AN RL VIEW OF OPD: LEAST SQUARE POLICY DISTILLATION FOR SAMPLE-EFFICIENT LLM REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

We study on-policy distillation (OPD) through the lens of reinforcement learning, establishing a connection between the reverse-KL objective in OPD and KL-regularized policy optimization. Building on this connection, we introduce Least-Square Policy Distillation (LSPD), an RL-inspired framework that brings optimistic exploration and off-policy data reuse from value-based RL into policy distillation. LSPD preserves policy diversity through exploration while improving rollout efficiency by repeatedly learning from previously collected trajectories. Our theoretical analysis connects LSPD to optimistic value-based learning and shows that its idealized formulation achieves a sharp $\widetilde{\mathcal{O}}(\log K)$ regret bound under online exploration. Empirically, LSPD consistently outperforms existing distillation baselines across six mathematical reasoning benchmarks and diverse teacher–student settings, with average gains of $+1.59$ points in Avg@16. Remarkably, through Pass@k evaluations up to $k = 64$, we found that LSPD better preserves policy diversity by achieving stronger performance as k grows. Its fully off-policy variant achieves comparable performance to vanilla OPD using only the first 25% of rollout batches. Together, these results provide an RL perspective on OPD that offers both a principled interpretation and a practical route toward more effective and rollout-efficient language model distillation.

1 INTRODUCTION

On-policy distillation (OPD) (Lu & Lab, 2025) has emerged as a promising approach to transferring and improving the capabilities of large language models (LLMs). Building on knowledge distillation (Hinton et al., 2015; Kim & Rush, 2016), OPD trains a student model using token-level teacher supervision on trajectories generated by the student itself. This allows the teacher to provide guidance at prefixes encountered under the student's own generation policy.

While OPD has recently been interpreted as a reinforcement learning (RL) approach (Yang et al., 2026a; Lu & Lab, 2025) and implemented using policy optimization frameworks such as PPO (Schulman et al., 2017b) and GRPO (Shao et al., 2024), the algorithmic implications of this formulation remain underexplored, particularly for exploration and data efficiency. These policy-based frameworks are typically implemented in an on-policy manner, requiring time-consuming fresh rollouts throughout training while offering limited support for historical data reuse and sufficient exploration.

On the other hand, KL-regularized RL (Yang et al., 2026a), with maximum-entropy RL as a special case, has been extensively studied both empirically (Haarnoja et al., 2018; 2017; Schulman et al., 2017a) and theoretically (Zhao et al., 2025; 2026). This literature provides principled algorithms for data-efficient off-policy updates that reuse previously collected experience (Watkins & Dayan, 1992; Degris et al., 2012; Munos et al., 2016), together with exploration strategies and regularization mechanisms that promote policy diversity and coverage. These advances provide a foundation for exploiting the structure of KL-regularized RL through value-based approaches to policy distillation.

Motivated by the success of value-based KL-regularized RL and the data-efficiency limitations of policy-based OPD implementations, we introduce *Least Square Policy Distillation (LSPD)*, a distillation framework motivated by optimistic KL-regularized value-based RL. Starting from the teacher-induced log-ratio reward, we consider reward estimation using accumulated student trajectories and

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

optimistic policy updates over statistically plausible reward functions. A Lagrangian relaxation of this formulation motivates quadratic matching between student and teacher log-probabilities. For practical token-level training, we combine a robust version of this matching loss with explicit entropy regularization to encourage policy diversity. Crucially, the resulting objective supports optimization on trajectories generated by earlier student policies, enabling both multiple updates per rollout batch and historical data reuse through a replay-buffer variant, *LSPD-RB*. Together, these components provide a practical approach to incorporating exploration-promoting regularization and off-policy learning into policy distillation. To summarize, our main contributions are threefold:

- **An RL-inspired policy distillation framework.** We introduce Least Square Policy Distillation (LSPD), motivated by optimistic KL-regularized policy optimization. LSPD combines robust quadratic matching of student and teacher log-probabilities with explicit entropy regularization, encouraging policy diversity while naturally supporting off-policy optimization. We complement this framework with a theoretical analysis of an idealized optimistic formulation showing that the LSPD enjoys a sharp convergence rate with regret by $\widetilde{\mathcal{O}}(\log K)$ where K is the update rounds.
- **Effective data reuse for sample-efficient distillation.** We demonstrate that LSPD effectively leverages both multiple updates per rollout batch and historical trajectories through its replay-buffer variant, LSPD-RB. In our off-policy experiments, LSPD-RB reaches saturated performance in approximately 10 rollout batches, compared with more than 40 for LSPD with one update per batch, highlighting the potential of historical data reuse to reduce rollout requirements.
- **Improved reasoning performance, diversity, and efficiency.** Across six benchmarks and three teacher–student settings, LSPD improves Avg@16 and Pass@16 by +1.59 and +1.87 points on average over baselines. LSPD with replay buffer further improves Pass@16 with only 10 training steps, while higher entropy and Pass@k demonstrate better policy diversity and solution coverage.

2 Related Works

Our work builds on language model distillation and reinforcement learning for LLM post-training. We connect reverse-KL distillation to optimistic KL-regularized RL, introduce a maximum-entropy least-square objective for off-policy data reuse, and establish a sharp theoretical guarantee.

Language Model Distillation. Language model distillation transfers knowledge from a larger teacher model to a smaller student model. Standard approaches typically minimize the forward KL divergence from the teacher to the student using teacher-generated data (Hinton et al., 2015; Kim & Rush, 2016). Subsequent work incorporates student-generated on-policy trajectories alongside teacher-generated data (Agarwal et al., 2024), or formulates on-policy distillation as an RL problem based on the reverse KL divergence between the student and teacher policies (Gu et al., 2024; Lu & Lab, 2025). Other studies investigate the effects of different divergence choices (Wu et al., 2025), interpret language model distillation through the lens of temporal-difference imitation learning (Yu et al., 2026b), and improve its data efficiency (Hsieh et al., 2023). DistiLLM combines skew KL objectives with adaptive off-policy reuse of student-generated responses (Ko et al., 2024), while DistiLLM-2 introduces a contrastive formulation that increases the likelihood of teacher responses and decreases that of student responses (Ko et al., 2025). Recent extensions of on-policy distillation further address reward extrapolation (Yang et al., 2026a), introduce entropy-aware training (Jin et al., 2026), and incorporate curriculum-level guidance (Li et al., 2026a).

Reinforcement Learning for LLM Post-Training. Reinforcement learning has been widely used to improve the instruction-following and reasoning capabilities of large language models. Ouyang et al. (2022) introduced reinforcement learning from human feedback for aligning language models with human preferences, while more recent work employs verifiable, rule-based rewards to improve mathematical and general reasoning capabilities (Shao et al., 2024; Guo et al., 2025; Yu et al., 2026a). Complementary to online RL, Identity Preference Optimization (IPO) learns directly from pairwise preferences using a squared loss on differences of policy-to-reference log-likelihood ratios (Gheshlaghi Azar et al., 2024), whereas our quadratic objective matches student and teacher log-probabilities. A growing theoretical literature has established sharp performance guarantees (Zhao et al., 2026) and logarithmic regret bounds (Zhao et al., 2025) for KL-regularized RL.

3 PRELIMINARIES

We formulate language model post-training from an RL view: at the sequence level, the post-training of LLM can be viewed as a contextual bandit. In particular, at position t, we denote the prefix $\mathbf{x}_{<t} = (\mathbf{q}, \mathbf{y}_{<t}) \in \mathcal{X}$ using the query $\mathbf{q}$ from the dataset $\mathcal{D}$ and the tokens generated by the model $\mathbf{y}_{<t}$. The model then generates the next token $y_t \in \mathcal{Y}$ from the policy $\pi(\cdot|\mathbf{x})$. Through this process, we define the reward $R : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ and it's class $\mathcal{R} \ni R$ which the policy seeks to maximize.

Maximum-Entropy and KL-Regularized Reinforcement Learning. Instead of greedy maximizing the reward by $\operatorname{argmax}_\pi \mathbb{E}_\pi[R(\mathbf{q}, \mathbf{y})]$, maximum-entropy reinforcement learning (Ziebart et al., 2008; Haarnoja et al., 2018), or generally, the KL regularized reinforcement learning Zhao et al. (2025), regularize the policy optimization with the KL divergence with the objective

$$\operatorname{argmax}_\pi \mathbb{E}_{\mathbf{y}\sim\pi(\cdot|\mathbf{q})}[R(\mathbf{q}, \mathbf{y})] - \eta^{-1} D_{\mathrm{KL}}\big(\pi(\cdot|\mathbf{q}) \,\|\, \pi^{\mathrm{ref}}(\cdot|\mathbf{q})\big)\,. \tag{3.1}$$

where the π^{ref} denotes a fixed reference policy. When π^{ref} is the uniform policy on the action set $\mathcal{A}$, it can be verified that $D_{\mathrm{KL}}\big(\pi(\cdot|\mathbf{q}) \,\|\, \pi^{\mathrm{ref}}(\cdot|\mathbf{q})\big)$ becomes the negative entropy $-\mathcal{H}(\pi(\cdot|\mathbf{q}))$ and Eq. 3.1 becomes the typical setting of maximum entropy RL $\operatorname{argmax}_\pi \mathbb{E}_{\mathbf{y}\sim\pi(\cdot|\mathbf{q})}[R(\mathbf{q}, \mathbf{y})] + \eta^{-1}\mathcal{H}(\pi(\cdot|\mathbf{q}))$.

On-Policy Distillation. On-policy distillation (OPD, Lu & Lab 2025) minimizes the token-level reverse KL divergence between the teacher policy π^E with the student policy π by

$$\mathcal{L}_{\mathrm{OPD}}(\pi) = \mathbb{E}\left[\sum_{t=1}^{T} D_{\mathrm{KL}}\big(\pi(\cdot|\mathbf{x}_{<t}) \| \pi^E(\cdot|\mathbf{x}_{<t})\big)\right] = \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{E}_{y_t\sim\pi(\cdot|\mathbf{x}_{<t})} \log \frac{\pi(y_t|\mathbf{x}_{<t})}{\pi^E(y_t|\mathbf{x}_{<t})}\right], \tag{3.2}$$

where the expectation is taken over prefix $\mathbf{x}_{<t}$ is sampled from student policy π from some query $\mathbf{q}$, which is dubbed as *on-policy* since it requires student's rollout. Notably, the current common implementation of OPD leverages the on-policy optimization framework like PPO (Schulman et al., 2017b) by taking the policy gradient (and additional clippings in PPO) by

$$\nabla\mathcal{L}_{\mathrm{OPD}}(\pi) = \mathbb{E}\left[\sum_{t=1}^{T} \nabla \log \pi(y_t|\mathbf{x}_{<t}) \cdot \log \frac{\pi^\perp(y_t|\mathbf{x}_{<t})}{\pi^E(y_t|\mathbf{x}_{<t})}\right] = \mathbb{E}\left[\sum_{t=1}^{T} \frac{\nabla\pi(y_t|\mathbf{x}_{<t})}{\pi^\perp(y_t|\mathbf{x}_{<t})} \cdot \log \frac{\pi^\perp(y_t|\mathbf{x}_{<t})}{\pi^E(y_t|\mathbf{x}_{<t})}\right],$$

where $\pi^\perp$ denotes the stopping gradient and $\log \frac{\pi^\perp(y_t|\mathbf{x}_{<t})}{\pi^E(y_t|\mathbf{x}_{<t})}$ is the advantage function for PPO process.

4 METHODOLOGY AND ANALYSIS

4.1 ON-POLICY DISTILLATION AS KL-REGULARIZED REINFORCEMENT LEARNING

We reinterpret the reverse-KL objective in Eq. 3.2 as token-level policy optimization with a teacher-induced reward. Following the notation in Section 3, we fix an arbitrary reference policy π^{ref} and define $R(\mathbf{x}_{<t}, y_t) = \log\big(\pi^E(y_t|\mathbf{x}_{<t})/\pi^{\mathrm{ref}}(y_t|\mathbf{x}_{<t})\big)$, assuming that the teacher and reference policies assign positive probability to tokens in the student's support at each prefix. Then, the OPD objective in Eq. 3.2 becomes

$$\begin{aligned}
\operatorname*{argmin}_\pi \mathcal{L}_{\mathrm{OPD}}(\pi) &= \operatorname*{argmin}_\pi \mathbb{E}\left[\sum_{t=1}^{T} \log \frac{\pi(y_t|\mathbf{x}_{<t})}{\pi^E(y_t|\mathbf{x}_{<t})}\right] \\
&= \operatorname*{argmin}_\pi \mathbb{E}\left[\sum_{t=1}^{T} \left(\log \frac{\pi(y_t|\mathbf{x}_{<t})}{\pi^{\mathrm{ref}}(y_t|\mathbf{x}_{<t})} - \log \frac{\pi^E(y_t|\mathbf{x}_{<t})}{\pi^{\mathrm{ref}}(y_t|\mathbf{x}_{<t})}\right)\right] \\
&= \operatorname*{argmax}_\pi \mathbb{E}\left[\sum_{t=1}^{T} \big(\mathbb{E}_{y_t\sim\pi(\cdot|\mathbf{x}_{<t})} R(\mathbf{x}_{<t}, y_t) - D_{\mathrm{KL}}\big(\pi(\cdot|\mathbf{x}_{<t}) \| \pi^{\mathrm{ref}}(\cdot|\mathbf{x}_{<t})\big)\big)\right].
\end{aligned} \tag{4.1}$$

Here, expectations are taken over $\mathbf{q} \sim \mathcal{D}$ and autoregressive generation from π, with $\mathbf{x}_{<t} = (\mathbf{q}, \mathbf{y}_{<t})$. Thus, OPD is the token-level counterpart of the KL-regularized policy optimization problem in Eq. 3.1 with $\eta = 1$. The reward R is fixed with respect to the student and measures the teacher's next-token log-likelihood relative to the reference, while the KL penalty discourages deviations from that reference at each prefix. Importantly, this is an exact reformulation of OPD rather than an additional regularization term. Choosing π^{ref} as a fixed initial model yields the reward–KL structure commonly used in RLHF (Ouyang et al., 2022; Rafailov et al., 2023), with the reward specified directly by the teacher-to-reference likelihood ratio. Prior work (Yang et al., 2026a) has also noted the connection between OPD and KL-regularized RL.

