Learning While Staying Curious: Entropy-Preserving Supervised Fine-Tuning via Adaptive Self-Distillation for Large Reasoning Models

Hao Wang[1,*] **Hao Gu**[2,*] **Hongming Piao**[1] **Kaixiong Gong**[3]
Yuxiao Ye[2] **Xiangyu Yue**[3] **Sirui Han**[2,†] **Yike Guo**[2,†] **Dapeng Wu**[1,†]
[1]City University of Hong Kong [2]The Hong Kong University of Science and Technology
[3]The Chinese University of Hong Kong
hao.wang@my.cityu.edu.hk siruihan@ust.hk yikeguo@ust.hk dapengwu@cityu.edu.hk

Abstract

The standard post-training recipe for large reasoning models, supervised fine-tuning followed by reinforcement learning (SFT-then-RL), may limit the benefits of the RL stage: while SFT imitates expert demonstrations, it often causes overconfidence and reduces generation diversity, leaving RL with a narrowed solution space to explore. Adding entropy regularization during SFT is not a cure-all; it tends to flatten token distributions toward uniformity, increasing entropy without improving meaningful exploration capability. In this paper, we propose **CurioSFT**, an entropy-preserving SFT method designed to enhance exploration capabilities through intrinsic curiosity. It consists of (a) *Self-Exploratory Distillation*, which distills the model toward a self-generated, temperature-scaled teacher to encourage exploration within its capability; and (b) *Entropy-Guided Temperature Selection*, which adaptively adjusts distillation strength to mitigate knowledge forgetting by amplifying exploration at reasoning tokens while stabilizing factual tokens. Extensive experiments on mathematical reasoning tasks demonstrate that, *in SFT stage*, CurioSFT outperforms the vanilla SFT by **2.5 points** on in-distribution tasks and **2.9 points** on out-of-distribution tasks. We also verify that exploration capabilities preserved during SFT successfully translate into concrete gains *in RL stage*, yielding an average improvement of **5.0 points**. Code is available at https://github.com/HaoooWang/CurioSFT.

1 Introduction

Recent breakthroughs (OpenAI et al., 2025; Guo et al., 2025) establish "SFT-then-RL" as the de-facto paradigm for enhancing large reasoning models on automatically verifiable tasks, such as mathematical reasoning (Shao et al., 2025b; Team et al., 2025; Zhang et al., 2025c), code generation (Liu et al., 2025b), and agentic search (Shao et al., 2025a; Jin et al., 2025; Wang et al., 2026b). In this paradigm, the Supervised Fine-tuning (SFT) stage aligns the model with domain-specific patterns and required knowledge, thereby providing a superior initialization for the subsequent Reinforcement Learning (RL) stage.

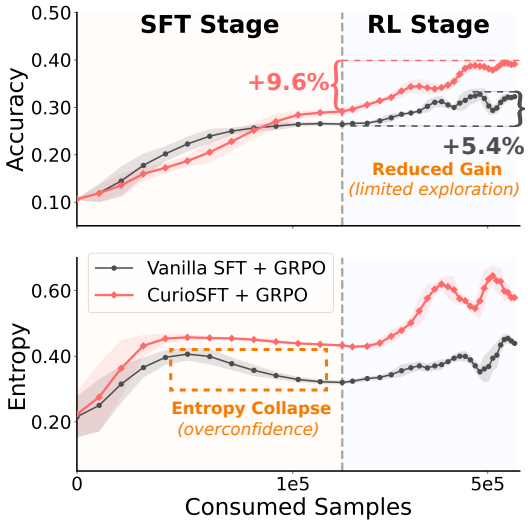


Figure 1: Evaluation entropy and accuracy (Avg@8 in AIME 2024) across the SFT and RL stages. CurioSFT mitigates entropy collapse during SFT, and yields larger accuracy gains in the RL stage.

However, this paradigm faces a critical challenge: the Cross Entropy loss in SFT rigidly maximizes the likelihood of expert tokens, which inevitably drives the model toward overconfidence (Desai and Durrett, 2020; Jiang et al., 2021; Wang et al., 2025b) and constricts the exploration space as training progresses. As shown in Figure 1, we use token entropy to quantify exploration capability and observe a rapid collapse over the SFT stage. Counter-intuitively, the SFT stage locks the model into a low-diversity mode, severely con-

*Equal contribution †Corresponding author

This paper was supported in part by the Hong Kong Research Grants Council (grant C1042-23GF) and the Hong Kong Innovation and Technology Commission (grant MHP/061/23).

straining the search space for the subsequent RL stage. This limitation often leads to marginal gains or even degradation compared to direct RL, aligning with recent findings (Zhang et al., 2025a,b; Tan et al., 2025).

A straightforward approach is to regularize the SFT stage with entropy loss (Jost, 2006) on each token. However, trivially maximizing entropy will indiscriminately smooth the token probability and introduce *ungrounded entropy*, damaging exploration capability and leading to unsatisfactory or degraded performance. Concretely, it fails to distinguish token roles: forcing entropy on factual tokens disrupts knowledge retention, while neglecting critical reasoning tokens (e.g., "wait") where exploration is truly beneficial. This discrepancy highlights the need for a method that *substantially enhances exploration capabilities without compromising the model's intrinsic knowledge*.

To achieve this, we introduce **CurioSFT**, a novel entropy-preserving SFT method designed to enhance exploration with knowledge retention. This method consists of two key components: *Self-Exploratory Distillation* and *Entropy-Guided Temperature Selection*. Building on self-distillation (Allen-Zhu and Li, 2020; Pham et al., 2022; Li et al., 2023b), *Self-Exploratory Distillation* exploits the monotonic relationship between token entropy and sampling temperature to construct a higher-entropy "teacher distribution" via an increased temperature. Aligning with this high-entropy teacher allows the model to selectively expand its search space under the guidance of its own curiosity. Crucially, to account for the distinct roles of tokens during reasoning, *Entropy-Guided Temperature Selection* dynamically modulates the temperature based on token-level uncertainty. This mechanism selectively encourages exploration at critical reasoning tokens while maintaining deterministic targets for factual tokens, thereby effectively mitigating the risk of knowledge forgetting.

Extensive experiments on mathematical reasoning benchmarks demonstrate that, CurioSFT not only effectively preserves entropy but also achieves superior performance across both in-distribution and out-of-distribution (OOD) tasks, outperforming vanilla SFT by an average of **2.5 points** and **2.9 points**, respectively. We empirically verify that the exploration capabilities preserved during SFT successfully translate into concrete gains in the RL stage. To this end, our contributions are three-fold:

- We empirically analyze the drawbacks of entropy loss in SFT, including *exploration degradation* and *knowledge forgetting*. To address these, we propose CurioSFT, which preserves entropy while improving overall performance during the SFT stage.
- We propose *Self-Exploratory Distillation* to preserve entropy while improving effective exploration by aligning with a self-generated, temperature-scaled teacher. We further introduce *Entropy-Guided Temperature Selection* to adapt token-level temperatures, selectively encouraging exploration and mitigating knowledge forgetting.
- Extensive experiments on mathematical reasoning benchmarks demonstrate that, CurioSFT not only improves performance in SFT but also enhances the exploration capability, significantly improving the performance of RL stage. We also verify the robustness of CurioSFT across models and hyperparameters.

2 Preliminaries

SFT Loss. Let $\mathcal{D}$ denote the SFT dataset, which contains multiple questions $\mathbf{x}$ and corresponding expert responses $\mathbf{y}$. The optimization objective during the SFT stage is to minimize the cross-entropy loss between the model distribution and a one-hot target distribution induced by expert tokens, as:

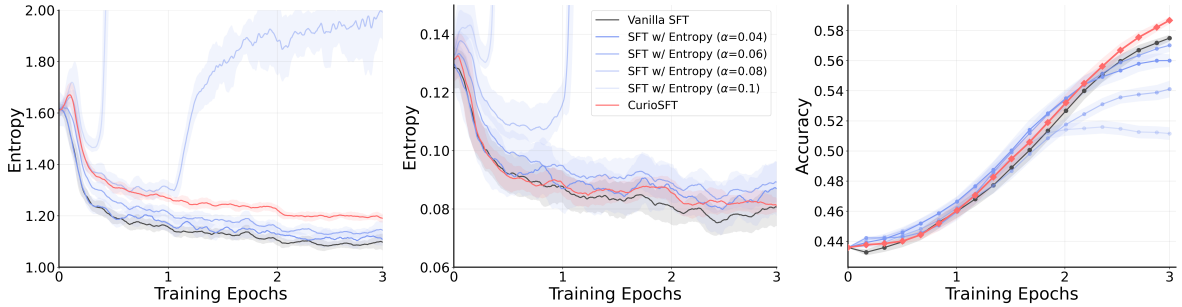
$$\mathcal{L}_{\text{SFT}}(\theta) = -\log \pi_\theta\big(y_t \mid \mathbf{s}_t\big), \tag{1}$$

where π_θ is the fine-tuned model and $\mathbf{s}_t$ is the concatenation of the question $\mathbf{x}$ and the previously generated tokens $\mathbf{y}_{<t}$.

