 \leq t$. On the other hand, a direct computation shows that

$$\mathrm{D}_{\mathrm{H}^2}\left(\pi^\star_{\eta,h}(\cdot|s_h), \hat{\pi}_{\eta,h}(\cdot|s_h)\right) = 1 - (1-\eta)(1-u_h) - \sqrt{\eta^2 + 2\eta(1-\eta)u_h} \leq \frac{(1-\eta)^2}{2\eta} \cdot u_h^2.$$

By Lemma 1 again we see that

$$\begin{aligned} D_h^\eta &= 1 - (1-\eta)(1-u_h) - \sqrt{\eta^2 + 2\eta(1-\eta)u_h} + (1-\eta)(1-u_h) \cdot D_{h+1}^\eta \\ &\leq \frac{(1-\eta)^2}{2\eta} \cdot u_h^2 + (1-\eta)(1-u_h) \cdot D_{h+1}^\eta. \end{aligned}$$

Thus, by induction, we have that

$$D_1^\eta \leq \frac{(1-\eta)^2}{2\eta} \cdot \sum_{h=1}^{H}(1-\eta)^{h-1} \cdot u_h^2 = \frac{(1-\eta)^2}{2\eta} \cdot \cdot \lambda^2 S.$$

Substituting in the definition of λ and S, we have that

$$D_1^\eta \leq \frac{1-\eta}{2\left((1-\eta)^{-H} - 1\right)} \cdot t^2 \leq \frac{2(1-\eta)}{(1-\eta)^{-H} - 1} \cdot D_1^2.$$

The result follows immediately. □

We now prove the main lower bound.

Proof of Proposition 1. This follows from combining Propositions 12 to 14. □

D.3 Implications for Offline Imitation Learning

We conclude this appendix by demonstrating that the exponential dependence on horizon in Theorem 6 must appear in any offline IL algorithm through a similar construction to the one used in Appendix D.2.

Proposition 15. *Let $\kappa \geq 1$ and $0 \leq \eta < 1$. For any $H \geq 2$ and*

$$\varepsilon < \frac{(1 - 1/2\kappa)^{H-1}}{128\kappa},$$

there exists an MDP, a policy class Π of size 2, and a corruption distribution ν such that both policies in Π κ-dominate ν, but any offline IL algorithm achieving $\mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\pi^\star}, \mathbb{P}^{\hat{\pi}}\right) \leq \varepsilon$ requires

$$n \gtrsim \frac{\eta \cdot \kappa}{(1-\eta)^{H+1} \cdot \varepsilon}$$

samples from the corrupted expert.

Proof. We use the identical construction as the proof in Proposition 13, except we suppose that

$$\pi_H(\cdot|s_H) = (1-u) \cdot \delta_{a_2} + u \cdot \delta_{a_1}, \quad \pi'_H(\cdot|s_H) = (1-u) \cdot \delta_{a_2} + u \cdot \delta_{a_3},$$

and

$$\nu_H(\cdot|s_H) = \kappa \cdot u \cdot \delta_{a_1} + (1 - 2\kappa \cdot u) \cdot \delta_{a_2} + \kappa \cdot u \cdot \delta_{a_3},$$

for some $u \leq 1/2\kappa$. This construction is clearly κ-dominated. Moreover,

$$\mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\pi}, \mathbb{P}^{\pi'}\right) = (1 - 1/2\kappa)^{H-1} \cdot u \quad \text{and} \quad \mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\pi_\eta}, \mathbb{P}^{\pi'_\eta}\right) \leq (1-\eta)^{H-1} \cdot (1 - 1/2\kappa)^{H-1} \cdot u \cdot \frac{(1-\eta)^2}{4\eta\kappa}.$$

Choosing

$$u = \frac{64 \cdot \varepsilon}{(1 - 1/2\kappa)^{H-1}},$$

which allows $u \leq 1/2\kappa$ by the assumption on ε, we see that

$$\mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\pi}, \mathbb{P}^{\pi'}\right) = 64 \cdot \varepsilon \quad \text{and} \quad \mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\pi_\eta}, \mathbb{P}^{\pi'_\eta}\right) \leq (1-\eta)^{H+1} \cdot \frac{16 \cdot \varepsilon}{\eta \cdot \kappa}.$$

A standard two point argument concludes the proof. □

E Proofs from Section 4

In this appendix, we prove the results related to online Imitation Learning stated in the main body. We begin in Appendix E.1 with the proof of Theorem 7, before continuing to prove the upper bound on NAIL in Appendix E.3, and concluding with a proof of the lower bound in Appendix E.4.

E.1 Proof of Theorem 7

We first state a slightly tighter version of the result, albeit with a more complicated expression in the upper bound. We break this into two separate results, one using forward KL and one using reverse KL, which together imply Theorem 7. First, under κ-domination, we have the following result.

Theorem 15. *Let π, π' be any two policies that κ-dominate the corruption ν. Then, for any $0 \leq \eta < 1$ it holds that*

$$\mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\pi}, \mathbb{P}^{\pi'}\right) \leq \frac{2(1 + \eta(2\kappa - 1))}{(1-\eta)^2} \cdot \left(\mathrm{D}_{\mathrm{KL}}\left(\mathbb{P}^{\pi', \pi_\eta} \middle\| \mathbb{P}^{\pi', \pi'_\eta}\right) \wedge \mathrm{D}_{\mathrm{KL}}\left(\mathbb{P}^{\pi', \pi'_\eta} \middle\| \mathbb{P}^{\pi', \pi_\eta}\right)\right).$$

Note that the first conclusion of Theorem 7 can be recovered immediately.