4.2 Least Square Policy Distillation

Optimistic KL-regularized Update. Given the KL-regularized objective introduced in the previous sub-section, the next question is how to solve the resulting policy optimization problem. A straightforward approach is to apply PPO (Schulman et al., 2017b), using advantage estimation together with the clipped surrogate objective. However, PPO is not an optimistic optimization method: its updates are driven by the current reward estimates and do not explicitly account for uncertainty in a way that encourages exploration toward potentially better tokens. Motivated by prior works on KL-regularized reinforcement learning (Zhao et al., 2026; 2025), which leverage optimistic reward estimation for policy optimization, we instead consider a more direct approach that admits an explicit, "closed-form" iterative update of the conditional token distribution. Specifically, after collecting rollouts through iteration k, we estimate the induced token-level reward using their prefix–token pairs:

$$\begin{aligned}\widehat{r}_k &\in \operatorname{argmin}_{r\in\mathcal{R}} \textstyle\sum_{i=1}^{k} \mathbb{E}_{\mathbf{q}\sim\mathcal{D},\, \mathbf{y}\sim\pi_i(\cdot|\mathbf{q})} \left[\sum_{t=1}^{T} \left(r(\mathbf{x}_{<t}, y_t) - R(\mathbf{x}_{<t}, y_t)\right)^2\right] \\ &= \operatorname{argmin}_{r\in\mathcal{R}} \textstyle\sum_{i=1}^{k} \mathbb{E}_{\mathbf{q}\sim\mathcal{D},\, \mathbf{y}\sim\pi_i(\cdot|\mathbf{q})} \left[\sum_{t=1}^{T} \left(r(\mathbf{x}_{<t}, y_t) - \log \tfrac{\pi^E(y_t|\mathbf{x}_{<t})}{\pi^{\text{ref}}(y_t|\mathbf{x}_{<t})}\right)^2\right]. \end{aligned} \tag{4.2}$$

For a stored rollout $(\mathbf{q}_i, \mathbf{y}_i)$, let $y_{i,t}$ denote its token at position t and $\mathbf{x}_{i,<t} = (\mathbf{q}_i, \mathbf{y}_{i,<t})$ its prefix. We then construct a least-squares confidence set containing reward functions that remain statistically consistent with the accumulated prefix–token pairs:

$$\mathcal{C}_k = \left\{ r \in \mathcal{R} : \textstyle\sum_{i=1}^{k} \sum_{t=1}^{T} \left[r(\mathbf{x}_{i,<t}, y_{i,t}) - R(\mathbf{x}_{i,<t}, y_{i,t})\right]^2 + \lambda \le \beta^2 \right\}. \tag{4.3}$$

Here, β denotes the confidence radius and $\lambda > 0$ is a regularization parameter. We directly find the pointwise optimistic token-level reward $r_k^+(\mathbf{x}_{<t}, y_t) = \sup_{r\in\mathcal{C}_k} r(\mathbf{x}_{<t}, y_t)$. At each fixed prefix, we then use the closed-form KL-regularized token update

$$\pi_{k+1}(y_t|\mathbf{x}_{<t}) \propto \pi^{\text{ref}}(y_t|\mathbf{x}_{<t}) \exp\left(\eta r_k^+(\mathbf{x}_{<t}, y_t)\right).$$

This update optimizes the conditional token distribution while holding the prefix fixed. Thus, each iteration uses stored prefix–token pairs to estimate the induced reward and constructs optimistic next-token distributions from the most favorable pointwise reward estimates compatible with the collected data. Algorithm 2 summarizes the corresponding sequence-level theoretical idealization.

Lagrangian Relaxation. The pointwise optimization $r_k^+(\mathbf{x}_{<t}, y_t) = \sup_{r\in\mathcal{C}_k} r(\mathbf{x}_{<t}, y_t)$ can be reformulated through a Lagrangian relaxation. Ignoring terms that are constant with respect to r, for each fixed prefix $\mathbf{x}_{<t}$ and token $y_t \in \mathcal{Y}$, we obtain

$$\arg\max_{r\in\mathcal{R}} r(\mathbf{x}_{<t}, y_t) - \mu \textstyle\sum_{i=1}^{k} \sum_{u=1}^{T} \left[r(\mathbf{x}_{i,<u}, y_{i,u}) - R(\mathbf{x}_{i,<u}, y_{i,u})\right]^2, \tag{4.4}$$

where $\mu \ge 0$ denotes the Lagrange multiplier and u indexes token positions in the stored rollouts. We further consider a centered reward function class parameterized by token-level policies $\mathcal{R} = \left\{r_\pi(\mathbf{x}_{<t}, y_t) = \log\left(\pi(y_t|\mathbf{x}_{<t})/\pi^{\text{ref}}(y_t|\mathbf{x}_{<t})\right) : \pi \in \Pi\right\}$. Since π^{ref} is fixed, the pointwise optimistic optimization at each prefix can therefore be equivalently written as

$$\forall y_t \in \mathcal{Y} : \arg\max_{\pi\in\Pi} \log \pi(y_t|\mathbf{x}_{<t}) - \mu \sum_{i=1}^{k} \sum_{u=1}^{T} \left(\log \pi(y_{i,u}|\mathbf{x}_{i,<u}) - \log \pi^E(y_{i,u}|\mathbf{x}_{i,<u})\right)^2. \tag{4.5}$$

Algorithm 1 Least Square Policy Distillation

Input: Student policy π, teacher policy π^E, replay buffer $\mathcal{B}$, batch size B, number of updates N.
Initialize $\mathcal{B} \leftarrow \emptyset$.
for $k = 1, \ldots, K$ **do**
  Sample queries $\{\mathbf{q}_b\}_{b=1}^{B}$ and responses $\{\mathbf{y}_b\} \sim \pi_k(\cdot|\mathbf{q}_b)$. ◁ *On-policy rollout collection*
  $\mathcal{B} \leftarrow \mathcal{B} \cup \{(\mathbf{q}_b, \mathbf{y}_b)\}$.
  for $i = 1, \ldots, N$ **do**
    Sample a minibatch from $\mathcal{B}$ and update the student using Eq. 4.6.
  end for
  if use LSPD **then**
    $\mathcal{B} \leftarrow \emptyset$. ◁ *No replay in LSPD; LSPD-RB reuses historical trajectories.*
  end if
end for

In this way, we obtain an optimistic token-level distillation formulation motivated by the reverse-KL objective of OPD (Lu & Lab, 2025). Importantly, the resulting objective is off-policy, allowing the algorithm to leverage prefix–token pairs from historical rollouts for policy updates rather than restricting training to samples generated by the current policy. Moreover, the mechanism through which optimism is incorporated into the objective is closely related to maximum-entropy objectives studied in the reinforcement learning literature (Ziebart et al., 2008; Haarnoja et al., 2018) and, more recently, in the LLM post-training literature (Xie et al., 2025).

4.3 Practical Implementation

In practice we are given a query dataset $\mathcal{D}$, a teacher policy π^E, and a student policy π. The student generates rollouts $\mathbf{y}$, conditioned on queries $\mathbf{q}$ sampled from $\mathcal{D}$, yielding prefix–token pairs $(\mathbf{x}_{<t}, y_t)$. Building on Eq. 4.5, we use a practical surrogate that combines robust token-level log-probability matching with an entropy bonus in place of the pointwise optimistic term:

$$\mathcal{J}(\pi) = \mathbb{E}_{(\mathbf{q},\mathbf{y})\sim\mathcal{B}}\left[\frac{1}{T}\sum_{t=1}^{T}\left(\left(\log \pi(y_t|\mathbf{x}_{<t}) - \log \pi^E(y_t|\mathbf{x}_{<t})\right)^2 - \mu^{-1}\,\mathcal{H}(\pi(\cdot|\mathbf{x}_{<t}))\right)\right], \quad (4.6)$$

where $\mathcal{B}$ denotes a replay buffer that stores query-response pairs, $\mathcal{H}(\pi(\cdot|\mathbf{x}_{<t}))$ denotes the entropy of the current student's next-token distribution at a stored prefix.

To improve training stability and reduce the influence of excessively large log-probability differences, in practical implementations, we replace the standard quadratic loss with a Huber-type robust penalty $\psi(\log \pi(y_t|\mathbf{x}_{<t}) - \log \pi^E(y_t|\mathbf{x}_{<t}))$, detailed in Appendix A.1. Notably, the objective in Eq. 4.6 supports both on-policy and off-policy optimization. The on-policy (or semi-on-policy) variant uses freshly generated responses from the current policy π for one or multiple optimization updates, whereas the off-policy variant additionally reuses responses generated by earlier policies through a replay buffer. This flexibility allows rollout data to be efficiently reused across multiple updates. In the following experiments, we refer to the former as Least Square Policy Distillation (LSPD) and the fully off-policy variant as Least Square Policy Distillation with Replay Buffer (LSPD-RB). The detailed practical implementation is summarized in Algorithm 1.

5 Experiments

We evaluate the empirical performance of our proposed method against a range of post-training baselines on mathematical reasoning tasks. We describe the experimental setup in Section 5.1, present the main results in Section 5.2, investigate off-policy learning and ablation studies in Section 5.3. Results on out-of-domain evaluation and more ablation studies are shown in Appendix B.

5.1 Experimental Setup

Model Setup. We consider three distillation setups based on the Qwen3 model family (Yang et al., 2025). Specifically, we use Qwen3-8B as the teacher and Qwen3-4B-Base as the student, Qwen3-4B

as the teacher and Qwen3-1.7B-Base as the student, and Qwen3-1.7B as the teacher and Qwen3-0.6B-Base as the student. For all teacher models, we disable thinking mode and use standard non-thinking model configuration.

Dataset and Evaluation Setup. For all experiments, we use DAPO-Math-17K (Yu et al., 2026a) as the training dataset for mathematical reasoning. For evaluation, we consider a diverse set of challenging mathematical reasoning benchmarks spanning different difficulty levels, including MATH-500 (Hendrycks et al., 2021), Minerva (Lewkowycz et al., 2022), Olympiad-Bench (He et al., 2024), AMC23 (Mathematical Association of America, 2023), AIME24 (Math-AI, 2024), and AIME25 (Math-AI, 2025). We report the average score over 16 generations (Avg@16) as the final result for all baselines and benchmarks in Table 1.

Generation Configurations. During training, we use top-p sampling with $p = 1.0$, a temperature of 1.0, a maximum rollout length of 7168 tokens, and four rollouts per query. During evaluation, we use top-p sampling with $p = 0.95$, a temperature of 0.7, and the same maximum rollout length of 7168 tokens. For both training and evaluation, we use the prompt template described in Appendix A.3.

Baselines. We compare LSPD and LSPD-RB against several language model distillation baselines: Knowledge Distillation (KD) (Hinton et al., 2015; Kim & Rush, 2016), On-Policy Distillation (OPD) (Lu & Lab, 2025), and Entropy-Aware On-Policy Distillation (EOPD) (Jin et al., 2026). KD performs off-policy distillation on teacher-generated trajectories using cross-entropy supervision together with forward KL divergence between the student and teacher distributions. In contrast, OPD trains on student-generated trajectories using the reverse-KL objective introduced in Section 3. EOPD further augments OPD with forward KL regularization at high-entropy teacher token positions to better preserve teacher uncertainty and output diversity.

5.2 Main Results

Results on Mathematical Reasoning Capabilities. Table 1 shows that LSPD consistently delivers strong mathematical reasoning performance across model scales and benchmarks. Averaged over all 18 model–benchmark combinations, LSPD achieves an Avg@16 of 31.60 and Pass@16 of 53.30, outperforming the strongest baseline, EOPD, by +0.91 and +0.85 points, respectively, and standard OPD by +1.99 and +2.27 points. Notably, LSPD-RB achieves a comparable Avg@16 of 31.51 while further improving Pass@16 to 54.86, exceeding EOPD by +0.82 Avg@16 and +2.42 Pass@16, and OPD by +1.89 and +3.84 points, respectively. When all five methods are considered, LSPD attains the best or tied-best Avg@16 on 11 of 18 settings and Pass@16 on 9 of 18 settings, while remaining top-2 on 17 of 18 and 16 of 18 settings, respectively. More importantly, considering LSPD and LSPD-RB together, at least one of the two achieves the best or tied-best result on 17 of 18 settings for both Avg@16 and Pass@16, and ranks second or tied-second in each of the two remaining cases. Thus, the better-performing LSPD variant is top-2 across all 36 metric–setting combinations, demonstrating robust improvements in both average-generation accuracy and multi-sample solution coverage. Remarkably, LSPD-RB achieves these results using only 10 training steps, whereas the baselines require more than 40 steps to converge stably, highlighting the substantial sampling efficiency enabled by off-policy reuse.

Results on Model Entropy. To examine the student's response diversity during distillation, we track its policy entropy throughout training across three teacher–student pairs in Figure 1. Both LSPD and EOPD (Jin et al., 2026) maintain student entropy at approximately 0.5, consistently higher than standard OPD (Lu & Lab, 2025). While EOPD augments the OPD objective with an entropy-gated forward KL term, LSPD encourages higher entropy directly through the entropy regularization term in Eq. 4.6, together with the robust log-probability matching objective. Despite these different formulations, both methods exhibit similar entropy behavior during training. This suggests that LSPD can maintain a higher level of student policy diversity than standard OPD with direct entropy-regularized maximization instead of relying on EOPD's entropy-gated forward KL formulation.

Results on Pass@k Scaling. To further evaluate the multi-sample solution coverage and useful generation diversity of the distilled student model, we study how Pass@k scales with the number of sampled responses. As reported in Table 2, we distill a Qwen3-4B-Base student from a Qwen3-8B non-thinking teacher using LSPD and three baselines, KD, OPD, and EOPD, and evaluate Pass@k over a broad range of sampling budgets, with $k \in \{1, 2, 4, 8, 16, 32, 64\}$. Remarkably, LSPD ex-

Table 1: **Main Results on Mathematical Reasoning.** We report Avg@16 and Pass@16 performance of our proposed LSPD and three baselines (KD, OPD, and EOPD) across six mathematical reasoning benchmarks and three teacher–student distillation settings based on the Qwen3 model family. The best-performing result within each setup is highlighted in **bold**, and the second-best result is underlined. Note that LSPD-RB results are reported with only *10 training steps*, whereas all other methods require at least 40 training steps to converge stably.