Entropy Loss. To encourage output diversity and prevent over-confidence, prior works (Shao et al., 2025b; Hu et al., 2025) introduce an entropy loss term as a regularizer during the RL stage, as:

$$\begin{aligned}\mathcal{L}_{\text{entropy}}(\theta) &= \alpha \cdot -H(\pi_\theta(\cdot \mid \mathbf{s}_t)) \\ &= \alpha \cdot \sum_{y \in \mathcal{V}} \pi_\theta(y \mid \mathbf{s}_t) \log \pi_\theta(y \mid \mathbf{s}_t),\end{aligned} \tag{2}$$

where $\mathcal{V}$ denotes the vocabulary of the fine-tuned model, α is the loss weight. However, applying entropy regularization solely at the RL stage often yields marginal benefits, as the preceding SFT stage has already driven the model into a low-entropy mode. Consequently, it is important to preserve



(a) Expert Token Entropy (Top 20%) (b) Expert Token Entropy (Bottom 80%) (c) Pass@32 for AIME 2024

Figure 2: **Expert token entropy and evaluation accuracy during the SFT stage.** Compared to vanilla entropy loss, which uniformly encourages entropy across tokens, CurioSFT selectively increases entropy on high-entropy tokens while preserving low-entropy ones. We further observe that the increased token entropy induced by entropy loss does not translate into actual improvements in Pass@32 performance in our experiments.

Table 1: OOD performance comparison.

Method	GPQA	MMLU-Pro	ARC-C	Avg.↑
Vanilla SFT	27.7	47.5	78.8	51.3
SFT w/ Entropy	28.0	45.9	77.2	50.4
SFT w/ Entropy (Top 20%)	**29.6**	**47.9**	**79.3**	**52.3**

entropy and encourage exploration during the SFT stage itself. Yet, deploying entropy loss in the SFT stage presents a fundamental challenge: *unlike the online nature of RL, SFT is an offline process where the model cannot judge whether the increased entropy leads to valid reasoning paths or merely introduces noise*. In the following section, we discuss two key limitations arising from this "blind" regularization through empirical observations.

3 The Pitfall of Entropy Loss in SFT

Ungrounded entropy degrades exploration capability. In the SFT stage, many tokens in the dataset are relatively *unfamiliar* to the current model, reflected by their lower output probability compared to online sampling tokens (offline 71% vs. online 76%). As a result, the entropy loss becomes highly sensitive to its weight α: as shown in Figure 2a and Figure 2b, when α is too small, entropy barely increases; when α is too large (e.g., $\alpha \geq 0.08$ in our setting), the objective can push some token distributions toward near-uniformity, causing an "entropy explosion" that destabilizes training. Even with a seemingly reasonable choice (e.g., $\alpha = 0.06$), entropy loss does not reliably improve performance (Figure 2c). The key reason is that the entropy loss indiscriminately pushes the token distribution toward higher entropy, without distinguishing between expanding a valid reasoning path and merely injecting noise. Consequently, increased token entropy does not translate into better reasoning performance and may even harm effective exploration.

Token-agnostic regularization amplifies knowledge forgetting. Recent works suggest that exploration in LLMs is driven by a relatively small subset of high-entropy tokens, while most tokens remain low-entropy to preserve knowledge (Wang et al., 2025a). As shown in the Figure 2a and Figure 2b, when we partition tokens by entropy (e.g., top 20% vs. the remaining 80%), naive entropy loss increases entropy in *both* groups. This is because maximizing entropy is equivalent to minimizing the KL divergence to a uniform distribution for *all* tokens (detailed proof in Appendix A). Such token-agnostic regularization is detrimental to the model's original knowledge and reasoning behavior. As shown in Table 1, restricting entropy loss to the top 20% high-entropy tokens yields significantly better performance on knowledge-intensive OOD tasks. Empirically, low-entropy tokens often correspond to deterministic factual content (e.g., nouns and numbers), where stability is crucial; forcing entropy at these positions weakens factual consistency and can induce knowledge forgetting. In contrast, high-entropy tokens tend to act as reasoning connectors (e.g., "wait", "alternatively"), which are the natural targets for exploration.

4 Proposed Solution: CurioSFT

To address the limitations of vanilla entropy loss, we introduce CurioSFT, an entropy-preserving SFT method that enables models to learn expert behaviors while maintaining exploration capability. As shown in Figure 3, our method consists of two key components: **Self-Exploratory Distillation**

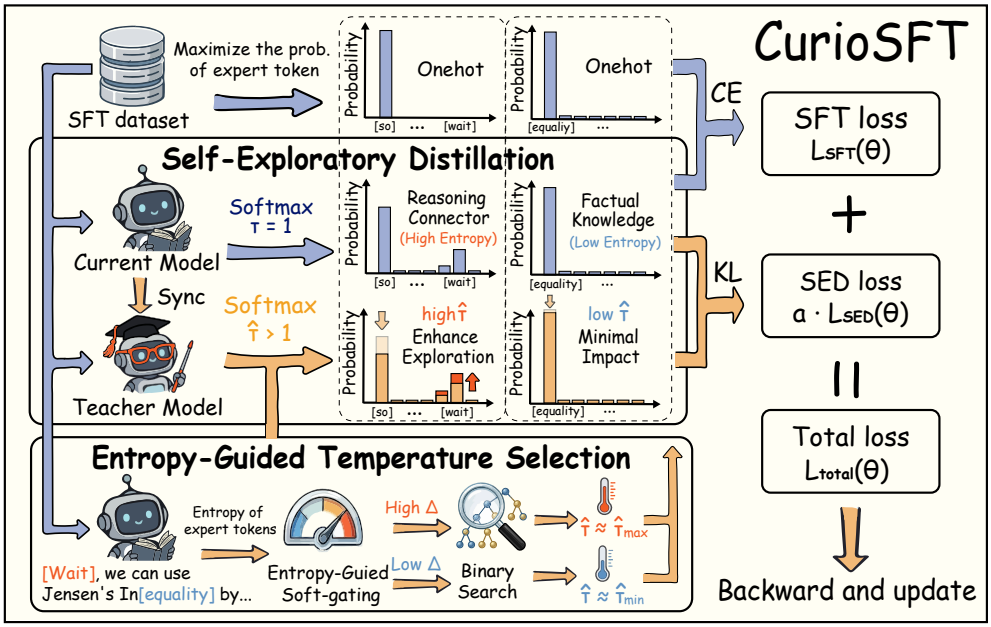


Figure 3: Proposed solution: CurioSFT

(Section 4.1) and **Entropy-Guided Temperature Selection** (Section 4.2).

4.1 Self-Exploratory Distillation

Frontier LLMs internalize extensive world knowledge during pre-training, and exhibit implicit exploration capabilities that enable the generation of diverse trajectories (Rafailov et al., 2024; Cui et al., 2025). Motivated by self-distillation (Kim et al., 2021), we exploit this intrinsic capacity by minimizing the divergence between the policy of current model and a self-generated, higher-entropy teacher distribution. Formally, an LLM policy is obtained by applying the softmax function to the output logits $z_\theta(\cdot \mid \mathbf{s}_t)$, as:

$$\pi_\theta(y \mid \mathbf{s}_t; \tau) = \frac{\exp\left(z_\theta(y \mid \mathbf{s}_t)/\tau\right)}{\sum_{y' \in \mathcal{V}} \exp\left(z_\theta(y' \mid \mathbf{s}_t)/\tau\right)}, \quad (3)$$

where τ denotes the sampling temperature and is typically fixed at 1.0 in standard SFT training. Leveraging the property that the entropy is monotonically increasing with respect to τ (see proof in Appendix B), we can construct a higher-entropy teacher distribution π^{tch} by simply rescaling the logits with a larger temperature $\hat{\tau} > \tau$, which satisfies $H\left(\pi^{\text{tch}}(\cdot \mid \mathbf{s}_t; \hat{\tau})\right) > H\left(\pi_\theta(\cdot \mid \mathbf{s}_t; \tau)\right)$. Subsequently, we can introduce a regularization loss that aligns the model with this higher-entropy teacher to achieve entropy preservation.

We theoretically prove that the constructed teacher distribution is the unique higher-entropy distribution that minimizes the KL divergence to the current policy under the entropy-increase constraint (see proof in Appendix C). Adopting this teacher offers two key advantages: *(a) Curiosity-driven exploration.* The temperature-scaled teacher strictly *preserves the relative order* of token probabilities. Distilling the student toward this teacher therefore encourages exploration only over tokens that lie within the model's *valid exploration space*, rather than injecting uninformative entropy to all tokens. *(b) Reduced Knowledge forgetting.* Prior works suggest that training data with lower divergence from the current model is associated with less knowledge forgetting (Shenfeld et al., 2025). By distilling toward a higher-entropy teacher that remains close to the current policy, the model can enhance its exploration ability while mitigating knowledge forgetting.

