When Does Online Imitation Learning Help in LLM Post-Training? The Role of (Non-)Realizability Beyond Horizon

Huaqing Zhang*
Tsinghua University
zhanghq22@mails.tsinghua.edu.cn

Jingchu Gai*
Carnegie Mellon University
jgai@cs.cmu.edu

Juno Kim
UC Berkeley
junokim@berkeley.edu

Bingbin Liu
Harvard University
bliu@g.harvard.edu

Andrej Risteski
Carnegie Mellon University
aristesk@cs.cmu.edu

Abstract

Online imitation learning (IL), particularly on-policy distillation, has emerged as a strong LLM post-training approach, often outperforming offline supervised fine-tuning (SFT). Yet a principled understanding of when and why online helps remains unclear. In this work, we challenge the view that error accumulation is the main source of online IL's advantage, and instead show that the benefits of online interaction depend critically on whether the setting is realizable, i.e., whether the student policy class can represent the expert policy. Under realizability, we empirically find that offline IL already matches expert performance. In contrast, in non-realizable (misspecified) settings, we prove that offline IL encounters an information-theoretic bottleneck even when horizon $H = 1$, and propose a structural characterization of misspecification relative to the reward, under which online IL provably achieves high performance despite large expert-student discrepancy.

1 Introduction

Imitation learning (IL) trains a student policy from expert demonstrations, enabling the student to achieve strong performance without access to explicit reward signals. For post-training of large language models (LLMs), the canonical offline imitation learning method is *supervised fine-tuning* (SFT), where the learner imitates an expert policy using a fixed dataset of expert demonstrations, such as human-written responses or traces generated by a stronger model. However, such offline approaches are reported to suffer from limitations such as poor generalization [8, 21, 31] and severe catastrophic forgetting [6].

Online imitation learning offers a promising alternative for mitigating these issues in post-training. In the online IL setting, the learner actively interacts with the environment and queries an expert to obtain feedback [50]. A prominent online IL approach for LLM post-training is *on-policy distillation* (OPD), where the student samples from its current policy and minimizes the reverse-KL divergence to the expert distribution [30, 3, 20]. Although recent studies report the empirical superiority of online IL in LLM post-training [61, 36], a principled understanding of when and why this advantage arises remains limited.

A classical explanation for the advantage of online IL is its ability to mitigate error accumulation: offline IL can suffer from errors that compound quadratically over the horizon, whereas DAgger-style online algorithms improve this dependence under a recoverability assumption [48, 50]. However, in *realizable* settings, where the expert policy is contained in the student policy class, recent theory

*Equal contribution.

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

shows that DAgger offers no sample complexity advantage over offline IL for the log-likelihood objective as used in SFT [16]. Our empirical results confirm this: in realizable settings with synthetic and real-world math reasoning tasks, SFT-trained students are able to match the expert's performance, and OPD offers no further gains in either accuracy or training speed (Section 4). These results show that empirically, offline and online IL match not only in statistical complexity, but also when considering other factors such as optimization and problem-specific structures.

Both Foster et al. [16]'s theory and our results motivate us to focus on *non-realizability* (or *misspecification*), rather than error accumulation, as a primary source of online IL's advantage in LLM training: the student policy class is too restricted, e.g., due to a limited model size, to represent the expert policy. This view is consistent with recent empirical findings [20, 62, 61] that OPD outperforms SFT in strong-to-weak distillation where a smaller student learns from a much larger expert, a setting that is inherently non-realizable.

To theoretically understand online IL's advantage in non-realizable settings, we study a simplified yet informative contextual-bandit framework with horizon $H = 1$. This setting rules out long-horizon error accumulation by construction, allowing us to isolate the advantages of online IL under misspecification. We first show, both theoretically and empirically, that prior analyses which control the performance gap via distributional discrepancies between the expert and student policies [16, 47, 45] can be overly pessimistic under misspecification, and that reward-dependent analyses are necessary. For offline IL, we identify an information-theoretic limitation specific to non-realizable settings: the expert policy may place too little probability mass on responses that the restricted student class can learn. For online IL, we identify a *structural condition* (Assumption 1) under which online IL can provably achieve small performance suboptimality despite large distributional discrepancy. The condition characterizes misspecification relative to the reward: the expert's signal should either be aligned with the actual reward, or deviate from it only in ways that the restricted student class cannot realize. We further show that the sample complexity advantage of online IL over offline IL relies on access to a strong base model that provides better coverage than the expert.

To summarize, our main contribution is identifying non-realizability as a key factor underlying the advantage of online over offline imitation learning in language model post-training. Specifically,

1. In realizable settings, we empirically verify that offline IL (SFT) fully recovers expert performance, and that online IL (on-policy distillation) yields neither accuracy gains nor faster learning (Section 4).
2. In non-realizable settings, we show that prior discrepancy-based analyses can be overly pessimistic (Section 5.1), and reward-dependent analyses are necessary to explain the success of IL.
3. By analyzing the reward-dependent structure of misspecification, we identify an information-theoretic limitation of offline IL in this regime (Section 5.2) and propose a structural condition characterizing when online IL can remain effective despite misspecification (Section 5.3). Under this condition, online IL achieves lower sample complexity than offline IL when the base policy provides better coverage of the optimal student policy $\pi^\star$ than the expert policy does (Section 5.3.1).

2 Preliminaries

For a set $\mathcal{X}$, let $\Delta(\mathcal{X})$ denote the set of probability distributions over $\mathcal{X}$. For distributions $P, Q \in \Delta(\mathcal{X})$, let $D_{\mathrm{KL}}(P\|Q) \coloneqq \mathbb{E}_{x\sim P}\log\frac{P(x)}{Q(x)}$ $D_{\mathrm{TV}}(P,Q) \coloneqq \frac{1}{2}\sum_{x\in\mathcal{X}}|P(x)-Q(x)|$, and $D_{\chi^2}(P\,\|\,Q) \coloneqq \mathbb{E}_{x\sim Q}\left[\left(P(x)/Q(x)\right)^2\right] - 1$. To provide a clean theoretical sandbox, we abstract language model post-training as a contextual bandit problem [44, 60, 17]. Let $\mathcal{X}$ represent the set of all possible prompts (contexts) and $\mathcal{Y}$ the set of generated responses (actions). A policy $\pi : \mathcal{X} \to \Delta(\mathcal{Y})$ specifies a conditional distribution $\pi(y \mid x)$ over responses y given a prompt x. Given a reward function $r(x, y) : \mathcal{X} \times \mathcal{Y} \to [0, 1]$, we aim to find a policy π within a policy class Π that achieves high expected reward $V(\pi) \coloneqq \mathbb{E}_{x\sim\mathcal{D}_x}\mathbb{E}_{y\sim\pi(\cdot|x)}r(x, y)$, where $\mathcal{D}_x$ is the distribution over prompts. We follow standard convention in RL theory and assume the policy class $|\Pi| < \infty$ is finite [15, 16, 47]. Realizability is a widely studied assumption in RL theory that asks whether the student policy class is expressive enough to represent the expert policy.

Definition 1 (Realizability and Misspecification)**.** *We say the expert policy π^{e} is realizable with respect to the student policy class Π if $\pi^{\mathrm{e}} \in \Pi$. Otherwise, the setting is non-realizable (misspecified).*

Imitation learning is a learning paradigm in which the learner improves its policy by leveraging information provided by an expert policy π^{e}, without direct access to the reward feedback during training. In this work, we study two settings: offline and online imitation learning.

Offline Imitation Learning. The learner is given a fixed dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ of i.i.d. expert demonstrations, where $x_i \sim \mathcal{D}_x$ and $y_i \sim \pi^{\mathrm{e}}(\cdot \mid x_i)$. The standard approach in this setting is **behavior cloning**, which reduces IL to minimizing a supervised learning objective:

$$L_{\mathrm{BC}}(\pi) \coloneqq \mathbb{E}_{x\sim\mathcal{D}_x}\, \mathbb{E}_{y\sim\pi^{\mathrm{e}}(\cdot|x)}\big[\ell(\pi(\cdot \mid x), y)\big], \tag{1}$$

where $\ell : \Delta(\mathcal{Y}) \times \mathcal{Y} \to \mathbb{R}$ is a loss function. A standard instantiation in LLM post-training uses the negative log-likelihood loss $\ell(\pi(\cdot \mid x), y) = -\log \pi(y \mid x)$, which corresponds to the **SFT** objective.

Online Imitation Learning. In the online setting, the learner can actively query the expert for feedback. Learning proceeds in episodes: in the ith episode, the learner observes $x_i \sim \mathcal{D}_x$, samples a response $y_i \sim \pi_i(\cdot \mid x_i)$ from a policy π_i, and receives expert feedback on this response. In existing MDP formulations (e.g., [48, 16]), the expert provides step-wise guidance along a trajectory. In order to isolate online IL's advantage due to *non-realizability* from the effects of error accumulation over long horizons [48], we instead use a simpler formulation for contextual bandits: in each episode, the learner produces a complete response y_i (a token sequence), and the expert reveals its conditional probability $\pi^{\mathrm{e}}(y_i \mid x_i)$ on that response. This form of probability access is natural in language model distillation, where the expert model's logits are available. Furthermore, it captures the core interaction pattern of widely used on-policy distillation algorithm [3, 20, 36]. While the instantiation of online IL can be flexible, we focus here on the following formulation:

Definition 2 (Population Objective of Online IL)**.** *Given an expert policy π^{e} and a* shaping function *$f : \mathbb{R} \to \mathbb{R}$ that maps expert density to a scalar score, online IL aims to maximize*

$$J_{\mathrm{on}}(\pi) \coloneqq \mathbb{E}_{x\sim\mathcal{D}_x}\mathbb{E}_{y\sim\pi(\cdot|x)}\big[f(\pi^{\mathrm{e}}(y \mid x))\big]. \tag{2}$$

The key difference to the offline objective (Eq. (1)) is that the inner expectation is taken over the student policy π, rather than the expert policy π^{e}. A prominent instantiation of online imitation learning in language model post-training is **on-policy distillation** [3, 20], which uses a reverse-KL objective and is optimized in an online, on-policy manner:

$$\max_{\pi\in\Pi} -\mathbb{E}_{x\sim\mathcal{D}_x}\big[D_{\mathrm{KL}}\left(\pi(\cdot \mid x) \| \pi^{\mathrm{e}}(\cdot \mid x)\right)\big]. \tag{3}$$

This reverse-KL objective is an instance of Definition 2 with $f\left(\pi^{\mathrm{e}}(y \mid x)\right) = \log\left(\pi^{\mathrm{e}}(y \mid x)\right)$, together with an additional entropy regularization term that promotes exploration and prevents policy collapse.[2]

Remark 1 (Contextual bandit formulation)**.** *Although IL is often analyzed in MDPs, the LLM post-training setting collapses to a contextual bandit at the response level: generation has deterministic transitions, and the reward (e.g., whether a math problem is solved correctly) is defined over the complete trajectory. In other words, given a context x, a policy (language model) π essentially specifies a distribution over entire responses y. This contextual bandit view also matches practical SFT (Eq.* (1)*) and on-policy distillation (Eq.* (3)*) objectives, which are defined over full responses. Moreover, we focus on* non-realizability *as a source of online IL's advantage. Unlike the classical error-accumulation view, which attributes this advantage to mitigating compounding errors over long horizons [48], the advantage from non-realizability already appears in contextual bandits with horizon $H = 1$. Removing the multi-step structure allows us to isolate this source of advantage.*

3 Related Work

Imitation Learning for LLM Post-Training. Imitation learning plays an important role in language-model post-training. Supervised fine-tuning (SFT) can be viewed as offline IL, where the model is trained with a log-likelihood objective on the demonstration data [4, 43, 40]. However, empirical studies show that SFT can struggle to improve a weak student when the expert is a substantially stronger reasoning model [33, 29]. Recently, online IL has become a prominent approach for LLM post-training, with **on-policy distillation** (OPD) as the representative instantiation [30, 20]. These methods have proven effective in practice [36, 62] and have been adopted in industrial-scale post-training pipelines [61, 58]. However, a theoretical understanding of when and why online methods outperform offline SFT remains limited, as discussed below.

[2] Our analysis can be generalized to a shaping function f depending on both $\pi^{\mathrm{e}}(y \mid x)$ and $\pi(y \mid x)$. This extension fully captures the reverse-KL objective. For simplicity, we focus on shaping functions that depend only on the expert density.

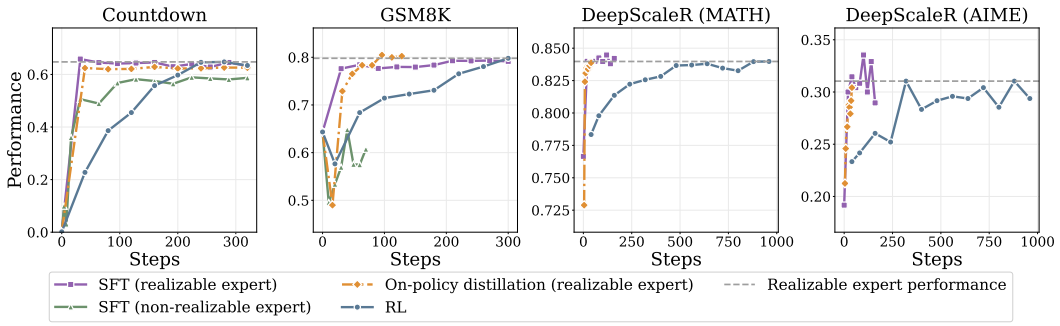


Figure 1: ***Offline IL suffices with realizable experts.*** *Across different tasks, SFT with a realizable expert fully matches expert performance, and on-policy distillation yields no accuracy gains or faster training. In contrast, when the expert is non-realizable, SFT exhibits a significant performance gap.*

Theoretical Analysis on Online and Offline IL. Seminal work by Ross & Bagnell [48] shows that offline imitation learning can incur error accumulation, specifically a quadratic dependence with respect to horizon, and DAgger-like algorithms can mitigate error amplification through online interactions under recoverability assumptions [50, 49]. However, this line of work gives only algorithm-dependent lower bounds, rather than an end-to-end separation between offline and online IL [16, 45]. Indeed, Foster et al. [16] show that, under *realizability*, DAgger-like algorithms offer no worst-case statistical improvement over offline IL with log-likelihood loss. While online interaction could theoretically yield improvements under additional assumptions (see the discussion in their Section 4), our empirical results do not provide evidence of such gains under realizability (Section 4).

In non-realizable settings, Rohatgi et al. [47] formulate IL as minimizing the Hellinger distance between student and expert. They show that an offline BC algorithm, `RhoEstimatorBC`, is already statistically optimal for this objective [47, Theorem 3.1], and no computational advantage of online IL is established. This suggests that discrepancy-based analyses can be overly pessimistic and may not explain the practical advantage of online IL. We discuss additional related work in Appendix A.

4 Offline IL Suffices Under Realizability

Classical analyses attribute online IL's advantage to mitigating long-horizon error accumulation [48, 50]. However, recent theoretical work shows that under realizable settings, offline behavior cloning already achieves optimal statistical complexity for the log-likelihood objective used in SFT [16]. While this means there is no worst-case statistical advantage for online IL under realizable settings, in practice, other factors such as optimization dynamics and problem-specific structures may still make online IL favorable. In this section, we show that this is not the case: as a sanity check for the realizable setting, we empirically demonstrate that SFT (offline IL) is indeed sufficient to fully recover expert performance in this regime, and on-policy distillation does not yield further gains.