Model	Method	MATH-500		Minerva		Olympiad		AMC23		AIME24		AIME25	
		Avg@16	Pass@16	Avg@16	Pass@16	Avg@16	Pass@16	Avg@16	Pass@16	Avg@16	Pass@16	Avg@16	Pass@16
8B ↓ 4B	KD	79.73	94.00	38.51	58.09	45.97	67.85	51.20	77.11	17.71	33.33	17.08	36.67
	OPD	79.31	95.20	35.73	58.46	44.64	67.85	48.49	79.97	17.92	30.00	15.83	**40.00**
	EOPD	80.50	95.00	37.78	**59.56**	46.36	68.00	51.05	80.72	17.92	36.67	17.92	**40.00**
	LSPD	**81.61**	**95.40**	**38.99**	**59.56**	47.22	69.04	**52.48**	81.93	18.96	36.67	18.13	36.67
	LSPD-RB	81.55	95.20	38.49	58.09	**47.41**	**69.33**	52.11	**84.34**	**21.25**	**50.00**	**18.75**	36.67
4B ↓ 1.7B	KD	66.75	91.00	27.76	52.94	30.52	58.52	35.54	66.27	8.13	26.67	5.21	13.33
	OPD	69.13	90.20	26.93	53.68	31.98	57.04	34.64	65.06	10.00	26.67	6.88	16.67
	EOPD	69.75	90.60	27.92	54.04	32.91	59.26	35.84	66.27	11.46	33.33	7.50	20.00
	LSPD	**70.86**	**91.80**	29.32	**55.15**	33.47	59.70	**36.97**	**67.47**	**11.88**	30.00	**8.33**	**23.33**
	LSPD-RB	70.69	91.00	**29.57**	54.41	**33.50**	**61.04**	36.60	**67.47**	11.67	**36.67**	**8.33**	**23.33**
1.7B ↓ 0.6B	KD	49.61	79.60	15.49	41.18	18.98	41.63	22.97	53.01	2.08	13.33	1.88	10.00
	OPD	51.40	79.80	11.28	37.50	**20.66**	41.93	22.89	51.81	4.17	13.33	1.25	**13.33**
	EOPD	51.98	79.40	15.85	38.60	19.84	42.81	23.12	53.01	3.33	16.67	1.46	10.00
	LSPD	**52.04**	**81.60**	**17.14**	40.81	20.39	43.56	**24.62**	54.22	**5.00**	**20.00**	1.46	10.00
	LSPD-RB	51.56	81.20	16.36	**41.54**	20.12	**44.59**	22.97	**65.06**	3.96	16.67	**2.29**	**13.33**

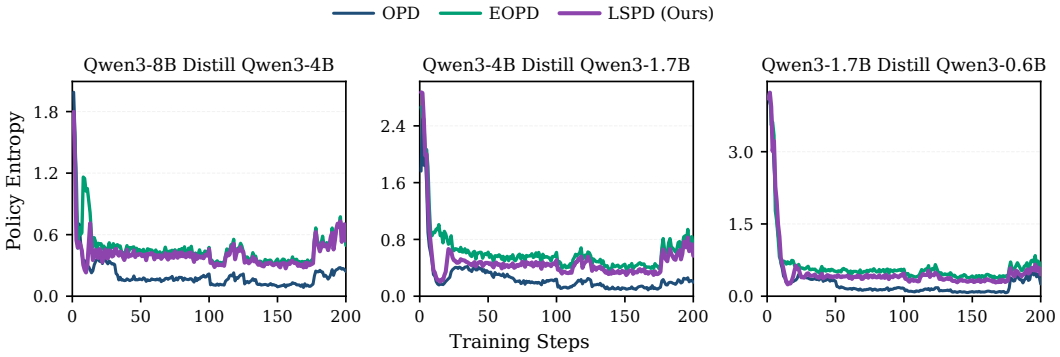


Figure 1: **Student Entropy during Training.** We plot the student policy entropy over 200 training steps for OPD, EOPD, and LSPD across three teacher–student pairs. LSPD maintains an entropy level comparable to EOPD and consistently higher than standard OPD, indicating improved preservation of student policy diversity during distillation.

hibits stronger scaling as the sampling budget increases. In particular, it achieves the best or tied-best Pass@32 and Pass@64 on all three benchmarks, reaching 85.54, 43.33, and 46.67 Pass@32, and 89.16, 46.67, and 50.00 Pass@64 on AMC23, AIME24, and AIME25, respectively. Averaged across the three benchmarks, LSPD achieves Pass@32/64 of 58.51/61.94, consistently outperforming the strongest competing method, EOPD (57.00/60.43), by +1.51 and +1.52 points, respectively. These results suggest that LSPD maintains competitive single-sample accuracy while providing stronger solution coverage under larger sampling budgets, indicating that its improvements are not limited to concentrating probability mass on a small set of high-probability responses.

5.3 Off-policy and LSPD-RB Replay Buffer Results

Since our proposed objective naturally supports off-policy optimization, we evaluate its ability to leverage off-policy data under two settings. First, we vary the number of optimization steps per rollout batch as $N \in \{1, 4, 16, 64\}$, resulting in increasingly off-policy updates while still training only on the most recently collected rollouts. Second, we consider a fully off-policy variant, LSPD-RB, which maintains a replay buffer containing all previously collected query–response pairs and performs updates using batches sampled from this buffer. Figure 2 reports the corresponding Avg@16 training curves on AMC23, AIME24, and AIME25 with a Qwen3-4B teacher model and a Qwen3-1.7B-Base student. For all settings in Figure 2, including different choices of N and the replay-buffer variant, each training step collects one rollout batch consisting of 64 prompts with 4 responses sampled per prompt.

Table 2: **Pass@k Scaling Results.** We report Pass@k results for $k \in \{1, 2, 4, 8, 16, 32, 64\}$ on Qwen3-4B-Base distilled from Qwen3-8B non thinking teacher. The best-performing result within each setup is highlighted in **bold**, and the second-best result is underlined.

Benchmark	Method	Pass@1	Pass@2	Pass@4	Pass@8	Pass@16	Pass@32	Pass@64
AMC23	KD	51.81	56.63	66.27	71.08	77.11	**85.54**	87.95
	OPD	51.81	55.42	63.86	73.49	79.97	83.13	86.75
	EOPD	51.81	60.24	**72.29**	75.90	80.72	84.34	87.95
	LSPD	**54.22**	**62.65**	71.08	**79.52**	**81.93**	**85.54**	**89.16**
AIME24	KD	**20.00**	**26.67**	**30.00**	**33.33**	33.33	36.67	40.00
	OPD	16.67	20.00	23.33	26.67	30.00	36.67	40.00
	EOPD	13.33	23.33	**30.00**	**33.33**	**36.67**	40.00	43.33
	LSPD	16.67	23.33	26.67	30.00	**36.67**	**43.33**	**46.67**
AIME25	KD	10.00	23.33	26.67	33.33	36.67	43.33	43.33
	OPD	**20.00**	**26.67**	**33.33**	**36.67**	**40.00**	43.33	46.67
	EOPD	13.33	20.00	**33.33**	**36.67**	**40.00**	**46.67**	**50.00**
	LSPD	13.33	20.00	26.67	33.33	36.67	**46.67**	**50.00**

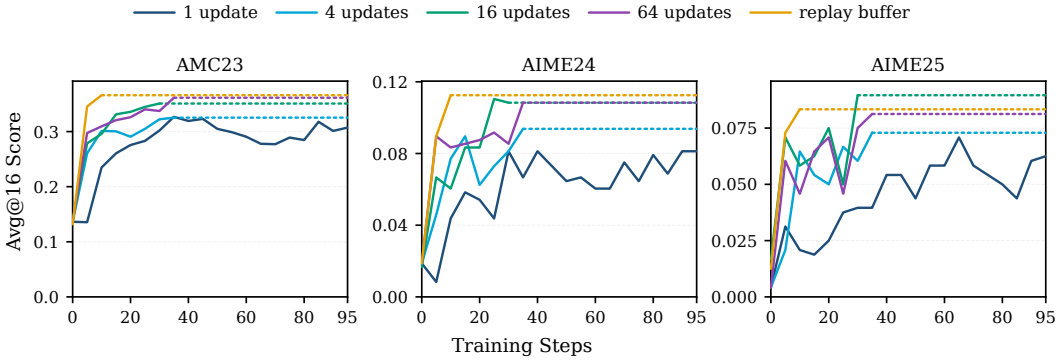


Figure 2: **Off-Policy Training Curves.** We plot the evolution of Avg@16 on AMC23, AIME24, and AIME25 for LSPD with $N \in \{1, 4, 16, 64\}$ optimization steps per rollout batch, together with the replay-buffer variant LSPD-RB. For LSPD-RB, after each newly collected rollout batch is added to the replay buffer, we perform 256 optimization steps on batches sampled from the buffer. We indicate the saturated performance after convergence using dotted horizontal lines. Each training step collects one rollout batch consisting of 64 prompts with 4 responses sampled per prompt.

For the semi-on-policy setting, increasing the number of updates per rollout batch consistently improves sample efficiency, requiring fewer rollout batches to reach saturated performance. This effect is particularly pronounced on AIME24 and AIME25: with $N \geq 4$, performance typically saturates after roughly 30 training steps, whereas $N = 1$ requires more than 50 steps. The gains become substantially smaller beyond $N = 4$, however, suggesting that the sample-efficiency benefit of additional updates eventually saturates. In contrast, the fully off-policy LSPD-RB variant reaches its saturated performance within approximately 10 training steps, substantially faster than all semi-on-policy variants. These results demonstrate that LSPD can effectively reuse historical data and that replay-based off-policy training can further improve its sample efficiency.

Ablation Studies. We ablate the entropy regularization component introduced in Eq. 4.6 to study its effect on reasoning performance and solution diversity. As shown in Table 3, removing entropy regularization has only a modest effect on Avg@16, but leads to a substantially larger degradation in Pass@16. Averaged across the six benchmarks, removing entropy regularization decreases Avg@16 by only 0.17 points for LSPD and 0.75 points for LSPD-RB, while Pass@16 drops by 2.26 and 1.64 points, respectively. Across both methods and all benchmarks, this corresponds to an average degradation of 0.46 points in Avg@16 compared with 1.95 points in Pass@16. Notably, Pass@16 decreases in all 12 method–benchmark comparisons without entropy regularization, whereas the changes in Avg@16 are comparatively small and mixed. These results suggest that entropy regularization primarily improves the diversity and coverage of generated solutions, rather than substantially changing the average per-sample accuracy.

Table 3: **Effect of Entropy Regularization.** Avg@16 and Pass@16 performance of LSPD and LSPD-RB with and without entropy regularization across six mathematical reasoning benchmarks.

Reg.	Method	MATH-500		Minerva		Olympiad		AMC23		AIME24		AIME25	
		Avg@16	Pass@16	Avg@16	Pass@16	Avg@16	Pass@16	Avg@16	Pass@16	Avg@16	Pass@16	Avg@16	Pass@16
w/ Ent.	LSPD	70.86	91.80	29.32	55.15	33.47	59.70	36.97	67.47	11.88	30.00	8.33	23.33
	LSPD-RB	70.69	91.00	29.57	54.41	33.50	61.04	36.60	67.47	11.67	36.67	8.33	23.33
w/o Ent.	LSPD	71.14	90.60	28.65	51.84	32.91	58.52	37.12	66.27	11.46	26.67	8.54	20.00
	LSPD-RB	69.39	90.60	28.84	53.31	33.65	60.59	36.67	66.27	10.00	33.33	7.29	20.00

6 Theoretical Analysis

In this section, we establish the theoretical advantage of the proposed algorithm by explicitly enforcing optimism and exploiting the value-based KL-regularized RL. The detailed proof is deferred to Appendix C. Following the notation in Section 3, we define the cumulative regret as

$$\begin{aligned}\mathrm{Regret}(K) &:= \sum_{k=1}^{K} \mathbb{E}_{\mathbf{q}\sim\mathcal{D},\,\mathbf{y}\sim\pi_k(\cdot|\mathbf{q})}\left[\sum_{t=1}^{T} D_{\mathrm{KL}}(\pi_k(\cdot|\mathbf{x}_{<t})\|\pi^*(\cdot|\mathbf{x}_{<t}))\right] \\ &= \sum_{k=1}^{K} \mathbb{E}_{\mathbf{q}\sim\mathcal{D}}\left[D_{\mathrm{KL}}(\pi_k(\cdot|\mathbf{q})\,\|\,\pi^*(\cdot|\mathbf{q}))\right]. \qquad (6.1)\end{aligned}$$

Here $\mathbf{x}_{<t} = (\mathbf{q}, \mathbf{y}_{<t})$, K counts rollout/update rounds, and T is the response-length and the second equality is due to the chain rule of KL divergence. The policy π^* denotes the optimal policy. Similarly with Sriraman et al. (2026), we assume the noisy teacher feedback in π^E as detailed in Assumption C.1. Then the following assumption is also crucial in understanding OPD:

Assumption 6.1. Let $\mathcal{R} \subseteq \{r : \mathcal{X} \times \mathcal{Y} \to [-B, B]\}$ be a reward class with $B < \infty$. For the optimal policy π^* and the reference policy π^{ref}, we assume $r^*(\mathbf{x}_{<t}, y_t) := \log \frac{\pi^*(y_t|\mathbf{x}_{<t})}{\pi^{\mathrm{ref}}(y_t|\mathbf{x}_{<t})} \in \mathcal{R}$.

In practice, one can choose π^{ref} as the initial student model π_0 thus Assumption 6.1 assumes the well coverage between the student and teacher model.

Theorem 6.2 (Logarithmic reverse-KL regret, informal)**.** Under Assumptions C.1 and 6.1, with high probability, Algorithm 2 satisfies $\mathrm{Regret}(K) = \mathcal{O}(d_E \log(N_{\mathcal{R}} K))$ where $d_E, N_{\mathcal{R}}$ are both complexity measurements in the order of $\widetilde{\mathcal{O}}(d)$ when the reward is a d-dimensional linear function.

We defer the detailed proof and further discussion to Appendix C, and highlight how Theorem 6.2 informs the comparison between reverse-KL-based OPD and forward-KL-based SFT.