To ensure the stability of the teacher model, we deploy a separate teacher model parameterized by ϕ, and the teacher distribution is denoted as:

$$\pi^{\text{tch}}_\phi(y \mid \mathbf{s}_t; \hat{\tau}) = \frac{\exp(z^{\text{tch}}_\phi(y \mid \mathbf{s}_t)/\hat{\tau})}{\sum_{y' \in \mathcal{V}} \exp(z^{\text{tch}}_\phi(y' \mid \mathbf{s}_t)/\hat{\tau})}, \quad (4)$$

where $\hat{\tau}$ is the teacher sampling temperature. Using a separate teacher updated more slowly than the student stabilizes the distillation target, preventing rapid fluctuations in the teacher distribution during training. The parameters of the teacher model ϕ are synchronized with the current policy θ every n steps using an exponential moving average with a

Algorithm 1 Training with CurioSFT

Require: SFT dataset $\mathcal{D}$, base model π_θ, teacher model π_ϕ^{tch}

1: Initialize teacher parameters: $\phi \leftarrow \theta$
2: **for** $step = 1, 2, \ldots$ **do**
3: Sample training data $(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}$
4: Compute model logits $z_\theta(\cdot \mid \mathbf{s}_t)$
5: // Entropy-guided temperature selection
6: Compute teacher logits $z_\phi^{\text{tch}}(\cdot \mid \mathbf{s}_t)$
7: Compute entropy H_t for each token t
8: Compute entropy increment Δ_t by Eq. (7)
9: $\hat{\tau}_t \leftarrow \textsc{BinarySearch}(z_\phi^{\text{tch}}(\cdot \mid \mathbf{s}_t), \Delta_t)$
10: // Self-exploratory distillation
11: $\pi_\phi^{\text{tch}}(\cdot \mid \mathbf{s}_t) \leftarrow \text{Softmax}\big(z_\phi^{\text{tch}}(\cdot \mid \mathbf{s}_t)/\hat{\tau}_t\big)$
12: Compute $\mathcal{L}_{\text{SED}}(\theta)$ by Eq. (5)
13: Compute $\mathcal{L}_{\text{total}}(\theta)$ by Eq. (6)
14: Update $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}_{\text{total}}$
15: // Teacher update
16: **if** $step \pmod n = 0$ **then**
17: $\phi \leftarrow (1-\mu)\phi + \mu\theta$
18: **end if**
19: **end for**

decay factor of μ (Algo. 1 Line 17). Finally, we formulate the self-exploratory distillation objective using the K2-loss (Liu et al., 2025a), defined as:

$$\mathcal{L}_{\text{SED}}(\theta) = \frac{1}{2} \sum_{t=1}^{T} \left(\log \frac{\pi_\theta(y \mid \mathbf{s}_t; \tau)}{\pi_\phi^{\text{tch}}(y \mid \mathbf{s}_t; \hat{\tau})} \right)^2. \quad (5)$$

Finally, the overall optimization objective is defined as a combination of the SFT loss and the self-distillation loss, with a coefficient α:

$$\mathcal{L}_{\text{total}}(\theta) = \mathcal{L}_{\text{SFT}}(\theta) + \alpha \cdot \mathcal{L}_{\text{SED}}(\theta). \quad (6)$$

4.2 Entropy-Guided Temperature Selection

The sampling temperature $\hat{\tau}$ for the teacher distribution is a crucial parameter in CurioSFT. A higher value encourages the model to align with a higher-entropy teacher distribution, whereas a lower value keeps the update closer to standard SFT. As discussed in Section 3, *high-entropy tokens* typically act as branching points that benefit from exploration, while *low-entropy tokens* encode deterministic facts, where stability is preferred. To respect such heterogeneity, we adaptively assign temperatures based on the token uncertainty. We first compute a token-level entropy increment Δ_t, then determine the temperature required via binary search. Finally, we use these specialized temperatures to construct the teacher distribution in Eq. (5).

Specifically, given the current token entropy $H_t = H(\pi_\phi^{\text{tch}}(\cdot \mid \mathbf{s}_t))$, we compute the entropy increment Δ_t via:

$$\Delta_t = \Delta_{\max} \cdot \text{Sigmoid}\big(\gamma \cdot (H_t - H_{\text{pivot}})\big), \quad (7)$$

where $\Delta_{\max}$ is the maximum allowable entropy increase, γ is a scaling factor, and H_{pivot} is the *entropy pivot* that decides the activation threshold of exploration: increasing H_{pivot} makes exploration more selective, while decreasing it expands the set of tokens receiving substantial entropy increase. We adopt soft-gating rather than a hard mask to avoid brittle thresholding and introduce a smooth, adaptive margin: for *high-entropy tokens* ($H_t \gg H_{\text{pivot}}$), the sigmoid term approaches 1, pushing the target entropy toward $H_t + \Delta_{\max}$ and thus strongly encouraging diversity at those positions. Conversely, for *low-entropy tokens* ($H_t \ll H_{\text{pivot}}$), the sigmoid term approaches 0, keeping the target entropy close to H_t and thereby minimizing interference with the model's established knowledge.

Given the entropy increment Δ_t, our goal is to find a temperature for teacher distribution that matches the desired entropy target, as:

$$\min_{\hat{\tau}_t} \left| H\big(\pi_\phi^{\text{tch}}(\cdot \mid \mathbf{s}_t; \hat{\tau}_t)\big) - \big(H_t + \Delta_t\big) \right| < \epsilon,$$

where ϵ is a small constant. Given that entropy is monotonically increasing with respect to temperature τ (see proof in Appendix B), we can efficiently solve for $\hat{\tau}_t \in [\hat{\tau}_{\min}, \hat{\tau}_{\max}]$ using a binary search. To minimize computational overhead during training, we implement the temperature search as a fully vectorized operation and approximate the entropy using only the top-k logits, which focuses computation on the most influential tokens while significantly accelerating the calculation. Details of binary search are provided in Appendix D.

5 Experiments

Training. We use OpenR1-Math-46K (Yan et al., 2025) as the training dataset for both SFT and RL stages, which contains 46K mathematics problems and corresponding answers generated by DeepSeek-R1 (Guo et al., 2025). We adopt Qwen2.5-Math-7B (Yang et al., 2024) as the base model, except in Section 5.3, where we study robustness across different models. For the SFT stage, we train for 3 epochs, and for the RL stage, we train for 500 steps using GRPO (Shao et al., 2024). We set the entropy pivot $H_{\text{pivot}} = 1.2$ nats, the scaling

Table 2: **Performance comparison in the SFT stage.**

Method	In-Distribution Tasks						Out-of-Distribution Tasks				Other	
	AIME25/24	AMC23	MATH.	Miner.	Olymp.	Avg.↑	GPQA	MMLU.	ARC-C	Avg.↑	Entropy↑	Speed↓
Base Model												
Qwen2.5-Math-7B	4.6/8.3	35.5	50.1	12.1	16.5	21.2	26.2	32.4	63.2	40.6	0.15	–
Vanilla SFT with Regularization												
Vanilla SFT	22.9/26.7	59.6	85.8	45.5	50.4	48.5	27.7	47.5	78.8	51.3	0.31	**43.2**
SFT with Entropy	23.3/25.4	60.3	86.1	44.2	49.2	48.1	28.0	45.9	77.2	50.4	0.36	44.6
SFT with KL	21.6/24.6	58.2	83.9	45.2	46.3	46.6	27.4	47.8	78.0	51.1	0.30	50.2
SFT Variants												
GEM (Li et al., 2024)	24.6/26.7	60.2	85.7	47.7	50.9	49.3	30.6	49.0	**81.4**	53.7	**0.66**	44.5
DFT (Wu et al., 2025)	23.3/25.0	59.3	86.6	46.4	49.9	48.4	31.1	48.8	79.0	53.0	0.29	43.7
PSFT (Zhu et al., 2025)	25.0/28.8	**60.4**	86.9	48.3	52.6	50.3	29.9	47.8	76.7	51.5	0.32	45.2
Our Method												
CurioSFT	**26.3/29.6**	59.9	**87.0**	**49.8**	**53.2**	**51.0**	**31.7**	**49.5**	81.3	**54.2**	0.43	53.9
Impr. vs SFT	**+3.4/+2.9**	**+0.3**	**+1.2**	**+4.3**	**+2.8**	**+2.5**	**+4.0**	**+2.0**	**+2.5**	**+2.9**	**+0.12**	+10.7
w/o Adaptive Temp.	24.6/26.7	59.0	86.4	48.8	53.1	49.8	29.5	47.3	79.8	52.2	0.45	51.9
w/o Separate Teacher	25.0/28.8	58.2	85.4	49.3	52.8	49.9	30.8	48.2	80.2	53.0	0.39	45.5

factor $\gamma = 2.0$, the maximum entropy increment $\Delta_{\max} = 0.5$ nats, and the loss weight $\alpha = 1$. The temperature clip range is $\hat{\tau}_{\min} = 1.1, \hat{\tau}_{\max} = 1.5$. Further training details and hyperparameters are provided in Appendix E.