Setup We conduct experiments on Countdown [41][3] and three math reasoning datasets: GSM8K [9], DeepScaleR [37], and DAPO [67]. We compare the following training paradigms:

- **RL:** Models are trained using GRPO [51] with verifiable outcome rewards.
- **SFT/OPD with non-realizable expert:** For the Countdown task, the student is trained directly on ground-truth solutions. For GSM8K, the student is supervised by a highly capable non-realizable expert, DeepSeek-V3.2-Exp [35], which achieves over 95% accuracy on GSM8K. For DAPO, we follow the setup of Li et al. [34] and use Skywork-7B [22] as the expert.
- **SFT/OPD with realizable expert:** For Countdown, GSM8K, and DeepScaleR, we use the RL-trained model as the expert. For DAPO, we use the model obtained by OPD from Skywork-7B. Each expert is trained from the same base model as its student and therefore belongs to the student's policy class, thus realizability is ensured by construction.

[3]Countdown asks the model to use each given integer exactly once, with operators $\{+, -, \times, \div\}$, to form an expression matching a target; e.g., for $[8, 3, 2, 1]$ and target 24, $8 \times 3 \times (2 - 1) = 24$.

Results When the expert is realizable, offline IL (SFT) matches expert performance on both Countdown and real-world math reasoning tasks, and online IL (OPD) improves neither final performance nor training efficiency (Figure 1). This confirms that the error-accumulation argument does not separate offline and online IL in the realizable setting. In contrast, when the expert is *non-realizable*, SFT falls well short of the expert's performance, and OPD outperforms SFT by a large margin (Figure 2), consistent with evidence in large-scale and industrial settings [20, 61, 62]. [4] We further evaluate out-of-distribution (OOD) generalization on the Countdown task and catastrophic forgetting during GSM8K training. We find that both SFT and on-policy distillation with a realizable expert achieve OOD performance close to that of the expert and exhibit minor forgetting. By contrast, SFT with a non-realizable expert exhibits substantially worse OOD generalization and more severe forgetting. Detailed settings and results are provided in Appendix C.

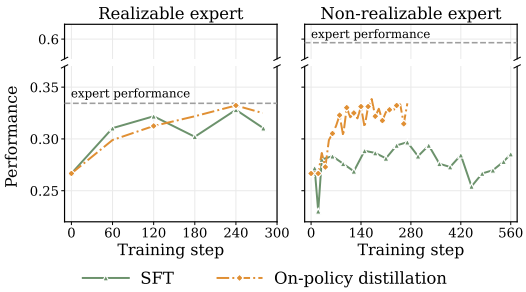


Figure 2: **OPD improves SFT performance in non-realizable settings.** Trained on the DAPO dataset. Average accuracy on AIME 2024 and 2025 reported. OPD achieves higher performance than SFT for a non-realizable expert, but does not improve for a realizable expert.

5 Non-Realizability as a Source of Online IL Gains

Since the advantage of online IL in LLM post-training does not arise under realizability (Section 4), we turn to the *non-realizable* setting, where the expert policy cannot be represented by the student policy class, i.e. $\pi^{\mathrm{e}} \notin \Pi$. In this setting, online on-policy distillation has been empirically shown to significantly outperform offline IL (SFT) [20, 61, 62]. However, despite these successes, a theoretical understanding of *why and when* online interaction helps in non-realizable settings remains limited.

Previous work framed imitation learning as minimizing the distributional discrepancy (e.g., Hellinger distance) between the student and expert policies [47]. In this section, we first discuss and empirically verify that such discrepancy-based analyses can be overly pessimistic under misspecification. Furthermore, discrepancy can be misaligned with the true reward objective, and therefore does not adequately explain the widely observed empirical advantage of online over offline IL methods.

Motivated by these limitations, we develop a new perspective on the advantage of online over offline IL in non-realizable settings by directly accounting for the *reward structure*. For offline IL, we identify a fundamental information-theoretic barrier under non-realizability even in contextual bandits – a barrier distinct from classical error-accumulation arguments [48, 50]. We show in Theorem 1 that the sample complexity of any offline IL algorithm that learns from i.i.d. expert demonstrations scales with a coverage coefficient C^{e}_{∞} (Eq. (4)), which can be interpreted as a measure of the mismatch between the expert and student policies. For online IL, we characterize the interplay between (i) the expert signal, (ii) the misspecified student policy class, and (iii) the reward function, and identify a general condition under which online IL is provably efficient despite severe misspecification (Theorem 2).

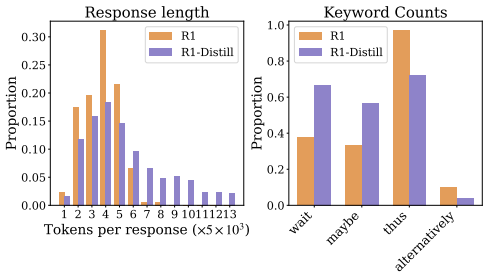
5.1 Limitations of Discrepancy-based Analyses in Non-Realizable Settings

A common approach to deriving IL guarantees is to upper bound the performance suboptimality by a distributional discrepancy between the expert policy π^{e} and the learned policy $\hat{\pi} \in \Pi$ [45, 16, 47]. For instance, one typically argues that $V(\pi^{\mathrm{e}}) - V(\hat{\pi}) \leq \mathbb{E}_{x\sim\mathcal{D}_x}[D_{\mathrm{TV}}(\pi^{\mathrm{e}}(\cdot \mid x), \hat{\pi}(\cdot \mid x))]$, where D_{TV} can be further controlled by other statistical distances or divergences.[5]

In realizable settings, the approximation term vanishes: $\inf_{\pi\in\Pi} \mathbb{E}_{x\sim\mathcal{D}_x}[D_{\mathrm{TV}}(\pi^{\mathrm{e}}(\cdot \mid x), \pi(\cdot \mid x))] = 0$, so minimizing discrepancy drives the performance suboptimality $V(\pi^{\mathrm{e}}) - V(\hat{\pi})$ to zero. In non-realizable settings, however, even the best-in-class discrepancy is strictly positive, so discrepancy-

[4] Gu et al. [20], Yang et al. [61], and Yang et al. [62] study strong-to-weak distillation (distilling a smaller model from a much larger expert), and their setup therefore falls into the non-realizable setting.

[5] Foster et al. [16] give a tighter upper bound based on Hellinger distance than the direct total variation bound for deterministic experts or experts with bounded reward variance (Theorems 2.1 and 3.1). Our discussion is not sensitive to this distinction.



(a) Distributional discrepancy proxies.

Metric	Value (%)
Length-induced TV distance	29.99
Keyword-frequency gap (*wait*)	29.12
Keyword-frequency gap (*maybe*)	23.29
Keyword-frequency gap (*thus*)	24.88
Keyword-frequency gap (*alternatively*)	6.20
Performance gap	5.46

(b) Discrepancy vs. performance gap.

Figure 3: ***Bounding performance gap by expert-student discrepancy can be overly pessimistic.*** *We compare DeepSeek-R1-0528 (expert) and DeepSeek-R1-0528-Qwen3-8B (student) on the AIME benchmark.* ***(a)*** *The expert and student differ substantially in response-length distributions and keyword frequencies.* ***(b)*** *Length-induced TV distance and keyword-frequency gaps are lower bounds of the TV distance between response distributions, yet already far exceed the performance gap. This suggests that discrepancy-based guarantees can be overly loose.*

based bounds cannot certify suboptimality below this irreducible distance. While such bounds are tight in the worst case over all possible reward functions, they can be overly pessimistic for "average-case" reward functions: a large expert-student discrepancy does not necessarily imply a large performance gap. This is especially relevant for language models, where discrepancies may arise from wording, style, or reasoning traces that are not related to correctness. For example, a strong expert may solve a math problem directly, while a weaker student may reach the same answer through trial and error; their response distributions can differ substantially despite similar success rates. We verify this empirically in LLM post-training below.

Furthermore, discrepancy-based analyses do not explain the practical gains of online IL over offline IL. Prior work has shown that offline ρ-estimator behavior cloning is already statistically optimal for minimizing the discrepancy even in non-realizable settings [47, Theorem 3.1]; see Section 3 for more details. This suggests that the advantage of online interaction, when it arises, is not in identifying the policy closest to the expert, but in identifying the student policy with higher reward under a particular reward function. These limitations motivate us to go beyond the discrepancy-based analysis and develop a finer-grained, reward-dependent analysis for non-realizable settings.

Empirical Validation We empirically validate that expert–student distributional discrepancy can be much larger than the corresponding performance gap. We take DeepSeek-R1-0528 (671B) as the expert and DeepSeek-R1-0528-Qwen3-8B as the student [10], which is distilled from DeepSeek-R1-0528. We collect 840 responses from each model on the AIME 2024 and 2025 benchmarks (details in Section D.6). As shown in Figure 3, the two models exhibit substantial differences in their response distributions, as reflected by two simple and interpretable features: response length (discretized into 5,000-token bins), and the frequency of reasoning-related keywords [68, 18]. These metrics are lower bounds of the total variation distance between the response distributions, since distinct responses may share the same length bin or selected keywords. Thus, the true expert-student discrepancy can only be larger than reported. Since even these lower bounds exceed the observed performance gap by a wide margin, it validates that distributional discrepancies can be much larger than performance differences.

These limitations motivate us to go beyond the reward-agnostic discrepancy-based analysis in non-realizable settings, and instead ask when offline IL becomes inefficient (Section 5.2) and when online IL can yield improvements (Section 5.3) under a specific reward function.

5.2 Information-theoretic Lower Bounds for Offline IL in Non-Realizable Settings

We now present an information-theoretic limitation of offline IL in non-realizable settings. Intuitively, the hard case arises when the expert's responses cannot be faithfully represented by the student policy class, while correct responses that are representable by the student are rarely revealed in expert samples.

We formalize this intuition in Theorem 1, showing that even when the student class Π contains a policy that has the same performance as the expert π^e, learning from i.i.d. expert demonstrations can still be statistically hard in non-realizable settings. Specifically, for any offline IL algorithm that observes only i.i.d. expert demonstrations, the sample complexity required to achieve a vanishing

performance suboptimality must scale with the coverage coefficient C^{e}_{∞} defined as

$$C^{\mathrm{e}}_{\infty} \coloneqq \sup_{x\in\mathcal{X},y\in\mathcal{Y}} \frac{\pi^{\star}(y \mid x)}{\pi^{\mathrm{e}}(y \mid x)}, \tag{4}$$

where $\pi^{\star} \in \arg\max_{\pi\in\Pi} V(\pi)$ [39, 7, 46]. This barrier is **specific to non-realizability**: $C^{\mathrm{e}}_{\infty} = 1$ under realizability since one can take $\pi^{\star} = \pi^{e}$, but can be arbitrarily large under misspecification, serving as a measure of the mismatch between the expert and $\pi^{\star}$.

Limitation of offline IL in non-realizable settings

Theorem 1 (Informal version of Theorem 4)**.** *For any coverage parameter $C^{\mathrm{e}}_{\infty} > 1$ and any (possibly randomized) offline IL learner $\mathcal{A}$ using N i.i.d. expert demonstrations, there exist a contextual bandit problem $(\mathcal{D}_x, r)$, a policy class Π, and an expert $\pi^{\mathrm{e}} \notin \Pi$ with coverage coefficient C^{e}_{∞} relative to the best student policy $\pi^{\star} \in \Pi$, such that $\pi^{\star}$ matches the expert's performance, $V(\pi^{\star}) = V(\pi^{\mathrm{e}})$, yet the learner's output $\hat{\pi} \in \Pi$ satisfies*

$$\mathbb{E}[V(\pi^{\mathrm{e}}) - V(\hat{\pi})] \gtrsim \min\left\{1, \sqrt{\frac{C^{\mathrm{e}}_{\infty} \log|\Pi|}{N}}\right\},$$

where the expectation is over the demonstrations and the learner's internal randomness.

The proof constructs a hard family via Assouad's method [66]; see Appendix B.1.

5.3 A Structural Condition for Effective Online IL in Non-Realizable Settings

Online interaction can alleviate the information bottleneck of learning from i.i.d. expert demonstrations by querying the expert on learner-induced inputs. However, these queries may fall in regions rarely sampled by the expert, where the expert signal can be unreliable. Thus, a high-performing expert is not necessarily a useful teacher for a misspecified student; its feedback must help the learner identify high-performing policies within the restricted student class Π. Motivated by Section 5.1, we seek structural conditions characterizing the interplay of misspecification structure and reward structure, under which online IL achieves small performance suboptimality $\max_{\pi\in\Pi} V(\pi) - V(\hat{\pi})$, even when the learned policy remains far from the expert in distributional discrepancy. We consider a setting in which the feedback on a response y is given by the expert's conditional probability $\pi^{\mathrm{e}}(y \mid x)$, and study a specific class of online imitation learning objectives (Definition 2):

$$\max_{\pi\in\Pi} J_{\mathrm{on}}(\pi) \coloneqq \mathbb{E}_{x\sim\mathcal{D}_x}\mathbb{E}_{y\sim\pi(\cdot|x)}[f(\pi^{\mathrm{e}}(y \mid x))]\,,$$

where $f : \mathbb{R} \to \mathbb{R}$ is a shaping function. The widely used on-policy distillation algorithm uses $f(\pi^{\mathrm{e}}(y \mid x)) = \log(\pi^{\mathrm{e}}(y \mid x))$ together with an additional entropy regularization term (Section 2).

A Motivating Synthetic Example. We use a simple toy 2D Gaussian example (Figure 4) to motivate our condition. Actions are $a = (x, y) \in \mathbb{R}^2$, policies are Gaussians $\pi_{\mu} = \mathcal{N}(\mu, I_2)$, and the reward is $r(x, y) = \mathbf{1}[y \geq 0]$. The expert is $\pi^{\mathrm{e}} = \mathcal{N}((-2, 2), I_2)$, and we compare two student classes $\Pi_A = \{\mathcal{N}(\mu, I_2) : \mu_x = 0\}$ and $\Pi_B = \{\mathcal{N}(\mu, I_2) : \mu_y = \mu_x\}$. Both classes are misspecified, yet the expected rewards resulting from IL (with reverse-KL objective as in on-policy distillation) differ sharply. Let $\hat{\pi}_A, \hat{\pi}_B$ be the reverse-KL projections of π^{e} onto Π_A and Π_B, respectively. Then $V(\hat{\pi}_A) \approx 0.977, V(\hat{\pi}_B) = 0.5$, corresponding to excess risks 0.023 and 0.5, respectively.

The gap between Π_A and Π_B arises from their different *misspecification structures relative to the reward*. The reward $r(x, y) = \mathbf{1}[y \geq 0]$ depends only on the y-coordinate. In Π_A, misspecification is confined to the reward-irrelevant x-direction: the class fixes $\mu_x = 0$ but leaves μ_y free. Thus, the reverse-KL projection matches the expert's reward-relevant mean $\mu^{\mathrm{e}}_y = 2$, yielding high expected reward. In contrast, Π_B couples the two coordinates through $\mu_y = \mu_x$. As a result, matching the expert distribution trades off the reward-relevant y-direction against the reward-irrelevant x-direction, leading to suboptimal expected reward.