Remark 6.3. Theorem 6.2 implies a $\widetilde{\mathcal{O}}(\epsilon^{-1})$ sample complexity for obtaining a mixture policy $\bar{\pi} = \frac{1}{K}\sum_{k=1}^{K} \pi_k$ whose reverse-KL divergence to π^* is at most ϵ under **bounded log-ratio reward**. By comparison, Foster et al. (2024) establish a $\widetilde{\mathcal{O}}(\epsilon^{-1})$ sample complexity for attaining an ϵ-small imitation performance gap through maximum likelihood estimation (MLE) with a **deterministic expert**. Although these guarantees concern different error criteria, they highlight complementary conditions supporting fast learning. In reverse KL minimization including OPD and our LSPD, the fast rates depends on the coverage between the student and teacher model with $B < \infty$ according to Assumption 6.1. This dependence on the coverage is also reported empirically in Li et al. (2026b); Xing et al. (2026); Fu et al. (2026). In contrast, the SFT with MLE exploits expert determinism and exhibits the "mass-covering" tendency in minimizing the forward KL. This comparison explains the practical implementations for SFT in establishing an initial policy followed by reverse-KL-based refinement with adequate coverage.

7 Conclusion

We presented Least Square Policy Distillation (LSPD), an RL-inspired framework that connects reverse-KL distillation to KL-regularized policy optimization through a teacher-induced reward. This perspective motivates an optimistic least-squares formulation that enables trajectory reuse with a purely off-policy replay buffer. Experiments across six mathematical reasoning benchmarks and multiple teacher–student settings show improved reasoning performance, stronger Pass@k scaling, and greater rollout efficiency. In particular, LSPD-RB achieves performance comparable to vanilla OPD using only the first 25% of rollout batches. Together with a sharp reverse-KL regret guarantee, these results show how value-based RL principles can improve policy distillation by preserving policy diversity and enhancing sample efficiency through off-policy learning.

Reproducibility Statement

To support reproducibility, we describe the teacher–student configurations, training dataset, evaluation benchmarks, and generation settings in Section 5.1, and summarize the practical training procedure in Algorithm 1. Training hyperparameters and replay-buffer settings are reported in Table 4, while the robust penalty and prompt template are detailed in Appendices A.1 and A.3, respectively. The assumptions and proofs underlying our theoretical guarantees are provided in Appendix C. We will include the anonymized source code for LSPD and LSPD-RB in the supplementary material to facilitate reproduction of our experiments.

Ethics Statement

This work studies language model distillation using existing pretrained models and established benchmark datasets. Although LSPD and LSPD-RB aim to improve reasoning performance and rollout efficiency, these improvements should not be interpreted as guarantees of safety, fairness, or factual reliability. Distillation may transfer undesirable behaviors from teacher models and training data, including biases and harmful or inaccurate outputs. Downstream applications should therefore undergo appropriate safety evaluation and incorporate safeguards suited to their intended use. Use and redistribution of the underlying models, datasets, and code should respect their applicable licenses and terms of use.

AI Use Statement

We used generative AI tools to assist with mathematical derivations and proof development for the theoretical analysis of Least Square Policy Distillation (LSPD), drafting and revising the manuscript, and implementing the methods and experimental code. All AI-assisted content was reviewed by the authors for correctness, including checking mathematical arguments against the stated assumptions, reviewing the text for accuracy and clarity, and inspecting the code for consistency with the described methods. The authors take full responsibility for the final content of this work, including all theoretical claims, proofs, experimental results, and associated code.

References

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, volume 2024, pp. 21246–21263, 2024.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Thomas Degris, Martha White, and Richard S. Sutton. Off-policy actor-critic. In *Proceedings of the 29th International Conference on Machine Learning*, 2012.

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

Zixuan Fu, Bingxiang He, Yuxin Zuo, Haohuan Huang, Jinqian Zhang, Ruhang Xiao, Cheng Qian, Qinyu Luo, Huan-ang Gao, Yudong Wang, et al. Rethinking on-policy distillation of large language models ii: One training example. *arXiv preprint arXiv:2609.04172*, 2026.

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. A general theoretical paradigm to understand learning from human preferences. In *Proceedings of the 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pp. 4447–4455. PMLR, 2024.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. In *International Conference on Learning Representations*, volume 2024, pp. 32694–32717, 2024.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In *International conference on machine learning*, pp. 1352–1361. PMLR, 2017.

Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3828–3850, 2024.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

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

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the association for computational linguistics: ACL 2023*, pp. 8003–8017, 2023.

Woogyeol Jin, Taywon Min, Yongjin Yang, Dennis Wei, Yi Zhou, Swanand Ravindra Kadhe, Nathalie Baracaldo, and Kimin Lee. Entropy-aware on-policy distillation of language models. *arXiv preprint arXiv:2603.07079*, 2026.

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

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. DistiLLM: Towards streamlined distillation for large language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 24872–24895. PMLR, 2024.

Jongwoo Ko, Tianyi Chen, Sungnyun Kim, Tianyu Ding, Luming Liang, Ilya Zharkov, and Se-Young Yun. DistiLLM-2: A contrastive approach boosts the distillation of LLMs. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 31044–31062. PMLR, 2025.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in neural information processing systems*, 35:3843–3857, 2022.

Gengsheng Li, Mao Zheng, Mingyang Song, Ruiqi Liu, Tianyu Yang, Jie Sun, Qiyong Zhong, Haiyun Guo, Junfeng Fang, Dan Zhang, et al. On-policy distillation with curriculum turn-level guidance for multi-turn agents. *arXiv preprint arXiv:2606.15912*, 2026a.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huanang Gao, Wenkai Yang, Zhiyuan Liu, et al. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe. *arXiv preprint arXiv:2604.13016*, 2026b.

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

Team Math-AI. American invitational mathematics examination (aime) 2024, 2024.

Team Math-AI. American invitational mathematics examination (aime) 2025, 2025.

Mathematical Association of America. American mathematics competitions. American Mathematics Competitions, 2023.

Rémi Munos, Tom Stepleton, Anna Harutyunyan, and Marc G. Bellemare. Safe and efficient off-policy reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 29, pp. 1046–1054, 2016.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744, 2022.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*, 2023.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

John Schulman, Xi Chen, and Pieter Abbeel. Equivalence between policy gradients and soft q-learning. *arXiv preprint arXiv:1704.06440*, 2017a.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017b.

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

Ved Sriraman, Peihan Liu, Daniel Hsu, and Adam Block. Behavior cloning is not all you need: The optimality of on-policy distillation for noisy expert feedback. *arXiv preprint arXiv:2606.30923*, 2026.

Christopher J. C. H. Watkins and Peter Dayan. Q-learning. *Machine Learning*, 8(3):279–292, 1992.

Taiqiang Wu, Chaofan Tao, Jiahao Wang, Runming Yang, Zhe Zhao, and Ngai Wong. Rethinking kullback-leibler divergence in knowledge distillation for large language models. In *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 5737–5755, 2025.

Tengyang Xie, Dylan Foster, Akshay Krishnamurthy, Corby Rosset, Ahmed H Awadallah, and Alexander Rakhlin. Exploratory preference optimization: Harnessing implicit q*-approximation for sample-efficient rlhf. In *International Conference on Learning Representations*, volume 2025, pp. 43632–43669, 2025.

Xingrun Xing, Haoqing Wang, Boyan Gao, Ziheng Li, and Yehui Tang. Trust region on-policy distillation. *arXiv preprint arXiv:2606.01249*, 2026.

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

Wenkai Yang, Weijie Liu, Ruobing Xie, Kai Yang, Saiyong Yang, and Yankai Lin. Learning beyond teacher: Generalized on-policy distillation with reward extrapolation. *arXiv preprint arXiv:2602.12125*, 2026a.

Yuxiao Yang, Tianrun Yu, Shangzhe Li, Kaixiang Zhao, Xuchao Zhang, Chetan Bansal, Huaxiu Yao, Taylor W Killian, and Weitong Zhang. When eos tokens disagree: Understanding length inflation in on-policy distillation. *arXiv preprint arXiv:2609.20511*, 2026b.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244, 2026a.

Zishun Yu, Shangzhe Li, and Xinhua Zhang. Language model distillation: A temporal difference imitation learning perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 34512–34520, 2026b.

Heyang Zhao, Chenlu Ye, Wei Xiong, Quanquan Gu, and Tong Zhang. Logarithmic regret for online KL-regularized reinforcement learning. In *Forty-second International Conference on Machine Learning*, 2025.

Heyang Zhao, Chenlu Ye, Quanquan Gu, and Tong Zhang. Sharp analysis for kl-regularized contextual bandits and rlhf. *Advances in Neural Information Processing Systems*, 38:107964–108002, 2026.

Brian D. Ziebart, Andrew Maas, J. Andrew Bagnell, and Anind K. Dey. Maximum entropy inverse reinforcement learning. AAAI'08. AAAI Press, 2008. ISBN 9781577353683.

Yuxin Zuo, Kaiyan Zhang, Li Sheng, Shang Qu, Ganqu Cui, Xuekai Zhu, Haozhan Li, Xinwei Long, Ermo Hua, Biqing Qi, et al. Ttrl: Test-time reinforcement learning. *Advances in Neural Information Processing Systems*, 38:131459–131483, 2026.

A Hyperparameters and Implementation Details

A.1 Details on Huber-Type Penalty in Practical Implementation

For improved numerical stability, we employ a Huber-type penalty in the practical implementations of LSPD and LSPD-RB. Specifically, let $\Delta_t(\pi, \pi^E) = \log \pi(y_t|\mathbf{x}_{<t}) - \log \pi^E(y_t|\mathbf{x}_{<t})$, we define

$$\psi\big(\Delta_t(\pi, \pi^E)\big) = \begin{cases} \Delta_t^2(\pi, \pi^E), & \big|\Delta_t(\pi, \pi^E)\big| \le c, \\ 2c\big|\Delta_t(\pi, \pi^E)\big| - c^2, & \big|\Delta_t(\pi, \pi^E)\big| > c, \end{cases} \tag{A.1}$$

where $c > 0$ denotes the threshold separating the quadratic and linear regimes. This formulation preserves the original quadratic objective when the log-probability discrepancy is moderate, while replacing the quadratic growth with a linear penalty for large discrepancies, thereby reducing the influence of extreme outliers and improving optimization stability. We provide an ablation study of this design choice in Appendix B.

A.2 Training Details

Our experimental implementation of LSPD and LSPD-RB builds upon the on-policy distillation codebase of Li et al. (2026b) and adopts the EOS corrections in Yang et al. (2026b). Detailed training hyperparameters are summarized in Table 4. For comparison, we reproduce the OPD (Lu & Lab, 2025), EOPD (Jin et al., 2026), and KD (Hinton et al., 2015; Kim & Rush, 2016) baselines within the same implementation framework and under the same experimental setup. Across all methods, we keep the training batch size, rollout and evaluation configurations, and optimizer settings fixed to ensure a controlled comparison. All experiments are conducted using four NVIDIA RTX PRO 6000 GPUs, each equipped with 96GB of VRAM.

A.3 Prompt Template

For both training rollouts and evaluation, we follow the mathematical reasoning prompt template used in Test-Time Reinforcement Learning (TTRL) (Zuo et al., 2026). Specifically, we append the following instruction to each problem:

Prompt Template

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

B Additional Experimental Results

Out-of-Domain Results. We further evaluate whether improvements from LSPD transfer beyond the mathematical reasoning domain used for distillation. Specifically, although the student is trained exclusively on mathematical dataset DAPO-Math-17K (Yu et al., 2026a), we evaluate KD (Hinton et al., 2015; Kim & Rush, 2016), OPD (Lu & Lab, 2025), EOPD (Jin et al., 2026), and our proposed LSPD and replay-buffer variant LSPD-RB on four out-of-domain benchmarks spanning general knowledge, commonsense reasoning, and code generation: GPQA (Rein et al., 2023), MMLU (Hendrycks et al., 2020), WinoGrande (Sakaguchi et al., 2021), and HumanEval (Chen et al., 2021). We report 0-shot accuracy on GPQA Main (448 questions), 5-shot accuracy on MMLU (14,042 questions), and 0-shot validation accuracy on WinoGrande (1,267 questions). For HumanEval, we report accuracy using one greedy completion for each of the 164 tasks. For all out-of-domain experiments, we leverage a Qwen3-4B non-thinking teacher and a Qwen3-1.7B-Base student.

Results are shown in Table 5. Averaged across the four out-of-domain benchmarks, LSPD achieves a score of 56.02, improving upon the strongest baseline, EOPD (55.01), by $+1.02$ percentage points. LSPD-RB further increases the average score to 56.84, corresponding to a $+1.84$ point improvement over EOPD. Notably, LSPD-RB achieves the best performance on GPQA, MMLU, and HumanEval, while LSPD obtains the best result on WinoGrande. These results indicate that the gains from LSPD are not restricted to the in-domain mathematical reasoning tasks used during training; instead, the

Table 4: **Hyperparameters for LSPD and LSPD-RB.** We report the training hyperparameters used for our proposed LSPD and LSPD-RB methods across all experiments. For the off-policy experiments without a replay buffer, we vary the number of optimizer updates per rollout iteration over $N \in \{1, 4, 16, 64\}$.