Evaluation. To evaluate the model's performance, we utilize six challenging and widely used mathematical reasoning benchmarks, including: AIME 2024, AIME 2025, AMC (LI et al., 2024), Math-500 (Hendrycks et al., 2021), Olympiad Bench (He et al., 2024), and Minerva (Lewkowycz et al., 2022). Furthermore, to assess the extent of knowledge retention and generalization ability, we evaluate the model on three OOD benchmarks: ARC-Challenge (Clark et al., 2018), GPQA-Diamond (Rein et al., 2024), and MMLU-Pro (Wang et al., 2024). We set the sampling temperature to 0.6 and Top_p $= 0.95$, and keep other settings consistent with the training. Due to the large number of questions in MMLU-Pro, we generate a single response per question for MMLU-Pro, while using 8 responses per question for all other benchmarks, and compute the average accuracy as the final reported metric.

5.1 Effectiveness of CurioSFT

Baselines. We compare CurioSFT against two categories of baselines: *(a) Vanilla SFT with Regularization*, which includes adding entropy loss and KL divergence constraints relative to the base model. *(b) Advanced SFT Variants,* which are designed to mitigate over-confidence and encourage diversity. Specifically, PSFT (Zhu et al., 2025) employs trust-region constraints to limit policy shift; DFT (Wu et al., 2025) re-weights token updates based on model internal knowledge; and GEM (Li et al., 2024) maintains diversity by encouraging the model to diverge from over-confident distributions.

Results. Table 2 shows that CurioSFT achieves the best overall performance on both in-distribution and OOD benchmarks. Compared to vanilla SFT, CurioSFT improves the ID average from 48.5% to 51.0% (**+2.5** points) and the OOD average from 51.3% to 54.2% (**+2.9** points), while preserving higher token entropy (0.31 → 0.43, **+0.12** nats). Enforcing a KL constraint slows the training while offering little benefit for entropy preservation. Among SFT variants, GEM achieves the highest entropy by keeping away from overconfident distributions. However, this occurs at the cost of ungrounded entropy, leading it to underperform CurioSFT on in-distribution tasks. PSFT and DFT improve training stability, but since they do not explicitly target exploration preservation, their overall improvements remain limited. As shown in Figure 4, we also report N-gram diversity (1 minus N-gram similarity) and confirm that entropy preservation consistently increases generation diversity. Overall, these results indicate that CurioSFT successfully preserves effective entropy while mitigating knowledge forgetting during the SFT stage.

Ablation Study. We ablate key components of CurioSFT to quantify their contributions. Removing the Entropy-Guided Temperature Selection module (Section 4.2) and using a fixed tempera-

Table 3: **Performance comparison in the RL stage.**

Model	In-Distribution Benchmarks							Out-of-Distribution Benchmarks			
	AIME25	AIME24	AMC23	MATH.	Miner.	Olymp.	Avg.	GPQA	MMLU.	ARC-C	Avg.
Vanilla GRPO											
Vanilla GRPO	18.8	20.8	62.8	84.7	46.7	50.0	47.3	40.3	50.1	84.1	58.2
Hybrid SFT with RL											
LUFFY (Yan et al., 2025)	29.4	23.1	65.6	87.6	49.5	57.2	52.1	39.9	53.0	80.5	57.8
Prefix-RFT (Huang et al., 2025)	26.4	31.8	68.2	88.4	50.9	55.7	53.6	39.1	52.1	84.0	58.4
RL-PLUS (Dong et al., 2025)	25.9	33.4	68.1	90.2	52.3	58.8	54.8	40.4	54.7	82.3	59.1
SFT-then-RL Paradigm											
SFT + RL	24.6	32.1	68.1	88.2	51.9	57.1	53.7	41.1	53.3	83.5	59.3
GEM (Li et al., 2024) + RL	27.5	34.6	71.1	90.8	52.1	61.3	56.2	40.3	54.1	83.7	59.4
DFT (Wu et al., 2025) + RL	24.2	31.3	69.8	91.3	50.5	59.0	54.4	40.4	54.0	84.9	59.8
PSFT (Zhu et al., 2025) + RL	26.7	36.7	71.5	91.2	52.0	62.7	56.8	42.8	55.4	85.1	61.1
CurioSFT+ RL	**30.4**	**39.2**	**72.7**	**91.7**	**54.9**	**63.2**	**58.7**	**43.2**	**56.0**	**85.9**	**61.7**
Impr. vs SFT+RL	**+5.8**	**+7.1**	**+4.6**	**+3.5**	**+3.0**	**+6.1**	**+5.0**	**+2.1**	**+2.7**	**+2.4**	**+2.4**

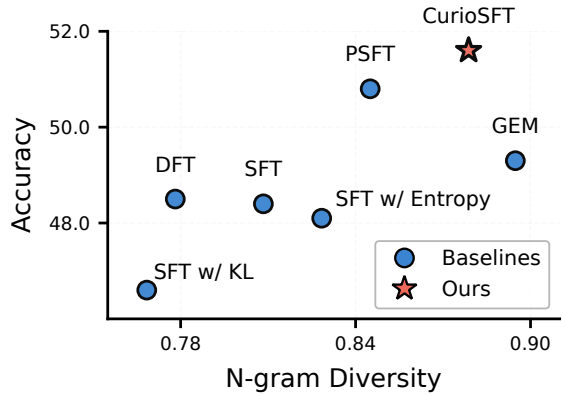


Figure 4: Accuracy vs. N-gram diversity.

ture $\hat{\tau} = 1.3$ leads to a clear OOD drop (54.2% $\rightarrow$ 52.2%, **-2.0** points), highlighting the importance of token-level adaptivity for retaining knowledge. Next, we remove the separate teacher model and directly use the current model as the teacher, which causes a modest performance degradation, supporting the role of a stable teacher signal in providing reliable entropy-preserving guidance.

Complexity. CurioSFT introduces additional computation due to an extra forward pass and token-wise temperature search, but the overhead remains within a practical and acceptable range (43.2 $\rightarrow$ 53.9 seconds per step). Moreover, most of the cost can be reduced by removing the separate teacher, at the risk of a small performance drop.

5.2 Unlocking the Potential of the SFT-then-RL Paradigm

Baselines. We next examine whether the preserved entropy during SFT leads to *meaningful* gains in the RL stage. We compare our solution against two families of baselines. First, we consider *single-stage hybrid SFT+RL* methods that fuse offline demonstrations with on-policy exploration into one single stage. Specifically, LUFFY (Yan et al., 2025) optimizes an RL objective on a mixture of online rollouts and offline demonstrations; Prefix-RFT (Huang et al., 2025) injects expert prefixes to steer exploration; and RL-PLUS (Dong et al., 2025) reuses expert examples during RL through multiple importance sampling. Second, we consider the standard *two-stage SFT-then-RL* paradigm, where we run GRPO from the SFT checkpoints in Section 5.1.

Results. Table 3 summarizes the results after the RL stage. CurioSFT + GRPO pipeline achieves the best overall performance, improving the two-stage baseline SFT+RL from 53.7% to 58.7% on the in-distribution tasks (**+5.0** points) and from 59.3% to 61.7% on the OOD tasks (**+2.4** points). The gains are most pronounced on the challenging AIME benchmarks, where CurioSFT+GRPO reach 39.2% on AIME24 (vs. 32.1% for SFT+RL) indicating that CurioSFT provides a substantially better initialization for RL exploration. Moreover, CurioSFT + GRPO consistently outperforms single-stage hybrid methods, demonstrating that a well-designed SFT stage that preserves *effective* exploration can unlock a higher RL performance ceiling.

5.3 Algorithm Robustness

We further evaluate CurioSFT on Qwen3-4B-Base (Yang et al., 2025) and Llama-3.1-8B-Instruct (Grattafiori et al., 2024). As shown in Table 4, CurioSFT consistently improves over vanilla

Table 4: Robustness across different backbones.

Method	AIME24/25	AMC	MATH.	Miner.	Olymp.	Avg.
Base model: Qwen3-4B-Base						
Base model	6.3 / 4.2	42.0	53.0	17.9	20.8	24.0
SFT	20.4 / 19.6	53.0	83.5	47.4	47.7	45.3
SFT + GRPO	27.5 / 22.9	62.9	89.0	50.2	57.9	51.7
CurioSFT	21.3 / 25.0	54.4	84.2	48.3	48.2	46.9
CurioSFT+ GRPO	**28.8 / 27.9**	**65.0**	**89.7**	**53.6**	**59.1**	**54.0**
Base model: Llama-3.1-8B-Instruct						
Base model	2.1 / 2.5	19.3	43.2	26.3	14.8	18.0
SFT	8.3 / 11.7	38.3	68.1	32.7	35.6	32.5
SFT + GRPO	9.6 / **12.1**	40.3	74.1	35.5	38.9	35.1
CurioSFT	10.0 / 10.4	38.9	69.0	33.0	36.6	33.0
CurioSFT+ GRPO	**14.2** / 11.7	**42.9**	**74.3**	**38.6**	**40.9**	**37.1**

Table 5: Performance on code tasks.