Proof of Theorem 15. We make use of the subadditivity of Hellinger squared divergence (Proposition 8). Indeed, it holds that

$$\mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\pi}, \mathbb{P}^{\pi'}\right) \leq \mathbb{E}^{\pi'}\left[\sum_{h=1}^{H} \mathrm{D}_{\mathrm{H}^2}\left(\pi(\cdot|s_h), \pi'(\cdot|s_h)\right)\right].$$

Now, applying Lemma 2, we have that for any s_h,

$$\mathrm{D}_{\mathrm{H}^2}\left(\pi(\cdot|s_h), \pi'(\cdot|s_h)\right) \leq \frac{2(1+\eta(2\kappa-1))}{(1-\eta)^2} \cdot \mathrm{D}_{\mathrm{H}^2}\left(\pi_\eta(\cdot|s_h), \pi'_\eta(\cdot|s_h)\right).$$

Combining this fact with the preceding display and applying Proposition 5, we have

$$\begin{aligned}\mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\pi}, \mathbb{P}^{\pi'}\right) &\leq \frac{2(1+\eta(2\kappa-1))}{(1-\eta)^2} \cdot \mathbb{E}^{\pi'}\left[\sum_{h=1}^{H} \mathrm{D}_{\mathrm{H}^2}\left(\pi_{\eta,h}(\cdot|s_h), \pi'_{\eta,h}(\cdot|s_h)\right)\right] \\ &\leq \frac{2(1+\eta(2\kappa-1))}{(1-\eta)^2} \cdot \mathbb{E}^{\pi'}\left[\sum_{h=1}^{H} \mathrm{D}_{\mathrm{KL}}\left(\pi_{\eta,h}(\cdot|s_h)\middle\|\pi'_{\eta,h}(\cdot|s_h)\right)\right].\end{aligned} \tag{8}$$

We may now apply the Chain rule for KL divergence (Proposition 4) to observe that

$$\mathbb{E}^{\pi'}\left[\sum_{h=1}^{H} \mathrm{D}_{\mathrm{KL}}\left(\pi_{\eta,h}(\cdot|s_h)\middle\|\pi'_{\eta,h}(\cdot|s_h)\right)\right] = \mathrm{D}_{\mathrm{KL}}\left(\mathbb{P}^{\pi',\pi_\eta}\middle\|\mathbb{P}^{\pi',\pi'_\eta}\right).$$

The first result follows. The second follows by the same argument by observing that the Hellinger distance is symmetric in (8). □

We now provide a similar comparison between the clean and noisy trajectory distributions absent κ-domination.

Proposition 16. *Let $\pi^\star, \hat{\pi}$ be two policies and let ν be an arbitrary corruption distribution. Then for any $0 \leq \eta < 1$ it holds that*

$$\mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\pi^\star}, \mathbb{P}^{\hat{\pi}}\right) \leq \frac{1}{1-\eta} \cdot \sqrt{2H \cdot \mathrm{D}_{\mathrm{KL}}\left(\mathbb{P}^{\hat{\pi},\pi^\star_\eta}\middle\|\mathbb{P}^{\hat{\pi},\hat{\pi}_\eta}\right) \wedge \mathrm{D}_{\mathrm{KL}}\left(\mathbb{P}^{\hat{\pi},\hat{\pi}_\eta}\middle\|\mathbb{P}^{\hat{\pi},\pi^\star_\eta}\right)}.$$

Proof. As in the proof of Theorem 7, we use the subadditivity of the Hellinger distance (Proposition 8) to get that

$$\begin{aligned}\mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\pi^\star}, \mathbb{P}^{\hat{\pi}}\right) &\leq \mathbb{E}^{\hat{\pi}}\left[\sum_{h=1}^{H} \mathrm{D}_{\mathrm{H}^2}\left(\pi^\star_h(\cdot|s_h), \hat{\pi}_h(\cdot|s_h)\right)\right] \\ &\leq \frac{\sqrt{2}}{1-\eta} \cdot \mathbb{E}^{\hat{\pi}}\left[\sum_{h=1}^{H} \sqrt{\mathrm{D}_{\mathrm{H}^2}\left(\pi^\star_{\eta,h}(\cdot|s_h), \hat{\pi}_{\eta,h}(\cdot|s_h)\right)}\right] \\ &\leq \frac{\sqrt{2H}}{1-\eta} \cdot \sqrt{\mathbb{E}^{\hat{\pi}}\left[\sum_{h=1}^{H} \mathrm{D}_{\mathrm{H}^2}\left(\pi^\star_{\eta,h}(\cdot|s_h), \hat{\pi}_{\eta,h}(\cdot|s_h)\right)\right]} \\ &\leq \frac{\sqrt{2H}}{1-\eta} \cdot \sqrt{\mathrm{D}_{\mathrm{KL}}\left(\mathbb{P}^{\hat{\pi},\pi^\star_\eta}\middle\|\mathbb{P}^{\hat{\pi},\hat{\pi}_\eta}\right)},\end{aligned}$$

where the second inequality comes from Lemma 3, the third inequality comes from Cauchy-Schwarz, and the final inequality comes from applying Pinsker's inequality (Proposition 5) and the chain rule for KL divergence (Proposition 4). The first claimed bound follows. The second comes from the identical argument but using the symmetry of the Hellinger distance to apply Pinsker's inequality in the reverse direction. The result follows. □

Finally, we note that Theorem 7 follows immediately from Theorem 15 and Proposition 16, concluding the proof of the main result.

E.2 Necessity of Horizon Dependence in the Absence of κ-Domination

We now show that the polynomial dependence in horizon that appears in Proposition 16 is necessary with the following lower bound. In particular, Proposition 16 is tight up to a polynomial dependence on $(1-\eta)$.

Proposition 17. *For any $H \geq 1$ and $0 < \eta < 1$, there exists a horizon H MDP with 2 actions, policies $\pi^\star, \hat{\pi}$, and corruption distribution ν such that*

$$\mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\pi^\star}, \mathbb{P}^{\hat{\pi}}\right) \geq 8^{-1} \cdot \sqrt{\frac{H \cdot \eta}{(1-\eta)} \cdot \mathrm{D}_{\mathrm{KL}}\left(\mathbb{P}^{\hat{\pi}, \pi^\star_\eta} \middle\| \mathbb{P}^{\hat{\pi}, \hat{\pi}_\eta}\right)}.$$

Proof. Let the MDP M have two actions, $\mathcal{A} = \{0, 1\}$ and H steps such that s_h transitions to s_{h+1} deterministically, independent of the action. Let

$$\pi^\star_h(0|s_h) = 1, \quad \hat{\pi}_h(0|s_h) = 1 - \varepsilon, \quad \text{and} \quad \nu_h(0|s_h) = 0$$

for every h. Thus actions are independent across time. We thus compute

$$\mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\pi^\star}, \mathbb{P}^{\hat{\pi}}\right) = 1 - (1-\varepsilon)^{H/2}$$

and thus, if $\varepsilon \leq 1/4H$, it holds that

$$\frac{H\varepsilon}{4} \leq \mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\pi^\star}, \mathbb{P}^{\hat{\pi}}\right) \leq H\varepsilon.$$

On the other hand, it holds that

$$\begin{aligned}
\mathrm{D}_{\mathrm{KL}}\left(\mathbb{P}^{\hat{\pi}, \pi^\star_\eta} \middle\| \mathbb{P}^{\hat{\pi}, \hat{\pi}_\eta}\right) &= H \cdot \mathrm{D}_{\mathrm{KL}}\left(\pi^\star_\eta(\cdot|s) \middle\| \hat{\pi}_\eta(\cdot|s)\right) \\
&= H \cdot \left((1-\eta) \cdot \log\left(\frac{1}{1-\varepsilon}\right) + \eta \cdot \log\left(\frac{\eta}{\eta + (1-\eta)\varepsilon}\right)\right).
\end{aligned}$$

Using the fact that $\log(1-x) \geq -x - x^2$ and $\log(1+x) \geq x - x^2/2$ for $x \in (0,1)$, we get that if $\varepsilon \leq \eta/e(1-\eta)$, then

$$\mathrm{D}_{\mathrm{KL}}\left(\mathbb{P}^{\hat{\pi}, \pi^\star_\eta} \middle\| \mathbb{P}^{\hat{\pi}, \hat{\pi}_\eta}\right) \leq \frac{3}{2} \cdot H \cdot \frac{1-\eta}{\eta} \cdot \varepsilon^2.$$

Thus for sufficiently small ε, the claimed relation holds. □

E.3 Proof of Theorem 8

The main content of the proof is to show the following result, which is that the on-policy KL divergence $\mathbb{E}\left[\mathrm{D}_{\mathrm{KL}}\left(\mathbb{P}^{\hat{\pi}, \pi^\star_\eta} \middle\| \mathbb{P}^{\hat{\pi}, \hat{\pi}_\eta}\right)\right]$ for $\hat{\pi}$ returned by Algorithm 1 is small. The result will then follow from Theorem 7. Indeed, we have the following result.

Lemma 4. *Let Π be a finite policy class, ν be an arbitrary corruption distribution, and $0 \leq \eta < 1$ be a corruption level. Let $\hat{\pi}$ denote the policy returned by Algorithm 1. Then it holds that*

$$\mathbb{E}\left[\mathrm{D}_{\mathrm{KL}}\left(\mathbb{P}^{\hat{\pi}, \pi^\star_\eta} \middle\| \mathbb{P}^{\hat{\pi}, \hat{\pi}_\eta}\right)\right] \leq \frac{H \cdot \log(|\Pi|)}{n}.$$

Proof. Let $\mu_t = \sum_{\pi \in \Pi} w_t(\pi) \cdot \pi$ denote the mixture policy defined in Algorithm 1. We first claim that for *any* sequence of trajectories $\tau^{(t)}, \tau^{(t)'}$, it holds that

$$\sum_{t=1}^{n} \sum_{h=1}^{H} \log\left(\frac{\pi^\star_{\eta,h}(a_h^{(t)'}|s_h^{(t)})}{\mu^{(t)}_{\eta,h}(a_h^{(t)'}|s_h^{(t)})}\right) \leq H \cdot \log(|\Pi|). \tag{9}$$

To see this, let

$$\ell(\pi, \tau') = -\sum_{h=1}^{H} \log\left(\pi_{\eta,h}(a'_h|s_h)\right)$$

denote a loss function. We claim that ℓ is $1/H$-mixable (Definition 9) and that Algorithm 1 amounts to running the exponential weights algorithm (Algorithm 2) with respect to ℓ over the policy class

$$\Pi_\eta = \{\pi_\eta | \pi \in \Pi\}.$$

Indeed, fix τ' and let $w \in \Delta(\Pi)$ and

$$\mu_w = \sum_{\pi \in \Pi} w(\pi) \cdot \pi.$$

Then, we note that $(\mu_w)_\eta = \sum_{\pi \in \Pi} w(\pi) \cdot \pi_\eta$ by linearity and thus

$$\begin{aligned}
e^{-H^{-1} \cdot \ell(\mu_w, \tau')} &= \left(\prod_{h=1}^{H} (\mu_w)_{\eta,h}(a'_h | s_h) \right)^{1/H} \\
&= \prod_{h=1}^{H} \left(\sum_{\pi \in \Pi} w(\pi) \cdot \pi_{\eta,h}(a'_h | s_h) \right)^{1/H} \\
&\geq \sum_{\pi \in \Pi} w(\pi) \cdot \left(\prod_{h=1}^{H} \pi_{\eta,h}(a'_h | s_h) \right)^{1/H} \\
&= \sum_{\pi \in \Pi} w(\pi) \cdot e^{-H^{-1} \cdot \ell(\pi, \tau')} \\
&= \mathbb{E}_{\pi \sim w} \left[e^{-H^{-1} \cdot \ell(\pi, \tau')} \right],
\end{aligned}$$

where the inequality follows from Hölder's inequality. Thus ℓ is $1/H$-mixable. Moreover, it is immediate that the update in Algorithm 1 corresponds to the exponential weights update with respect to ℓ over the policy class Π_η with learning rate $\lambda = H^{-1}$. Thus it holds by Proposition 10 that (9) holds for any sequence $\tau^{(t)}$.

Now, for fixed $1 \leq t \leq n$ and temporarily suppressing the notational dependence on t, note that

$$\begin{aligned}
\mathbb{E}\left[\sum_{h=1}^{H} \log \left(\frac{\pi^\star_{\eta,h}(a'_h | s_h)}{\mu_{\eta,h}(a'_h | s_h)} \right) \right] &= \mathbb{E}^\mu \mathbb{E}_{a'_h \sim \pi^\star_{\eta,h}(\cdot | s_h)} \left[\sum_{h=1}^{H} \log \left(\frac{\pi^\star_{\eta,h}(a'_h | s_h) \cdot P_h(s_{h+1} | s_h, a_h)}{\mu_{\eta,h}(a'_h | s_h) \cdot P_h(s_{h+1} | s_h, a_h)} \right) \right] \\
&= \mathrm{D}_{\mathrm{KL}} \left(\mathbb{P}^{\mu, \pi^\star_\eta} \middle\| \mathbb{P}^{\mu, \mu_\eta} \right), \qquad (10)
\end{aligned}$$

where the last equality follows from the chain rule for KL divergence (Proposition 4). Combining (9) and (10) and renormalizing, we have that

$$\mathbb{E}\left[\frac{1}{n} \sum_{t=1}^{n} \mathrm{D}_{\mathrm{KL}} \left(\mathbb{P}^{\mu^{(t)}, \pi^\star_\eta} \middle\| \mathbb{P}^{\mu^{(t)}, \mu^{(t)}_\eta} \right) \right] \leq \frac{H \cdot \log(|\Pi|)}{n}.$$

Since $\hat{\pi} = \mu_T$ for $T \sim \mathrm{Unif}([n])$, independent of the training randomness, the left-hand side is exactly the average over t. The desired bound follows. □

We can now prove the main result.

Proof of Theorem 8. The result follows immediately by combining Lemma 4 and Theorem 7. □

E.4 Proof of Proposition 2

In this section we prove the lower bound Proposition 2 demonstrating that even with online access, noisy, stochastic experts necessitate the linear in horizon dependence in sample complexity that is present in Theorem 8. We first state a more formal version of the lower bound, which is stated in the main body as Proposition 2.

Proposition 18. *For any $H \geq 2$, corruption level $0 < \eta < 1$, and $\varepsilon < 1/32$, there exists a horizon H MDP with three actions, deterministic transitions, a known corruption distribution ν, and a policy class Π of size $|\Pi| = 2$ such that ν is κ-dominated by Π with $\kappa \asymp H/\varepsilon$ and any online IL algorithm must observe at least*

$$n \geq \frac{\eta \cdot H}{512(1-\eta)^2 \cdot \varepsilon^2}$$

trajectories of interaction with the noisy expert $\pi^\star_\eta$ in order to achieve regret $J(\pi^\star) - J(\hat{\pi}) \leq \varepsilon$.

Proof. By Foster et al. [17], it suffices to show a lower bound against learning in trajectory-wise Hellinger distance to order ε. Let an MDP have states $s_1, \ldots, s_H$ with transitions $s_h \to s_{h+1}$ deterministically independent of the action. Let the action space be $\mathcal{A} = \{a_1, a_2, a_3\}$ and the policy space be $\Pi = \{\pi_+, \pi_-\}$. For some u to be determined such that $Hu \leq 1/2$ and $u \leq 1/4$, let

$$\pi_{+,h}(\cdot|s_h) = u \cdot \delta_{a_1} + (1-u) \cdot \delta_{a_2}, \quad \pi_{-,h}(\cdot|s_h) = u \cdot \delta_{a_3} + (1-u) \cdot \delta_{a_2}, \quad \text{and} \quad \nu_h = \frac{1}{4} \cdot (\delta_{a_1} + \delta_{a_3}) + \frac{1}{2} \cdot \delta_{a_2}.$$

Note that ν_h is κ-dominated by Π with $\kappa = 1/4u$.

Now observe that

$$\mathrm{D}_{\mathrm{H}^2}\left(\pi_{+,h}(\cdot|s_h), \pi_{-,h}(\cdot|s_h)\right) = 1 - \sqrt{(1-u)^2} = u.$$

Moreover, because the action distributions are independent,

$$\mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\pi_+}, \mathbb{P}^{\pi_-}\right) = 1 - (1-u)^H \geq \frac{Hu}{2},$$

where the inequality used the assumption that $2Hu \leq 1$. On the other hand, an elementary computation reveals that

$$\begin{aligned}
\pi_{+,\eta,h} &= \left(\frac{\eta}{4} + (1-\eta)u\right) \cdot \delta_{a_1} + \left(\frac{\eta}{2} + (1-\eta)(1-u)\right) \cdot \delta_{a_2} + \frac{\eta}{4} \cdot \delta_{a_3} \\
\pi_{-,\eta,h} &= \frac{\eta}{4} \cdot \delta_{a_1} + \left(\frac{\eta}{2} + (1-\eta)(1-u)\right) \cdot \delta_{a_2} + \left(\frac{\eta}{4} + (1-\eta)u\right) \cdot \delta_{a_3}.
\end{aligned}$$

Thus,

$$\begin{aligned}
\mathrm{D}_{\mathrm{KL}}\left(\pi_{+,\eta,h} \| \pi_{-,\eta,h}\right) &= \left(\frac{\eta}{4} + (1-\eta)u\right) \cdot \log\left(1 + \frac{4(1-\eta)u}{\eta}\right) + (1-\eta)u \cdot \log\left(\frac{\eta}{4(1-\eta)u + \eta}\right) \\
&= (1-\eta)u \cdot \log\left(1 + \frac{4(1-\eta)u}{\eta}\right) \\
&\leq \frac{4(1-\eta)^2 u^2}{\eta}.
\end{aligned}$$

Thus, after n rounds of interaction, the chain rule for KL divergence (Proposition 4) ensures that the KL divergence between the distributions over trajectories induced by π_+ and π_- is at most

$$nH \cdot \frac{4(1-\eta)^2 u^2}{\eta}$$

and thus if

$$n \leq \frac{\eta}{8H(1-\eta)^2 u^2},$$

then by Le Cam's inequality no algorithm can identify $\pi^\star$ with probability greater than $3/4$, which is required in order to achieve regret at most ε if $Hu/2 \geq 4\varepsilon$. Thus, setting $u = 8\varepsilon/H$, we see that as long as $\varepsilon \leq 1/32$, then $Hu \leq 1/2$ and $u \leq 1/4$. Plugging in concludes the proof. □

We now show the necessity of κ-domination in order to shave off the quadratic dependence on ε in the sample complexity. We have the following result.

Proposition 19. *For any $\kappa \geq 1$, any corruption level $0 < \eta < 1$, and any $\varepsilon < 2^{-4} \cdot \kappa^{-1}$, there exists a horizon $H = 1$ MDP with three actions, a policy class Π of size $|\Pi| = 2$, and a known corruption distribution ν such that:*

Algorithm 3 NAILGUN: Noise-robust Aggregation for Imitation Learning with Greedy UNcertainty

Require: Number of rounds n, deterministic policy class Π, noisy expert $\pi^\star_\eta$, noise ceiling $\alpha < 1$, contamination ceiling $\rho > 0$.

Set $r = \sqrt{(1-\alpha)/\alpha\rho}$ and $w_1 = \mathrm{Unif}(\Pi)$.
for $t = 1$ to n **do**
  Define $\mu_t = \sum_{\pi\in\Pi} w_t(\pi)\cdot\pi$ and $\overline{\mu}_t(\cdot|s) = \mathrm{argmax}_a\, \mu_t(a|s)$.
  Deploy $\overline{\mu}_t$ to get trajectory $\tau^{(t)} \sim \mathbb{P}^{\overline{\mu}_t}$.
  Query noisy expert $\pi^\star_\eta$ on $\tau^{(t)}$ to obtain augmented trajectory $\tau'^{(t)}$.
  Update $w_{t+1}(\pi) \propto w_t(\pi)\cdot r^{\sum_{h=1}^H \mathbb{I}\left\{\pi(s_h^{(t)}) = a_h^{(t)'}\right\}}$.
end for
return $\hat{\pi} = \overline{\mu}_T$ for $T \sim \mathrm{Unif}([n])$.

(a) ν is κ-dominated by Π and $\pi^\star \in \Pi$, and

(b) in order for an algorithm to achieve regret $J(\pi^\star) - J(\hat{\pi}) \le \varepsilon$, the learner must observe at least $n \gtrsim \eta\cdot\kappa/\varepsilon(1-\eta)^2$ trajectories of interaction with the noisy expert $\pi^\star_\eta$.

Proof. Suppose there is a single state s and three actions a_1, a_2, a_3. Fix some $0 < u \le 1/2\kappa$ and suppose that $\Pi = \{\pi_+, \pi_-\}$ where

$$\pi_+ = u\cdot\delta_{a_1} + (1-u)\cdot\delta_{a_2}, \quad \pi_- = u\cdot\delta_{a_3} + (1-u)\cdot\delta_{a_2}, \quad \text{and} \quad \nu = \kappa u\cdot\delta_{a_1} + \kappa u\cdot\delta_{a_3} + (1-2\kappa u)\cdot\delta_{a_2}.$$

It is immediate that ν is κ-dominated by Π from the construction. Moreover, observe that

$$\mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\pi_+}, \mathbb{P}^{\pi_-}\right) = 1 - \sqrt{(1-u)^2} = u.$$

Thus, if $u > 4\varepsilon$, then any algorithm returning a policy $\hat{\pi}$ such that $\mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\hat{\pi}}, \mathbb{P}^{\pi^\star}\right) \le \varepsilon$ must identify $\pi^\star$. We now observe that an elementary computation reveals that

$$\mathrm{D}_{\mathrm{KL}}\left(\mathbb{P}^{\pi_+,\eta} \| \mathbb{P}^{\pi_-,\eta}\right) = u(1-\eta)\cdot\log\left(1 + \frac{1-\eta}{\eta\kappa}\right) \le u\cdot\frac{(1-\eta)^2}{\eta\kappa}.$$

Thus, if we were to set $u = 8\varepsilon$, then we would have by Le Cam's inequality that if

$$n\cdot\frac{8\varepsilon\cdot(1-\eta)^2}{\eta\kappa} \le \frac{1}{8}$$

then with constant probability, $\hat{\pi} \neq \pi^\star$. If $\varepsilon \le 1/16\kappa$, then we can find such a u and thus the stated sample complexity is required to achieve regret at most ε. The result follows. □

F Proofs from Section 5

In this appendix, we provide the proofs of the results stated in Section 5. We begin by describing the algorithm NAILGUN and providing intuition for its design in Appendix F.2. We then provide the proof of Theorem 9 in Appendix F.3. Finally, we provide the proof of the lower bound Proposition 3 in Appendix F.4.

F.1 The Problem of Identifiability

In the unknown corruption setting, identifiability can be a major concern, i.e., it may be the case that there are multiple policies π that are consistent with the observed noisy expert $\pi^\star_\eta$ and thus it is impossible to identify the clean expert $\pi^\star$. We now provide a simple example of this phenomenon.

Let M be a horizon $H = 1$ MDP with a single state and action space $\mathcal{A}$. Let $\pi^\star = \delta_{a^\star}$ be a deterministic expert and let ν be an arbitrary policy. For $\eta \le \alpha$, let

$$\pi = \frac{1-\alpha}{1-\eta}\cdot\delta_{a^\star} + \frac{\alpha-\eta}{1-\eta}\cdot\nu.$$

Then it holds that $\pi^\star_\alpha = \pi_\eta$, so the noisy expert $\pi^\star_\alpha$ is consistent with both $\pi^\star$ and π as the underlying expert policy, and thus it is impossible to distinguish between $\pi^\star$ and π. In particular, if $J(\pi^\star) \gg J(\pi)$, then it is impossible to learn a policy with good performance without additional assumptions on the corruption or the feedback.

F.2 Description of the Algorithm

In Theorem 8, we saw that we could get regret bounds that scale polynomially in the horizon H by appealing to exponential weights and in particular the mixability of the trajectory-level log-loss. Unfortunately, this strategy fundamentally relies on the learner having access to the loss at each round of interaction, which depends on knowing both η and ν. In the unknown corruption setting, we do not have access to this information and thus we cannot directly apply this strategy. Instead we use a surrogate loss function specifically adapted to the assumption that the clean expert $\pi^\star$ is *deterministic*. In particular, we use the following loss function:

$$\ell(\pi, \tau') = \log(r) \cdot \sum_{h=1}^{H} \mathbb{I}\left\{\pi(s_h) \neq a'_h\right\}. \tag{11}$$

While this loss function itself can grow linearly in horizon, we demonstrate that its induced behavior on the probability of error is substantially nicer. We eliminate this concern by rolling out our estimated policies *greedily*. This leads naturally to Algorithm 3, which we call NAILGUN. At each step, we roll out our current policy greedily, collect noisy expert feedback, and then run Algorithm 2 with the loss function defined in (11). The key step in the analysis, provided below, is the observation that the loss in (11) has the property that when we make a mistake at any point along the trajectory, the weight we place on a policy that is not the expert goes down by a constant factor in expectation, where critically this constant factor is independent of horizon. This analysis is thus substantially different from the standard analysis of exponential weights, which relies on the mixability of the loss function to control the sum of losses across rounds, and is more similar to the analysis of the *halving algorithm* for online learning with expert advice [39].

F.3 Proof of Theorem 9

We begin by restating the theorem with a slightly tighter dependence on the parameters in question.

Theorem 16. *Let Π be a class of* deterministic *policies and suppose that $\pi^\star \in \Pi$. Let ν denote a corruption distribution such that for all $h \in [H]$, $a \in \mathcal{A}$, and $s \in \mathcal{S}$ it holds that $\nu_h(a|s) \leq \rho$ for some $0 < \rho < 1$. Let $\alpha > 0$ such that $(1+\rho)\alpha < 1$ and suppose that the corruption noise in the noisy expert η satisfies $\eta \leq \alpha$. If $\hat{\pi}$ is the policy returned by Algorithm 3, then it holds that*

$$\mathbb{E}\left[\mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\hat{\pi}}, \mathbb{P}^{\pi^\star}\right)\right] \leq \frac{4 \cdot \log(|\Pi|)}{n\left(\sqrt{1-\alpha} - \sqrt{\alpha\rho}\right)^2}.$$

We now derive Theorem 9 directly from this result.

Proof of Theorem 9. We compute

$$\left(\sqrt{1-\alpha} - \sqrt{\alpha\rho}\right)^2 = \frac{\left(1-\alpha(1+\rho)\right)^2}{\left(\sqrt{1-\alpha} + \sqrt{\alpha\rho}\right)^2} \geq \frac{\left(1-\alpha(1+\rho)\right)^2}{2}.$$

Plugging this into Theorem 16 concludes the proof. □

In order to prove Theorem 16, we must introduce some notation. First, for any $1 \leq t \leq n$, let $\mathcal{F}_t$ denote the σ-algebra generated by all randomness before time t. Thus w_t, W_t, μ_t, and $\overline{\mu}_t$ are $\mathcal{F}_{t-1}$-measurable. We recall that

$$r = \sqrt{\frac{1-\alpha}{\alpha\rho}} \geq 1, \quad \mu_t = \sum_{\pi \in \Pi} w_t(\pi) \cdot \pi, \quad \text{and} \quad w_{t+1}(\pi) \propto w_t(\pi) r^{\sum_{h=1}^{H} \mathbb{I}\left\{\pi\left(s_h^{(t)}\right) = a_h^{(t)\prime}\right\}}.$$

Moreover, $\overline{\mu}_t$ denotes the greedy policy associated with μ_t. We introduce the following notation:

$$R_t(\pi) = \frac{w_t(\pi)}{w_t(\pi^\star)} \quad \text{and} \quad W_t = \frac{1}{w_t(\pi^\star)} = 1 + \sum_{\pi \neq \pi^\star} R_t(\pi). \tag{12}$$

Note that if W_t is small, then μ_t places a lot of weight on the correct policy $\pi^\star$ and, moreover, if W_t is very small, then we expect $\overline{\mu}_t$ to agree with $\pi^\star$ on a trajectory. We let

$$M_t = \left\{\text{there exists } h \leq H \text{ such that } \overline{\mu}_{t,h}(s_h^{(t)}) \neq \pi_h^\star(s_h^{(t)})\right\} \tag{13}$$

denote the event that the rolled out policy $\overline{\mu}_t$ disagrees with the *clean* expert $\pi^\star$ on at least one action. The key lemma that we will show is that, in expectation, any time a mistake is made, W_t goes down by a constant factor.

Lemma 5. *Let W_t be as in (12) and let M_t be as in (13). Then it holds for any $1 \le t \le n$ that*

$$\mathbb{E}\left[W_{t+1}|\mathcal{F}_{t-1}\right] \le W_t\left(1 - \frac{\left(\sqrt{1-\alpha} - \sqrt{\alpha\rho}\right)^2}{2} \cdot \mathbb{E}\left[M_t|\mathcal{F}_{t-1}\right]\right).$$

We will now prove Theorem 16 assuming Lemma 5 and return to the proof of this key result below.

Proof of Theorem 16. Let M_t be as in (13). Applying the fact that $\hat{\pi} = \mu_T$ for $T \sim \mathrm{Unif}([n])$ as well as Foster et al. [17, Lemma D.3], we see that

$$\mathbb{E}\left[\mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\hat{\pi}}, \mathbb{P}^{\pi^\star}\right)\right] \le \frac{1}{n}\sum_{t=1}^{n}\mathbb{E}\left[\mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\overline{\mu}_t}, \mathbb{P}^{\pi^\star}\right)\right] \le \frac{2}{n}\sum_{t=1}^{n}\mathbb{P}\left(M_t\right).$$

We now apply Jensen's inequality and Lemma 5 to observe that

$$\begin{aligned}
\mathbb{E}\left[\log(W_{t+1})|\mathcal{F}_{t-1}\right] &\le \log\left(\mathbb{E}\left[W_{t+1}|\mathcal{F}_{t-1}\right]\right) \\
&\le \log\left(W_t\left(1 - \frac{\left(\sqrt{1-\alpha} - \sqrt{\alpha\rho}\right)^2}{2} \cdot \mathbb{E}\left[M_t|\mathcal{F}_{t-1}\right]\right)\right) \\
&= \log(W_t) + \log\left(1 - \frac{\left(\sqrt{1-\alpha} - \sqrt{\alpha\rho}\right)^2}{2} \cdot \mathbb{P}\left(M_t|\mathcal{F}_{t-1}\right)\right) \\
&\le \log(W_t) - \frac{\left(\sqrt{1-\alpha} - \sqrt{\alpha\rho}\right)^2}{2} \cdot \mathbb{P}\left(M_t|\mathcal{F}_{t-1}\right).
\end{aligned}$$

The last inequality follows due to the fact that

$$0 \le \frac{\left(\sqrt{1-\alpha} - \sqrt{\alpha\rho}\right)^2}{2}\mathbb{P}\left(M_t|\mathcal{F}_{t-1}\right) \le 1$$

almost surely and the numerical inequality $\log(1-x) \le -x$ for $0 < x < 1$. Rearranging, taking expectations, and applying the tower property of conditional expectation, we see that

$$\begin{aligned}
\frac{\left(\sqrt{1-\alpha} - \sqrt{\alpha\rho}\right)^2}{2} \cdot \sum_{t=1}^{n}\mathbb{P}\left(M_t\right) &\le \sum_{t=1}^{n}\mathbb{E}\left[\log(W_t) - \log(W_{t+1})\right] \\
&\le \log(W_1) \\
&= \log\left(|\Pi|\right),
\end{aligned}$$

where the second inequality follows because $W_t \ge 1$ by construction in (12) and the equality follows by assuming a uniform prior w_1. Rearranging concludes the proof. □

Thus it remains to prove the key intermediate result, Lemma 5.

Proof of Lemma 5. We fix a time t and omit the t from the notation of the trajectories τ and τ' in order to simplify the presentation. Let $\mathcal{G}_t$ denote the sigma-algebra generated by $\mathcal{F}_{t-1}$ and the states $s_1^{(t)}, \dots, s_H^{(t)}$, but not the noisy labels $a_1^{(t)\prime}, \dots, a_H^{(t)\prime}$. That is, $\mathcal{G}_t = \mathcal{F}_{t-1} \vee \sigma(s_{1:H}^{(t)})$. Observe that for $\pi \neq \pi^\star$, letting $a_h^\star = \pi_h^\star(s_h)$ and $\overline{a}_h = \pi_h(s_h)$, we have

$$R_{t+1}(\pi) = R_t(\pi) \cdot \exp\left(\log(r) \cdot \sum_{h=1}^{H}\left(\mathbb{I}\left\{\overline{a}_h = a_h'\right\} - \mathbb{I}\left\{a_h^\star = a_h'\right\}\right)\right).$$

In the event that $\overline{a}_h = a_h^\star$, clearly $\mathbb{I}\left\{\overline{a}_h = a_h'\right\} - \mathbb{I}\left\{a_h^\star = a_h'\right\} = 0$. On the other hand, if $\overline{a}_h \neq a_h^\star$, then by the margin assumption,

$$\mathbb{P}\left(a_h' = a_h^\star|s_h\right) \ge 1 - \alpha \quad \text{and} \quad \mathbb{P}\left(a_h' = \overline{a}_h|s_h\right) \le \alpha\rho.$$

Thus,

$$\begin{aligned}\mathbb{E}\left[r^{\mathbb{I}\left\{\pi_h(s_h)=a'_h\right\}-\mathbb{I}\left\{\pi^\star_h(s_h)=a'_h\right\}}|s_h\right] &= 1-\mathbb{P}\left(a'_h=a^\star_h|s_h\right)-\mathbb{P}\left(a'_h=\overline{a}_h|s_h\right)\\ &\quad+\frac{\mathbb{P}\left(a'_h=a^\star_h|s_h\right)}{r}+r\cdot\mathbb{P}\left(a'_h=\overline{a}_h|s_h\right)\\ &= 1-\left(1-\frac{1}{r}\right)\cdot\mathbb{P}\left(a'_h=a^\star_h|s_h\right)+(r-1)\cdot\mathbb{P}\left(a'_h=\overline{a}_h|s_h\right).\end{aligned}$$

By the assumptions on α,ρ, it holds that $r>1$ and thus combining the preceding two displays, we have

$$\begin{aligned}\mathbb{E}\left[r^{\mathbb{I}\left\{\pi_h(s_h)=a'_h\right\}-\mathbb{I}\left\{\pi^\star_h(s_h)=a'_h\right\}}|s_h\right] &\le 1-\left(1-\frac{1}{r}\right)\cdot(1-\alpha)+(r-1)\cdot\alpha\rho\\ &= 1-\left(\sqrt{1-\alpha}-\sqrt{\alpha\rho}\right)^2,\end{aligned}$$

where we used the fact that $r=\sqrt{(1-\alpha)/\alpha\rho}$ in the last equality.

Letting

$$N_t(\pi)=\sum_{h=1}^{H}\mathbb{I}\left\{\pi_h(s_h^{(t)})\neq\pi^\star_h(s_h^{(t)})\right\},$$

we thus conclude that

$$\begin{aligned}\mathbb{E}\left[R_{t+1}(\pi)|\mathcal{G}_t\right] &\le R_t(\pi)\cdot\left(1-\left(\sqrt{1-\alpha}-\sqrt{\alpha\rho}\right)^2\right)^{N_t(\pi)}\\ &\le R_t(\pi)\left(1-\left(\sqrt{1-\alpha}-\sqrt{\alpha\rho}\right)^2\cdot\mathbb{I}\left\{N_t(\pi)\ge 1\right\}\right).\end{aligned}$$

Summing over $\pi\neq\pi^\star$, we have that

$$\begin{aligned}\mathbb{E}\left[W_{t+1}-1|\mathcal{G}_t\right] &= \mathbb{E}\left[\sum_{\pi\neq\pi^\star}R_{t+1}(\pi)|\mathcal{G}_t\right]\\ &\le \sum_{\pi\neq\pi^\star}R_t(\pi)\left(1-\left(\sqrt{1-\alpha}-\sqrt{\alpha\rho}\right)^2\cdot\mathbb{I}\left\{N_t(\pi)\ge 1\right\}\right)\\ &= W_t-1-\left(\sqrt{1-\alpha}-\sqrt{\alpha\rho}\right)^2\cdot\sum_{N_t(\pi)\ge 1}R_t(\pi).\end{aligned}\tag{14}$$

Now let $a_h=\overline{\mu}_{t,h}(s_h)$ be the rolled out action and suppose that M_t occurs; then there is some minimal $h\le H$ such that $a_h\neq a^\star_h$. Because $\overline{\mu}_t$ is the greedy policy, it must then hold that

$$\sum_{\substack{\pi\in\Pi\\ \pi(s_h)=a_h}}w_t(\pi)\ge\sum_{\substack{\pi\in\Pi\\ \pi(s_h)=a^\star_h}}w_t(\pi)$$

and thus $\mu_{t,h}(a^\star_h|s_h)\le 1/2$ and, in particular,

$$\sum_{N_t(\pi)\ge 1}w_t(\pi)\ge\frac{1}{2}.$$

Dividing by $w_t(\pi^\star)$, we see that

$$\sum_{N_t(\pi)\ge 1}R_t(\pi)\ge\frac{1}{2\cdot w_t(\pi^\star)}=\frac{W_t}{2}.$$

By (14), it clearly holds that W_{t+1} is a supermartingale. Moreover, combining that equation with the preceding display, we see that the result holds. □

F.4 Proof of Proposition 3

We now prove that NAILGUN is essentially optimal up to constants, the content of Proposition 3. We first state a more formal version of the result before giving the construction.

Proposition 20. *For any $\varepsilon < 1/8$, any $0 < \alpha < 1$ and any $0 < \rho < 1$ such that $\alpha(1+\rho) < 1$, there exists a horizon $H = 2$ MDP and a* deterministic *policy class Π of size $|\Pi| = M$ with $M \leq 1 + 1/\rho$ such that there exists a family of ρ-smooth measures $\{\nu_i\}_{i=1}^M$ such that for some $\eta \leq \alpha$, any online IL algorithm with access to a noisy expert $\pi^\star_\eta$ must observe*

$$n \gtrsim \frac{1}{\varepsilon\left(1-\alpha(1+\rho)\right) \cdot \log\left(1 + \frac{1-\alpha(1+\rho)}{\alpha\rho}\right)}$$

trajectories in order to guarantee expected regret $J(\pi^\star) - J(\hat{\pi}) \leq \varepsilon$.

Proof. Let $m = \lceil 1/\rho \rceil$ and let $\Pi = \left\{\pi^1, \ldots, \pi^m\right\}$ be a class of m deterministic policies. Let $\mathcal{A} = [m] \cup \{\perp\}$ and take the corruption level to be $\eta = \alpha$. We suppose there is a deterministic initial state s_1 and all policies map to the same action, which transitions to state s_2 with probability q and s_2' with probability $1-q$ for some q to be determined. For $1 \leq i \leq m$, let

$$\pi^i(s) = \begin{cases} \perp & s = s_1 \\ i & s = s_2 \\ \perp & s = s_2' \end{cases},$$

and let

$$\nu_i(i) = 0, \quad \nu_i(j) = \rho \text{ for } j \in [m] \setminus \{i\}, \quad \text{and} \quad \nu_i(\perp) = 1 - (m-1)\rho.$$

In other words, π^i is informative only on state s_2 and reveals the 'correct' action, whereas ν_i places no mass on i and distributes mass evenly otherwise. Note that by construction ν_i is always ρ-smooth.

We first note that π_i and π_j differ only upon transitioning to s_2 and thus

$$\mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\pi_i}, \mathbb{P}^{\pi_j}\right) = q$$

and so for any possibly randomized $\hat{\pi}$,

$$\mathbb{E}\left[\mathrm{D}_{\mathrm{H}^2}\left(\mathbb{P}^{\pi_i}, \mathbb{P}^{\hat{\pi}}\right)\right] = q \cdot \mathbb{P}\left(\hat{\pi}(s_2) \neq i\right).$$

Thus if $q = 4\varepsilon$, it suffices to show that for insufficiently many rounds of interaction, $\mathbb{P}\left(\hat{\pi}(s_2) \neq i\right) \geq 1/4$.

Now, we will let Q_i denote the trajectory distribution obtained by rolling out $(1-\alpha) \cdot \pi_i + \alpha \cdot \nu_i$ and we will denote by $p_i = (1-\alpha)\delta_i + \alpha \cdot \nu_i$ and note that

$$p_i(j) = \begin{cases} 1-\alpha & j = i \\ \alpha\rho & j \in [m] \setminus \{i\} \\ \alpha(1-(m-1)\rho) & j = \perp \\ 0 & \text{otherwise.} \end{cases}$$

Thus,

$$\mathrm{D}_{\mathrm{KL}}\left(p_i \| p_j\right) = (1-\alpha) \cdot \log\left(\frac{1-\alpha}{\alpha\rho}\right) + \alpha\rho \cdot \log\left(\frac{\alpha\rho}{1-\alpha}\right) = \left(1-\alpha(1+\rho)\right) \cdot \log\left(1 + \frac{1-\alpha(1+\rho)}{\alpha\rho}\right).$$

Moreover, because p_i and p_j differ from each other only upon transitioning to s_2, it holds by the chain rule (Proposition 4) that

$$\mathrm{D}_{\mathrm{KL}}\left(Q_i \| Q_j\right) = q \cdot \left(1-\alpha(1+\rho)\right) \cdot \log\left(1 + \frac{1-\alpha(1+\rho)}{\alpha\rho}\right) = 4\varepsilon \cdot \left(1-\alpha(1+\rho)\right) \cdot \log\left(1 + \frac{1-\alpha(1+\rho)}{\alpha\rho}\right).$$

Applying Le Cam's two-point method to any pair $i \neq j$ and the chain rule again, it holds that in order for $\hat{\pi}$ to have a better than $3/4$ probability of selecting the correct index, it must hold that

$$n \gtrsim \frac{1}{4\varepsilon \cdot \left(1-\alpha(1+\rho)\right) \cdot \log\left(1 + \frac{1-\alpha(1+\rho)}{\alpha\rho}\right)}.$$

The result follows.

□

Finally, we can prove the main result.

Proof of Proposition 3. We observe that

$$\log\left(1+\frac{1-\alpha(1+\rho)}{\alpha\rho}\right)\leq\frac{1-\alpha(1+\rho)}{\alpha\rho}$$

and apply Proposition 20. □

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction state the main theoretical claims, including the offline exponential separation, the online augmented-trajectory guarantee, the unknown-corruption extension, and the empirical scope. The assumptions and scope are stated in the problem setup and theorem statements.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We discuss limitations in the discussion, including the stylized corruption model, the focus on learning from deterministic experts in the unknown corruption setting, the gap between the theoretical algorithms and practical OPD implementations, and the limited scale of the empirical validation due to our compute budget.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

   Justification: All theoretical results state their assumptions, including realizability, known or unknown corruption, deterministic versus stochastic experts, and domination/smoothness conditions. Full and correct proofs are provided in the appendix.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: The experimental setup, methods, task construction, noise model instantiation, architectures, training details, and evaluation metrics are described in Section 6 and Appendix B.

   Guidelines:

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

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [No]

   Justification: We plan to release code and scripts for the camera ready version, but they are not included with the initial anonymous submission. The appendix provides implementation details sufficient to understand the main experimental results.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: The paper specifies the training objectives, task setup, corruption process, model architecture, optimizer settings, dataset sizes, other hyperparameters, and evaluation protocol in Section 6 and Appendix B.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: The plots report variability across independent runs using shaded error regions. The appendix specifies the number of seeds and how the error regions are computed.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: The appendix reports the compute resources used for the synthetic and GSM8K experiments, including GPU type and approximate runtime (Appendix B.1).

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research conforms to the NeurIPS Code of Ethics. The work is primarily theoretical and uses synthetic tasks and standard public benchmarks, with no human-subject experiments or private data.

   Guidelines:

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

10. **Broader impacts**

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

Answer: [Yes]

Justification: The paper discusses potential positive impacts, such as more reliable learning from imperfect teachers, and potential risks, such as improving distillation methods for language models that could be misused if deployed without safeguards (Appendix A.4).

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

Answer: [N/A]

Justification: The paper does not release high-risk pretrained models, scraped datasets, or systems intended for deployment. The experiments use small synthetic tasks and standard benchmark data.

Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: Existing datasets, models, and codebases used in the experiments are cited in the paper and appendix. We exclusively used open-source code and datasets for all experiments.

Guidelines:

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

13. **New assets**

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

Answer: [N/A]

Justification: The paper does not introduce a new benchmark dataset or pretrained model as a standalone released asset. Any released code will be documented with reproduction instructions.

Guidelines:

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

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

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

Answer: [N/A]

Justification: The paper does not involve crowdsourcing, human-subject experiments, or collection of human participant data.

Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or human-subject research, so IRB approval is not applicable.

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: LLMs are studied as part of the experimental domain, but no LLM is used as an important, original, or non-standard component of the research methodology beyond the models explicitly described in the experiments.

    Guidelines:

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