A General Sufficient Condition: Misspecification–Reward Alignment. As illustrated in the above toy example, to characterize when IL is effective in non-realizable settings, one must account for misspecification structure between the student and expert under a particular reward function. We now propose a general condition, which we term the *Misspecification–Reward Alignment Condition* for

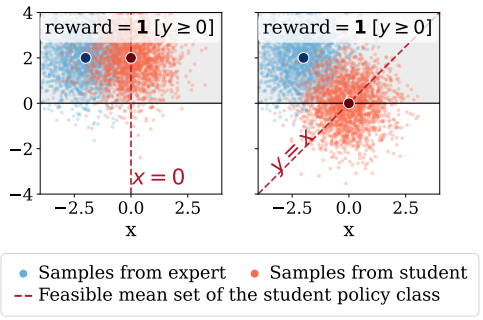


Figure 4: **Synthetic example** with 2d actions, Gaussian policies, and reward $r(x, y) = \mathbf{1}[y \geq 0]$. Expert $\pi^{\mathrm{e}} = \mathcal{N}((-2, 2), I_2)$. *Left:* student class Π_A: $\mu_x = 0$ yields a reverse-KL projection with $V(\hat{\pi}_A) \approx 0.977$. *Right:* student class Π_B: $\mu_y = \mu_x$ yields $V(\hat{\pi}_B) = 0.5$. The performance gap arises from different **misspecification structures relative to the reward**.

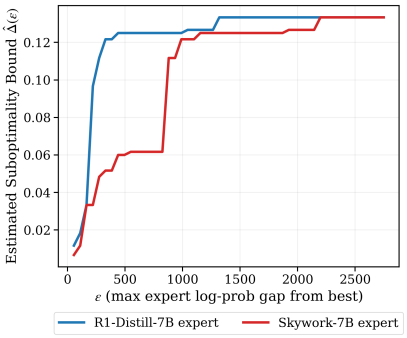


Figure 5: **Estimated suboptimality bound.** We empirically estimate the suboptimality bound in Theorem 2 $\hat{\Delta}(\epsilon) := \min_{\alpha>0} \frac{\epsilon+\hat{\delta}_\alpha(\epsilon)}{\alpha}$ through Monte Carlo. Consistent with Theorem 2, the expert leading to a better student (i.e., Skywork-7B) has a smaller suboptimality bound at every ϵ.

the online IL objective considered in Definition 2. Under this condition, approximate optimality for the online IL objective translates into a guarantee on the expected reward suboptimality.

Misspecification-Reward Alignment Condition

Without loss of generality, we can decompose the expert signal as:

$$f(\pi^{\mathrm{e}}(y \mid x)) = \alpha\, r(x, y) + h(x, y) + b(x), \tag{5}$$

where $\alpha > 0$, $h : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, and $b : \mathcal{X} \to \mathbb{R}$. The term $\alpha r(x, y)$ is the reward-aligned component, with α quantifying the strength of alignment. The residual $h(x, y)$ captures expert preferences beyond reward, and $b(x)$ is an action-independent offset.

Assumption 1. *Let $\pi^\star \in \Pi$ be a reference policy. For any $\epsilon > 0$, assume that for all $\pi \in \Pi$ satisfying $J_{\mathrm{on}}(\pi) \geq J_{\mathrm{on}}(\pi^\star) - \epsilon$,*

$$\mathbb{E}_{x\sim\mathcal{D}_x}\left[\mathbb{E}_{y\sim\pi(\cdot|x)}h(x, y) \; - \mathbb{E}_{y\sim\pi^\star(\cdot|x)}h(x, y)\right] \leq \delta_\alpha(\epsilon).$$

The decomposition in Eq. (5) can be made for any fixed $\alpha > 0$: for any expert signal $f(\pi^{\mathrm{e}}(y \mid x))$ and any x-dependent offset $b(x)$, one can define $h(x, y) := f(\pi^{\mathrm{e}}(y \mid x)) - \alpha r(x, y) - b(x)$. Therefore, the decomposition is without loss of generality; the actual assumption lies in the alignment condition on the residual term h (Assumption 1): within the ϵ-optimal set under J_{on}, the residual term cannot inflate $J_{\mathrm{on}}(\pi)$ for a reward-suboptimal policy. This condition allows us to translate near-optimality for the online IL objective into a guarantee on the reward performance of the learned policy.

Theorem 2. *Under Assumption 1, let $\pi \in \Pi$ be any ϵ-suboptimal policy for the online IL objective Eq. (2), i.e., $J_{\mathrm{on}}(\pi) \geq \max_{\pi'\in\Pi} J_{\mathrm{on}}(\pi') - \epsilon$. Then π satisfies:*

$$V(\pi) \geq V(\pi^\star) - \tfrac{\epsilon+\delta_\alpha(\epsilon)}{\alpha}.$$

The proof of Theorem 2 is provided in Appendix B.2. To better understand when the Misspecification–Reward Alignment condition (Assumption 1) holds, we highlight two representative scenarios.

1. When the reshaped expert signal $f(\pi^{\mathrm{e}}(y \mid x))$ is uniformly aligned well with the reward $r(x, y)$, Assumption 1 holds essentially independently of the misspecified student class. Concretely, if $|h(x, y)| \ll \alpha$ for all $x \in \mathcal{X}, y \in \mathcal{Y}$, then the suboptimality bound satisfies $\frac{\epsilon+\delta_\alpha(\epsilon)}{\alpha} \ll 1$ as $\epsilon \to 0$. Direct-OPD [14] provides a concrete example: it uses the log-policy shift $f(\pi^{\mathrm{e}}(y \mid x)) := \log \frac{\pi^{\mathrm{e}}(y|x)}{\pi_0^{\mathrm{e}}(y|x)}$ as the expert signal, where π_0^{e} and π^{e} denote the policies before and after KL-regularized RL, respectively. If π^{e} is the exact optimizer for reward r with KL coefficient

$\beta > 0$ and reference policy π_0^{e}, then

$$\log \frac{\pi^{\mathrm{e}}(y \mid x)}{\pi_0^{\mathrm{e}}(y \mid x)} = \frac{1}{\beta} r(x, y) - \log Z(x), \qquad Z(x) = \mathbb{E}_{y \sim \pi_0^{\mathrm{e}}(\cdot|x)} \left[\exp\left(\frac{r(x, y)}{\beta} \right) \right].$$

Thus, Eq. (5) holds with $\alpha = 1/\beta$, $b(x) = -\log Z(x)$, and $h(x, y) = 0$.

2. Although the expert signal $f(\pi^{\mathrm{e}}(y \mid x))$ encodes preferences beyond the reward, captured by the residual $h(x, y)$, the misspecified student is unable to represent these preferences. For example, the expert may prefer more concise responses, while any student policy that achieves high performance requires longer chains of reasoning due to limited capability. In this case, the reward-irrelevant preferences (e.g., response length) in the expert signal do not substantially distinguish among near-optimal policies in the student class. Formally, for any $\pi_1, \pi_2 \in \Pi$ satisfying $J_{\mathrm{on}}(\pi_i) \geq \max_{\pi \in \Pi} J_{\mathrm{on}}(\pi) - \epsilon$ for $i = 1, 2$, we have

$$\left| \mathbb{E}_{x \sim \mathcal{D}_x} \left[\mathbb{E}_{y \sim \pi_1(\cdot|x)} h(x, y) - \mathbb{E}_{y \sim \pi_2(\cdot|x)} h(x, y) \right] \right| \ll 1.$$

This regime also explains why the expert signal is effective for Π_A in the 2D Gaussian example.

We discuss the comparison between Theorem 2 and the classical view of online IL as mitigating error accumulation in Appendix B.3.

Empirical Validation. We examine whether the suboptimality bound in Theorem 2 reflects differences in distillation performance across experts. Following Li et al. [34], we perform OPD from the same base model, DeepSeek-R1-Distill-Qwen-1.5B [10], with two different experts: DeepSeek-R1-Distill-Qwen-7B (R1-Distill-7B) [10] and Skywork-OR1-Math-7B (Skywork-7B) [22]. On AIME 2024 and AIME 2025, distillation from Skywork-7B improves the student's average performance from 25.0% to 32.8%, whereas distillation from R1-Distill-7B improves it to 27.1% (averaged over three runs).

To empirically assess Assumption 1, we estimate the suboptimality bound in Theorem 2. The decomposition in Eq. (5) is not unique: each choice of $\alpha > 0$ induces a residual upper bound $\delta_\alpha(\epsilon)$. Since the bound in Theorem 2 depends on this choice through $(\epsilon + \delta_\alpha(\epsilon))/\alpha$, we further minimize the estimated bound over α. We estimate the bound using the following procedure. For each expert, we perform three on-policy distillation runs. We collect checkpoints from all distillation trajectories across the two experts, with checkpoints saved every 10 steps, and use their union as a proxy policy class $\widehat{\Pi}$. We take the checkpoint with the highest empirical reward as the reference policy $\widehat{\pi}^\star$. Then, for each expert separately and for each $\epsilon > 0$, we consider the set of checkpoints whose empirical online IL objective is within ϵ of the maximum over the collected checkpoints: $\widehat{\Pi}_\epsilon := \left\{ \pi \in \widehat{\Pi} : \widehat{J}_{\mathrm{on}}(\pi) \geq \max_{\pi' \in \widehat{\Pi}} \widehat{J}_{\mathrm{on}}(\pi') - \epsilon \right\}$. We estimate the residual upper bound $\delta_\alpha(\epsilon)$ in Assumption 1 over $\widehat{\Pi}_\epsilon$ by Monte Carlo, and denote the resulting estimate by $\widehat{\delta}_\alpha(\epsilon)$. We then report the estimated bound

$$\widehat{\Delta}(\epsilon) := \min_{\alpha > 0} \frac{\epsilon + \widehat{\delta}_\alpha(\epsilon)}{\alpha}.$$

Details are provided in Section D.7.

As shown in Figure 5, Skywork-7B yields a much smaller estimated bound than R1-Distill-7B across ϵ, consistent with its larger distillation improvement. The relatively large range of ϵ is expected, since the online IL objective is based on sequence-level log probabilities, whose scale grows approximately linearly with response length.

5.3.1 Finite-Sample Guarantee

In practice, on-policy distillation is implemented using policy-gradient methods, and finite-sample analysis of such methods beyond tabular and linear settings remains highly challenging [2]. To isolate the potential statistical benefits of online interaction from these optimization challenges, we instead analyze an online policy-elimination algorithm for the online IL objective Eq. (2). The algorithm estimates each policy's objective difference relative to the base policy $J_{\mathrm{on}}(\pi) - J_{\mathrm{on}}(\pi^{\mathrm{base}})$, rather than $J_{\mathrm{on}}(\pi)$. We defer details and proof to Appendix B.4.

The efficient training of online and on-policy algorithms for LLM post-training crucially relies on a strong base model to obtain positive feedback with nontrivial probability [17, 68]. Following prior work, we formalize the role of the base model through *coverage coefficients* [59, 25, 17, 26].

Our guarantee depends on a weaker χ^2-coverage coefficient, which is upper-bounded by L_∞-type coverage coefficients (as in C^{e}_∞):

$$D^{\mathrm{base}}(\pi^\star) := \mathbb{E}_{x\sim\mathcal{D}_x} D_{\chi^2}\big(\pi^\star(\cdot \mid x) \,\big\|\, \pi^{\mathrm{base}}(\cdot \mid x)\big), \tag{6}$$

where $\pi^\star \in \arg\max_{\pi\in\Pi} J_{\mathrm{on}}(\pi)$.

Besides coverage, we also require a boundedness condition on the shaped expert signal, which plays the same role as the standard bounded-reward condition in finite-sample RL and off-policy evaluation analyses.

Assumption 2 (Bounded shaped expert signal)**.** *For all $x \in \mathcal{X}$ and $y \in \mathcal{Y}$,*

$$|f(\pi^{\mathrm{e}}(y \mid x))| \le V_{\max}.$$

Theorem 3 (Informal version of Theorem 5)**.** *Suppose Assumption 2 holds. Run Algorithm 1 with a total budget of N expert queries. Then, with probability at least $1-\delta$, its output $\hat{\pi}$ satisfies*

$$J_{\mathrm{on}}(\pi^\star) - J_{\mathrm{on}}(\hat{\pi}) \le \widetilde{O}\left(V_{\max}\sqrt{\frac{2\,D^{\mathrm{base}}(\pi^\star)}{N}\,\max\{\bar{\rho}^\star_N,\ \tfrac{1}{N^2}\}}\right)$$

where $\bar{\rho}^\star_N$ is the average coverage improvement factor defined in Eq. (9) *and satisfies $\bar{\rho}^\star_N \le 1$.*

The bound depends on $D^{\mathrm{base}}(\pi^\star)\bar{\rho}^\star_N$, where $\bar{\rho}^\star_N \le 1$ captures potential coverage gains as the student improves and the candidate set shrinks. Even in the worst case, it depends on base coverage D^{base} rather than the expert coverage coefficient C^{e}_∞ in the offline lower bound (Theorem 1). Since the base policy is the model before fine-tuning, it is expected to cover policies reachable during training better than an external expert. We stress, however, that improved coverage is a structural condition we identify for efficient online IL, not a universal guarantee. Indeed, prior empirical studies find that on-policy distillation does not outperform SFT without a sufficiently strong base model [28].

We also note that the hard instances in Theorem 1 can also be constructed to satisfy Assumption 1 and Assumption 2, yielding a clean offline–online separation within the same regime. See Remark 3 for further discussion.

6 Conclusion and Discussion

In this work, we show that misspecification is a key factor driving the empirical performance gap between offline and online IL methods in LLM post-training. Empirically, under realizability, offline IL already recovers expert performance, leaving no room for online interaction to help. Thus, the gap often observed between SFT and online methods such as RLVR or on-policy distillation need not reflect an inherent limitation of SFT. Instead, it can arise from the offline data distribution induced by a non-realizable expert, which may be poorly suited for the target student class. On the theoretical front, we characterize an information-theoretic limitation of offline IL in non-realizable settings, and provide structural conditions under which online IL can still be effective even when the discrepancy between expert and student is large. This perspective complements prior explanations based on error accumulation. One question beyond our current framework is why on-policy distillation can be more efficient than RLVR. Addressing this may require analyzing the multi-step structure of language generation and the role of dense reward signals.

Acknowledgments

The authors thank Audrey Huang, Abhishek Panigrahi, Aditi Raghunathan, Dhruv Rohatgi, Kaiyue Wen and Chen Wu for valuable discussions and feedback. B.L. is supported by the Kempner Fellowship from the Kempner Institute at Harvard. A.R. and J.G. are supported in part by NSF awards IIS-2211907, CCF-2238523, IIS-2403275, an Amazon Research Award, ONR award N000142512124, a Google Research Scholar Award, and an OpenAI Superalignment Fast Grant.

References

[1] Abbeel, P. and Ng, A. Y. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, pp. 1, 2004.

[2] Agarwal, A., Kakade, S. M., Lee, J. D., and Mahajan, G. On the theory of policy gradient methods: Optimality, approximation, and distribution shift. *Journal of Machine Learning Research*, 22(98):1–76, 2021.

[3] Agarwal, R., Vieillard, N., Zhou, Y., Stanczyk, P., Garea, S. R., Geist, M., and Bachem, O. On-policy distillation of language models: Learning from self-generated mistakes. In *The Twelfth International Conference on Learning Representations*, 2024.