Method	HumanEval+	MBPP+	LiveCodeBench	Avg.↑
Qwen2.5-7B-Instruct	75.0	66.5	25.5	55.6
Vanilla SFT	78.6	70.3	26.1	58.3
CurioSFT	**80.1**	**71.2**	**27.4**	**59.6**
Impr. vs SFT	**+1.5**	**+0.9**	**+1.3**	**+1.3**

SFT on both backbones, improving the averaged accuracy from 45.3% to 46.9% on Qwen3-4B (**+1.6** points) and from 32.5% to 33.0% on Llama-3.1-8B-Instruct (**+0.5** points). These results validate that CurioSFT generalizes across model families and sizes.

5.4 Performance Beyond Math Tasks

To examine whether the benefits of CurioSFT extend beyond mathematical reasoning, we conduct a comparison experiment on *code task*, another reasoning-intensive generation task. We construct the training set by combining TACO (Li et al., 2023a) and LeetCodeDataset (Xia et al., 2025), and perform SFT for 3 epochs on Qwen2.5-7B-Instruct. We then compare CurioSFT against Vanilla SFT on three widely used code benchmarks: HumanEval+ (Chen et al., 2021), MBPP+ (Austin et al., 2021), and LiveCodeBench v1–5 (May 2023 to February 2025) (Jain et al., 2024). As shown in Table 5, CurioSFT consistently outperforms Vanilla SFT across all three benchmarks, improving the average score from 58.3 to 59.6 (**+1.3** points). Although preliminary, these results suggest that the advantage of preserving effective exploration during SFT is not limited to mathematics, and can also transfer to code generation tasks.

5.5 Hyperparameter Tuning

Unlike entropy loss, which typically requires careful tuning of the loss weight, CurioSFT adaptively computes a token-wise temperature and thus avoids introducing an additional sensitive coefficient. In practice, the key hyperparameter is the entropy pivot H_{pivot} in Eq. (7), which controls the overall strength of the entropy regularization. Increasing H_{pivot} weakens the overall encouragement (fewer tokens receive a large entropy increment), while decreasing it makes the entropy increment larger for more tokens, thereby preserving more entropy. We sweep H_{pivot} from 0.4 to 1.4. As shown in Figure 5, performance peaks around $H_{\text{pivot}} = 1.2$. Importantly, across the entire range, CurioSFT consistently maintains performance gains over the baseline, indicating that the proposed method is robust to the choice of H_{pivot}.

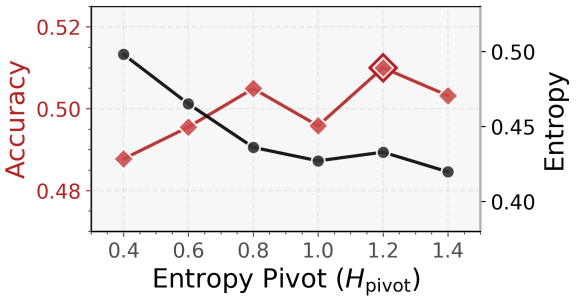


Figure 5: Sensitivity to the entropy pivot H_{pivot}.

Table 6: Impact of the EMA update frequency.

Freq.	AIME25/24	AMC23	MATH.	Miner.	Olymp.	Avg.↑
$n = 1$	25.2/28.4	58.6	85.1	48.3	52.8	49.7
n = 5	**26.3/29.6**	**59.9**	**87.0**	**49.8**	53.2	**51.0**
$n = 10$	26.1/28.2	59.3	86.5	48.1	**53.4**	50.3
$n = 50$	20.7/24.4	54.6	85.1	46.3	48.4	46.6

We further study the effect of the EMA update frequency for the teacher model in CurioSFT. Recall that the teacher parameters are updated every n steps using EMA. This update interval controls the balance between *teacher stability* and *teacher responsiveness*: if the teacher is updated too frequently, the distillation target may change too rapidly and become unstable; if it is updated too infrequently, the teacher may lag behind the student and provide a weaker self-exploratory distillation signal. To quantify this trade-off, we vary the EMA update interval $n \in \{1, 5, 10, 50\}$ while keeping all other settings fixed.

As shown in Table 6, updating the teacher every 5 steps yields the best overall performance. When the update interval is too small ($n = 1$), the teacher changes too quickly, which reduces the stability of the distillation target and slightly harms performance. In contrast, when the interval is too

large ($n = 50$), the teacher becomes excessively stale and can no longer provide sufficiently effective guidance, leading to a substantial performance drop.

6 Related Work

SFT in Post-Training. SFT plays an irreplaceable role in enhancing model capabilities during post-training (OpenAI et al., 2023; Team et al., 2023). Generally, SFT serves two primary functions: (a) serving as a large-scale *knowledge injection* (Mecklenburg et al., 2024), which significantly enhances zero-shot performance on OOD tasks (Wei et al., 2021); and (b) serving as a *cold-start initialization* that enables the model to rapidly adapt to specific response patterns, thereby raising the performance ceiling for subsequent post-training stages (typically RL). For example, DeepSeek-R1 (Guo et al., 2025) utilizes a small set of high-quality reasoning data to activate the model's inherent chain-of-thought capabilities before RL. ReTool (Feng et al., 2025) constructs a diverse, high-quality tool-use dataset to instruct the model on when to invoke specific tools. In this paper, we focus on the latter role, investigating how SFT can be optimized to provide a *superior initialization point* for the subsequent RL stage.

SFT-then-RL vs. Hybrid SFT with RL. RL plays a pivotal role in LLM post-training (Gu et al., 2026; Wang et al., 2026a). Several studies argue that the two-stage "SFT-then-RL" paradigm may underperform compared to applying RL directly to the base model (Zhang et al., 2025a,b; Yan et al., 2025; Lv et al., 2025; Fu et al., 2025). Hence, a line of research has explored fusing SFT and RL into a single-stage **hybrid paradigm**. For example, LUFFY (Yan et al., 2025) updates the model using both expert demonstrations and self-exploration rollouts via a weighted RL loss. RL-PLUS (Dong et al., 2025) introduces an exploration-based advantage function to balance SFT and RL losses. HPT (Lv et al., 2025) integrates the two objectives through a unified theoretical perspective. In contrast to these works, we empirically demonstrate that when intrinsic exploration capabilities are preserved during the SFT phase, the SFT-then-RL paradigm achieves a higher performance ceiling than hybrid approaches.

Diversity Regularization in SFT. The central challenge in SFT lies in its inherent susceptibility to overconfidence and diversity collapse. To address this, several works have explored regularization techniques. DFT (Wu et al., 2025) re-weights token updates based on generation probabilities; GEM (Li et al., 2024) employs reverse KL divergence to prevent the distribution from converging to a collapsed mode; and PSFT (Zhu et al., 2025) imposes trust-region constraints to limit policy drift. However, none of these approaches address the problem from the perspective of entropy collapse, which is the key for effective exploration in the "SFT-then-RL" paradigm.

7 Conclusion

In this paper, we address a critical bottleneck in the SFT-then-RL paradigm: standard SFT induces entropy collapse that severely constricts downstream exploration. We propose **CurioSFT**, an entropy-preserving SFT method utilizing adaptive self-distillation to maintain diverse yet valid exploration spaces. Experiments demonstrate that CurioSFT consistently outperforms vanilla SFT in both in-distribution and out-of-distribution tasks. More importantly, we verify that the preserved exploration effectively transfers to the RL stage, unlocking a significantly higher performance ceiling.

Limitations

While CurioSFT effectively preserves exploration during SFT, we acknowledge two limitations. First, our method incurs additional training overhead compared to vanilla SFT, as it requires an extra forward pass to compute the teacher distribution and token-wise temperature selection. Nevertheless, the added cost remains within a practical range in our experiments. Second, our approach is bounded by the base model's intrinsic capabilities. Since we rely on self-distillation to preserve and amplify the model's latent exploration, the benefit may be smaller when the base model lacks sufficient prior knowledge. Future work could incorporate external signals to further enhance the exploration capability beyond what self-distillation alone can provide.

References

Zeyuan Allen-Zhu and Yuanzhi Li. 2020. Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. *arXiv preprint arXiv:2012.09816*.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen

Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. 2021. Program synthesis with large language models.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, and 39 others. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv:1803.05457v1*.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Yuchen Zhang, Jiacheng Chen, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, Jiarui Yuan, Huayu Chen, Kaiyan Zhang, Xingtai Lv, Shuo Wang, Yuan Yao, Xu Han, and 6 others. 2025. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, and 181 others. 2025. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Shrey Desai and Greg Durrett. 2020. Calibration of pre-trained transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 295–302, Online. Association for Computational Linguistics.