Hyperparameter	w/o Replay Buffer	w/ Replay Buffer
Entropy regularization coefficient μ^{-1}	0.1	0.1
Squared-loss robustification	Huber	Huber
Huber transition threshold c	5.0	5.0
Optimizer	AdamW	AdamW
Learning rate	10^{-6}	10^{-6}
Learning-rate schedule	Constant	Constant
Warmup steps	0	0
Optimizer momentum coefficients	(0.9, 0.999)	(0.9, 0.999)
Weight decay	0.01	0.01
Maximum gradient norm	1.0	1.0
Maximum prompt length (tokens)	1,024	1,024
Maximum response length (tokens)	7,168	7,168
Rollout sampling temperature	1.0	1.0
Teacher distribution temperature	1.0	1.0
Vocabulary truncation	None	None
Prompts per rollout iteration	64	64
Responses per prompt	4	4
New trajectories per rollout iteration	256	256
Trajectories per optimizer update	64	64
Optimizer updates per rollout iteration	4	256
Rollout iterations	100	100
Replay buffer capacity (trajectories)	—	65,536
Replay eviction policy	—	First in, first out
Replay sampling	—	Uniform

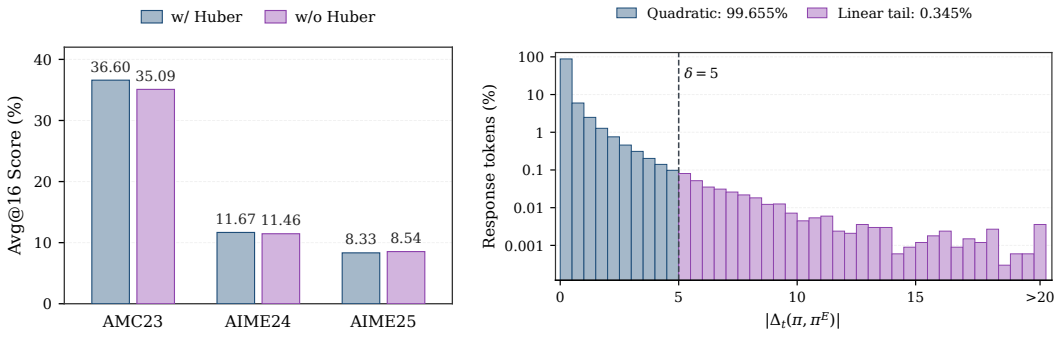
Table 5: **Results on Out-of-Domain Benchmarks.** Performance comparison of KD, OPD, EOPD, LSPD, and LSPD-RB on GPQA, MMLU, WinoGrande, and HumanEval. The student is trained only on mathematical reasoning data. The best result on each benchmark is shown in **bold**, and the second-best result is underlined.

Method	GPQA	MMLU	WinoGrande	HumanEval
KD	<u>30.13</u>	61.24	63.85	64.63
OPD	27.68	61.14	63.61	67.07
EOPD	27.90	61.47	<u>64.80</u>	65.85
LSPD	29.91	<u>61.61</u>	**64.88**	<u>67.68</u>
LSPD-RB	**30.58**	**62.03**	64.64	**70.12**

learned policy retains and, in several cases, improves general reasoning and coding capabilities, suggesting better preservation of the student's out-of-domain capabilities during distillation.

Ablation on Huber-Type Robust Penalty. We ablate the Huber-type robust penalty introduced in Eq. A.1. Specifically, we distill a Qwen3-1.7B-Base student from a Qwen3-4B non-thinking teacher using LSPD-RB, with and without the Huber-type penalty. The results are shown in Figure 3. Across AMC23, AIME24, and AIME25, the Avg@16 results show that incorporating the Huber-type penalty consistently improves performance, yielding an average gain of $+0.50$ points.

We further analyze the distribution of token-level log-probability differences using the step-10 checkpoint of the distilled Qwen3-1.7B-Base model. We sample 128 responses from 32 training prompts in DAPO-Math-17K, resulting in approximately 335K evaluated tokens. We find that



(a) Performance with and without Huber.

(b) Log-probability difference magnitudes.

Figure 3: **Huber-Type Penalty Ablation.** We evaluate the effect of the Huber-type penalty on AMC23, AIME24, and AIME25. The left figure compares Avg@16 performance with and without the Huber-type penalty, while the right figure shows the distribution of the log-ratio discrepancy $\Delta_t(\pi, \pi^E)$. Overall, incorporating the Huber-type penalty yields a small but consistent improvement in performance. We further observe that the objective operates almost entirely in its quadratic regime: only 0.345% of the evaluated token-level discrepancies satisfy $\Delta_t(\pi, \pi^E) > 5$ and therefore fall into the linear tail. This suggests that the Huber formulation primarily behaves as a quadratic penalty in practice, while its linear tail provides additional robustness to the small fraction of large log-ratio discrepancies.

Algorithm 2 Least Square Policy Distillation (Theoretical)

Input: Reference policy π^{ref}, reward class $\mathcal{R}$, teacher feedback π_k^E, rounds K, response-length bound T, parameters λ, β; $\eta = 1$.

Initialize $\mathcal{D}_0 = \emptyset$, $\mathcal{C}_0 = \mathcal{R}$, and $\pi_1 = \pi_{r_0^+}$ with $r_0^+ = \sup_{r \in \mathcal{R}} r$ pointwise.
for $k = 1, \ldots, K$ **do**
  Sample $\mathbf{q}_k \sim \mathcal{D}$ and initialize $\mathbf{x}_{k,<1} = (\mathbf{q}_k, \emptyset)$.
  for $t = 1, \ldots, T$ **do**
    Sample $y_{k,t} \sim \pi_k(\cdot|\mathbf{x}_{k,<t})$.
    Observe $\ell_{k,t} = \log\big(\pi_k^E(y_{k,t}|\mathbf{x}_{k,<t}) / \pi^{\text{ref}}(y_{k,t}|\mathbf{x}_{k,<t})\big)$.
    Append the token: $\mathbf{x}_{k,<t+1} = (\mathbf{x}_{k,<t}, y_{k,t})$; stop at termination.
  end for
  Store the prefix–token pairs and targets from rollout k in $\mathcal{D}_k$.
  Fit $\widehat{r}_k$ to all stored token targets by Eq. C.8.
  Construct the estimator-centered confidence set $\mathcal{C}_k$ in Eq. C.13.
  Set $r_k^+(\mathbf{x}_{<t}, y_t) = \sup_{r \in \mathcal{C}_k} r(\mathbf{x}_{<t}, y_t)$ for every prefix–token pair.
  Set $\pi_{k+1}(y_t|\mathbf{x}_{<t}) \propto \pi^{\text{ref}}(y_t|\mathbf{x}_{<t}) \exp\big(r_k^+(\mathbf{x}_{<t}, y_t)\big)$.
end for

99.655% of tokens fall within the quadratic region of Eq. A.1, whereas only 0.345% exhibit sufficiently large log-probability discrepancies to enter the linear region. This suggests that the objective behaves predominantly as a quadratic penalty during training, while the linear tail robustly handles rare large discrepancies and improves training stability.

C Theoretical Analysis and Proofs

This appendix proves Theorem 6.2 by adapting the contextual-bandit analysis of Zhao et al. (2025) to conditional next-token distributions. We apply the sharp bandit inequality at each fixed prefix and then sum it along student-generated responses using the KL chain rule. No transition model, action-value function, or Bellman recursion is required. The reduction uses the special reward $r^* = \log(\pi^*/\pi^{\text{ref}})$, whose target partition function equals one at every prefix; deterministic token concatenation alone would not justify a bandit analysis for an arbitrary sequential reward.

We first assume $N_{\mathcal{R}} = |\mathcal{R}| < \infty$ and give the covering-number extension in Appendix C.5. Throughout, k indexes rollout/update rounds and t indexes tokens in the finite vocabulary $\mathcal{Y}$. Queries are drawn from the dataset distribution $\mathcal{D}$ of Section 3. The policy π_k remains fixed while generating $\mathbf{y}_k$, and all its token observations become available for the update to π_{k+1}. Responses have at most T tokens. Equivalently, completed responses may be padded with an absorbing token on which all policies agree and all rewards and regression errors are zero.

C.1 Setup and Reverse-KL Identity

Let $\mathcal{F}_{k-1}$ be the history before sampling query $\mathbf{q}_k$. Within rollout k, let $\mathcal{F}_{k,t-1}$ contain this history, the current query, and the tokens and teacher feedback preceding position t. Thus $\mathbf{x}_{k,<t} = (\mathbf{q}_k, \mathbf{y}_{k,<t})$ is known before drawing $y_{k,t}$. We use $\mathcal{F}_k$ for the history after the complete rollout and its feedback.

Assumption C.1 (Conditionally unbiased teacher)**.** There exists a target policy π^* such that, for every observed prefix–token pair $(\mathbf{x}_{k,<t}, y_{k,t})$,

$$\log \pi_k^E(y_{k,t}|\mathbf{x}_{k,<t}) = \log \pi^*(y_{k,t}|\mathbf{x}_{k,<t}) + \epsilon_{k,t},$$
$$\mathbb{E}[\epsilon_{k,t}|\mathcal{F}_{k,t-1}, y_{k,t}] = 0.$$

Moreover, the noise is conditionally σ-sub-Gaussian: for every $u \in \mathbb{R}$,

$$\mathbb{E}[\exp(u\epsilon_{k,t}) \,|\, \mathcal{F}_{k,t-1}, y_{k,t}] \le \exp(\sigma^2 u^2/2).$$

Remark C.2. Explicit modeling of teacher noise has precedent in the theoretical study of on-policy distillation. In particular, Sriraman et al. (2026) distinguish a clean expert π^* from an observed teacher corrupted at the token level according to $\pi^E(y_t|\mathbf{x}_{<t}) = (1-\zeta)\pi^*(y_t|\mathbf{x}_{<t}) + \zeta\nu_t(y_t|\mathbf{x}_{<t})$, where $\zeta \in [0, 1)$ is the corruption level and ν_t is a corruption policy. Their analysis considers known corruption and, for unknown corruption, imposes a deterministic clean expert together with smoothness and margin conditions on the corruption. Assumption C.1 follows the same motivation of distinguishing target behavior from imperfect teacher feedback, but adopts a different statistical model: the observed token-level feedback $\log \pi_k^E(y_{k,t}|\mathbf{x}_{k,<t})$ is conditionally unbiased for $\log \pi^*(y_{k,t}|\mathbf{x}_{k,<t})$, with sub-Gaussian deviations. Conditional unbiasedness centers the induced reward-regression target at r^*, while the tail condition enables concentration under adaptive prefix and token sampling. Thus, introducing explicit structural assumptions on teacher noise is consistent with prior theoretical work. Our particular condition is a complementary analytical idealization tailored to least-squares reward estimation, rather than a consequence of their mixture-corruption model.

Definition C.3 (Reward representation and regularized objective)**.** For a fixed reference policy π^{ref}, define the target token reward by

$$r^*(\mathbf{x}_{<t}, y_t) := \log \frac{\pi^*(y_t|\mathbf{x}_{<t})}{\pi^{\mathrm{ref}}(y_t|\mathbf{x}_{<t})}. \tag{C.1}$$

For any reward function r, define its prefix-wise partition function and conditional Gibbs policy by

$$Z_r(\mathbf{x}_{<t}) := \sum_{y_t \in \mathcal{Y}} \pi^{\mathrm{ref}}(y_t|\mathbf{x}_{<t}) \exp\big(r(\mathbf{x}_{<t}, y_t)\big), \tag{C.2}$$

$$\pi_r(y_t|\mathbf{x}_{<t}) := \frac{\pi^{\mathrm{ref}}(y_t|\mathbf{x}_{<t}) \exp\big(r(\mathbf{x}_{<t}, y_t)\big)}{Z_r(\mathbf{x}_{<t})}. \tag{C.3}$$

The sequence distribution is the autoregressive product of these conditionals, not a single sequence-level Gibbs normalization. With $\eta = 1$, the regularized objective is

$$J(\pi) := \mathbb{E}_{\mathbf{q}\sim\mathcal{D},\, \mathbf{y}\sim\pi(\cdot|\mathbf{q})} \left[\sum_{t=1}^{T} \left(r^*(\mathbf{x}_{<t}, y_t) - \log \frac{\pi(y_t|\mathbf{x}_{<t})}{\pi^{\mathrm{ref}}(y_t|\mathbf{x}_{<t})} \right) \right]. \tag{C.4}$$

Reverse-KL identity. Substituting Eq. C.1 into Eq. C.4 and conditioning on each prefix gives

$$\begin{aligned} J(\pi) &= -\mathbb{E}_{\mathbf{q}\sim\mathcal{D},\, \mathbf{y}\sim\pi(\cdot|\mathbf{q})} \left[\sum_{t=1}^{T} D_{\mathrm{KL}}(\pi(\cdot|\mathbf{x}_{<t}) \,\|\, \pi^*(\cdot|\mathbf{x}_{<t})) \right] \\ &= -\mathbb{E}_{\mathbf{q}\sim\mathcal{D}} \left[D_{\mathrm{KL}}(\pi(\cdot|\mathbf{q}) \,\|\, \pi^*(\cdot|\mathbf{q})) \right], \end{aligned} \tag{C.5}$$

where the second equality is the autoregressive KL chain rule. In particular, $Z_{r^*}(\mathbf{x}_{<t}) = 1$ and $\pi_{r^*} = \pi^*$ at every prefix. Hence $J(\pi^*) = 0$, and

$$J(\pi^*) - J(\pi) = \mathbb{E}_{\mathbf{q}\sim\mathcal{D},\,\mathbf{y}\sim\pi(\cdot|\mathbf{q})}\left[\sum_{t=1}^{T} D_{\mathrm{KL}}(\pi(\cdot|\mathbf{x}_{<t}) \,\|\, \pi^*(\cdot|\mathbf{x}_{<t}))\right]. \tag{C.6}$$

Thus $\mathrm{Regret}(K) = \sum_{k=1}^{K}[J(\pi^*) - J(\pi_k)]$ is exactly the regret in Eq. 6.1, not a token-averaged surrogate.

Definition C.4 (Regression target and reward estimator)**.** At token t of rollout k, the observed regression target is

$$\ell_{k,t} := \log \frac{\pi_k^E(y_{k,t}|\mathbf{x}_{k,<t})}{\pi^{\mathrm{ref}}(y_{k,t}|\mathbf{x}_{k,<t})}. \tag{C.7}$$

Let $\mathcal{D}_k$ store the prefix–token pairs $\{(\mathbf{x}_{i,<u}, y_{i,u}) : 1 \le i \le k,\ 1 \le u \le T\}$ and their observed targets. The empirical version of Eq. 4.2 is

$$\widehat{r}_k \in \arg\min_{r\in\mathcal{R}} \sum_{i=1}^{k}\sum_{u=1}^{T}\big(r(\mathbf{x}_{i,<u}, y_{i,u}) - \ell_{i,u}\big)^2. \tag{C.8}$$

We use $\mathcal{D}_0 = \emptyset$ and any $\widehat{r}_0 \in \mathcal{R}$.