[4] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[5] Chang, J., Uehara, M., Sreenivas, D., Kidambi, R., and Sun, W. Mitigating covariate shift in imitation learning via offline data with partial coverage. *Advances in Neural Information Processing Systems*, 34:965–979, 2021.

[6] Chen, H., Razin, N., Narasimhan, K., and Chen, D. Retaining by doing: The role of on-policy data in mitigating forgetting. *arXiv preprint arXiv:2510.18874*, 2025.

[7] Chen, J. and Jiang, N. Information-theoretic considerations in batch reinforcement learning. In *International conference on machine learning*, pp. 1042–1051. PMLR, 2019.

[8] Chu, T., Zhai, Y., Yang, J., Tong, S., Xie, S., Schuurmans, D., Le, Q. V., Levine, S., and Ma, Y. Sft memorizes, rl generalizes: A comparative study of foundation model post-training. *arXiv preprint arXiv:2501.17161*, 2025.

[9] Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., and Schulman, J. Training verifiers to solve math word problems. *arXiv preprint arXiv: 2110.14168*, 2021.

[10] DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL `https://arxiv.org/abs/2501.12948`.

[11] Dudik, M., Hsu, D., Kale, S., Karampatziakis, N., Langford, J., Reyzin, L., and Zhang, T. Efficient optimal learning for contextual bandits. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, UAI'11, pp. 169–178, 2011.

[12] Espinosa-Dice, N., Choudhury, S., Sun, W., and Swamy, G. Efficient imitation under misspecification. *arXiv preprint arXiv:2503.13162*, 2025.

[13] Face, H. Open r1: A fully open reproduction of deepseek-r1, 2025.

[14] Feng, S., Gao, H.-a., Chi, H., Wu, H., Zhang, Z., Jiang, Z., He, B., Ma, W.-Y., Zhang, Y.-Q., and Zhou, H. Weak-to-strong generalization via direct on-policy distillation. *arXiv preprint arXiv:2607.05394*, 2026.

[15] Foster, D. J. and Rakhlin, A. Foundations of reinforcement learning and interactive decision making. *arXiv preprint arXiv:2312.16730*, 2023.

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

[17] Foster, D. J., Mhammedi, Z., and Rohatgi, D. Is a good foundation necessary for efficient reinforcement learning? the computational role of the base model in exploration. *arXiv preprint arXiv:2503.07453*, 2025.

[18] Gandhi, K., Chakravarthy, A., Singh, A., Lile, N., and Goodman, N. D. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective stars. *arXiv preprint arXiv:2503.01307*, 2025.

[19] Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[20] Gu, Y., Dong, L., Wei, F., and Huang, M. Minillm: Knowledge distillation of large language models. In *The Twelfth International Conference on Learning Representations*, 2024.

[21] Gudibande, A., Wallace, E., Snell, C., Geng, X., Liu, H., Abbeel, P., Levine, S., and Song, D. The false promise of imitating proprietary language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=Kz3yckpCN5.

[22] He, J., Liu, J., Liu, C. Y., Yan, R., Wang, C., Cheng, P., Zhang, X., Zhang, F., Xu, J., Shen, W., et al. Skywork open reasoner 1 technical report. *arXiv preprint arXiv:2505.22312*, 2025.

[23] Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[24] Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., and Steinhardt, J. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

[25] Huang, A., Block, A., Foster, D. J., Rohatgi, D., Zhang, C., Simchowitz, M., Ash, J. T., and Krishnamurthy, A. Self-improvement in language models: The sharpening mechanism. In *The Thirteenth International Conference on Learning Representations*, 2025.

[26] Huang, A., Zhan, W., Xie, T., Lee, J. D., Sun, W., Krishnamurthy, A., and Foster, D. J. Correcting the mythos of kl-regularization: Direct alignment without overoptimization via chi-squared preference optimization. In *The Thirteenth International Conference on Learning Representations*, 2025.

[27] Jain, S., Kirk, R., Lubana, E. S., Dick, R. P., Tanaka, H., Grefenstette, E., Rocktäschel, T., and Krueger, D. S. Mechanistically analyzing the effects of fine-tuning on procedurally defined tasks. *arXiv preprint arXiv:2311.12786*, 2023.

[28] Jia, N., Yang, H., Ma, X., Lian, J., Zhang, S., Zhang, W., Zeng, K., Cai, X., and Sun, Z. Asymmetric on-policy distillation: Bridging exploitation and imitation at the token level. *arXiv preprint arXiv:2605.06387*, 2026.

[29] Jiang, W., Lu, Y., Lin, H., Han, X., and Sun, L. Teach small models to reason by curriculum distillation. In Christodoulopoulos, C., Chakraborty, T., Rose, C., and Peng, V. (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 7412–7422, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.376. URL https://aclanthology.org/2025.emnlp-main.376/.

[30] Kim, Y. and Rush, A. M. Sequence-level knowledge distillation. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pp. 1317–1327, 2016.

[31] Kumar, A., Raghunathan, A., Jones, R. M., Ma, T., and Liang, P. Fine-tuning can distort pretrained features and underperform out-of-distribution. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=UYneFzXSJWh.

[32] Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C. H., Gonzalez, J. E., Zhang, H., and Stoica, I. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

[33] Li, Y., Yue, X., Xu, Z., Jiang, F., Niu, L., Lin, B. Y., Ramasubramanian, B., and Poovendran, R. Small models struggle to learn from strong reasoners. *arXiv preprint arXiv:2502.12143*, 2025.

[34] Li, Y., Zuo, Y., He, B., Zhang, J., Xiao, C., Qian, C., Yu, T., Gao, H.-a., Yang, W., Liu, Z., et al. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe. *arXiv preprint arXiv:2604.13016*, 2026.

[35] Liu, A., Mei, A., Lin, B., Xue, B., Wang, B., Xu, B., Wu, B., Zhang, B., Lin, C., Dong, C., et al. Deepseek-v3. 2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*, 2025.

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

[37] Luo, M., Tan, S., Wong, J., Shi, X., Tang, W. Y., Roongta, M., Cai, C., Luo, J., Zhang, T., Li, L. E., et al. Deepscaler: Surpassing o1-preview with a 1.5 b model by scaling rl. *Notion Blog*, 2025.

[38] Muennighoff, N., Yang, Z., Shi, W., Li, X. L., Fei-Fei, L., Hajishirzi, H., Zettlemoyer, L., Liang, P., Candès, E., and Hashimoto, T. B. s1: Simple test-time scaling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 20286–20332, 2025.

[39] Munos, R. Error bounds for approximate policy iteration. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, pp. 560–567, 2003.

[40] Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[41] Pan, J., Zhang, J., Wang, X., Yuan, L., Peng, H., and Suhr, A. Tinyzero. https://github.com/Jiayi-Pan/TinyZero, 2025. Accessed: 2025-01-24.

[42] Prakash, N., Shaham, T. R., Haklay, T., Belinkov, Y., and Bau, D. Fine-tuning enhances existing mechanisms: A case study on entity tracking. *arXiv preprint arXiv:2402.14811*, 2024.

[43] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

[44] Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

[45] Rajaraman, N., Yang, L., Jiao, J., and Ramchandran, K. Toward the fundamental limits of imitation learning. *Advances in Neural Information Processing Systems*, 33:2914–2924, 2020.

[46] Rashidinejad, P., Zhu, B., Ma, C., Jiao, J., and Russell, S. Bridging offline reinforcement learning and imitation learning: A tale of pessimism. *Advances in Neural Information Processing Systems*, 34:11702–11716, 2021.

[47] Rohatgi, D., Block, A., Huang, A., Krishnamurthy, A., and Foster, D. J. Computational-statistical tradeoffs at the next-token prediction barrier: Autoregressive and imitation learning under misspecification. *arXiv preprint arXiv:2502.12465*, 2025.

[48] Ross, S. and Bagnell, D. Efficient reductions for imitation learning. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 661–668. JMLR Workshop and Conference Proceedings, 2010.

[49] Ross, S. and Bagnell, J. A. Reinforcement and imitation learning via interactive no-regret learning. *arXiv preprint arXiv:1406.5979*, 2014.

[50] Ross, S., Gordon, G., and Bagnell, D. 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*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011.

[51] Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y. K., Wu, Y., and Guo, D. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv: 2402.03300*, 2024.

[52] Shenfeld, I., Pari, J., and Agrawal, P. Rl's razor: Why online reinforcement learning forgets less. *arXiv preprint arXiv:2509.04259*, 2025.

[53] Sheng, G., Zhang, C., Ye, Z., Wu, X., Zhang, W., Zhang, R., Peng, Y., Lin, H., and Wu, C. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*, 2024.

[54] Song, Y., Rohatgi, D., Singh, A., and Bagnell, J. A. To distill or decide? understanding the algorithmic trade-off in partially observable reinforcement learning. *arXiv preprint arXiv: 2510.03207*, 2025.

[55] Syed, U. and Schapire, R. E. A game-theoretic approach to apprenticeship learning. *Advances in neural information processing systems*, 20, 2007.

[56] Team, Q. Qwen2. 5: A party of foundation models, september 2024. *URL https://qwenlm. github. io/blog/qwen2*, 5(4), 2024.

[57] von Werra, L., Belkada, Y., Tunstall, L., Beeching, E., Thrush, T., Lambert, N., Huang, S., Rasul, K., and Gallouédec, Q. TRL: Transformers Reinforcement Learning, 2020. URL `https://github.com/huggingface/trl`.

[58] Xiao, B., Xia, B., Yang, B., Gao, B., Shen, B., Zhang, C., He, C., Lou, C., Luo, F., Wang, G., et al. Mimo-v2-flash technical report. *arXiv preprint arXiv:2601.02780*, 2026.

[59] Xie, T., Foster, D. J., Krishnamurthy, A., Rosset, C., Awadallah, A. H., and Rakhlin, A. Exploratory preference optimization: Harnessing implicit q*-approximation for sample-efficient rlhf. In *The Thirteenth International Conference on Learning Representations*, 2025.

[60] Xiong, W., Dong, H., Ye, C., Wang, Z., Zhong, H., Ji, H., Jiang, N., and Zhang, T. Iterative preference learning from human feedback: Bridging theory and practice for RLHF under KL-constraint. volume 235 of *Proceedings of Machine Learning Research*. PMLR, 2024.

[61] Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[62] Yang, W., Liu, W., Xie, R., Yang, K., Yang, S., and Lin, Y. Learning beyond teacher: Generalized on-policy distillation with reward extrapolation. *arXiv preprint arXiv:2602.12125*, 2026.

[63] Ye, T., Dong, L., Chi, Z., Wu, X., Huang, S., and Wei, F. Black-box on-policy distillation of large language models. *CoRR*, abs/2511.10643, 2025. doi: 10.48550/ARXIV.2511.10643. URL `https://doi.org/10.48550/arXiv.2511.10643`.

[64] Ye, Y., Huang, Z., Xiao, Y., Chern, E., Xia, S., and Liu, P. Limo: Less is more for reasoning. *arXiv preprint arXiv:2502.03387*, 2025.

[65] Yeo, E., Tong, Y., Niu, M., Neubig, G., and Yue, X. Demystifying long chain-of-thought reasoning in llms. *arXiv preprint arXiv:2502.03373*, 2025.

[66] Yu, B. *Assouad, Fano, and Le Cam*, pp. 423–435. Springer New York, New York, NY, 1997. ISBN 978-1-4612-1880-7. doi: 10.1007/978-1-4612-1880-7_29. URL https://doi.org/10.1007/978-1-4612-1880-7_29.

[67] Yu, Q., Zhang, Z., Zhu, R., Yuan, Y., Zuo, X., Yue, Y., Dai, W., Fan, T., Liu, G., Liu, L., et al. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244, 2026.

[68] Zeng, W., Huang, Y., Liu, Q., Liu, W., He, K., Ma, Z., and He, J. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild. *arXiv preprint arXiv:2503.18892*, 2025.

[69] Zhao, S., Xie, Z., Liu, M., Huang, J., Pang, G., Chen, F., and Grover, A. Self-distilled reasoner: On-policy self-distillation for large language models. *CoRR*, abs/2601.18734, 2026. doi: 10.48550/ARXIV.2601.18734. URL https://doi.org/10.48550/arXiv.2601.18734.

[70] Zheng, Y., Zhang, R., Zhang, J., Ye, Y., Luo, Z., Feng, Z., and Ma, Y. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand, 2024. Association for Computational Linguistics. URL http://arxiv.org/abs/2403.13372.

[71] Zhou, C., Liu, P., Xu, P., Iyer, S., Sun, J., Mao, Y., Ma, X., Efrat, A., Yu, P., Yu, L., et al. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36: 55006–55021, 2023.

[72] Ziebart, B. D., Maas, A. L., Bagnell, J. A., Dey, A. K., et al. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, pp. 1433–1438. Chicago, IL, USA, 2008.

A Additional Related Work

More works on Imitation Learning for LLM Post-Training.

Previous work shows that SFT amplifies existing mechanisms and knowledge acquired during pretraining [27, 42]. With a small amount of carefully curated data, SFT can already yield strong instruction-following performance and even unlock more advanced capabilities such as reasoning [71, 64, 38]. Recent studies comparing SFT with reinforcement-learning (RL) fine-tuning suggest that SFT can exhibit weaker out-of-domain generalization [8] and more severe catastrophic forgetting than RL [6, 52]. However, for relatively weak base models, SFT can be more sample-efficient than RL for improving reasoning performance [10]. Moreover, SFT (or mid-training) often serves as a warm-start for subsequent RL fine-tuning by improving the initial policy [65, 18].

Recent work has further extended on-policy distillation to black-box settings without access to expert logits [63], multi-expert distillation [58], and self-distillation [69]. Feng et al. [14] use the RL-induced policy shift $\log \frac{\pi^{\mathrm{e}}(y|x)}{\pi_0^{\mathrm{e}}(y|x)}$ as the expert signal, where π_0^{e} and π^{e} denote the policies before and after KL-regularized RL, respectively. If π^{e} exactly maximizes the KL-regularized RL objective with reward r, KL coefficient $\beta > 0$, and reference policy π_0^{e}, then

$$\log \frac{\pi^{\mathrm{e}}(y \mid x)}{\pi_0^{\mathrm{e}}(y \mid x)} = \frac{1}{\beta} r(x, y) - \log Z(x), \qquad Z(x) = \mathbb{E}_{y \sim \pi_0^{\mathrm{e}}(\cdot|x)} \left[\exp\left(\frac{r(x, y)}{\beta} \right) \right].$$

Thus in this case, Eq. (5) holds with $\alpha = 1/\beta$, $b(x) = -\log Z(x)$, and $h(x, y) = 0$.

These methods have proven effective in practice [36, 62] and have been adopted in industrial-scale post-training pipelines [61, 58]. More recent work studies the mechanisms and failure modes of on-policy distillation. Li et al. [34] find that successful OPD depends on compatible teacher–student thinking patterns and genuinely new capabilities from the teacher. Jia et al. [28] show that standard OPD suffers from high-variance updates, vanishing gradients, and exploration bottlenecks, and may even perform worse than the initial SFT checkpoint when the student is insufficiently warmed up.

More works on Theoretical Analysis on Online and Offline IL. Another line of work is inverse reinforcement learning (IRL), which aims to recover an underlying reward function from expert demonstrations [1, 72, 55, 5]. Recent work shows that interaction with the environment can enable efficient imitation learning under misspecification through IRL, given certain structural assumptions [12]. However, these results typically assume that the reward function lies in a finite and realizable hypothesis class, and the IRL algorithms are not commonly adopted in LLM post-training. Finally, a recent study [54] also empirically identifies misspecification, rather than sampling error, as a more fundamental cause of behavior cloning's suboptimal performance in partially observed MDPs.

B Proofs and Additional Results

B.1 Lower Bound for Offline IL in Non-Realizable Settings

Theorem 4 (Formal version of Theorem 1)**.** *For any integers $N \geq 1$ and $K \geq 2$, any coverage parameter $C_\infty^{\mathrm{e}} > 1$, and any (possibly randomized) offline IL learner $\mathcal{A}$ that is given only N i.i.d. demonstrations $(x_i, y_i)_{i=1}^N$ with $x_i \sim \mathcal{D}_x$ and $y_i \sim \pi^{\mathrm{e}}(\cdot \mid x_i)$ and whose output is in Π, there exist a contextual bandit instance $(\mathcal{X}, \mathcal{Y}, \mathcal{D}_x, r)$, a policy class Π with $|\Pi| = K$ and an expert policy $\pi^{\mathrm{e}} \notin \Pi$ such that the following holds:*

1. *An optimal student policy $\pi^\star \in \arg\max_{\pi \in \Pi} V(\pi)$ matches the expert's performance: $V(\pi^\star) = V(\pi^{\mathrm{e}})$.*
2. *The pointwise coverage coefficient of π^{e} with respect to $\pi^\star$ equals C_∞^{e}:*
$$C_\infty^{\mathrm{e}} = \sup_{x \in \mathcal{X},\, y \in \mathcal{Y}} \frac{\pi^\star(y \mid x)}{\pi^{\mathrm{e}}(y \mid x)}.$$
3. *The output policy $\hat{\pi}$ of $\mathcal{A}$ satisfies*
$$\mathbb{E}[V(\pi^{\mathrm{e}}) - V(\hat{\pi})] \;\gtrsim\; \min\Big\{1,\, \sqrt{\tfrac{C_\infty^{\mathrm{e}} \log |\Pi|}{N}}\Big\},$$
where the expectation is over the demonstrations and the learner's internal randomness.

Proof. Set $\ell \coloneqq \lfloor \log_2 K \rfloor, \Delta \coloneqq \min\left\{\frac{1}{4}, \sqrt{\frac{C_\infty^{\mathrm{e}}\,\ell}{256N}}\right\}, a \coloneqq \frac{1}{2}+\Delta, b \coloneqq \frac{1}{2}-\Delta$. Let $\mathcal{D}_x$ be uniform on $\mathcal{X} = \{x_1, \ldots, x_\ell\}$ and let $\mathcal{Y} = \{y_1, y_2, y_3\}$. Index student policies by the hypercube $\mathcal{S} = \{y_2, y_3\}^\ell$. For $s \in \mathcal{S}$, write $\bar{s}_i$ for the other element of $\{y_2, y_3\}$ and define

$$\pi_s(y_1 \mid x_i) = 0, \pi_s(s_i \mid x_i) = a, \pi_s(\bar{s}_i \mid x_i) = b.$$

Take student policy class $\Pi = \{\pi_s : s \in \mathcal{S}\}$. We assume without loss of generality that K is a power of two, so that $\ell = \log_2 K$ and $|\Pi| = 2^\ell = K$. For general K, let $K' = 2^{\lfloor \log_2 K \rfloor}$, and let Π_0 be the class of K' policies constructed above. Introduce $K - K'$ additional actions $z_1, \ldots, z_{K-K'}$ with $r_s(x, z_j) = 0$ for every instance s, context x, and action z_j. Extend all original student and expert policies by assigning zero probability to these actions, and augment Π_0 with the deterministic policies $\pi_j^{\mathrm{dum}}(\cdot \mid x) = \delta_{z_j}$. The resulting policy class has exactly K elements. Since the original hard instance is embedded in this larger class, this augmentation cannot decrease the minimax risk.

For each $s \in \mathcal{S}$, we define the corresponding non-realizable expert policy π_s^{e} by

$$\pi_s^{\mathrm{e}}(y_1 \mid x_i) = 1 - \frac{1}{C_\infty^{\mathrm{e}}}, \pi_s^{\mathrm{e}}(s_i \mid x_i) = \frac{a}{C_\infty^{\mathrm{e}}}, \pi_s^{\mathrm{e}}(\bar{s}_i \mid x_i) = \frac{b}{C_\infty^{\mathrm{e}}},$$

and reward r_s by

$$r_s(x_i, y_1) = a, r_s(x_i, s_i) = 1, r_s(x_i, \bar{s}_i) = 0.$$

Write $V_s(\pi) \coloneqq \mathbb{E}_{x\sim\mathcal{D}_x}\mathbb{E}_{y\sim\pi(\cdot|x)} r_s(x, y)$ for the expected reward of π under r_s. By construction,

$$V_s(\pi_s^{\mathrm{e}}) = \left(1 - \frac{1}{C_\infty^{\mathrm{e}}}\right) a + \frac{a}{C_\infty^{\mathrm{e}}} = a = V_s(\pi_s) = \max_{\pi\in\Pi} V_s(\pi),$$

and the coverage coefficient equals C_∞^{e}, attained at $\pi^\star = \pi_s$. Moreover, for every $t \in \mathcal{S}$,

$$V_s(\pi_s^{\mathrm{e}}) - V_s(\pi_t) = \frac{2\Delta}{\ell}\, d_{\mathrm{H}}(s, t), \tag{7}$$

where d_{H} denotes Hamming distance. Let $\pi_{\widehat{T}}$ be the policy output by the learner, where index $\widehat{T} \in \mathcal{S}$. It therefore suffices to lower bound the expected Hamming loss $d_{\mathrm{H}}(s, \widehat{T})$.

Let $s^{(i)}$ be the index obtained by flipping coordinate i of s. Let P_s be the law of one expert demonstration under instance s. P_s and $P_{s^{(i)}}$ differ only at x_i, where a/C_∞^{e} and b/C_∞^{e} are interchanged. Thus

$$\begin{aligned} D_{\mathrm{KL}}\left(P_s \| P_{s^{(i)}}\right) &= \frac{1}{C_\infty^{\mathrm{e}}\,\ell}\left(a \log\frac{a}{b} + b \log\frac{b}{a}\right) \\ &= \frac{2\Delta}{C_\infty^{\mathrm{e}}\,\ell} \log\frac{1+2\Delta}{1-2\Delta} \le \frac{16\Delta^2}{C_\infty^{\mathrm{e}}\,\ell}. \end{aligned}$$

The last step follows from $\Delta \le 1/4$ and $\log((1+2\Delta)/(1-2\Delta)) \le 4\Delta/(1-2\Delta) \le 8\Delta$. Independence of the N demonstrations gives

$$D_{\mathrm{KL}}\left(P_s^{\otimes N} \| P_{s^{(i)}}^{\otimes N}\right) \le \frac{16N\Delta^2}{C_\infty^{\mathrm{e}}\,\ell} \le \frac{1}{16}.$$

Let $\mathbb{P}_s$ and $\mathbb{E}_s$ include the randomness of both the demonstrations and the learner. For each i and each s with $s_i = y_2$, the binary testing inequality yields

$$\begin{aligned} &\mathbb{P}_s(\widehat{T}_i \ne s_i) + \mathbb{P}_{s^{(i)}}(\widehat{T}_i \ne s_i^{(i)}) \\ &\qquad \ge 1 - \sqrt{\frac{1}{2} D_{\mathrm{KL}}\left(P_s^{\otimes N} \| P_{s^{(i)}}^{\otimes N}\right)} \ge \frac{3}{4}. \end{aligned}$$

For fixed i, the $2^{\ell-1}$ pairs $(s, s^{(i)})$ partition $\mathcal{S}$. Averaging over these pairs, and then summing over coordinates, gives

$$\frac{1}{2^\ell}\sum_{s\in\mathcal{S}} \mathbb{P}_s(\widehat{T}_i \ne s_i) \ge \frac{3}{8}, \qquad \frac{1}{2^\ell}\sum_{s\in\mathcal{S}} \mathbb{E}_s[d_{\mathrm{H}}(s, \widehat{T})] \ge \frac{3\ell}{8}.$$

Combining this with equation 7, we obtain

$$\frac{1}{2^\ell}\sum_{s\in\mathcal{S}}\mathbb{E}_s[V_s(\pi^{\mathrm{e}}_s)-V_s(\hat{\pi})]\geq\frac{2\Delta}{\ell}\cdot\frac{3\ell}{8}=\frac{3\Delta}{4}.$$

Therefore, at least one fixed instance s satisfies the same expected lower bound. Finally, since $\ell\geq(\log K)/2$,

$$\frac{3\Delta}{4}\geq\frac{3}{64}\min\left\{1,\sqrt{\frac{C^{\mathrm{e}}_\infty\,\ell}{N}}\right\}\geq\frac{1}{64}\min\left\{1,\sqrt{\frac{C^{\mathrm{e}}_\infty\log K}{N}}\right\}.$$

Together with $|\Pi|=K$, this proves the theorem. □

B.2 Proof of Theorem 2

Fix any $\pi\in\Pi$ that is ϵ-suboptimal for the online IL objective Eq. (2), i.e.,

$$J_{\mathrm{on}}(\pi)\geq\max_{\pi'\in\Pi}J_{\mathrm{on}}(\pi')-\epsilon\geq J_{\mathrm{on}}(\pi^\star)-\epsilon.$$

By Assumption 1, there exist $\alpha>0$, h, and b such that for all (x,y),

$$f(\pi^{\mathrm{e}}(y\mid x))=\alpha\,r(x,y)+h(x,y)+b(x).$$

Taking expectation over $x\sim\mathcal{D}_x$ and $y\sim\pi(\cdot\mid x)$ yields

$$\begin{aligned}J_{\mathrm{on}}(\pi)&=\mathbb{E}_{x\sim\mathcal{D}_x}\mathbb{E}_{y\sim\pi(\cdot|x)}f(\pi^{\mathrm{e}}(y\mid x))\\&=\alpha\,\mathbb{E}_{x\sim\mathcal{D}_x}\mathbb{E}_{y\sim\pi(\cdot|x)}r(x,y)+\mathbb{E}_{x\sim\mathcal{D}_x}\mathbb{E}_{y\sim\pi(\cdot|x)}h(x,y)+\mathbb{E}_{x\sim\mathcal{D}_x}b(x)\\&=\alpha\,V(\pi)+\mathbb{E}_{x\sim\mathcal{D}_x}\mathbb{E}_{y\sim\pi(\cdot|x)}h(x,y)+\mathbb{E}_{x\sim\mathcal{D}_x}b(x).\end{aligned}$$

The same decomposition holds for $\pi^\star$:

$$J_{\mathrm{on}}(\pi^\star)=\alpha\,V(\pi^\star)+\mathbb{E}_{x\sim\mathcal{D}_x}\mathbb{E}_{y\sim\pi^\star(\cdot|x)}h(x,y)+\mathbb{E}_{x\sim\mathcal{D}_x}b(x).$$

Subtracting the two identities cancels the $b(x)$ term and gives

$$J_{\mathrm{on}}(\pi)-J_{\mathrm{on}}(\pi^\star)=\alpha\big(V(\pi)-V(\pi^\star)\big)+\mathbb{E}_{x\sim\mathcal{D}_x}\Big[\mathbb{E}_{y\sim\pi(\cdot|x)}h(x,y)-\mathbb{E}_{y\sim\pi^\star(\cdot|x)}h(x,y)\Big].$$

Rearranging,

$$\alpha\big(V(\pi^\star)-V(\pi)\big)=J_{\mathrm{on}}(\pi^\star)-J_{\mathrm{on}}(\pi)+\mathbb{E}_{x\sim\mathcal{D}_x}\Big[\mathbb{E}_{y\sim\pi(\cdot|x)}h(x,y)-\mathbb{E}_{y\sim\pi^\star(\cdot|x)}h(x,y)\Big].$$

Now use $J_{\mathrm{on}}(\pi)\geq J_{\mathrm{on}}(\pi^\star)-\epsilon$, which implies $J_{\mathrm{on}}(\pi^\star)-J_{\mathrm{on}}(\pi)\leq\epsilon$. Moreover, since $J_{\mathrm{on}}(\pi)\geq J_{\mathrm{on}}(\pi^\star)-\epsilon$, the second part of Assumption 1 applies and yields

$$\mathbb{E}_{x\sim\mathcal{D}_x}\Big[\mathbb{E}_{y\sim\pi(\cdot|x)}h(x,y)-\mathbb{E}_{y\sim\pi^\star(\cdot|x)}h(x,y)\Big]\leq\delta_\alpha(\epsilon).$$

Combining these two bounds,

$$\alpha\big(V(\pi^\star)-V(\pi)\big)\leq\epsilon+\delta_\alpha(\epsilon),$$

and dividing by $\alpha>0$ finishes the proof. □

B.3 Comparison between Theorem 2 and the classical view of online IL as mitigating error accumulation

A classic account is that online interaction mitigates the compounding of per-step prediction errors over long horizons, leading to DAgger-style algorithms [48, 50, 36]. In contrast, our work studies the *information bottleneck* that arises in non-realizable settings, an orthogonal limitation of offline imitation learning that already appears in contextual bandits, i.e., when $H=1$. We further characterize how the reward function interacts with the misspecification structure to enable online imitation learning in non-realizable settings, a perspective absent from previous DAgger-style analyses.

Algorithm 1 Base-Centered Active Exploration

Require: Policy class Π; base policy $\pi^{\mathrm{base}} \in \Pi$; budget N; failure probability δ.
Ensure: Output policy $\widehat{\pi}$.

Remove every $\pi \neq \pi^{\mathrm{base}}$ with $D^{\mathrm{base}}(\pi) = 0$.
Set $K = \lceil \log_2 N \rceil$ and $\ell = \log\big(4|\Pi|(1 + \lceil \log_2 N \rceil)(1 + \lceil 2\log_2 N \rceil)/\delta\big)$.
Set $\mathcal{C}_0 = \Pi$, $\pi_0 = \pi^{\mathrm{base}}$, and $T_{-1} = 0$.
for $k = 0, \dots, K$ **do**
  Set $T_k = \min\{2^k, N\}$ and $m_k = T_k - T_{k-1}$.
  ▷ Choose sampling distribution
  Set $q_k = \textsc{SamplingDist}\big(\mathcal{C}_k \setminus \{\pi^{\mathrm{base}}\},\ \pi^{\mathrm{base}},\ \pi_k\big)$ via Algorithm 2.
  ▷ Sample
  for $t = T_{k-1} + 1, \dots, T_k$ **do**
    Draw $x_t \sim \mathcal{D}_x$ and $y_t \sim q_k(\cdot \mid x_t)$; query $u_t = u(x_t, y_t) \in [0, 1]$; set $q_t = q_k$.
  end for
  ▷ Update confidence intervals
  for each $\pi \in \mathcal{C}_k \setminus \{\pi^{\mathrm{base}}\}$ **do**
    Set $(\tau_{k,\pi},\ r_k(\pi)) = \textsc{ClipAndRadius}\big(\sum_{j=0}^{k} m_j V_{q_j}(\pi; \pi^{\mathrm{base}}),\ D^{\mathrm{base}}(\pi),\ T_k,\ \ell\big)$ via Algorithm 3.
    Set $\widehat{\mu}_k(\pi) = \frac{1}{T_k} \sum_{t=1}^{T_k} \mathrm{clip}_{\tau_{k,\pi}}\big(\big(\pi(y_t \mid x_t) - \pi^{\mathrm{base}}(y_t \mid x_t)\big)\, u_t / q_t(y_t \mid x_t)\big)$.
    Set $L_k(\pi) = \widehat{\mu}_k(\pi) - r_k(\pi)$ and $U_k(\pi) = \widehat{\mu}_k(\pi) + r_k(\pi)$.
  end for
  Set $\widehat{\mu}_k(\pi^{\mathrm{base}}) = L_k(\pi^{\mathrm{base}}) = U_k(\pi^{\mathrm{base}}) = 0$.
  ▷ Update candidate set
  Set $\mathcal{C}_{k+1} = \{\pi \in \mathcal{C}_k : U_k(\pi) \geq \max\{0,\ \max_{\pi' \in \mathcal{C}_k} L_k(\pi')\}\}$.
  Set $\pi_{k+1} \in \arg\max_{\pi \in \mathcal{C}_{k+1} \cup \{\pi^{\mathrm{base}}\}} L_k(\pi)$.
end for
return $\widehat{\pi} = \pi_{K+1}$.

Algorithm 2 SamplingDist: base-anchored minimax exploration distribution

Require: Active policies $\mathcal{A}$; base policy π^{base}; anchor policy π'.
Ensure: Sampling distribution q.

For $\gamma \in [0, 1]$, set $q_\gamma = (1 - \gamma)\pi^{\mathrm{base}} + \gamma\pi'$.
Set $\gamma^\star \in \arg\min_{\gamma \in [0,1]} \max_{\pi \in \mathcal{A}} V_{q_\gamma}(\pi; \pi^{\mathrm{base}}) / D^{\mathrm{base}}(\pi)$.
return $q = q_{\gamma^\star}$.

B.4 Finite Sample Guarantee of Learning Objective Eq. (2)

The online IL objective in Eq. (2) is typically optimized using policy-gradient methods in LLM post-training, as in on-policy distillation [20, 36]. However, analyzing the sample complexity of these methods remains challenging beyond structured settings such as tabular or linear parameterizations [2]. Moreover, we consider a general finite policy class without assuming a differentiable parameterization, so policy-gradient methods do not directly apply. To characterize the statistical complexity of optimizing Eq. (2) through online learning, we therefore present a simple algorithm (Algorithm 1) based on policy elimination [11].

Specifically, Algorithm 1 uses clipped importance weighting to estimate each student policy's expected reward difference relative to the base policy, rather than its expected reward itself. At the beginning of each epoch, it selects the policy with the largest lower confidence bound on this difference and collects online rollouts from a mixture of this policy and the base policy. The mixture coefficient is chosen to minimize the worst importance-weight variance over the surviving policies (Algorithm 2). It then uses these rollouts to update the confidence bounds and eliminates policies whose upper confidence bounds fall below the largest lower confidence bound.

Choosing the mixture coefficient γ requires no additional expert queries, since the objective in Line 2 of Algorithm 2 depends only on policy probabilities and the context distribution. The objective is convex in γ (Lemma 1), so this step is a one-dimensional convex optimization problem. Moreover, exact minimization is unnecessary: any approximate solution whose objective value is no larger than that of the base-only choice preserves the statistical guarantees, since the confidence bounds account for the variances under the chosen sampling policy.

Algorithm 3 CLIPANDRADIUS: variance-adaptive clipping threshold and confidence radius

Require: Second-moment proxy S; base-policy coverage D; sample count T; confidence parameter ℓ.
Ensure: Clipping threshold τ and confidence radius r.

Set $\widetilde{S} = \max\{S,\ D/T\}$.
Set $h = \lfloor \log_2\big(TD/\widetilde{S}\big) \rfloor$ and $\overline{S} = 2^{-h}\, TD$. ▷ snap the variance proxy to a dyadic grid
return $\tau = \sqrt{\overline{S}/\ell}$ and $r = 4\sqrt{\overline{S}\,\ell}/T$.

For any policy $\pi \in \Pi$, define its χ^2 base-policy coverage as

$$D^{\mathrm{base}}(\pi) := \mathbb{E}_{x\sim\mathcal{D}_x}\, D_{\chi^2}\big(\pi(\cdot \mid x) \,\big\|\, \pi^{\mathrm{base}}(\cdot \mid x)\big).$$

For a sampling distribution q, define

$$V_q(\pi; \pi^{\mathrm{base}}) := \mathbb{E}_{x\sim\mathcal{D}_x} \sum_y \frac{\big(\pi(y \mid x) - \pi^{\mathrm{base}}(y \mid x)\big)^2}{q(y \mid x)}.$$

We write $u(x, y) := f(\pi^{\mathrm{e}}(y \mid x))$ for the shaped expert signal in the online objective equation 2. We then analyze the sample complexity of Algorithm 1.

Theorem 5 (Finite-sample guarantee for Algorithm 1)**.** *Suppose that Π is finite with $M = |\Pi|$, and that $\pi^{\mathrm{base}} \in \Pi$. Suppose Assumption 2 holds. Assume that $D^{\mathrm{base}}(\pi) < \infty$ for every $\pi \in \Pi$. Let q_k be the sampling distribution selected at epoch k by Algorithm 1. For every $\pi \in \mathcal{C}_k$, define*

$$S_k(\pi) := \sum_{j=0}^{k} m_j V_{q_j}(\pi; \pi^{\mathrm{base}}).$$

Let $\ell = \log \frac{4M(\lceil \log_2 N\rceil+1)(\lceil 2\log_2 N\rceil+1)}{\delta}$. Let $\pi^\star \in \arg\max_{\pi\in\Pi} J_{\mathrm{on}}(\pi)$. Then, with probability at least $1-\delta$,

$$J_{\mathrm{on}}(\pi^\star) - J_{\mathrm{on}}(\hat{\pi}) \le 16 V_{\max} \sqrt{\frac{D^{\mathrm{base}}(\pi^\star)\ell}{N} \max\left\{\bar{\rho}_N^\star, \frac{1}{N^2}\right\}}, \tag{8}$$

where

$$\bar{\rho}_N^\star := \frac{S_K(\pi^\star)}{N D^{\mathrm{base}}(\pi^\star)} = \frac{1}{N}\sum_{k=0}^{K} m_k \frac{V_{q_k}(\pi^\star; \pi^{\mathrm{base}})}{D^{\mathrm{base}}(\pi^\star)} \le 1. \tag{9}$$

Proof. Let $K = \lceil \log_2 N\rceil, H = \lceil 2\log_2 N\rceil$. Without loss of generality, assume $u(x, y) \in [0, 1]$. Write

$$\mu(\pi) := J_{\mathrm{on}}(\pi) - J_{\mathrm{on}}(\pi^{\mathrm{base}}).$$

For a sample collected at time t in epoch j, let $q_t = q_j$ and define

$$X_t(\pi) := \frac{\pi(y_t \mid x_t) - \pi^{\mathrm{base}}(y_t \mid x_t)}{q_t(y_t \mid x_t)} u_t.$$

Let $\mathcal{F}_{t-1}$ contain all observations and all algorithmic choices made before drawing (x_t, y_t). The distribution q_t is $\mathcal{F}_{t-1}$-measurable. Importance weighting therefore gives

$$\mathbb{E}_t[X_t(\pi)] = \mu(\pi),$$

where $\mathbb{E}_t[\cdot] = \mathbb{E}[\cdot \mid \mathcal{F}_{t-1}]$. Since $u_t \in [0, 1]$,

$$\begin{aligned}\mathbb{E}_t[X_t(\pi)^2] &\le \mathbb{E}_{x\sim\mathcal{D}_x} \sum_y \frac{\big(\pi(y \mid x) - \pi^{\mathrm{base}}(y \mid x)\big)^2}{q_t(y \mid x)} \\ &= V_{q_t}(\pi; \pi^{\mathrm{base}}).\end{aligned}$$

We first prove that all confidence intervals used by the algorithm are valid. The dyadic construction is needed because $S_k(\pi)$ is predictable but random: the earlier sampling policies depend on the observed history.

Fix an epoch k, a non-base policy π, and a deterministic bucket $h \in \{0, \dots, H\}$. Write

$$s_{k,h}(\pi) := 2^{-h} T_k D^{\text{base}}(\pi), \qquad \tau_{k,h}(\pi) := \sqrt{\frac{s_{k,h}(\pi)}{\ell}}.$$

For notational brevity, set

$$\widetilde{X}_{t,k,h}(\pi) := \text{clip}_{\tau_{k,h}(\pi)}(X_t(\pi)).$$

Consider the clipped estimator

$$\widehat{\mu}_{k,h}(\pi) := \frac{1}{T_k} \sum_{t=1}^{T_k} \widetilde{X}_{t,k,h}(\pi).$$

We analyze this estimator on the event that π remains active at epoch k and $S_k(\pi) \le s_{k,h}(\pi)$.

The clipping bias satisfies

$$\begin{aligned}
\left| \frac{1}{T_k} \sum_{t=1}^{T_k} \mathbb{E}_t[\widetilde{X}_{t,k,h}(\pi)] - \mu(\pi) \right| &\le \frac{1}{T_k} \sum_{t=1}^{T_k} \mathbb{E}_t[|X_t(\pi)| \mathbf{1}\{|X_t(\pi)| > \tau_{k,h}(\pi)\}] \\
&\le \frac{1}{T_k \tau_{k,h}(\pi)} \sum_{t=1}^{T_k} \mathbb{E}_t[X_t(\pi)^2] \\
&\le \frac{s_{k,h}(\pi)}{T_k \tau_{k,h}(\pi)} = \frac{\sqrt{s_{k,h}(\pi)\ell}}{T_k}. \qquad (10)
\end{aligned}$$

Define the martingale differences

$$Y_{t,k,h}(\pi) := \widetilde{X}_{t,k,h}(\pi) - \mathbb{E}_t[\widetilde{X}_{t,k,h}(\pi)].$$

They satisfy

$$|Y_{t,k,h}(\pi)| \le 2\tau_{k,h}(\pi)$$

and

$$\begin{aligned}
\sum_{t=1}^{T_k} \mathbb{E}_t[Y_{t,k,h}(\pi)^2] &\le \sum_{t=1}^{T_k} \mathbb{E}_t[\widetilde{X}_{t,k,h}(\pi)^2] \\
&\le \sum_{t=1}^{T_k} \mathbb{E}_t[X_t(\pi)^2] \le S_k(\pi) \le s_{k,h}(\pi).
\end{aligned}$$

The two-sided Freedman inequality therefore gives, except on an event of probability at most $2e^{-\ell}$,

$$\left| \sum_{t=1}^{T_k} Y_{t,k,h}(\pi) \right| \le \sqrt{2 s_{k,h}(\pi)\ell} + \frac{4\tau_{k,h}(\pi)\ell}{3} \qquad (11)$$

whenever the policy is active at epoch k and $S_k(\pi) \le s_{k,h}(\pi)$.

Combining equation 10 and equation 11, and using $1 + \sqrt{2} + 4/3 < 4$, yields

$$|\widehat{\mu}_{k,h}(\pi) - \mu(\pi)| \le \frac{4\sqrt{s_{k,h}(\pi)\ell}}{T_k}. \qquad (12)$$

A union bound over at most $M(K+1)(H+1)$ policy–epoch–bucket triples shows that, with probability at least

$$1 - 2M(K+1)(H+1)e^{-\ell} \ge 1 - \delta,$$

equation 12 holds simultaneously for every applicable triple.

We now verify that the bucket selected by the algorithm is among those covered by this event. Since π^{base} is feasible in the sampling distribution selection problem (Line 2 in Algorithm 2),

$$V_{q_k}(\pi; \pi^{\text{base}}) \le D^{\text{base}}(\pi)$$

for every policy active at epoch k. Hence, for $\pi \in \mathcal{C}_k$,

$$S_k(\pi) \le T_k D^{\mathrm{base}}(\pi).$$

Recall from Algorithm 3 that, applied with $S = S_k(\pi)$, $D = D^{\mathrm{base}}(\pi)$, and $T = T_k$, the algorithm sets

$$\widetilde{S}_k(\pi) := \max\left\{S_k(\pi), \frac{D^{\mathrm{base}}(\pi)}{T_k}\right\}, \qquad h_k(\pi) := \left\lfloor \log_2 \frac{T_k D^{\mathrm{base}}(\pi)}{\widetilde{S}_k(\pi)} \right\rfloor,$$

$$\overline{S}_k(\pi) := 2^{-h_k(\pi)} T_k D^{\mathrm{base}}(\pi), \qquad r_k(\pi) := \frac{4\sqrt{\overline{S}_k(\pi)\ell}}{T_k}.$$

It follows that $h_k(\pi) \ge 0$. On the other hand, $\widetilde{S}_k(\pi) \ge D^{\mathrm{base}}(\pi)/T_k$, so $h_k(\pi) \le 2\log_2 T_k \le H$. Then

$$\widetilde{S}_k(\pi) \le \overline{S}_k(\pi) < 2\widetilde{S}_k(\pi). \tag{13}$$

In particular, $S_k(\pi) \le \overline{S}_k(\pi)$, and the algorithm's estimator is exactly $\widehat{\mu}_{k,h_k(\pi)}(\pi)$. Therefore, on the uniform event,

$$|\widehat{\mu}_k(\pi) - \mu(\pi)| \le r_k(\pi)$$

for every epoch k and every $\pi \in \mathcal{C}_k$. For the base policy the same statement holds exactly because $\widehat{\mu}_k(\pi^{\mathrm{base}}) = L_k(\pi^{\mathrm{base}}) = U_k(\pi^{\mathrm{base}}) = 0$.

We next show that the optimal policy is never eliminated. Initially, $\pi^\star \in \mathcal{C}_0$. Suppose inductively that $\pi^\star \in \mathcal{C}_k$. Because $\pi^{\mathrm{base}} \in \Pi$ and $\pi^\star$ is optimal,

$$\mu(\pi^\star) \ge 0.$$

Moreover, for every $\pi \in \mathcal{C}_k$,

$$L_k(\pi) \le \mu(\pi) \le \mu(\pi^\star) \le U_k(\pi^\star).$$

Thus

$$U_k(\pi^\star) \ge \max\left\{0, \max_{\pi \in \mathcal{C}_k} L_k(\pi)\right\},$$

which implies $\pi^\star \in \mathcal{C}_{k+1}$. Induction proves $\pi^\star \in \mathcal{C}_{K+1}$.

It remains to quantify the coverage improvement and the final error. Since

$$V_{\pi^{\mathrm{base}}}(\pi; \pi^{\mathrm{base}}) = D^{\mathrm{base}}(\pi),$$

the minimax sampling design satisfies

$$\rho_k = \max_{\pi \in \mathcal{C}_k \setminus \{\pi^{\mathrm{base}}\}} \frac{V_{q_k}(\pi; \pi^{\mathrm{base}})}{D^{\mathrm{base}}(\pi)} \le 1.$$

Because $\pi^\star$ remains active,

$$\begin{aligned} S_K(\pi^\star) &= \sum_{k=0}^{K} m_k V_{q_k}(\pi^\star; \pi^{\mathrm{base}}) \\ &\le D^{\mathrm{base}}(\pi^\star) \sum_{k=0}^{K} m_k \rho_k \le N D^{\mathrm{base}}(\pi^\star). \end{aligned}$$

This proves $\bar{\rho}_N^\star \le 1$.

Finally, since $\pi^\star \in \mathcal{C}_{K+1}$, the same output rule gives

$$L_K(\hat{\pi}) \ge L_K(\pi^\star) \ge \mu(\pi^\star) - 2r_K(\pi^\star).$$

Using $\mu(\hat{\pi}) \ge L_K(\hat{\pi})$ and $T_K = N$,

$$\begin{aligned} J_{\mathrm{on}}(\pi^\star) - J_{\mathrm{on}}(\hat{\pi}) &= \mu(\pi^\star) - \mu(\hat{\pi}) \\ &\le 2r_K(\pi^\star) = \frac{8\sqrt{\overline{S}_K(\pi^\star)\ell}}{N}. \end{aligned} \tag{14}$$

By equation 13,

$$\begin{aligned}\overline{S}_K(\pi^\star) &< 2\max\left\{S_K(\pi^\star), \frac{D^{\mathrm{base}}(\pi^\star)}{N}\right\} \\ &= 2ND^{\mathrm{base}}(\pi^\star)\max\left\{\bar{\rho}_N^\star, \frac{1}{N^2}\right\}.\end{aligned}$$

Substitution into equation 14 proves equation 8. □

Lemma 1 (Convexity of the mixture-selection objective)**.** *Fix the base policy π^{base}, the current student policy π_k, and the surviving set $\mathcal{C}_k$. Assume that $D^{\mathrm{base}}(\pi) < \infty$ for every $\pi \in \mathcal{C}_k$, and let*

$$\mathcal{C}_k^+ \coloneqq \{\pi \in \mathcal{C}_k : D^{\mathrm{base}}(\pi) > 0\}$$

be nonempty. For $\gamma \in [0,1]$, define

$$q_{k,\gamma} \coloneqq (1-\gamma)\pi^{\mathrm{base}} + \gamma\pi_k$$

and

$$F_k(\gamma) \coloneqq \max_{\pi\in\mathcal{C}_k^+} \frac{1}{D^{\mathrm{base}}(\pi)}\mathbb{E}_x\left[\sum_y \frac{\big(\pi(y \mid x) - \pi^{\mathrm{base}}(y \mid x)\big)^2}{q_{k,\gamma}(y \mid x)}\right],$$

with the conventions $0/0 = 0$ and $a/0 = +\infty$ for $a > 0$. Then F_k is convex on $[0,1]$. Consequently, minimizing F_k over $\gamma \in [0,1]$ is a one-dimensional convex optimization problem.

Proof. Fix π, x, and y, and write

$$a = \big(\pi(y \mid x) - \pi^{\mathrm{base}}(y \mid x)\big)^2, \qquad b = \pi^{\mathrm{base}}(y \mid x), \qquad d = \pi_k(y \mid x) - \pi^{\mathrm{base}}(y \mid x).$$

The corresponding summand is $a/(b+\gamma d)$, where $a \geq 0$ and $b + \gamma d \geq 0$. Whenever $b + \gamma d > 0$,

$$\frac{\mathrm{d}^2}{\mathrm{d}\gamma^2}\frac{a}{b+\gamma d} = \frac{2ad^2}{(b+\gamma d)^3} \geq 0.$$

With the conventions $0/0 = 0$ and $a/0 = +\infty$ for $a > 0$, each summand is convex on $[0,1]$. Summation, expectation, and division by the positive constant $D^{\mathrm{base}}(\pi)$ preserve convexity. Finally, the pointwise maximum over $\pi \in \mathcal{C}_k^+$ also preserves convexity, proving the claim. □

Remark 2 (The Necessity of Coverage Condition)**.** *The coverage coefficients in Theorem 3 are not merely artifacts of the analysis: without coverage or additional structural assumptions, an explicit dependence on the action-space cardinality is information-theoretically unavoidable.*

To make this precise, let $K = |\mathcal{Y}|$ and consider the following family of single-context bandit tasks. Let

$$\mathcal{X} = \{x\}, \qquad \Pi = \{\pi_y : y \in \mathcal{Y}\}, \qquad \pi_y(\cdot \mid x) = \delta_y.$$

For each $y^\star \in \mathcal{Y}$, define a task $\mathcal{M}_{y^\star}$ with reward and expert policy

$$r_{y^\star}(x,y) = \mathbf{1}\{y = y^\star\}, \qquad \pi^{\mathrm{e}}_{y^\star}(\cdot \mid x) = \delta_{y^\star}.$$

Taking $f(p) = p$, the online objective on task $\mathcal{M}_{y^\star}$ satisfies

$$J^{y^\star}_{\mathrm{on}}(\pi_y) = \mathbf{1}\{y = y^\star\}.$$

Thus, $\pi_{y^\star}$ is the unique optimal policy, with $J^{y^\star}_{\mathrm{on}}(\pi_{y^\star}) = V_{\max} = 1$.

Proposition 1 (Action-space lower bound without coverage)**.** *Consider any possibly randomized and adaptive learner $\mathcal{A}$ with access to the expert signal. At round t, the learner chooses an action $y_t \in \mathcal{Y}$ and observes only*

$$Z_t = f\big(\pi^{\mathrm{e}}_{y^\star}(y_t \mid x)\big) = \mathbf{1}\{y_t = y^\star\}.$$

After at most N such queries, it returns $\hat{\pi} \in \Pi$. Then, for every learner $\mathcal{A}$, there exists a task $\mathcal{M}_{y^\star}$ such that

$$\mathbb{E}_{\mathcal{A}}\big[J_{\mathrm{on}}^{y^\star}(\hat{\pi})\big] \le \min\left\{1, \frac{N+1}{|\mathcal{Y}|}\right\}.$$

Consequently, guaranteeing expected suboptimality at most any fixed $\epsilon < 1$ requires $N \ge (1-\epsilon)|\mathcal{Y}| - 1 = \Omega(|\mathcal{Y}|)$.

Proof. Let $Y^\star$ be uniformly distributed over $\mathcal{Y}$, independently of the learner's internal randomness. We first condition on this internal randomness, so that the learner is deterministic.

Consider the sequence of actions queried by the learner along the transcript in which every observed signal is zero, and let $Q \subseteq \mathcal{Y}$ be the set of distinct actions appearing in this sequence. Clearly, $|Q| \le N$. If $y^\star \in Q$, we use the trivial bound $J_{\mathrm{on}}^{y^\star}(\hat{\pi}) \le 1$. If $y^\star \notin Q$, every observation is zero, so the learner follows the same all-zero transcript and returns the same deterministic policy. Therefore, among all $y^\star \notin Q$, its output can be correct for at most one value of $y^\star$. It follows that

$$\sum_{y^\star \in \mathcal{Y}} J_{\mathrm{on}}^{y^\star}(\hat{\pi}) \le |Q| + 1 \le N + 1.$$

Averaging over the uniform choice of $Y^\star$, and then over the learner's internal randomness, gives

$$\frac{1}{|\mathcal{Y}|}\sum_{y^\star \in \mathcal{Y}} \mathbb{E}_{\mathcal{A}}\big[J_{\mathrm{on}}^{y^\star}(\hat{\pi})\big] \le \min\left\{1, \frac{N+1}{|\mathcal{Y}|}\right\}.$$

Hence, at least one $y^\star \in \mathcal{Y}$ satisfies the claimed upper bound. Since $J_{\mathrm{on}}^{y^\star}(\pi_{y^\star}) = 1$, the suboptimality lower bound follows immediately. □

On the other hand, when the base policy assigns non-negligible probability to $y^\star$, the coverage coefficients can be much smaller than $|\mathcal{Y}|$, avoiding an explicit dependence on the size of the action space in online IL.

Remark 3 (Online–Offline Separation under the Structural Conditions)**.** *The hard family in Theorem 1 can also be chosen to satisfy conditions required by the online upper bound, including Assumption 1 and the $V_{\max}$-bounded signal condition, by taking a threshold shaping function and choosing α appropriately. Furthermore, for any base model $\pi^{\mathrm{base}} \in \Pi$, the χ^2 base-policy coverage $D^{base}(\pi^\star)$ is upper-bounded by a constant, while the expert coverage coefficient C_∞^{e} can be arbitrarily large.*

Write $C := C_\infty^{\mathrm{e}} > 1$, and recall the construction in the proof of Theorem 4:

$$d := \lfloor \log_2 K \rfloor, \qquad \Delta := \min\left\{\frac{1}{4}, \sqrt{\frac{Cd}{256N}}\right\}, \qquad a := \frac{1}{2} + \Delta, \qquad b := \frac{1}{2} - \Delta.$$

For $s \in \{y_2, y_3\}^d$, the expert assigns probabilities $1 - 1/C$, a/C, and b/C to y_1, s_i, and $\bar{s}_i$ at context x_i, respectively. The corresponding rewards are a, 1, and 0. The student policy π_s assigns probabilities a and b to s_i and $\bar{s}_i$, and zero probability to y_1.

Take the threshold shaping function

$$f(p) := \mathbf{1}\Big\{p \ge \frac{a}{C}\Big\}.$$

Then $V_{\max} := 1$. Since $a > b$, we have

$$f(\pi_s^{\mathrm{e}}(s_i \mid x_i)) = 1, \qquad f(\pi_s^{\mathrm{e}}(\bar{s}_i \mid x_i)) = 0.$$

Thus $0 \le f(\pi_s^{\mathrm{e}}(y \mid x)) \le 1$ for every context and action, and the shaped signal equals r_s on the support $\{y_2, y_3\}$ of every student policy. Consequently, writing V_s and $J_{\mathrm{on}}^{(s)}$ for the reward and online objectives in instance s, we have

$$J_{\mathrm{on}}^{(s)}(\pi) = V_s(\pi) \qquad \textit{for every } \pi \in \Pi. \tag{15}$$

To verify Assumption 1, set $b(x) = 0$ and define $h_s(x, y) := f(\pi_s^{\mathrm{e}}(y \mid x)) - r_s(x, y)$. Every student policy has expected residual zero, so Assumption 1 holds with $\alpha = 1$ and $\delta_1(\epsilon) = 0$ for all $\epsilon > 0$.

Fix an index $t_0 \in \{y_2, y_3\}^d$, take $\pi^{\mathrm{base}} := \pi_{t_0} \in \Pi$. Every student policy has finite coverage under this base. More precisely,

$$D^{base}(\pi_s) = \frac{d_H(s, t_0)}{d} \left(\frac{(a-b)^2}{a} + \frac{(a-b)^2}{b} \right) = \frac{4\Delta^2}{ab} \frac{d_H(s, t_0)}{d} \leq \frac{4\Delta^2}{ab} \leq \frac{4}{3}.$$

Here $\Delta \leq 1/4$ and $ab = 1/4 - \Delta^2 \geq 3/16$.

Run Algorithm 1 with N queries, failure probability δ, and feedback $u(x, y) = f(\pi_s^{\mathrm{e}}(y \mid x)) \in \{0, 1\}$. Let

$$\ell := \log \left(\frac{4K(1 + \lceil \log_2 N \rceil)(1 + \lceil 2 \log_2 N \rceil)}{\delta} \right).$$

Using Theorem 5, with probability at least $1 - \delta$ we obtain

$$\begin{aligned} V_s(\pi_s^{\mathrm{e}}) - V_s(\widehat{\pi}_{\mathrm{on}}) &= J_{\mathrm{on}}^{(s)}(\pi_s) - J_{\mathrm{on}}^{(s)}(\widehat{\pi}_{\mathrm{on}}) \\ &\leq 16 \sqrt{\frac{D^{base}(\pi_s)\ell}{N}} \\ &\leq \frac{32}{\sqrt{3}} \sqrt{\frac{\ell}{N}} = \widetilde{O}\left(\sqrt{\frac{1}{N}} \right). \end{aligned} \tag{16}$$

In contrast, Theorem 1 shows that, on the same hard family, every offline learner incurs an additional dependence on the expert-coverage coefficient C:

$$\sup_s \mathbb{E}[V_s(\pi_s^{\mathrm{e}}) - V_s(\widehat{\pi}_{\mathrm{off}})] \gtrsim \min \left\{ 1, \sqrt{\frac{C \log K}{N}} \right\}.$$

This establishes an offline–online separation under the structural conditions of the online guarantee.

C Additional Experiments

Out-of-distribution generalization and catastrophic forgetting. We evaluate the model's out-of-distribution (OOD) generalization on the Countdown task, where test instances use a larger number range than those seen during training. We also assess catastrophic forgetting during GSM8K training by measuring general language capability on the MMLU benchmark [23]. Figure 6 shows that both SFT and on-policy distillation with a realizable expert achieve OOD performance matching that of the expert, whereas SFT with a non-realizable expert performs substantially worse and exhibits a larger gap between in-distribution and OOD performance. We observe a similar pattern for forgetting: under SFT with a realizable expert or on-policy distillation, the MMLU score decreases only slightly throughout training, with a final drop within 0.5 points; in contrast, under a non-realizable expert, it drops more substantially, by more than 2 points.

These findings refine prior observations that SFT tends to generalize worse OOD [8] and forget more than RL [52, 6]. Instead, our results suggest this gap is not inherent to the algorithm, but arises primarily under non-realizability. It is the training data distribution, rather than the SFT algorithm itself, that drives inferior OOD generalization and severe forgetting.

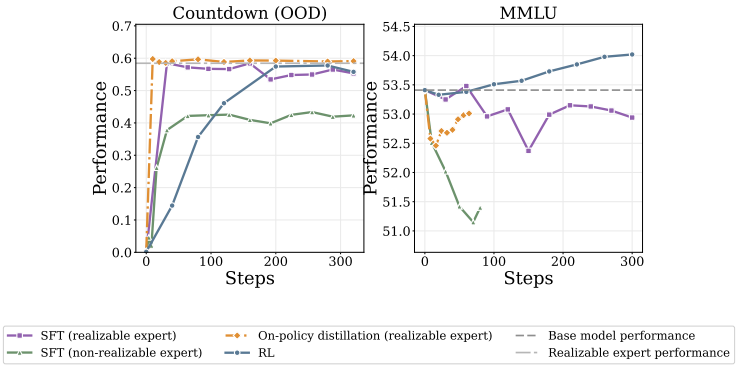


Figure 6: ***OOD generalization and catastrophic forgetting.*** *Left: Evaluation on Countdown instances with larger number ranges than training. Right: Forgetting during GSM8K training, measured on MMLU. Overall, SFT from a realizable expert achieves strong OOD performance while exhibiting little to no MMLU degradation, whereas SFT from a non-realizable expert harms OOD generalization during training and leads to severe forgetting.*

D Experimental Details

D.1 Countdown Task

Countdown Task Example

[INST] Using the numbers [5, 94, 9, 44], create an equation that equals 93. You can use basic arithmetic operations (+, -, *, /) and each number can only be used once. Show your work in `<think> </think>` tags. And return the final answer in `<answer> </answer>` tags, for example `<answer>`(1 + 2) / 3`</answer>`. **[/INST]**
Let me solve this step by step.

Experimental Setup. We utilize the dataset introduced by Pan et al. [41], which comprises $327{,}680$ training instances and $1{,}024$ test samples.[6] To illustrate the task format, we provide a representative training prompt.

Our implementation is built upon the official codebase of Pan et al. [41],[7] incorporating specific adaptations for NVIDIA A100 execution from a community fork.[8]

[6] `https://huggingface.co/datasets/Jiayi-Pan/Countdown-Tasks-3to4`
[7] `https://github.com/Jiayi-Pan/TinyZero`
[8] `https://github.com/JerryWu-code/TinyZero`

RL Training Details. We follow the training setup of Pan et al. [41] and provide the details below for completeness. We fine-tune the Qwen2.5-3B-Instruct model [56] using Reinforcement Learning (RL) for 320 steps on 2 NVIDIA A100 GPUs. The training configuration employs a global batch size of 128 with 5 rollouts per prompt, and a mini-batch size of 64 for optimization. We set the learning rate to 1×10^{-6} and the KL penalty coefficient to $\beta_{\mathrm{KL}} = 1 \times 10^{-3}$. The maximum generation length is capped at 1024 tokens. We use a reward function that assigns 1.0 to correct responses, 0.1 to incorrect but format-compliant responses, and 0 to invalid outputs.

Table 1: RL training details for the Countdown task.

Parameter	Value
Max prompt length	2,048
Clip ratio	0.2
Validation temperature	1
Rollout temperature	1
Validation top-p	0.7
Validation top-k	50

SFT Training Details. For training with both realizable and non-realizable experts, we use the same SFT configuration. We perform one-pass supervised fine-tuning with LlamaFactory on Qwen2.5-3B-Instruct using full-parameter training. The model is trained for 3,200 optimization steps with a maximum sequence length of 2,048 tokens. We use a batch size of 64, which is equal to the RL mini-batch size, a learning rate of 1×10^{-5}, a cosine learning rate schedule, and a warmup ratio of 0.1. For each question, we generate 5 different solutions, matching our RL setup, where 5 rollouts are generated for each question. For SFT with a realizable (RL-trained) expert, we roll out the expert with temperature 1.0 and disable both top-k and top-p filtering.

For the GSM8K and DeepScaleR experiments, we use the same SFT configuration, except that the batch size and maximum sequence length are matched to the corresponding RL setting. We therefore do not repeat these details separately.

One RL training step consists of multiple gradient updates. Specifically, if the global batch contains B_{global} prompts, each prompt produces R rollouts, and each gradient update uses a mini-batch of B_{mini} trajectories, then one RL step contains $B_{\mathrm{global}} R / B_{\mathrm{mini}}$ gradient updates. This number is 10 for Countdown and 16 for GSM8K and DeepScaleR. Figure 1 reports training progress in units of RL steps. To align SFT and RL by the number of gradient updates, we divide the number of SFT steps by 10 for Countdown and by 16 for GSM8K and DeepScaleR in Figure 1. Thus, one unit on the horizontal axis represents the same number of gradient updates for SFT and RL.

On-Policy Distillation Training Details. We use TRL's MiniLLM trainer [57, 20]. We train for 3,200 optimization steps on 8 GPUs. We use a batch size of 64. For each prompt, the student samples 4 completions on-policy. The maximum prompt length is 256 tokens and the maximum completion length is 1,024 tokens. We use a learning rate of 1×10^{-5}. Unless otherwise noted, other optimization settings follow the default TRL MiniLLM configuration, including AdamW, a linear learning-rate schedule with no warmup, and dropout disabled.

For the GSM8K experiments, we use a learning rate of 5×10^{-6}, while for DeepScaleR we use 1×10^{-5}. The sequence length in each setup is matched to that used in the corresponding RL training. All other configurations follow the Countdown setup.

Evaluation Details. During evaluation, we sample with temperature 1.0, top-p 0.7, and top-k 50. For each test example, we generate 8 responses and report the averaged accuracy. We set the maximum generation length to 1024 tokens.

OOD Evaluation Task. In Section 4 and Appendix C, we evaluate both in-distribution (ID) and out-of-distribution (OOD) performance for the Countdown task. For OOD evaluation, we use instances with four operands only, with larger input and answer ranges.

Table 2: Detailed settings for Countdown tasks used in ID and OOD evaluation.

Setting	Number of Operands	Range of Input	Range of Answer
Train / ID	3–4	100	100
OOD	4	125	300

D.2 GSM8K Task

Experimental Setup. We conduct experiments on the GSM8K benchmark [9], using the original train/test split, which contains approximately 7.47K training examples and 1.32K test examples. We use Llama-3.2-3B [19] as the base model.

Warmup Supervised Fine-Tuning. To enable RL training, we first perform supervised fine-tuning on the OpenR1 dataset [13] to strengthen the model's reasoning capabilities. We use the LlamaFactory [70] framework and train for $3{,}064$ steps with batch size 64 and learning rate 10^{-5} using the AdamW optimizer.

RL Training Details. We then perform RL training on the GSM8K training split for 300 steps using the DeepScaleR codebase [37], which is built upon verl [53]. A response receives reward 1 if its extracted final answer exactly matches the ground-truth answer, and 0 otherwise.

Table 3: RL training details for GSM8K experiments.

Parameter	Value
Train batch size	128
PPO mini-batch size	64
Max prompt length	512
Max response length	4,096
Rollout temperature	0.6
Rollout top-k	50
Rollout top-p	0.95
Responses per prompt	8
KL coefficient	0.001

Evaluation Details. During evaluation, we sample with temperature 0.6, top-p 0.95, and top-k 50. For MMLU evaluation, we use temperature 0.95, top-p 0.7, and top-k 50, and report the category-averaged MMLU accuracy.

D.3 DeepScaleR Task

Experimental Setup. We follow the setup of DeepScaleR [37] training with their 40k dataset, and evaluating with MATH [24] and AIME 2024 and AIME 2025 benchmarks. We use DeepSeek-R1-Distill-Qwen-1.5B [10] as the base model.

RL Training Details. We perform RL training using the DeepScaleR codebase [37], which is built on top of verl [53]. We follow the training configuration of the first stage (8k context length) of Luo et al. [37]. Specifically, for each prompt, we sample 8 responses with temperature 0.6. The maximum prompt length is 1,024 tokens and the maximum response length is 8,192 tokens. We use a train batch size of 128, a PPO mini-batch size of 64, an actor learning rate of 1×10^{-6}, and a KL coefficient of 0.001. We use binary reward.

Evaluation Details. During evaluation, we sample with temperature 0.6, top-p 0.95, and top-k 50.

D.4 DAPO task

On-Policy Distillation Training Details. For OPD with a non-realizable expert, we follow the DeepSeek-family setup in Figure 4 (left) of Li et al. [34], using DeepSeek-R1-Distill-Qwen-1.5B as the student and Skywork-OR1-Math-7B as the expert. For OPD with a realizable expert, we use the same student model and take the final student checkpoint from the preceding non-realizable OPD run as the expert. In both settings, we train on DAPO-Math-17K [67] for 270 steps and evaluate on AIME 2024 and AIME 2025.

We use the OPD training pipeline implemented with verl, using the PPO trainer in the verl [53] codebase together with vLLM [32] for rollout generation. We disable KL regularization and use token-mean loss aggregation.

Both runs share the same optimization hyperparameters and differ only in the expert model. We train with learning rate 10^{-6} and global batch size 256 (64 prompts and 4 rollouts per prompt) for 270 steps. The rollout temperature is 1.0, and we sample 4 responses per prompt during training. We use top-k truncation with $k = 16$. We use a maximum prompt length of 1024, a maximum response length of 7168, and a maximum generation length of 31744 for validation.

SFT Training Details. In each setting, SFT uses the same expert as OPD: Skywork-OR1-Math-7B [22] in the non-realizable setting, and the final student checkpoint from OPD with this expert in the realizable setting. We train with a learning rate of 10^{-6}, a global batch size of 256, and a maximum sequence length of $16{,}384$ tokens.

Evaluation Details. During evaluation, we sample with temperature 0.7, top-p 0.95, and no top-k sampling. We report mean@16 performance.

D.5 Computation Resources

We conduct experiments using A100 and H100 GPUs. The DeepScaleR experiments require 8 A100 GPUs for approximately one week. All other experiments are completed within two days.

D.6 Experimental Details for Section 5.1

For both models, we generate responses with temperature 0.6, top-p 0.95, and a maximum generation length of $100{,}000$ tokens. We evaluate both models on the AIME 2024 and AIME 2025 benchmarks. For each benchmark, each model generates 14 responses per question, yielding 840 responses per model in total. The response length is measured using the Qwen3-8B tokenizer for both models.

D.7 Experimental Details for Section 5.3

Training Setup. We reproduce the DeepSeek-family setting in Figure 4 (left) of Li et al. [34]. For completeness, we repeat the experimental details below. We use DeepSeek-R1-Distill-Qwen-1.5B as the student model. We consider two expert policies: DeepSeek-R1-Distill-Qwen-7B and Skywork-OR1-Math-7B. We train on the DAPO-Math-17K dataset [67] for 270 steps and evaluate performance on AIME24 and AIME25.

We use the OPD training pipeline implemented with verl, using the PPO trainer in the verl [53] codebase together with vLLM [32] for rollout generation. We disable KL regularization and use token-mean loss aggregation.

Both runs share the same optimization hyperparameters and differ only in the expert model. We train with learning rate 10^{-6} for 270 steps on 8 GPUs. The rollout temperature is 1.0, and we sample 4 responses per prompt during training. We use top-k truncation with $k = 16$. We use a maximum prompt length of 1024, a maximum response length of 7168, and a maximum generation length of 31744 for validation.

Estimating the suboptimality bound. We perform three distillation runs for each expert and collect checkpoints every 10 steps, obtaining 162 checkpoints in total. We denote this checkpoint set by $\widehat{\Pi}$. For each checkpoint policy $\pi \in \widehat{\Pi}$, we sample $M = 600$ responses on AIME 2024 and AIME 2025.

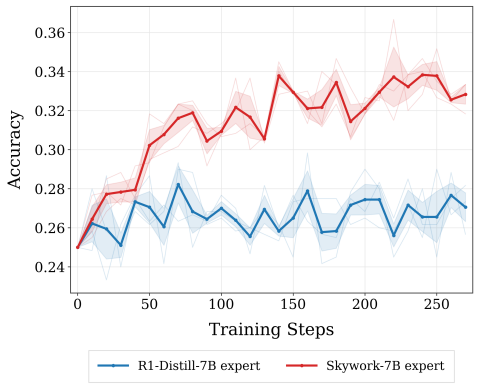


Figure 7: **The on-policy distillation results in Section 5.3.** Starting from the same base model, DeepSeek-R1-Distill-Qwen-1.5B, we perform on-policy distillation using two 7B experts: R1-Distill-7B and Skywork-7B. Distillation from Skywork-7B improves average performance on AIME 2024 and 2025 from 25.0% to 32.8%, whereas distillation from R1-Distill-7B improves it to 27.1% (averaged over three runs).

Table 4: Empirical estimation of quantities in Assumption 1.

Quantity	**Empirical proxy**
$V(\pi) = \mathbb{E}_x \mathbb{E}_{y\sim\pi(\cdot\|x)} r(x,y)$	$\widehat{V}(\pi) = \frac{1}{M}\sum_{i=1}^{M} r(x_i, y_i)$
$J_{\mathrm{on}}(\pi) = \mathbb{E}_x \mathbb{E}_{y\sim\pi(\cdot\|x)} f(\pi^{\mathrm{e}}(y \mid x))$	$\widehat{J}_{\mathrm{on}}(\pi) = \frac{1}{M}\sum_{i=1}^{M} \widehat{f}(x_i, y_i)$
policy class Π	Collected 162 checkpoints $\widehat{\Pi}$
ϵ-optimal policies w.r.t. J_{on}	$\widehat{\Pi}_\epsilon := \{\pi \in \widehat{\Pi} : \widehat{J}_{\mathrm{on}}(\pi) \geq \max_{\pi'\in\widehat{\Pi}} \widehat{J}_{\mathrm{on}}(\pi') - \epsilon\}$
Reference policy $\pi^\star$	$\hat{\pi}^\star = \arg\max_{\pi\in\widehat{\Pi}} \widehat{V}(\pi)$

Each sample consists of a benchmark prompt x_i and a response $y_i \sim \pi(\cdot \mid x_i)$. The reward value is estimated over these M responses, i.e.,

$$\widehat{V}(\pi) \coloneqq \frac{1}{M}\sum_{i=1}^{M} r(x_i, y_i),$$

where $r(x_i, y_i)$ is the correctness reward. For the expert score, we use the expert log-probability, i.e. $f(\pi^{\mathrm{e}}(y \mid x)) = \log \pi^{\mathrm{e}}(y \mid x)$, as is commonly done in on-policy distillation for training stability. This gives

$$\widehat{J}_{\mathrm{on}}(\pi) \coloneqq \frac{1}{M}\sum_{i=1}^{M} \log \pi^{\mathrm{e}}(y_i \mid x_i).$$

The resulting estimates are summarized in Table 4.

We estimate the suboptimality bound in Theorem 2 separately for each expert. To simplify notation, we omit the expert index in the estimators below. Let $\widehat{\pi}^\star \in \arg\max_{\pi\in\widehat{\Pi}} \widehat{V}(\pi)$ be the empirically best checkpoint policy in terms of reward value. Using the decomposition in Eq. (5), for a fixed $\alpha > 0$ we define the empirical residual

$$\widehat{H}_\alpha(\pi) \coloneqq \widehat{J}_{\mathrm{on}}(\pi) - \alpha \widehat{V}(\pi).$$

The action-independent offset $b(x)$ cancels when comparing two policies, and therefore does not need to be estimated. We then estimate the residual variation over $\widehat{\Pi}_\epsilon$:

$$\widehat{\delta}_\alpha \coloneqq \sup_{\pi\in\widehat{\Pi}_\epsilon} \left[\widehat{H}_\alpha(\pi) - \widehat{H}_\alpha(\widehat{\pi}^\star)\right].$$

Finally, we report the estimated suboptimality bound $\widehat{\Delta}(\epsilon) \coloneqq \min_{\alpha>0}(\epsilon + \widehat{\delta}_\alpha)/\alpha$.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The main claims in the abstract and introduction are supported by our theoretical results and empirical validation, and are stated within the scope of our assumptions and experimental settings.

   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 state the assumptions of our analysis clearly and discuss limitations and future directions in Section 6.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [Yes]

Justification: We clearly state the assumptions for each theoretical result and provide complete proofs.

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: We provide the experimental details needed to reproduce the main results in Appendix D.

   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: Our experiments are built on open-source codebases, and we provide detailed training configurations in Appendix D. We plan to release our implementation in the future.

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: We have specified all training and test details in Appendix D.

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: [No]

Justification: We do not report error bars because repeating the training experiments would be computationally expensive. Instead, we evaluate across multiple tasks and base models and observe consistent trends.

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: We report computer resources in Appendix D.

   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: This research follows the NeurIPS Code of Ethics in all aspects.

   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: We discuss the potential positive societal impact of improving the understanding of LLM post-training methods, as well as related future directions in Section 6. We do not identify direct negative societal impacts beyond the general risks associated with ML research.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We cite and acknowledge the creators of the existing assets used in this work.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

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

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

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

    Answer: [N/A]

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

    Guidelines:

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

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

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

    Answer: [N/A]

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

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [N/A]

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

    Guidelines:

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