Yihong Dong, Xue Jiang, Yongding Tao, Huanyu Liu, Kechi Zhang, Lili Mou, Rongyu Cao, Yingwei Ma, Jue Chen, Binhua Li, Zhi Jin, Fei Huang, Yongbin Li, and Ge Li. 2025. Rl-plus: Countering capability boundary collapse of llms in reinforcement learning with hybrid-policy optimization. *arXiv preprint arXiv:2508.00222*.

Jiazhan Feng, Shijue Huang, Xingwei Qu, Ge Zhang, Yujia Qin, Baoquan Zhong, Chengquan Jiang, Jinxin Chi, and Wanjun Zhong. 2025. Retool: Reinforcement learning for strategic tool use in llms. *arXiv preprint arXiv:2504.11536*.

Yuqian Fu, Tinghong Chen, Jiajun Chai, Xihuai Wang, Songjun Tu, Guojun Yin, Wei Lin, Qichao Zhang, Yuanheng Zhu, and Dongbin Zhao. 2025. Srft: A single-stage method with supervised and reinforcement fine-tuning for reasoning. *arXiv preprint arXiv:2506.19767*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Hao Gu, Hao Wang, Jiacheng Liu, Lujun Li, Qiyuan Zhu, Bei Liu, Binxing Xu, Lei Wang, Xintong Yang, Sida Lin, Sirui Han, and Yike Guo. 2026. Qarl: Rollout-aligned quantization-aware rl for fast and stable training under training–inference mismatch. *Preprint*, arXiv:2604.07853.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, and 175 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. 2024. 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*, pages 3828–3850.

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

Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. 2025. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model. *Preprint*, arXiv:2503.24290.

Zeyu Huang, Tianhao Cheng, Zihan Qiu, Zili Wang, Yinghui Xu, Edoardo M Ponti, and Ivan Titov. 2025. Blending supervised and reinforcement fine-tuning with prefix sampling. *arXiv preprint arXiv:2507.01679*.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. 2024. Livecodebench: Holistic and contamination free evaluation of large language models for code.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 2021. How can we know when language models know? on the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics*, 9:962–977.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*.

Lou Jost. 2006. Entropy and diversity. *Oikos*, 113(2):363–375.

Kyungyul Kim, ByeongMoon Ji, Doyoung Yoon, and Sangheum Hwang. 2021. Self-knowledge distillation with progressive refinement of targets. *Preprint*, arXiv:2006.12000.

Hynek Kydlíček. 2025. Math-verify: Math verification library.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. 2022. Solving quantitative reasoning problems with language models. *Advances in neural information processing systems*, 35:3843–3857.

Jia LI, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Costa Huang, Kashif Rasul, Longhui Yu, Albert Jiang, Ziju Shen, Zihan Qin, Bin Dong, Li Zhou, Yann Fleureau, Guillaume Lample, and Stanislas Polu. 2024. Numinamath.

Rongao Li, Jie Fu, Bo-Wen Zhang, Tao Huang, Zhihong Sun, Chen Lyu, Guang Liu, Zhi Jin, and Ge Li. 2023a. Taco: Topics in algorithmic code generation dataset. *arXiv preprint arXiv:2312.14852*.

Zheng Li, Xiang Li, Lingfeng Yang, Borui Zhao, Renjie Song, Lei Luo, Jun Li, and Jian Yang. 2023b. Curriculum temperature for knowledge distillation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 1504–1512.

Ziniu Li, Congliang Chen, Tian Xu, Zeyu Qin, Jiancong Xiao, Zhi-Quan Luo, and Ruoyu Sun. 2024. Preserving diversity in supervised fine-tuning of large language models. *arXiv preprint arXiv:2408.16673*.

Kezhao Liu, Jason Klein Liu, Mingtao Chen, and Yiming Liu. 2025a. Rethinking kl regularization in rlhf: From value estimation to gradient optimization. *Preprint*, arXiv:2510.01555.

Zihan Liu, Zhuolin Yang, Yang Chen, Chankyu Lee, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2025b. Acereason-nemotron 1.1: Advancing math and code reasoning through sft and rl synergy. *arXiv preprint arXiv:2506.13284*.

Xingtai Lv, Yuxin Zuo, Youbang Sun, Hongyi Liu, Yuntian Wei, Zhekai Chen, Xuekai Zhu, Kaiyan Zhang, Bingning Wang, Ning Ding, and Bowen Zhou. 2025. Towards a unified view of large language model post-training. *arXiv preprint arXiv:2509.04419*.

Nick Mecklenburg, Yiyou Lin, Xiaoxiao Li, Daniel Holstein, Leonardo Nunes, Sara Malvar, Bruno Silva, Ranveer Chandra, Vijay Aski, Pavan Kumar Reddy Yannam, Tolga Aktas, and Todd Hendry. 2024. Injecting new knowledge into large language models via supervised fine-tuning. *arXiv preprint arXiv:2404.00213*.

OpenAI, :, Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K. Arora, Yu Bai, Bowen Baker, Haiming Bao, Boaz Barak, Ally Bennett, Tyler Bertao, Nivedita Brett, Eugene Brevdo, Greg Brockman, Sebastien Bubeck, and 108 others. 2025. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Minh Pham, Minsu Cho, Ameya Joshi, and Chinmay Hegde. 2022. Revisiting self-distillation. *arXiv preprint arXiv:2206.08491*.

Rafael Rafailov, Joey Hejna, Ryan Park, and Chelsea Finn. 2024. From r to q^*: Your language model is secretly a q-function. *arXiv preprint arXiv:2404.12358*.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2024. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

Rulin Shao, Akari Asai, Shannon Zejiang Shen, Hamish Ivison, Varsha Kishore, Jingming Zhuo, Xinran Zhao, Molly Park, Samuel G. Finlayson, David Sontag, Tyler Murray, Sewon Min, Pradeep Dasigi, Luca Soldaini, Faeze Brahman, Wen tau Yih, Tongshuang Wu, Luke Zettlemoyer, Yoon Kim, and 2 others. 2025a. Dr tulu: Reinforcement learning with evolving rubrics for deep research. *arXiv preprint arXiv:2511.19399*.

Zhihong Shao, Yuxiang Luo, Chengda Lu, ZZ Ren, Jiewen Hu, Tian Ye, Zhibin Gou, Shirong Ma, and Xiaokang Zhang. 2025b. Deepseekmath-v2: Towards self-verifiable mathematical reasoning. *arXiv preprint arXiv:2511.22570*.

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

Idan Shenfeld, Jyothish Pari, and Pulkit Agrawal. 2025. Rl's razor: Why online reinforcement learning forgets less. *Preprint*, arXiv:2509.04259.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2024. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*.

Hongze Tan, Zihan Wang, Jianfei Pan, Jinghao Lin, Hao Wang, Yifan Wu, Tao Chen, Zhihang Zheng, Zhihao Tang, and Haihua Yang. 2025. Gtpo and grpo-s: Token and sequence-level reward shaping with policy entropy. *arXiv preprint arXiv:2508.04349*.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy Lillicrap, Angeliki Lazaridou, and 1332 others. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Kimi Team, Yifan Bai, Yiping Bao, Y. Charles, Cheng Chen, Guanduo Chen, Haiting Chen, Huarong Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, Zhuofu Chen, Jialei Cui, Hao Ding, Mengnan Dong, Angang Du, and 181 others. 2025. Kimi k2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*.

Jiacheng Wang, Tianle Chen, Pengyu Cheng, Xiaofeng Hou, and Jiacheng Liu. 2026a. Adareason: Progressive training of multi-lora adapters for budget-adaptive language reasoning models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 26242–26250.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, Yuqiong Liu, An Yang, Andrew Zhao, Yang Yue, Shiji Song, Bowen Yu, Gao Huang, and Junyang Lin. 2025a. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *arXiv preprint arXiv:2506.01939*.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. 2024. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *arXiv preprint arXiv:2406.01574*.

Zihan Wang, Hao Wang, Shi Feng, Xiaocui Yang, Daling Wang, Yiqun Zhang, Jinghao Lin, Haihua Yang, and Xiaozhong Ji. 2026b. Deepmed: Building a medical deepresearch agent via multi-hop med-search data and turn-controlled agentic training & inference. *arXiv preprint arXiv:2601.18496*.

Zihan Wang, Xingle Xu, Hao Wang, Yiwen Ye, Yuchen Li, Linhao Wang, Hongze Tan, Peidong Wang, Shi Feng, Guoqing Chen, Jinghao Lin, Zijing Wang, Yiqun Zhang, Yongkang Liu, Xiaocui Yang, Tao Yan, Shengzhi Wang, Yuhang Wu, and Ge Yu. 2025b. A survey on entropy mechanism in large reasoning models. *TechRxiv*, 2025(1220).

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Yongliang Wu, Yizhou Zhou, Zhou Ziheng, Yingzhe Peng, Xinyu Ye, Xinting Hu, Wenbo Zhu, Lu Qi, Ming-Hsuan Yang, and Xu Yang. 2025. On the generalization of sft: A reinforcement learning perspective with reward rectification. *arXiv preprint arXiv:2508.05629*.