Under Assumption C.1,

$$\ell_{k,t} = r^*(\mathbf{x}_{k,<t}, y_{k,t}) + \epsilon_{k,t}, \qquad \mathbb{E}[\epsilon_{k,t}|\mathcal{F}_{k,t-1}, y_{k,t}] = 0. \tag{C.9}$$

The conditional formulation permits adaptive prefixes and does not require independent observations within a response.

Definition C.5 (Cumulative eluder complexity)**.** Fix $\lambda > 0$. After k completed rollouts, define

$$U_k(\mathbf{x}_{<t}, y_t) := \sup_{r_1,r_2\in\mathcal{R}} \frac{|r_1(\mathbf{x}_{<t}, y_t) - r_2(\mathbf{x}_{<t}, y_t)|}{\sqrt{\lambda + \sum_{i=1}^{k}\sum_{u=1}^{T}\big(r_1(\mathbf{x}_{i,<u}, y_{i,u}) - r_2(\mathbf{x}_{i,<u}, y_{i,u})\big)^2}}, \tag{C.10}$$

including $k = 0$ with an empty denominator sum. For K rollout rounds, define

$$d_E(\mathcal{R}, \lambda, K, T) := \sup_{(\mathbf{q}_k,\mathbf{y}_k)_{k=1}^{K}} \sum_{k=1}^{K}\sum_{t=1}^{T} \min\big\{1,\, U_{k-1}(\mathbf{x}_{k,<t}, y_{k,t})^2\big\}, \tag{C.11}$$

where the supremum ranges over valid query–response sequences, $\mathbf{x}_{k,<t} = (\mathbf{q}_k, \mathbf{y}_{k,<t})$, and U_{k-1} uses only their first $k-1$ completed rollouts. We abbreviate this quantity as d_E.

This is the rollout-batched token-level version of the cumulative contextual-bandit complexity. It bounds the clipped squared widths along every realized adaptive sequence. Importantly, its denominator does not use tokens from the current rollout before the policy is updated. At $T = 1$, it is the usual bandit definition. Retaining only historical observations at the same token position can only enlarge each width; consequently, d_E is at most the sum of the T corresponding K-round bandit complexities. Thus response-length dependence is explicit in the complexity definition, without introducing a transition-estimation or Bellman-error term.

Definition C.6 (Confidence set, bonus, and optimistic reward)**.** For $\delta \in (0, 1)$, set

$$\beta := \max\left\{2B,\ \sqrt{16\sigma^2 \log\frac{2N_{\mathcal{R}}K}{\delta} + 2\lambda}\right\}. \tag{C.12}$$

Define the estimator-centered confidence set by

$$\mathcal{C}_k := \left\{r \in \mathcal{R} : \lambda + \sum_{i=1}^{k}\sum_{u=1}^{T}\big(r(\mathbf{x}_{i,<u}, y_{i,u}) - \widehat{r}_k(\mathbf{x}_{i,<u}, y_{i,u})\big)^2 \le \beta^2\right\}. \tag{C.13}$$

Its pointwise optimistic reward and an auxiliary confidence bonus are

$$r_k^+(\mathbf{x}_{<t}, y_t) := \sup_{r\in\mathcal{C}_k} r(\mathbf{x}_{<t}, y_t), \qquad b_k(\mathbf{x}_{<t}, y_t) := \min\{2B,\, \beta U_k(\mathbf{x}_{<t}, y_t)\}. \tag{C.14}$$

For noisy feedback, Eq. C.13 makes the theoretical confidence construction precise by centering at $\widehat{r}_k$: an uncentered squared residual against noisy targets includes the accumulated noise energy and is not, by itself, a logarithmic-radius confidence set. Algorithm 2 and the proof both use the same pointwise optimistic reward r_k^+; b_k is only a bound on its estimation error. Since $\widehat{r}_k \in \mathcal{C}_k$, the set is always nonempty. The policy update is

$$\pi_{k+1}(y_t|\mathbf{x}_{<t}) = \pi_{r_k^+}(y_t|\mathbf{x}_{<t}). \tag{C.15}$$

In particular, $\mathcal{C}_0 = \mathcal{R}$ and $\pi_1 = \pi_{r_0^+}$. These are conditional token updates; they do not assert that a prefix-wise Gibbs policy globally maximizes an arbitrary sequence-reward objective.

C.2 Objective Decomposition and the Sharp One-Step Bound

For a fixed prefix $\mathbf{x}_{<t}$, define

$$\Delta(\mathbf{x}_{<t}, r) := -\log Z_r(\mathbf{x}_{<t}) + \mathbb{E}_{y_t\sim\pi_r(\cdot|\mathbf{x}_{<t})}\left[r(\mathbf{x}_{<t}, y_t) - r^*(\mathbf{x}_{<t}, y_t)\right]. \tag{C.16}$$

Lemma C.7 (Objective decomposition)**.** For every $k \geq 1$,

$$J(\pi^*) - J(\pi_k) = \mathbb{E}_{\mathbf{q}\sim\mathcal{D},\,\mathbf{y}\sim\pi_k(\cdot|\mathbf{q})}\left[\sum_{t=1}^{T}\left(\Delta(\mathbf{x}_{<t}, r_{k-1}^+) - \Delta(\mathbf{x}_{<t}, r^*)\right)\right]. \tag{C.17}$$

Proof. For a fixed prefix, Eq. C.3 gives

$$\log\frac{\pi_k(y_t|\mathbf{x}_{<t})}{\pi^{\mathrm{ref}}(y_t|\mathbf{x}_{<t})} = r_{k-1}^+(\mathbf{x}_{<t}, y_t) - \log Z_{r_{k-1}^+}(\mathbf{x}_{<t}). \tag{C.18}$$

Using $r^* = \log(\pi^*/\pi^{\mathrm{ref}})$, we obtain the prefix-wise identity

$$\begin{aligned} D_{\mathrm{KL}}(\pi_k(\cdot|\mathbf{x}_{<t}) \,\|\, \pi^*(\cdot|\mathbf{x}_{<t})) &= -\log Z_{r_{k-1}^+}(\mathbf{x}_{<t}) \\ &\quad + \mathbb{E}_{y_t\sim\pi_k(\cdot|\mathbf{x}_{<t})}\left[r_{k-1}^+(\mathbf{x}_{<t}, y_t) - r^*(\mathbf{x}_{<t}, y_t)\right]. \end{aligned} \tag{C.19}$$

Because $Z_{r^*}(\mathbf{x}_{<t}) = 1$ and $\pi_{r^*} = \pi^*$, the right-hand side is $\Delta(\mathbf{x}_{<t}, r_{k-1}^+) - \Delta(\mathbf{x}_{<t}, r^*)$, with $\Delta(\mathbf{x}_{<t}, r^*) = 0$. Sum over tokens and average over prefixes generated by π_k, using Eq. C.6. □

Lemma C.8 (Gradient of the functional gap)**.** For every fixed prefix $\mathbf{x}_{<t}$ and token y_t,

$$\begin{aligned} \frac{\partial\Delta(\mathbf{x}_{<t}, r)}{\partial r(\mathbf{x}_{<t}, y_t)} &= \pi_r(y_t|\mathbf{x}_{<t})\big(r(\mathbf{x}_{<t}, y_t) - r^*(\mathbf{x}_{<t}, y_t)\big) \\ &\quad - \pi_r(y_t|\mathbf{x}_{<t})\mathbb{E}_{y_t'\sim\pi_r(\cdot|\mathbf{x}_{<t})}\left[r(\mathbf{x}_{<t}, y_t') - r^*(\mathbf{x}_{<t}, y_t')\right]. \end{aligned} \tag{C.20}$$

Proof. Direct differentiation of Eqs. C.2–C.3, holding the prefix fixed, yields

$$\frac{\partial Z_r(\mathbf{x}_{<t})}{\partial r(\mathbf{x}_{<t}, y_t)} = \pi^{\mathrm{ref}}(y_t|\mathbf{x}_{<t})e^{r(\mathbf{x}_{<t}, y_t)}, \tag{C.21}$$

$$\frac{\partial\pi_r(y_t|\mathbf{x}_{<t})}{\partial r(\mathbf{x}_{<t}, y_t)} = \pi_r(y_t|\mathbf{x}_{<t})\big(1 - \pi_r(y_t|\mathbf{x}_{<t})\big), \tag{C.22}$$

$$\frac{\partial\pi_r(y_t'|\mathbf{x}_{<t})}{\partial r(\mathbf{x}_{<t}, y_t)} = -\pi_r(y_t'|\mathbf{x}_{<t})\pi_r(y_t|\mathbf{x}_{<t}), \qquad y_t' \neq y_t. \tag{C.23}$$

The derivative of $-\log Z_r(\mathbf{x}_{<t})$ is $-\pi_r(y_t|\mathbf{x}_{<t})$. Differentiating the expectation in Eq. C.16 produces a direct term $+\pi_r(y_t|\mathbf{x}_{<t})$, which cancels it. The remaining terms give Eq. C.20. No derivative of a prefix distribution is taken. □

Define the uniform optimism event

$$\mathcal{E}_{\mathrm{opt}} := \left\{r_k^+(\mathbf{x}_{<t}, y_t) \geq r^*(\mathbf{x}_{<t}, y_t),\ \forall k \in \{0, \ldots, K\},\ \forall(\mathbf{x}_{<t}, y_t) \in \mathcal{X}\times\mathcal{Y}\right\}. \tag{C.24}$$

Lemma C.9 (Sharp one-step bound under optimism)**.** On $\mathcal{E}_{\mathrm{opt}}$, for every $k \geq 1$,

$$J(\pi^*) - J(\pi_k) \leq \mathbb{E}_{\mathbf{q}\sim\mathcal{D},\,\mathbf{y}\sim\pi_k(\cdot|\mathbf{q})}\left[\sum_{t=1}^{T}\left(r_{k-1}^+(\mathbf{x}_{<t}, y_t) - r^*(\mathbf{x}_{<t}, y_t)\right)^2\right]. \tag{C.25}$$

Proof. Fix a prefix $\mathbf{x}_{<t}$ and write

$$e(y_t) := r^+_{k-1}(\mathbf{x}_{<t}, y_t) - r^*(\mathbf{x}_{<t}, y_t) \geq 0, \qquad r_\alpha := r^* + \alpha e, \quad \alpha \in [0,1], \tag{C.26}$$

where the interpolation is only over the token-reward vector at this fixed prefix. Apply the one-dimensional mean-value theorem to $h(\alpha) := \Delta(\mathbf{x}_{<t}, r_\alpha)$. For some $\bar{\alpha} \in [0,1]$, Lemma C.8 yields

$$\begin{aligned}\Delta(\mathbf{x}_{<t}, r^+_{k-1}) - \Delta(\mathbf{x}_{<t}, r^*) &= \bar{\alpha}\left(\mathbb{E}_{\pi_{r_{\bar{\alpha}}}}[e^2] - \mathbb{E}_{\pi_{r_{\bar{\alpha}}}}[e]^2\right) \\ &\leq \bar{\alpha}\,\mathbb{E}_{\pi_{r_{\bar{\alpha}}}}[e^2]. \end{aligned} \tag{C.27}$$

Here and below, the token expectations in this proof are conditional on the fixed prefix. To compare the intermediate policy with π_k, define

$$m(\alpha) := \alpha\,\mathbb{E}_{\pi_{r_\alpha}}[e^2]. \tag{C.28}$$

The exponential-family derivative identity gives

$$m'(\alpha) = \mathbb{E}_{\pi_{r_\alpha}}[e^2] + \alpha\left(\mathbb{E}_{\pi_{r_\alpha}}[e^3] - \mathbb{E}_{\pi_{r_\alpha}}[e^2]\,\mathbb{E}_{\pi_{r_\alpha}}[e]\right). \tag{C.29}$$

The covariance term is nonnegative because $e \geq 0$. Indeed, for independent copies ξ, ξ' of $e(y_t)$ under π_{r_α},

$$\mathbb{E}[\xi^3] - \mathbb{E}[\xi^2]\mathbb{E}[\xi] = \frac{1}{2}\mathbb{E}\left[(\xi - \xi')^2(\xi + \xi')\right] \geq 0. \tag{C.30}$$

Consequently $m'(\alpha) \geq 0$, and

$$\bar{\alpha}\,\mathbb{E}_{\pi_{r_{\bar{\alpha}}}}[e^2] = m(\bar{\alpha}) \leq m(1) = \mathbb{E}_{y_t \sim \pi_k(\cdot|\mathbf{x}_{<t})}[e(y_t)^2]. \tag{C.31}$$

Combining Eqs. C.27 and C.31 proves the sharp contextual-bandit inequality at this prefix. Only after obtaining this pointwise bound do we sum over t and average under the student π_k. Lemma C.7 and conditional expectation then give Eq. C.25. This avoids comparing prefix distributions under different interpolated policies. □

C.3 Least-Squares Confidence and Uniform Optimism

Lemma C.10 (Finite-class least-squares confidence)**.** Under Assumptions C.1 and 6.1, with probability at least $1 - \delta/2$, simultaneously for every $k \in \{1, \ldots, K\}$,

$$\sum_{i=1}^{k}\sum_{u=1}^{T}\left(\widehat{r}_k(\mathbf{x}_{i,<u}, y_{i,u}) - r^*(\mathbf{x}_{i,<u}, y_{i,u})\right)^2 \leq 8\sigma^2 \log\frac{2N_{\mathcal{R}}K}{\delta}. \tag{C.32}$$

Proof. For $r \in \mathcal{R}$, write $D_{i,u}(r) := r(\mathbf{x}_{i,<u}, y_{i,u}) - r^*(\mathbf{x}_{i,<u}, y_{i,u})$. By Eq. C.9, the excess squared loss at a sampled token is

$$\begin{aligned}&\left(r(\mathbf{x}_{i,<u}, y_{i,u}) - \ell_{i,u}\right)^2 - \left(r^*(\mathbf{x}_{i,<u}, y_{i,u}) - \ell_{i,u}\right)^2 \\ &\quad = D_{i,u}(r)^2 - 2D_{i,u}(r)\epsilon_{i,u}. \end{aligned} \tag{C.33}$$

For $\sigma > 0$, conditional sub-Gaussianity implies that, for every fixed r, the exponential process obtained by revealing tokens and their feedback in chronological order is a nonnegative supermartingale. At the end of rollout k, it equals

$$\exp\left(\frac{1}{2\sigma^2}\sum_{i=1}^{k}\sum_{u=1}^{T} D_{i,u}(r)\epsilon_{i,u} - \frac{1}{8\sigma^2}\sum_{i=1}^{k}\sum_{u=1}^{T} D_{i,u}(r)^2\right). \tag{C.34}$$

The conditional noise assumption is applied after the sampled token is revealed, so adaptivity of the prefix and token does not invalidate this supermartingale. Markov's inequality and a union bound over $r \in \mathcal{R}$ and the K rollout endpoints yield, with probability at least $1 - \delta/2$,

$$2\sum_{i=1}^{k}\sum_{u=1}^{T} D_{i,u}(r)\epsilon_{i,u} \leq \frac{1}{2}\sum_{i=1}^{k}\sum_{u=1}^{T} D_{i,u}(r)^2 + 4\sigma^2 \log\frac{2N_{\mathcal{R}}K}{\delta} \tag{C.35}$$

for every such r and k. Only K fitting checkpoints are union-bounded, not KT separate estimators. Since $\widehat{r}_k$ minimizes the empirical loss and $r^* \in \mathcal{R}$,

$$\sum_{i=1}^{k}\sum_{u=1}^{T} D_{i,u}(\widehat{r}_k)^2 \le 2\sum_{i=1}^{k}\sum_{u=1}^{T} D_{i,u}(\widehat{r}_k)\epsilon_{i,u}. \tag{C.36}$$

Substitute $r = \widehat{r}_k$ into the uniform event in Eq. C.35 and rearrange to obtain Eq. C.32. For $\sigma = 0$, the feedback is noiseless and the same claim follows directly from least-squares optimality. □

Lemma C.11 (Uniform optimism)**.** Under Assumptions C.1 and 6.1, with probability at least $1 - \delta/2$, simultaneously for all $k \in \{0, \dots, K\}$ and all prefix–token pairs,

$$r^* \in \mathcal{C}_k, \qquad 0 \le r_k^+(\mathbf{x}_{<t}, y_t) - r^*(\mathbf{x}_{<t}, y_t) \le 2b_k(\mathbf{x}_{<t}, y_t). \tag{C.37}$$

In particular, $\mathcal{E}_{\text{opt}}$ holds on this event.

Proof. Work on the event of Lemma C.10. For $k \ge 1$, Eqs. C.12 and C.32 imply

$$\lambda + \sum_{i=1}^{k}\sum_{u=1}^{T}\big(\widehat{r}_k(\mathbf{x}_{i,<u}, y_{i,u}) - r^*(\mathbf{x}_{i,<u}, y_{i,u})\big)^2 \le \beta^2/2 \le \beta^2. \tag{C.38}$$

Thus $r^* \in \mathcal{C}_k$. This also holds for $k = 0$, since $\mathcal{C}_0 = \mathcal{R}$.

For every $r \in \mathcal{C}_k$, both r and r^* lie in the same empirical ball centered at $\widehat{r}_k$. Using $(a+b)^2 \le 2a^2 + 2b^2$ gives

$$\lambda + \sum_{i=1}^{k}\sum_{u=1}^{T}\big(r(\mathbf{x}_{i,<u}, y_{i,u}) - r^*(\mathbf{x}_{i,<u}, y_{i,u})\big)^2 \le \lambda + 4(\beta^2 - \lambda) \le 4\beta^2. \tag{C.39}$$

The definition of U_k therefore implies $|r(\mathbf{x}_{<t}, y_t) - r^*(\mathbf{x}_{<t}, y_t)| \le 2\beta U_k(\mathbf{x}_{<t}, y_t)$. Also, all functions in $\mathcal{R}$ take values in $[-B, B]$. Taking the pointwise supremum over $\mathcal{C}_k$ yields

$$r_k^+(\mathbf{x}_{<t}, y_t) - r^*(\mathbf{x}_{<t}, y_t) \le \min\{2B,\, 2\beta U_k(\mathbf{x}_{<t}, y_t)\} \le 2b_k(\mathbf{x}_{<t}, y_t). \tag{C.40}$$

Finally, target membership gives

$$r_k^+(\mathbf{x}_{<t}, y_t) - r^*(\mathbf{x}_{<t}, y_t) = \sup_{r \in \mathcal{C}_k} r(\mathbf{x}_{<t}, y_t) - r^*(\mathbf{x}_{<t}, y_t) \ge 0. \tag{C.41}$$

Together these prove Eq. C.37, including the empty-design case. □

C.4 Proof of Theorem 6.2

Proof. For the actual rollout k, define the accumulated squared token-reward error

$$S_k := \sum_{t=1}^{T}\big(r_{k-1}^+(\mathbf{x}_{k,<t}, y_{k,t}) - r^*(\mathbf{x}_{k,<t}, y_{k,t})\big)^2, \qquad \mu_k := \mathbb{E}[S_k | \mathcal{F}_{k-1}]. \tag{C.42}$$

On the event in Lemma C.11, the reverse-KL identity and Lemma C.9 imply

$$\mathrm{Regret}(K) \le \sum_{k=1}^{K} \mu_k, \qquad S_k \le 4\sum_{t=1}^{T} b_{k-1}(\mathbf{x}_{k,<t}, y_{k,t})^2. \tag{C.43}$$

Since $\beta \ge 2B$, the clipped bonus satisfies

$$b_{k-1}(\mathbf{x}_{<t}, y_t)^2 \le \beta^2 \min\{1,\, U_{k-1}(\mathbf{x}_{<t}, y_t)^2\}. \tag{C.44}$$

For every realized sequence of rollouts, Definition C.5 gives

$$\sum_{k=1}^{K}\sum_{t=1}^{T} \min\{1,\, U_{k-1}(\mathbf{x}_{k,<t}, y_{k,t})^2\} \le d_E. \tag{C.45}$$

Consequently, $\sum_{k=1}^{K} S_k \le 4\beta^2 d_E$ on the confidence event.

From observed errors to predictable regret. A pathwise bound on $\sum_k S_k$ does not by itself bound $\sum_k \mathbb{E}[S_k|\mathcal{F}_{k-1}]$ with high probability. We supply the required concentration step while keeping the same logarithmic regret order. Define the pointwise diameter

$$\omega(\mathbf{x}_{<t}, y_t) := \sup_{r_1, r_2 \in \mathcal{R}} |r_1(\mathbf{x}_{<t}, y_t) - r_2(\mathbf{x}_{<t}, y_t)|, \qquad c_0 := \max\{\lambda, 4B^2\}. \tag{C.46}$$

The set $\mathcal{C}_{k-1}$ is nonempty and contained in $\mathcal{R}$, and $r^* \in \mathcal{R}$. Thus, even outside the confidence event, $|r^+_{k-1} - r^*| \le \omega$ pointwise. Since $U_0 = \omega/\sqrt{\lambda}$ and $\omega \le 2B$,

$$\omega(\mathbf{x}_{<t}, y_t)^2 \le c_0 \min\{1, U_0(\mathbf{x}_{<t}, y_t)^2\}. \tag{C.47}$$

Any valid response can appear as the first rollout in the supremum defining d_E. Therefore

$$0 \le S_k \le c_0 \sum_{t=1}^{T} \min\{1, U_0(\mathbf{x}_{k,<t}, y_{k,t})^2\} \le c_0 d_E =: M \tag{C.48}$$

for every k, without conditioning on optimism. If $M = 0$, all these errors and the regret are zero. Otherwise, for $z \in [0, 1]$, the convexity bound $e^{-z} \le 1 - (1 - e^{-1})z$ yields

$$\mathbb{E}\Big[e^{-S_k/M} \mid \mathcal{F}_{k-1}\Big] \le \exp\big(-(1 - e^{-1})\mu_k/M\big). \tag{C.49}$$

Hence $\exp\Big(M^{-1}[(1 - e^{-1})\sum_{k=1}^{j} \mu_k - \sum_{k=1}^{j} S_k]\Big)$ is a nonnegative supermartingale in j. Markov's inequality gives a second event, of probability at least $1 - \delta/2$, on which

$$\sum_{k=1}^{K} \mu_k \le \frac{\sum_{k=1}^{K} S_k + M \log(2/\delta)}{1 - e^{-1}} \le 2 \sum_{k=1}^{K} S_k + 2M \log(2/\delta). \tag{C.50}$$

Intersecting this event with that of Lemma C.11 gives probability at least $1-\delta$. On their intersection,

$$\mathrm{Regret}(K) \le \left[8\beta^2 + 2\max\{\lambda, 4B^2\} \log \frac{2}{\delta}\right] d_E. \tag{C.51}$$

Substituting Eq. C.12 yields

$$\mathrm{Regret}(K) = \mathcal{O}\left(\left[B^2 + \lambda + \sigma^2 \log \frac{2N_{\mathcal{R}}K}{\delta} + (B^2 + \lambda) \log \frac{2}{\delta}\right] d_E\right). \tag{C.52}$$

For fixed B, σ, λ, this is

$$\mathrm{Regret}(K) = \mathcal{O}\left(d_E \log \frac{N_{\mathcal{R}}K}{\delta}\right), \tag{C.53}$$

as claimed. If $N_{\mathcal{R}}K = 1$, the reward class is a singleton and $d_E = \mathrm{Regret}(K) = 0$; otherwise constants inside the logarithm are absorbed into $\mathcal{O}$. The entire argument estimates immediate token log-ratio rewards and sums fixed-prefix bandit inequalities, rather than propagating errors through an MDP. □

C.5 Covering-Number Interpretation

For a token-level policy class Π, define its log-policy class and induced reward class by

$$\mathcal{L}_\Pi := \{(\mathbf{x}_{<t}, y_t) \mapsto \log \pi(y_t|\mathbf{x}_{<t}) : \pi \in \Pi\}, \tag{C.54}$$

$$\mathcal{R}_\Pi := \left\{(\mathbf{x}_{<t}, y_t) \mapsto \log \frac{\pi(y_t|\mathbf{x}_{<t})}{\pi^{\mathrm{ref}}(y_t|\mathbf{x}_{<t})} : \pi \in \Pi\right\}. \tag{C.55}$$

The fixed reference term cancels in every pairwise difference:

$$\left\|\log \frac{\pi_1}{\pi^{\mathrm{ref}}} - \log \frac{\pi_2}{\pi^{\mathrm{ref}}}\right\|_\infty = \|\log \pi_1 - \log \pi_2\|_\infty, \tag{C.56}$$

where the supremum is over prefix–token pairs. Thus these two classes have identical uniform covering numbers. The same cancellation holds in the numerator and denominator of Eq. C.10, so their token-level uncertainty values and cumulative eluder complexities also coincide.

Finite-cover extension. For completeness, the discretization can be made explicit without changing the regret order. Suppose $\mathcal{R} \subseteq [-B, B]^{\mathcal{X}\times\mathcal{Y}}$ admits a finite uniform ε-net with centers in $\mathcal{R}$, and let $N_{\mathcal{R}} = \mathcal{N}_\infty(\varepsilon, \mathcal{R})$. The estimator and confidence set may still be formed over the original class, assuming the empirical minimum is attained. Define

$$L_\varepsilon := \log \frac{4N_{\mathcal{R}}K}{\delta}, \qquad a_\delta := \sigma\sqrt{2\log\frac{8KT}{\delta}}, \qquad \Gamma_\varepsilon := 4KT\varepsilon(B + a_\delta). \tag{C.57}$$

Applying the supermartingale argument to the net centers with failure probability $\delta/4$, and a conditional sub-Gaussian union bound to all observed noises with failure probability $\delta/4$, gives simultaneously

$$2\sum_{i=1}^{k}\sum_{u=1}^{T} D_{i,u}(r)\epsilon_{i,u} \le \frac{1}{2}\sum_{i=1}^{k}\sum_{u=1}^{T} D_{i,u}(r)^2 + 4\sigma^2 L_\varepsilon + 2kT\varepsilon(B + a_\delta) \tag{C.58}$$

for all $r \in \mathcal{R}$ and $k \le K$, with probability at least $1 - \delta/2$. Indeed, replacing r by a net center changes $D_{i,u}(r)$ by at most ε, its square by at most $4B\varepsilon$, and the noise cross-term by at most $2\varepsilon|\epsilon_{i,u}|$, while $|\epsilon_{i,u}| \le a_\delta$ on the noise event. Least-squares optimality then yields

$$\sum_{i=1}^{k}\sum_{u=1}^{T} D_{i,u}(\widehat{r}_k)^2 \le 8\sigma^2 L_\varepsilon + \Gamma_\varepsilon. \tag{C.59}$$

Consequently, using

$$\beta^2 := \max\left\{4B^2,\ 16\sigma^2 L_\varepsilon + 2\Gamma_\varepsilon + 2\lambda\right\} \tag{C.60}$$

in Algorithm 2 proves the same optimism and explicit regret bound as before. Choosing $0 < \varepsilon \le [KT(1 + B + a_\delta)]^{-1}$ ensures $\Gamma_\varepsilon \le 4$. Thus Theorem 6.2 retains its logarithmic form with the indicated covering number, while any additional dependence on K, T through the covering resolution and d_E remains explicit. The covering number is therefore evaluated on prefix–token pairs at the stated resolution.

C.6 Comparison with On-Policy Distillation under Noisy Expert Feedback

Building on the discussion following Assumption C.1, we compare Theorem 6.2 with the unknown-corruption guarantee of Sriraman et al. (2026). We specialize their sequential formulation to token generation: states are prefixes $\mathbf{x}_{<t} = (\mathbf{q}, \mathbf{y}_{<t})$, actions are tokens $y_t \in \mathcal{Y}$, the horizon is T, and K counts rollout rounds. Write P^π for the joint distribution of $\mathbf{q} \sim \mathcal{D}$ and $\mathbf{y} \sim \pi(\cdot|\mathbf{q})$. Their error criterion is the squared Hellinger distance, with normalization

$$D_{\mathrm{H}^2}(P^\pi, P^{\pi'}) := \mathbb{E}_{\mathbf{q}\sim\mathcal{D}}\left[1 - \sum_{\mathbf{y}} \sqrt{\pi(\mathbf{y}|\mathbf{q})\pi'(\mathbf{y}|\mathbf{q})}\right].$$

Theorem C.12 (NAILGUN with unknown corruption, Sriraman et al., 2026, Theorems 9 and 15)**.** Let Π be a finite class of deterministic token-level policies containing the clean expert π^*. Suppose the observed teacher satisfies

$$\pi^E(y_t|\mathbf{x}_{<t}) = (1 - \zeta)\pi^*(y_t|\mathbf{x}_{<t}) + \zeta\nu_t(y_t|\mathbf{x}_{<t}), \tag{C.61}$$

where $0 \le \zeta \le \alpha$, $0 < \alpha, \rho < 1$, and $\nu_t(y_t|\mathbf{x}_{<t}) \le \rho$ for every prefix–token pair. Assume $\gamma := 1 - \alpha(1 + \rho) > 0$. Given the bounds α and ρ, but without knowing ζ or ν, NAILGUN uses sampled teacher-token labels at learner-visited prefixes and returns a uniformly selected rollout policy $\widehat{\pi}$ after K rounds such that

$$\mathbb{E}\left[D_{\mathrm{H}^2}(P^{\pi^*}, P^{\widehat{\pi}})\right] \le \frac{4\log|\Pi|}{K\left(\sqrt{1-\alpha} - \sqrt{\alpha\rho}\right)^2} \le \frac{8\log|\Pi|}{K\gamma^2}. \tag{C.62}$$

The expectation includes the training randomness and the selection of $\widehat{\pi}$.

Remark C.13 (Why vanilla OPD is insufficient under corruption)**.** NAILGUN is a noise-aware algorithm, not vanilla reverse-KL distillation: directly matching the corrupted teacher need not recover the clean expert (Sriraman et al., 2026). Under Assumption C.1, our feedback is instead conditionally unbiased. Theorem 6.2 establishes a finite-sample guarantee for optimistic least-squares learning, rather than proving that vanilla OPD must fail under our assumptions.

Remark C.14 (On-policy interaction versus optimistic exploration)**.** NAILGUN updates a weighted aggregate of deterministic policies using noisy expert labels (Sriraman et al., 2026, Algorithm 3). In contrast, Algorithm 2 combines optimistic token selection with least-squares estimation of log-ratio rewards. Our analysis shows how this explicit exploration mechanism yields fast learning with general reward-function approximation; ordinary on-policy sampling alone does not provide the same guarantee.

Remark C.15 (Different noise dependence and identifiability)**.** NAILGUN's bound depends inversely on the squared corruption margin γ^2, reflecting the difficulty of distinguishing clean and corrupted labels. Our bound instead depends on σ^2 through estimation uncertainty in Eq. C.52. As discussed following Assumption C.1, this difference relies on conditionally unbiased feedback, not robustness to arbitrary corruption. Our analysis explicitly characterizes how feedback uncertainty affects distillation under this noise model.

Remark C.16 (Comparing rates, support, and rollout budgets)**.** NAILGUN bounds expected final-policy Hellinger error, whereas Theorem 6.2 gives high-probability logarithmic cumulative reverse-KL regret and a $\widetilde{\mathcal{O}}(K^{-1})$ guarantee for the response-level mixture policy. Both have a $1/K$ output-error factor up to complexity and logarithmic terms, with K counting rollouts. Our analysis directly controls the distillation objective throughout training and accommodates stochastic targets under bounded log-ratio rewards. Different support assumptions and horizon-dependent complexities make these complementary guarantees, rather than a strict improvement in sample complexity.

D GENERATED SAMPLES

In this section, we present representative samples generated on the AMC23, AIME24, and AIME25 benchmarks using our proposed LSPD and LSPD-RB methods, under the Qwen3-8B non-thinking teacher to Qwen3-4B-Base student distillation setup.

LSPD Solutions in AMC23

Question. What is the degree measure of the acute angle formed by lines with slopes 2 and $\frac{1}{3}$?

Solution.
We are given two lines with slopes $m_1 = 2$ and $m_2 = \frac{1}{3}$, and we are asked to find the **acute angle** formed between them.
Step 1: Use the formula for the angle between two lines
The formula to find the angle θ between two lines with slopes m_1 and m_2 is:

$$\tan(\theta) = \left| \frac{m_1 - m_2}{1 + m_1 m_2} \right|$$

This gives the **acute angle** between the lines.
Step 2: Plug in the values
We have:

- $m_1 = 2$
- $m_2 = \frac{1}{3}$

Compute:

$$\tan(\theta) = \left| \frac{2 - \frac{1}{3}}{1 + 2 \cdot \frac{1}{3}} \right| = \left| \frac{\frac{6}{3} - \frac{1}{3}}{1 + \frac{2}{3}} \right| = \left| \frac{\frac{5}{3}}{\frac{5}{3}} \right| = 1$$

So:

$$\tan(\theta) = 1$$

Step 3: Find the angle

$$\theta = \tan^{-1}(1) = 45^\circ$$

Final Answer:

$\boxed{45^\circ}$

LSPD Solutions in AIME24

Question. There exist real numbers x and y, both greater than 1, such that $\log_x(y^x) = \log_y(x^{4y}) = 10$. Find xy.

Solution.
We are given the equations:

$$\log_x(y^x) = 10 \quad \text{and} \quad \log_y(x^{4y}) = 10$$

We are to find xy, where both x and y are **real numbers greater than 1**.
Step 1: Use logarithmic identities
Recall the logarithmic identity:

$$\log_b(a^c) = c \cdot \log_b(a)$$

So we can rewrite the given equations using this identity.
First equation:

$$\log_x(y^x) = x \cdot \log_x(y) = 10 \quad \text{(Equation 1)}$$

Second equation:

$$\log_y(x^{4y}) = 4y \cdot \log_y(x) = 10 \quad \text{(Equation 2)}$$

Let's denote:

$$\log_x(y) = a \quad \text{and} \quad \log_y(x) = b$$

Then from Equation 1:

$$x \cdot a = 10 \Rightarrow a = \frac{10}{x}$$

From Equation 2:

$$4y \cdot b = 10 \Rightarrow b = \frac{10}{4y} = \frac{5}{2y}$$

Now, recall the identity:

$$\log_x(y) = \frac{1}{\log_y(x)} \Rightarrow a = \frac{1}{b}$$

Substitute the expressions for a and b:

$$\frac{10}{x} = \frac{1}{\frac{5}{2y}} = \frac{2y}{5}$$

Step 2: Solve the equation

$$\frac{10}{x} = \frac{2y}{5}$$

Multiply both sides by $5x$:

$$50 = 2xy \Rightarrow xy = \boxed{25}$$

Final Answer:

$$\boxed{25}$$

LSPD Solutions in AIME25

Question. Find the sum of all integer bases $b > 9$ for which 17_b is a divisor of 97_b.

Solution.
We are given the problem:
Find the sum of all integer bases $b > 9$ for which 17_b is a divisor of 97_b.
Step 1: Convert the numbers from base b to base 10
We are given:

- 17_b is the number $1 \cdot b + 7 = b + 7$
- 97_b is the number $9 \cdot b + 7 = 9b + 7$

So the problem becomes:
Find all integers $b > 9$ such that $b + 7$ divides $9b + 7$, and then sum those values of b.
Step 2: Set up the divisibility condition
We want:

$$b + 7 | 9b + 7$$

This means:

$$9b + 7 \equiv 0 \pmod{b + 7}$$

Let's perform the division:

$$9b + 7 = 9(b + 7) - 63 + 7 = 9(b + 7) - 56$$

So:

$$9b + 7 \equiv -56 \pmod{b + 7}$$

Therefore:

$$b + 7 | 56$$

Step 3: Find all divisors of 56
We need to find all integers d such that:

$$b + 7 = d \Rightarrow b = d - 7$$

So, we list the **positive divisors** of 56:

$$\text{Divisors of } 56 : \{1, 2, 4, 7, 8, 14, 28, 56\}$$

Now, we want $b > 9$, so $d - 7 > 9 \Rightarrow d > 16$.
Check which of these divisors are greater than 16:

- 1: too small
- 2: too small
- 4: too small
- 7: too small
- 8: too small
- 14: $b = 14 - 7 = 7 \rightarrow$ too small
- 28: $b = 28 - 7 = 21 \rightarrow$ OK
- 56: $b = 56 - 7 = 49 \rightarrow$ OK

So the valid values of b are:

$$b = 21, 49$$

Step 4: Compute the sum

$$21 + 49 = \boxed{70}$$

Final Answer:

$$\boxed{70}$$

LSPD-RB Solutions in AMC23

Question. What is the degree measure of the acute angle formed by lines with slopes 2 and $\frac{1}{3}$?

Solution.
We are given two lines with slopes $m_1 = 2$ and $m_2 = \frac{1}{3}$, and we are asked to find the **acute angle** formed between them.
Step 1: Use the formula for the angle between two lines
The formula for the angle θ between two lines with slopes m_1 and m_2 is:

$$\tan\theta = \left|\frac{m_2 - m_1}{1 + m_1 m_2}\right|$$

We are interested in the **acute angle**, so we take the **absolute value** of the right-hand side.
Step 2: Plug in the values

$$\tan\theta = \left|\frac{\frac{1}{3} - 2}{1 + 2 \cdot \frac{1}{3}}\right|$$

Simplify the numerator and denominator:

- Numerator: $\frac{1}{3} - 2 = \frac{1}{3} - \frac{6}{3} = -\frac{5}{3}$
- Denominator: $1 + \frac{2}{3} = \frac{5}{3}$

So,

$$\tan\theta = \left|\frac{-\frac{5}{3}}{\frac{5}{3}}\right| = |-1| = 1$$

Step 3: Find the angle

$$\tan\theta = 1 \Rightarrow \theta = \tan^{-1}(1) = 45^\circ$$

Since the angle is acute (less than 90°), the answer is:

$$\boxed{45^\circ}$$

Final Answer:

$$\boxed{45^\circ}$$

LSPD-RB Solutions in AIME24

Question. There exist real numbers x and y, both greater than 1, such that $\log_x\left(y^x\right) = \log_y\left(x^{4y}\right) = 10$. Find xy.

Solution.
We are given the following two equations involving logarithms:

$$\log_x(y^x) = 10 \quad \text{and} \quad \log_y(x^{4y}) = 10$$

We are to find the value of xy, given that $x, y > 1$.
Step 1: Simplify the first equation

$$\log_x(y^x) = 10$$

Using the logarithm power rule:

$$\log_x(y^x) = x \cdot \log_x(y) = 10$$

So,

$$x \cdot \log_x(y) = 10 \tag{1}$$

Step 2: Simplify the second equation

$$\log_y(x^{4y}) = 10$$

Again using the logarithm power rule:

$$\log_y(x^{4y}) = 4y \cdot \log_y(x) = 10$$

So,

$$4y \cdot \log_y(x) = 10 \tag{2}$$

Step 3: Express $\log_x(y)$ in terms of $\log_y(x)$
We recall the change of base formula:

$$\log_x(y) = \frac{1}{\log_y(x)}$$

Let's denote:

$$\log_y(x) = a \quad \Rightarrow \quad \log_x(y) = \frac{1}{a}$$

Now substitute into equations (1) and (2):

- From (1): $x \cdot \frac{1}{a} = 10 \Rightarrow x = 10a$
- From (2): $4y \cdot a = 10 \Rightarrow y = \frac{10}{4a} = \frac{5}{2a}$

Step 4: Substitute into xy
We now have expressions for x and y in terms of a:

$$x = 10a, \quad y = \frac{5}{2a}$$

So,

$$xy = (10a) \cdot \left(\frac{5}{2a}\right) = \frac{50a}{2a} = 25$$

Step 5: Verify that $x, y > 1$
We have:

- $x = 10a$
- $y = \frac{5}{2a}$

We want $x, y > 1$, so:

- $10a > 1 \Rightarrow a > \frac{1}{10}$
- $\frac{5}{2a} > 1 \Rightarrow a < \frac{5}{2}$

So, $a \in \left(\frac{1}{10}, \frac{5}{2}\right)$, which is valid.
Final Answer:

$$\boxed{25}$$

LSPD-RB Solutions in AIME25

Question. Find the sum of all integer bases $b > 9$ for which 17_b is a divisor of 97_b.

Solution.
We are given that 17_b divides 97_b, and we are to find the **sum of all integer bases** $b > 9$ for which this is true.
Step 1: Convert both numbers from base b to base 10

We convert 17_b and 97_b to base 10:

- $17_b = 1 \cdot b + 7 = b + 7$
- $97_b = 9 \cdot b + 7 = 9b + 7$

So, the condition becomes:

$$b + 7 | 9b + 7$$

Step 2: Use divisibility condition

We are told that $b + 7$ divides $9b + 7$, so:

$$b + 7 | 9b + 7$$

To simplify this, we use the fact that if $d|a$ and $d|b$, then $d|(a - kb)$ for any integer k.

Let's subtract $9 \cdot (b + 7)$ from $9b + 7$:

$$(9b + 7) - 9(b + 7) = 9b + 7 - 9b - 63 = -56$$

So:

$$b + 7 | -56$$

Since $b + 7$ divides -56, it also divides 56. So we are looking for all **positive integers** $b > 9$ such that:

$$b + 7 | 56$$

Step 3: Find all divisors of 56

List all positive divisors of 56:

$$\text{Divisors of 56: } 1, 2, 4, 7, 8, 14, 28, 56$$

Now, we want $b + 7$ to be one of these, and $b > 9$, so:

$$b = d - 7 \quad \text{and} \quad b > 9 \Rightarrow d - 7 > 9 \Rightarrow d > 16$$

So we filter the divisors of 56 that are greater than 16:

$$\text{Divisors of 56 > 16: } 28, 56$$

So possible values of $b + 7$ are 28 and 56, and thus:

- $b = 28 - 7 = 21$
- $b = 56 - 7 = 49$

Both values are greater than 9, so they are valid.

Step 4: Final Answer

We now sum all such b:

$$21 + 49 = \boxed{70}$$

Bonus: Check that these work

- For $b = 21$:
  - $17_{21} = 21 + 7 = 28$
  - $97_{21} = 9 \cdot 21 + 7 = 189 + 7 = 196$
  - $28 | 196 \rightarrow$ True (196 ÷ 28 = 7)
- For $b = 49$:
  - $17_{49} = 49 + 7 = 56$
  - $97_{49} = 9 \cdot 49 + 7 = 441 + 7 = 448$
  - $56 | 448 \rightarrow$ True (448 ÷ 56 = 8)

Both values satisfy the condition.

Final Answer:

$$\boxed{70}$$