Yunhui Xia, Wei Shen, Yan Wang, Jason Klein Liu, Huifeng Sun, Siyue Wu, Jian Hu, and Xiaolong Xu. 2025. Leetcodedataset: A temporal dataset for robust evaluation and efficient training of code llms. *arXiv preprint arXiv:2504.14655*.

Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang. 2025. Learning to reason under off-policy guidance. *arXiv preprint arXiv:2504.14945*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. 2024. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*.

Shaokun Zhang, Yi Dong, Jieyu Zhang, Jan Kautz, Bryan Catanzaro, Andrew Tao, Qingyun Wu, Zhiding Yu, and Guilin Liu. 2025a. Nemotron-research-tool-n1: Exploring tool-using language models with reinforced reasoning. *arXiv preprint arXiv:2505.00024*.

Wenhao Zhang, Yuexiang Xie, Yuchang Sun, Yanxi Chen, Guoyin Wang, Yaliang Li, Bolin Ding, and Jingren Zhou. 2025b. On-policy rl meets off-policy experts: Harmonizing supervised fine-tuning and reinforcement learning via dynamic weighting. *https://arxiv.org/abs/2508.11408*.

Yiqun Zhang, Peng Ye, Xiaocui Yang, Shi Feng, Shufei Zhang, Lei Bai, Wanli Ouyang, and Shuyue Hu. 2025c. Nature-inspired population-based evolution of large language models. *Preprint*, arXiv:2503.01155.

Wenhong Zhu, Ruobing Xie, Rui Wang, Xingwu Sun, Di Wang, and Pengfei Liu. 2025. Proximal supervised fine-tuning. *arXiv preprint arXiv:2508.17784*.

A Derivation of Entropy Loss

Let $\mathcal{U}$ be the uniform distribution, i.e., $\mathcal{U}(y) = \frac{1}{|\mathcal{V}|}$. The KL divergence from $\pi_\theta(\cdot \mid \mathbf{s})$ to $\mathcal{U}$ is:

$$\begin{aligned} &D_{\mathrm{KL}}\Big(\pi_\theta(\cdot \mid \mathbf{s}) \,\Big\|\, \mathcal{U}\Big) \\ &\quad= \sum_{y\in\mathcal{V}} \pi_\theta(y \mid \mathbf{s}) \log \frac{\pi_\theta(y \mid \mathbf{s})}{\mathcal{U}(y)} \\ &\quad= \sum_{y\in\mathcal{V}} \pi_\theta(y \mid \mathbf{s}) \Big(\log \pi_\theta(y \mid \mathbf{s}) - \log \tfrac{1}{|\mathcal{V}|} \Big) \\ &\quad= \sum_{y\in\mathcal{V}} \pi_\theta(y \mid \mathbf{s}) \log \pi_\theta(y \mid \mathbf{s}) + \log|\mathcal{V}| \\ &\quad= -H\big(\pi_\theta(\cdot \mid \mathbf{s})\big) + \log|\mathcal{V}|. \end{aligned}$$

Since $\log|\mathcal{V}|$ is constant w.r.t. θ, minimizing $D_{\mathrm{KL}}\big(\pi_\theta(\cdot \mid \mathbf{s}) \,\|\, \mathcal{U}\big)$ is equivalent to maximizing $H\big(\pi_\theta(\cdot \mid \mathbf{s})\big)$. Therefore, naive entropy regularization implicitly encourages the model distribution to move toward the uniform distribution over $\mathcal{V}$.

B Proof of Monotonicity

Theorem 1. *Let $\pi_\theta(\cdot \mid \mathbf{s}; \tau) = \mathrm{softmax}\big(z_\theta(\cdot \mid \mathbf{s})/\tau\big)$ be the distribution derived from logits $z_\theta(\cdot \mid \mathbf{s})$ with temperature τ. Then the entropy $H\big(\pi_\theta(\cdot \mid \mathbf{s}; \tau)\big)$ is non-decreasing with respect to τ.*

Proof. For brevity, let π_y denote $\pi_\theta(y \mid \mathbf{s}; \tau)$, and let z_y denote the logit for token y (i.e., $z_y := z_\theta(y \mid \mathbf{s})$). Recall that:

$$\log \pi_y = \frac{z_y}{\tau} - \log Z(\tau),$$

where $Z(\tau)$ is the partition function. The derivative of the entropy $H(\pi) = -\mathbb{E}_\pi[\log \pi]$ with respect to τ is derived as follows:

$$\begin{aligned} \frac{\partial H}{\partial \tau} &= -\sum_y \frac{\partial \pi_y}{\partial \tau} \log \pi_y - \sum_y \pi_y \frac{\partial \log \pi_y}{\partial \tau} \\ &= -\sum_y \frac{\partial \pi_y}{\partial \tau} \log \pi_y \quad (\text{since } \sum_y \pi_y = 1) \\ &= -\sum_y \frac{\partial \pi_y}{\partial \tau} \Big(\frac{z_y}{\tau} - \log Z \Big) \\ &= -\frac{1}{\tau} \sum_y z_y \frac{\partial \pi_y}{\partial \tau}. \end{aligned} \tag{8}$$

Using the standard derivative of the softmax function $\frac{\partial \pi_y}{\partial \tau} = \frac{\pi_y}{\tau^2}\Big(\mathbb{E}_\pi[z] - z_y\Big)$ and substituting this into Eq. (8), we have:

$$\begin{aligned} \frac{\partial H}{\partial \tau} &= -\frac{1}{\tau^3} \sum_y \pi_y z_y \Big(\mathbb{E}_\pi[z] - z_y\Big) \\ &= \frac{1}{\tau^3} \left[\sum_y \pi_y z_y^2 - \left(\sum_y \pi_y z_y \right)^2 \right] \\ &= \frac{1}{\tau^3} \left(\mathbb{E}_\pi[z^2] - (\mathbb{E}_\pi[z])^2 \right) \\ &= \frac{1}{\tau^3} \mathrm{Var}_\pi[z]. \end{aligned} \tag{9}$$

Since $\mathrm{Var}_\pi[z] \geq 0$ and $\tau > 0$, the derivative is always non-negative. Notably, the variance $\mathrm{Var}_\pi[z]$ vanishes if and only if the distribution π is uniform. Thus, $H\big(\pi_\theta(\cdot \mid \mathbf{s}; \tau)\big)$ is *strictly non-decreasing* in τ, except in the case of a uniform distribution. □

C Temperature scaling as the KL-closest higher-entropy teacher

To reduce overconfidence while staying close to π, we aim to construct a teacher distribution π^{tch} that satisfies two requirements: (i) it has *higher entropy* than the current policy, so it encourages exploration; and (ii) it remains *KL-close* to π, so the supervision signal is stable and capability-aware. This naturally leads to the following constrained optimization problem:

$$\begin{aligned} \mathbf{P1}: \min_{\pi^{\text{tch}}} \quad & D_{\mathrm{KL}}\Big(\pi^{\text{tch}}(\cdot \mid \mathbf{s}) \,\Big\|\, \pi(\cdot \mid \mathbf{s})\Big) \\ \text{s.t.} \quad & H\big(\pi^{\text{tch}}(\cdot \mid \mathbf{s})\big) \;\geq\; H\big(\pi(\cdot \mid \mathbf{s})\big) + \Delta. \end{aligned} \tag{10}$$

Theorem 2 (Temperature scaling is the optimum of **P1**)**.** *Given an entropy increment Δ, the unique optimum π_*^{tch} of (10) is a temperature-scaled distribution: there exists a unique $\hat{\tau} > 1$ such that*

$$\begin{aligned} \pi_*^{tch}(y \mid \mathbf{s}) &= \pi(y \mid \mathbf{s}; \hat{\tau}) \\ &= \frac{\exp\big(z(y \mid \mathbf{s})/\hat{\tau}\big)}{\sum_{y' \in \mathcal{V}} \exp\big(z(y' \mid \mathbf{s})/\hat{\tau}\big)}, \end{aligned} \tag{11}$$

and it satisfies $H\big(\pi_*^{tch}(\cdot \mid \mathbf{s})\big) = H\big(\pi(\cdot \mid \mathbf{s})\big) + \Delta$.

Proof. We solve **P1** by forming its Lagrangian and applying the KKT conditions. For convenience, denote the target entropy as $H^{\text{tar}} = H\big(\pi(\cdot \mid \mathbf{s})\big) + \Delta$. Rewrite the inequality constraint in the standard KKT form:

$$g(\pi^{\text{tch}}) := H^{\text{tar}} - H\big(\pi^{\text{tch}}(\cdot \mid \mathbf{s})\big) \leq 0,$$

with KKT multiplier $\lambda_H \geq 0$. Since $H(\pi^{\text{tch}}) = -\sum_{y\in\mathcal{V}} \pi^{\text{tch}}(y \mid \mathbf{s}) \log \pi^{\text{tch}}(y \mid \mathbf{s})$, we have

$g(\pi^{\text{tch}}) = H^{\text{tar}} + \sum_{y\in\mathcal{V}} \pi^{\text{tch}}(y \mid \mathbf{s}) \log \pi^{\text{tch}}(y \mid \mathbf{s})$. The Lagrangian of **P1** is:

$$\begin{aligned}\mathcal{L}(\pi^{\text{tch}}, \lambda, \lambda_H) = & \\ & \sum_{y\in\mathcal{V}} \pi^{\text{tch}}(y \mid \mathbf{s}) \log \frac{\pi^{\text{tch}}(y \mid \mathbf{s})}{\pi(y \mid \mathbf{s})} \\ & + \lambda\Big(\sum_{y\in\mathcal{V}} \pi^{\text{tch}}(y \mid \mathbf{s}) - 1\Big) \\ & + \lambda_H\Big(H^{\text{tar}} + \sum_{y\in\mathcal{V}} \pi^{\text{tch}}(y \mid \mathbf{s}) \log \pi^{\text{tch}}(y \mid \mathbf{s})\Big),\end{aligned} \tag{12}$$

where λ is the multiplier for normalization and $\lambda_H \geq 0$ is the KKT multiplier for $g(\pi^{\text{tch}}) \leq 0$. Taking the derivative w.r.t. $\pi^{\text{tch}}(y \mid \mathbf{s})$ and setting it to zero:

$$\begin{aligned}(1 + \log \pi^{\text{tch}}(y \mid \mathbf{s})) - \log \pi(y \mid \mathbf{s}) + \lambda \\ + \lambda_H\big(1 + \log \pi^{\text{tch}}(y \mid \mathbf{s})\big) = 0.\end{aligned} \tag{13}$$

Simplifying the above equation gives:

$$\log \pi^{\text{tch}}(y \mid \mathbf{s}) = \frac{1}{1+\lambda_H} \log \pi(y \mid \mathbf{s}) + C, \tag{14}$$

where C is a normalization constant independent of y. Define $\hat{\tau} := 1 + \lambda_H$, then we obtain a power-form solution:

$$\pi^{\text{tch}}(y \mid \mathbf{s}) \propto \pi(y \mid \mathbf{s})^{1/\hat{\tau}}. \tag{15}$$

When $\Delta > 0$, the entropy constraint must be active at optimum; otherwise one could move π^{tch} closer to π and strictly decrease $D_{\text{KL}}(\pi^{\text{tch}} \| \pi)$ while remaining feasible. Thus, by complementary slackness, $\lambda_H > 0$ and hence $\hat{\tau} > 1$. Using $\pi(y \mid \mathbf{s}) \propto \exp(z(y \mid \mathbf{s}))$, we have:

$$\begin{aligned}\pi^{\text{tch}}(y \mid \mathbf{s}) \propto \pi(y \mid \mathbf{s})^{1/\hat{\tau}} \ \propto \ \Big(\exp(z(y \mid \mathbf{s}))\Big)^{1/\hat{\tau}} \\ \propto \ \exp\big(z(y \mid \mathbf{s})/\hat{\tau}\big),\end{aligned} \tag{16}$$

which is exactly the temperature-scaled softmax form in (11) after normalization. Finally, by Appendix B, under the full-support assumption the entropy $H\big(\pi(\cdot \mid \mathbf{s}; \tau)\big)$ is continuous and strictly increasing in τ, so there exists a unique $\hat{\tau} > 1$ such that $H\big(\pi(\cdot \mid \mathbf{s}; \hat{\tau})\big) = H^{\text{tar}}$. Since $D_{\text{KL}}(\cdot \| \pi)$ is convex in its first argument and the feasible set is convex because entropy is concave, **P1** is a convex optimization problem. Therefore, any distribution that satisfies the KKT conditions is globally optimal. Consequently, the teacher distribution constructed by temperature scaling is the KL-closest distribution to π among all distributions whose entropy is increased by at least Δ. □

Table 7: Hyperparameters in SFT stage

Parameter Name	Value
Epochs	3
Batch Size	256
Max Response Length	8192
Learning Rate	1e-5
Warm Up Style	cosine
Warm Up Steps	60

Table 8: Hyperparameters in RL stage

Parameter Name	Value
Training Steps	500
Batch Size	128
Mini Batch Size	64
Learning Rate	1e-6
Clip Higher	0.28
Clip Lower	0.2

D Efficient Implementation of Entropy-Guided Temperature Search

Leveraging the monotonic relationship between entropy and the sampling temperature, we can find a unique temperature $\hat{\tau}_t$ whose entropy matches a desired target via binary search. To make this procedure efficient in large-scale LLM training, we implement it with two key design choices:

- **Vectorized binary search.** Instead of searching for $\hat{\tau}_t$ token by token, we perform a batched binary search over all tokens in a mini-batch using vectorized PyTorch operations.
- **Top-k entropy approximation.** The softmax distribution over the vocabulary is typically heavy-tailed, so the entropy is dominated by the highest-probability tokens. We therefore approximate the full entropy using only the top-k logits:

$$H\big(\pi_\theta(\cdot \mid \mathbf{s}_t)\big) \approx -\sum_{y\in\mathcal{V}_{:k}} \hat{\pi}_\theta(y \mid \mathbf{s}_t) \log \hat{\pi}_\theta(y \mid \mathbf{s}_t), \tag{17}$$

  where $\mathcal{V}_{:k}$ denotes the top-k tokens and $\hat{\pi}_\theta$ is the distribution renormalized over this subset.

In all experiments, we set $k = 512$, which reduces the complexity of the entropy computation from $O(|\mathcal{V}|)$ to $O(k)$ and leads to only a modest training overhead.

Table 9: Training Prompt

<lim_start|>system \nYou are an exceptional mathematician. Your task is to solve mathematical questions through a systematic and thorough reasoning process. This involves careful analysis, exploration of possible approaches, verification of intermediate steps, critical reassessment, and iterative refinement of your reasoning process. Structure your response in two distinct sections: "Thought" and "Solution". In the "Thought" section, present your detailed reasoning process in the following format: \n<think>\nYour detailed reasoning, including brainstorming, logical deductions, verification, and refinement of ideas.\n</think>\nThis section must conclude with "</think>", and should reflect deep, reflective, and self-correcting thinking process. In the "Solution" section, following the "</think>", concisely draw the final, logical, and accurate answer from your reasoning. Please output your final answer within \\boxed{}.<lim_end|> \n<lim_start|>user \n Here is the question:{question}<lim_end|> \n<lim_start|>assistant \n<think>

E Experiment Setting

Training. For both SFT and RL, we use `Verl` (Sheng et al., 2024) as the training framework. All experiments are conducted on 8× NVIDIA H800 GPUs. The full SFT hyperparameters are provided in Table 7. For the RL stage, we use a binary reward: a response receives reward 1 if it matches the ground truth (verified by `Math-Verify` (Kydlíček, 2025)) and follows the required format, and 0 otherwise. The full RL hyperparameters are listed in Table 8. We use the prompt template in Table 9 for both SFT and RL training.

Dataset Processing. We use OpenR1-Math (Yan et al., 2025) for both SFT and RL training. We observe that a large portion of the questions contain irrelevant or distracting information. To reduce such noise, we rewrite the questions using DeepSeek-V3 (DeepSeek-AI et al., 2025) and keep the original ground-truth answers unchanged. Example rephrase prompts are shown in Table 10.

Table 10: Question Rephrase Prompt

I will provide a post from a math-related forum that contains a math problem. Your task is to extract only the math problem statement and remove any irrelevant or noisy content (e.g., commentary, solutions, chat, metadata). Keep the original wording and question type intact, and present the extracted problem clearly and concisely.
Remove any redundant context, personal commentary, anecdotes, or unrelated information. But make sure not to change the meaning of the problem and keep all necessary mathematical or technical details.

Here are a few examples.
Example 1:
Input:
What is the remainder of $8^6 + 7^7 + 6^8$ is divided by 5?
no calculator, of course, paper isn't needed either, but sure.

Output:
What is the remainder of $8^6 + 7^7 + 6^8$ when divided by 5?

Example 2:
Input:
(20 points) Let x, y be non-zero real numbers, and satisfy $\frac{x \sin \frac{\pi}{5} + y \cos \frac{\pi}{5}}{x \cos \frac{\pi}{5} - y \sin \frac{\pi}{5}} = \tan \frac{9\pi}{20}$.(1) Find the value of $\frac{y}{x}$;

Output:
Let x, y be non-zero real numbers, and satisfy $\frac{x \sin \frac{\pi}{5} + y \cos \frac{\pi}{5}}{x \cos \frac{\pi}{5} - y \sin \frac{\pi}{5}} = \tan \frac{9\pi}{20}$. Find the value of $\frac{y}{x}$.

Now, here is the text you need to extract the problem.

Input:
{question}

Output: