Entropy-KL Divergence-based Token Masking: A Novel Approach for Selective Fine-tuning of Large Language Models

Anonymous ACL submission

Abstract

Supervised fine-tuning (SFT) followed by reinforcement learning (RL) has become a standard post-training paradigm for large language models. This paradigm provides a cold-start for RL exploration, avoiding the inefficiency of pure RL where on-policy sampling yields insufficient positive samples. However, in practice, existing approaches often use a small amount of data for SFT initialization compared to the RL phase, which can cause the model to fit the limited samples and shift away from its pre-trained distribution. This distribution shift impedes the model's ability to effectively explore during subsequent RL training. To address this challenge, we propose that in low-data regimes, SFT should prioritize *activating* task-relevant capabilities rather than *memorizing* specific content. Along this line, we propose **EKSFT** (Entropy-KL Selective Fine-Tuning), which selectively masks tokens that exhibit either high entropy or high KL divergence from a reference model. By excluding these high-uncertainty, distribution-shifting tokens from imitation, EKSFT injects task-specific knowledge while preserving the integrity of the model's pre-trained distribution. Empirical evaluations on mathematical reasoning benchmarks demonstrate that EKSFT consistently outperforms standard SFT. Further RL fine-tuning from the EKSFT model yields consistently better post-RL performance, indicating improved exploration for the RL stage. Our codes and datasets are available at https://anonymous.4open.science/r/EK-SFT.

1 Introduction

Recently, the two-stage training pipeline of Supervised Fine-Tuning (SFT) then Reinforcement Learning (RL) has demonstrated remarkable capabilities across various downstream applications of large language models (LLMs) (Dao and Vu, 2025; Zhi et al., 2025; Pang et al., 2025; Yang et al., 2025; OpenAI, 2023), though its limitations are increasingly apparent. A reasonable viewpoint is that SFT provides a cold-start, as pure RL often yields insufficient positive samples for efficient learning. However, recent studies suggest that SFT may impair the model's exploration capacity, potentially affecting the performance of subsequent RL training (Chen et al., 2025b; Zhang et al., 2025b,c). This limitation stems from the fact that SFT essentially performs behavior cloning (Chu et al., 2025; Qin and Springenberg, 2025; Zhu et al., 2025b) on expert data, and when the dataset is limited or distributionally misaligned with the pretraining corpus, the model may suffer substantial parameter drift, leading to degraded generalization. Meanwhile, token-by-token imitation tends to sharpen the policy distribution, reducing rollout diversity, which is harmful for collecting sufficiently diverse rollouts during the RL phase. As shown in Figure 1, SFT models exhibit lower Pass@K (Chen et al., 2025d) at large K compared to Base models, suggesting that SFT narrows the effective output distribution and reduces sampling diversity.

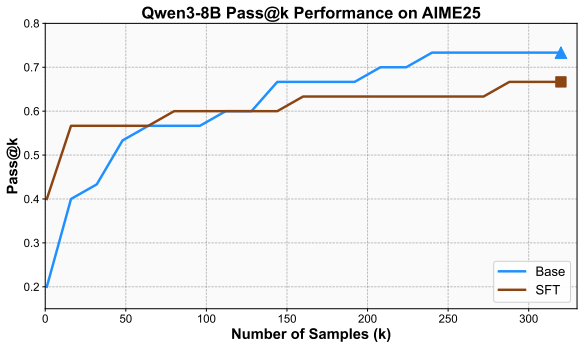

Figure 1: We train Qwen3-8B on the OpenR1 dataset and evaluate the Pass@K performance on AIME25. These results show that SFT models can yield suboptimal performance compared to Base models when K exceeds a threshold, e.g., K=140 as shown.

To address the above issues, existing methods can be categorized into two main research lines. A category of methods combines the SFT training

objective and RL training objective to leverage the strengths of both paradigms (Zhang et al., 2025c; Wu et al., 2025; Yan et al., 2025; Fu et al., 2025; Chen et al., 2025c; He et al., 2025). For example, DFT (Wu et al., 2025) reformulates SFT within an RL objective, while methods like CHORD (Zhang et al., 2025c), BRIDGE (Chen et al., 2025c) and AMFT (He et al., 2025) use static or dynamic weighting to balance the two objectives during training. However, in practice, mixing these two objectives often requires delicate tuning and shows high sensitivity to weighting schedules. Another line of work focuses on revising the SFT objective to improve generalization in the following RL stage. For example, PSFT (Zhu et al., 2025b) adopts a clipped surrogate objective to prevent excessive policy update, while IW-SFT (Qin and Springenberg, 2025) and ASFT (Zhu et al., 2025a) leverage importance weighting (Rubinstein and Kroese, 2016) to control the distributional shift that maintains the training stability for the RL phase. However, these methods still supervise all tokens uniformly and rely on global regularization to control deviation. Moreover, they do not explicitly address entropy collapse (Cui et al., 2025), which may diminish the model's exploration capacity (Xie et al., 2024).

To address these challenges, we propose that SFT should regard *activating* task-relevant capabilities rather than *memorizing* specific content (Chu et al., 2025; Xie et al., 2024) when SFT serves as an initialization stage for subsequent RL, improving the generalization of SFT models and enhancing the exploration for the subsequent RL phase. Along this line, we propose Entropy-KL Selective Fine-Tuning (EKSFT), which selectively masks tokens that exhibit either high entropy or high KL divergence from a reference model, while high entropy tokens usually indicate greater model uncertainty and high KL-divergence tokens represent deviation from the reference model. Additionally, EKSFT incorporates entropy loss (Zhang et al., 2025d; Cheng et al., 2025) and KL divergence loss (Li et al., 2025) constraints to further mitigate entropy collapse and maintain proximity to the pretrained distribution. By combining selective masking with explicit regularization, EKSFT injects task-relevant knowledge while maintaining small parameter drift and mitigating entropy collapse. The contributions of this paper can be summarized as follows:

- We propose a new perspective on the SFT-then-RL paradigm: SFT should regard *activating* task-relevant capabilities rather than *memorizing* specific content when SFT serves as an initialization stage for subsequent RL.
- We propose EKSFT, a novel approach that refines SFT by selectively masking high-entropy and high-KL-divergence tokens and incorporating entropy and KL regularization loss, thereby enhancing generalization and exploration for the subsequent RL phase.
- Empirical evaluations against multiple state-of-the-art (SOTA) baselines across several benchmarks demonstrate the effectiveness and superior performance of our proposed EKSFT.

2 Related Work

2.1 Supervised Fine-Tuning

SFT adapts pre-trained models to downstream tasks by training on task-specific (Chu et al., 2025; Xu et al., 2025a, 2026) datasets. It is often used as an initialization stage in the multi-stage post-training pipeline. However, recent studies have shown that the standard Cross-Entropy (CE) loss is not the best fit for SFT (Li et al., 2025; Xiao, 2024), pointing out that fine-tuning with standard CE loss often significantly reduces diversity. Moreover, SFT essentially performs behavior cloning (Chu et al., 2025), and when used as a cold-start for subsequent RL, it may cause distribution sharpening, reducing exploration capacity. GEM (Li et al., 2025) proposes a game-theoretic SFT algorithm with reverse Kullback-Leibler (KL) divergence minimization to preserve diversity and mitigate overfitting. IW-SFT (Qin and Springenberg, 2025) and ASFT (Zhu et al., 2025a) involve importance-weighted mechanisms to be further generalized. PSFT (Zhu et al., 2025b) is inspired by TRPO (Schulman et al., 2015) to adopt a clipped surrogate objective, leaving room for further optimization. Nonetheless, these approaches still supervise all tokens uniformly with global regularization, leading to suboptimal generalization for the subsequent RL phase.

2.2 Reinforcement Learning for LLM Alignment

Reinforcement learning (RL) has been widely adopted to enhance the alignment of large language models (LLMs) with human preferences (Bai et al., 2022; Ouyang et al., 2022). Recent advancements show remarkable success in complex reasoning tasks like mathematics and code generation (DeepSeek-AI, 2024) through Reinforcement

Learning from Verifiable Reward (RLVR) (Guo et al., 2025; Zhang et al., 2025a). RLVR leverages programmatic verifiers, such as unit tests (Poznanski et al., 2025) or answer checkers (Yu et al., 2025), to provide reward signals to achieve superior performance. However, RL-based exploration is often constrained by the base model's insufficient positive samples, leading to high computational expense and learning inefficiency (Wu et al., 2025; Zhu et al., 2025b). This limitation motivates an initialization phase that activates the base model's relevant capabilities before the RL phase.

2.3 Combining Supervised Fine-Tuning and Reinforcement Learning

To exploit the complementary strengths of SFT and RL, recent studies have explored their combination (Ouyang et al., 2022; Sheng et al., 2025; Liu et al., 2025a). SFT-then-RL paradigm (Lambert et al., 2024; Liu et al., 2025b) is widely adopted, which is a two-stage training pipeline where SFT is first employed to initialize the model, followed by RL for further optimization, and has been widely adopted in LLM applications, e.g., medical and financial domains (Zhi et al., 2025; Pang et al., 2025). The concurrent DFT (Wu et al., 2025) views that SFT's gradient update is equivalent to a policy gradient to rescale the SFT objective. CHORD (Zhang et al., 2025c), AMFT (He et al., 2025), and BRIDGE (Chen et al., 2025c) unify SFT and RL by designing a joint loss function between SFT and RL objectives with the weighted mechanism. SRFT (Fu et al., 2025) integrates both fine-tuning paradigms through entropy-aware weighting mechanisms. However, joint optimization is often highly sensitive to weighting schedules and can be unstable in practice, while CE-based imitation may still cause distribution sharpening and reduce exploration. In contrast, we improve the standard SFT via the selective token masking mechanism and fine-grained regularization, aiming to enhance the generalization and exploration for the subsequent RL phase.

3 Preliminary

Supervised Fine-Tuning. SFT is a common approach to adapt LLMs to downstream tasks. Given a supervised dataset $\mathcal{D} = \{(x_i, y_i{}^*)\}_{i=1}^N$, where x_i denotes the prompt and $y_i{}^* = (y_{i,1}, y_{i,2}, ...)$ denotes the corresponding response of $|y_i{}^*|$ tokens. The objective is to train the policy model π_θ to maximize the likelihood of the response $y_i{}^*$ given the corresponding prompt x_i:

$$\mathcal{L}_{\text{SFT}}(\theta) = -\mathbb{E}_{(x_i, y_i{}^*)\sim\mathcal{D}} \left[\sum_{t=1}^{|y_i{}^*|} \log \pi_\theta(x_i, y_{i,t} | y^*_{i,<t}) \right]. \tag{1}$$

where, $y^*_{i,<t} = (y_{i,1}, y_{i,2}, ..., y_{i,t-1})$ denotes the previous tokens before the index t.

Token-level Entropy. Token-level entropy quantifies the uncertainty of the next-token probability distribution produced by a language model. Given a prompt x and a language model distribution π_θ, the token-level entropy at time step t is defined as:

$$H(\pi_\theta(\cdot|x, y_{<t})) = -\sum_{v\in\mathcal{V}} \pi_\theta(v|x, y_{<t}) \log \pi_\theta(v|x, y_{<t}) \tag{2}$$

where $\mathcal{V}$ denotes the vocabulary and $y_{<t}$ denotes the previous tokens before the index t.

Token-level Kullback-Leibler divergence in LLMs. The KL is a statistical divergence that quantifies how one probability distribution differs from another. Given two language models π_θ and π_{ref}, the token-level KL divergence from π_θ to π_{ref} is defined as:

$$\text{KL}(\pi_\theta(\cdot|x, y_{<t}) || \pi_{\text{ref}}(\cdot|x, y_{<t})) = \sum_{v\in\mathcal{V}} \pi_\theta(v|x, y_{<t}) \log \frac{\pi_\theta(v|x, y_{<t})}{\pi_{\text{ref}}(v|x, y_{<t})} \tag{3}$$

where $\mathcal{V}$ denotes the vocabulary and $y_{<t}$ denotes the previous tokens before the index t.

4 Methodology

In this section, we propose EKSFT, aiming to better leverage task-relevant knowledge of LLMs on the expert dataset and mitigate the generalization decline and entropy collapse by the standard Supervised Fine-Tuning, which is beneficial for the subsequent RL phase.

4.1 Token Selection

Large language models (LLMs) follow an autoregressive fashion, where the smallest unit during training or inference is the token. To enhance the model's generalization after training, recent studies have shown that the effectiveness of designing entropy-preserving mechanisms and incorporating KL-divergence constraints.

High-entropy tokens represent positions where the model is uncertain. Forcing the model to imitate these uncertain predictions may sharpen the policy distribution and reducing output diversity instead of maintaining the distribution and activating the task-relevant knowledge. **High-KL tokens** indicate positions where the current policy has already diverged significantly from the reference model. Continuously supervising these tokens would further amplify distributional drift. Therefore, during the supervised learning, we propose to mask tokens that exhibit high entropy or high KL divergence, allowing the model to activate the task-relevant knowledge while preserving its exploration capacity and proximity to the pretrained distribution. A formal token-level gradient analysis is provided in Appendix E.

To avoid setting specific thresholds for determining the category of a token, we introduce a top-k ratio strategy, which ranks tokens within a given set. Given a policy model π_θ, a reference model π_{ref}, the Top-K ratio ρ ($\rho \in (0, 1]$) and sequential tokens $T = \{t_1, t_2, ..., t_{|T|}\}$, we can easily calculate the token-level entropy at the m-th token t_m:

$$H(t_m) = -\sum_{v \in \mathcal{V}} \pi_\theta(v|x, y_{<t}) \log \pi_\theta(v|x, y_{<t}), \tag{4}$$

where $\mathcal{V}$ denotes the vocabulary of the policy model. Then, we obtain all tokens entropy in T:

$$H(T) = \left\{H(t_1), H(t_2), ..., H(t_{|T|})\right\}. \tag{5}$$

And the rank of the m-th token by entropy can be formulated as:

$$r(t_m, H(T)) = \sum_{i=1}^{|T|} \mathbf{I}\left[H(t_i) \geq H(t_m)\right], \tag{6}$$

where $\mathbf{I}$ is the indicator function. Subsequently, we select the top-k ratio tokens, defined as follows:

$$\text{Top-K}(H(T), \rho) = \{t_i | r(t_i, H(T)) \leq k\}, \tag{7}$$

where $k = \lceil \rho \cdot |T| \rceil$ denotes the number of selected elements.

Beyond entropy, we found that tokens with large KL divergence have a pronounced effect on the drift of model parameters. Similar to token-level entropy computation, the KL divergence at the m-th token t_m can be calculated as:

$$\text{KL}(t_m) = \sum_{v \in \mathcal{V}} \pi_\theta(v|x, y_{<t}) \log \frac{\pi_\theta(v|x, y_{<t})}{\pi_{\text{ref}}(v|x, y_{<t})}. \tag{8}$$

Then, all tokens KL divergence in T are as follows:

$$\text{KL}(T) = \left\{\text{KL}(t_1), \text{KL}(t_2), ..., \text{KL}(t_{|T|})\right\}. \tag{9}$$

Following the same steps in Equation 6 and Equation 7, we can identify the tokens with the highest Top-K ratio among the KL divergence tokens, given by:

$$\text{Top-K}(\text{KL}(T), \rho) = \{t_i \,|\, r(t_i, \text{KL}(T)) \leq k\}. \tag{10}$$

Combining Entropy and KL Selection. Given the two Top-K sets, we adopt a *union* strategy to construct the final selected token set:

$$\begin{aligned} \mathcal{M}_{\text{KL}} &= \text{Top-K}(\text{KL}(T), \rho); \\ \mathcal{M}_H &= \text{Top-K}(H(T), \rho); \\ \mathcal{M} &= \mathcal{M}_{\text{KL}} \cup \mathcal{M}_H. \end{aligned} \tag{11}$$

This union covers tokens that are either high entropy or high KL, ensuring that our selective masking mitigates both entropy collapse and distributional shift during supervised learning. The two sets are largely complementary (average IoU ≈ 0.50); see Appendix F for details.

4.2 Regularization Objectives

KL-divergence Regularization. To minimize the fine-tuned model and the base policy model, it is common for a training paradigm to add the KL-divergence regularization in the total loss function. Although we have masked the high entropy and KL tokens during the training phase, we found that these tokens still suffer from entropy decline and KL increase by the gradient backpropagation in practice. Therefore, following prior work, we employ a KL-divergence regularization on the masked tokens to prevent undesirable drift, without altering the core design of EKSFT. The masked KL-divergence regularization is defined as:

$$\mathcal{L}_{\text{KL}}^{\text{mask}}(\theta) = \frac{1}{|\mathcal{M}|} \sum_{t \in \mathcal{M}} \text{KL}\big(\pi_\theta(\cdot \mid x, y_{<t}) \,\big\|\, \pi_{\text{ref}}(\cdot \mid x, y_{<t})\big) \tag{12}$$

where t, π_θ and π_{ref} represent the masked token in $\mathcal{M}$, the policy model and the base model.

Entropy Regularization. Aligning with the intuition behind KL regularization, to mitigate the entropy collapse on the fine-tuned model, EKSFT also introduces an entropy regularization on the

Algorithm 1: EKSFT: Entropy-KL Selective Fine-Tuning

Input: Policy model π_θ, Reference model π_{ref}, Dataset $\mathcal{D}$, Top-K ratio $\rho \in (0, 1]$, Coefficients $\lambda_H, \lambda_{\text{KL}}$, Learning rate η

for *each epoch* **do**

  Sample a batch $\mathcal{B} = \{(x_i, y_i)\}_{i=1}^{B} \sim \mathcal{D}$;

  Collect all valid tokens $T \leftarrow \{t_1, t_2, \ldots, t_{|T|}\}$;

  for *each token* $t \in T$ **do**

    $H(t) \leftarrow -\sum_{v \in \mathcal{V}} \pi_\theta(v|x, y_{<t}) \log \pi_\theta(v|x, y_{<t})$;

    $\text{KL}(t) \leftarrow \sum_{v \in \mathcal{V}} \pi_\theta(v|x, y_{<t}) \log \frac{\pi_\theta(v|x, y_{<t})}{\pi_{\text{ref}}(v|x, y_{<t})}$;

  Determine masked token number: $k \leftarrow \lceil \rho \cdot |T| \rceil$;

  $\mathcal{M}_H \leftarrow \text{Top-K}(H(T), \rho)$; $\mathcal{M}_{\text{KL}} \leftarrow \text{Top-K}(\text{KL}(T), \rho)$;

  Union masked token set: $\mathcal{M} \leftarrow \mathcal{M}_H \cup \mathcal{M}_{\text{KL}}$;

  Compute masked cross-entropy loss: $\mathcal{L}_{\text{CE}}^{\text{mask}} \leftarrow -\frac{1}{|\overline{\mathcal{M}}|} \sum_{t \notin \mathcal{M}} \log \pi_\theta(y_t|x_{<t})$;

  Compute masked entropy regularization: $\mathcal{L}_H^{\text{mask}} \leftarrow \frac{1}{|\mathcal{M}|} \sum_{t \in \mathcal{M}} H(\pi_\theta(\cdot|x, y_{<t}))$;

  Compute masked KL regularization: $\mathcal{L}_{\text{KL}}^{\text{mask}} \leftarrow \frac{1}{|\mathcal{M}|} \sum_{t \in \mathcal{M}} \text{KL}(\pi_\theta(\cdot|x, y_{<t}) \| \pi_{\text{ref}}(\cdot|x, y_{<t}))$;

  Combine objectives: $\mathcal{L}_{\text{EKSFT}} \leftarrow \mathcal{L}_{\text{CE}}^{\text{mask}} - \lambda_H \mathcal{L}_H^{\text{mask}} + \lambda_{\text{KL}} \mathcal{L}_{\text{KL}}^{\text{mask}}$;

  $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}_{\text{EKSFT}}$;

Output: Fine-tuned policy model π_θ

masked tokens to control the entropy decline. This entropy regularization can be formulated as:

$$\mathcal{L}_H^{\text{mask}}(\theta) = \frac{1}{|\mathcal{M}|} \sum_{t \in \mathcal{M}} H(\pi_\theta(\cdot|x, y_{<t})). \quad (13)$$

4.3 Entropy-KL Selective Fine-Tuning

The key insight of Entropy-KL Selective Fine-Tuning (EKSFT) is to activate the task-relevant knowledge while preserving the generalization of the fine-tuned model. To achieve this goal, EKSFT modifies the standard Cross-Entropy loss by masking Top-K of the high entropy and high KL-divergence tokens. The main loss function can be reformulated as:

$$\mathcal{L}_{\text{CE}}^{\text{mask}}(\theta) = -\frac{1}{|\overline{\mathcal{M}}|} \sum_{t \notin \mathcal{M}} \log \pi_\theta(y_t \mid x_{<t}), \quad (14)$$

where $\overline{\mathcal{M}}$ denotes the complement of $\mathcal{M}$.

In addition, to further control these masked tokens' entropy and KL divergence, we supply two weighted regularization: **Entropy Regularization** and **KL Regularization**. Specifically, we define the EKSFT loss by combining the Equation 12, Equation 13 and Equation 14 as:

$$\mathcal{L}_{\text{EKSFT}} = \mathcal{L}_{\text{CE}}^{\text{mask}} - \lambda_H \mathcal{L}_H^{\text{mask}} + \lambda_{\text{KL}} \mathcal{L}_{\text{KL}}^{\text{mask}}, \quad (15)$$

where λ_H and λ_{KL} represent the hyperparameter weight of entropy regularization and KL regularization, respectively. Notably, on masked tokens this yields a label-free gradient that drops the one-hot term $-\nabla \log \pi_\theta(y_t)$ of standard SFT, mitigating entropy collapse and drift without negating capability activation. The derivation is given in Appendix E.

4.4 Implementation details

As shown in Algorithm 1, EKSFT first calculates the token-level entropy and KL divergence in a batch $\mathcal{B}$ by using the Equation 2 and Equation 3. And then we determine the current quantity k of masked tokens by multiplying the Top-K ratio ρ and the number of the valid tokens, which do not include system prompt tokens. Subsequently, we can obtain two index sets $\mathcal{M}_H$ and $\mathcal{M}_{\text{KL}}$ by taking the Top-K operation for all token-level entropy and KL divergence, respectively. The final mask set $\mathcal{M}$ aggregates tokens that satisfy either high entropy or high KL divergence. Based on this mask set, we can compute a masked cross-entropy loss on the complement positions $\overline{\mathcal{M}}$ by applying the Equation 14, together with entropy and KL regularization evaluated on the masked positions by individually using Equation 13 and Equation 12. By combining the masked loss with two weighted regularization, we can obtain the final training objective for EKSFT.

5 Experiments

5.1 Experiment Setup

Training Datasets. We performed all training methods based on OpenR1-Math-46k-8192 (Yan et al., 2025), which contains 46k prompts and off-policy reasoning traces by filtering out the wrong and longer than 8192 tokens' generations in OpenR1-Math-220k (Hugging Face, 2025). In our setting, we sample 3k samples for the SFT and its variants and the remaining 43k samples for the subsequent RL phase, ensuring no overlap.

Method	AIME		AIME25		AMC		HMMT25		Avg	
	pass@1	pass@32	pass@1	pass@32	pass@1	pass@32	pass@1	pass@32	pass@1	pass@32
Qwen3-4B										
Base	25.6	60.0	21.3	46.7	60.5	86.7	11.5	30.0	29.7	55.8
SFT	**46.3**	**73.3**	32.1	56.7	68.7	86.7	16.9	36.7	41.0	63.3
DFT	36.5	70.0	29.5	56.7	63.1	85.5	12.7	26.7	35.5	59.7
IW-SFT	45.3	**73.3**	31.9	56.7	**69.2**	89.1	16.8	**50.0**	40.8	67.2
PSFT	40.9	66.7	29.8	46.7	68.2	83.1	15.5	33.3	38.6	57.4
EKSFT (Ours)	45.7	**73.3**	**33.4**	**60.0**	68.9	**90.4**	**18.5**	**50.0**	**41.7**	**68.4**
Qwen3-8B										
Base	27.1	60.0	22.4	53.3	62.0	89.2	13.1	40.0	31.2	60.6
SFT	41.4	**83.3**	27.8	56.7	70.4	**91.6**	16.9	**50.0**	39.1	70.4
DFT	37.3	73.3	29.7	53.3	63.7	83.1	14.2	40.0	36.2	62.4
IW-SFT	43.9	76.7	32.3	60.0	70.4	86.7	17.4	**50.0**	41.0	68.4
PSFT	29.0	66.7	20.5	53.3	62.1	88.0	12.9	36.7	31.1	61.2
EKSFT (Ours)	**47.2**	80.0	**34.8**	**66.7**	**70.8**	90.4	**19.6**	**50.0**	**43.1**	**71.8**

Table 1: **Overall performance comparison (Stage 1: without RL).** We compared the pass@1 and pass@32 performance between different methods using the Qwen3-4B and Qwen3-8B models on four mathematical reasoning benchmarks.

Models. We conducted fine-tuning experiments using two models from Qwen3 family (Yang et al., 2025), Qwen3-4B and Qwen3-8B. Both models have demonstrated robust capabilities (Zhao et al., 2025; Xu et al., 2025b; Zhang et al., 2025e; Lee et al., 2025) across various tasks, showing strong performance whether fine-tuned or used directly.

Evaluation benchmarks. We conducted a comprehensive evaluation on several widely-adopted mathematical reasoning benchmarks, including AIME24 (Li et al., 2024), AIME25 (Li et al., 2024), AMC (Li et al., 2024) and HMMT25 (Balunović et al., 2025). We reported pass@1 to evaluate performance and pass@32 to evaluate the boundaries of LLM capabilities (Yue et al., 2025; Chen et al., 2025d; Brown et al., 2024).

Baselines Methods. We compared the proposed EKSFT with a comprehensive set of baselines, including three categories: base model (without training), SFT-variants, and combined SFT with RL methods. For the base model, we consider the Original Model, which is the original Qwen3-4B/Qwen3-8B model. For SFT-variants methods, we consider (1) Standard SFT trains the base model using the standard CE loss function on the SFT dataset. (2) PSFT (Zhu et al., 2025b) is a trust-region–inspired supervised fine-tuning objective that clips the policy ratio to constrain policy drift. For combined SFT with RL methods, we consider (1) IW-SFT (Qin and Springenberg, 2025) is an importance-weighted variant that interprets SFT on curated data as a lower bound on a sparse-reward RL objective. (2) DFT (Wu et al., 2025) rescales the SFT objective by using a simple probability-based reweighting mechanism.

Training Details. For a fair comparison, all training methods use the same experimental data and hyperparameter settings. To further compare the cold-start performance of different methods, we initialize the second-stage reinforcement learning with models trained by each method under uniform configurations. All hyperparameters remain consistent during RL training. Details are in Appendix A.

5.2 Main Results

Our main results can be divided into two parts according to the training stages. In the first stage, we performed supervised learning on the 3k dataset. In the second stage, starting from the supervised fine-tuned models, we further continued training with reinforcement learning on the remaining 43k dataset. The results from the two stages are summarized in Table 1 and Table 2, respectively.

Performance after the supervised learning. Across both model scales, **EKSFT** generally achieves higher performance (pass@1) and stronger exploration capability (pass@32) than baselines. For the **Qwen3-4B model**, compared to the standard SFT, EKSFT yields an average **+0.7%** improvement in pass@1 and **+5.1%** improvement in pass@32, which demonstrates that the masking mechanism of high entropy and high KL divergence tokens in EKSFT not only enhances its downstream performance (pass@1), but also improves its exploratory capability (pass@32) on downstream tasks. In addition, compared to the competitive baseline IW-SFT, EKSFT still yields an average **+0.9%** improvement in pass@1 and

Method(+DAPO)	AIME		AIME25		AMC		HMMT25		Avg	
	pass@1	pass@32	pass@1	pass@32	pass@1	pass@32	pass@1	pass@32	pass@1	pass@32
Qwen3-4B										
Base	43.4	73.3	37.6	66.7	74.9	91.6	17.8	33.3	43.4	66.2
SFT	48.1	80.0	39.5	66.7	76.4	94.0	19.4	40.0	45.8	70.2
DFT	39.3	73.3	31.5	63.3	72.1	90.4	13.7	33.3	39.1	65.1
IW-SFT	46.0	66.7	36.3	63.3	76.2	91.6	**21.4**	**50.0**	45.0	67.9
PSFT	46.3	76.7	39.3	60.0	**78.2**	90.3	20.0	46.7	45.9	68.4
EKSFT (Ours)	**48.3**	**83.3**	**41.6**	**73.3**	**78.2**	**96.4**	**21.4**	**50.0**	**47.4**	**75.8**
Qwen3-8B										
Base	50.6	73.3	37.4	63.3	78.9	90.4	19.6	**53.3**	46.6	70.1
SFT	52.6	80.0	40.0	66.7	80.5	93.9	20.0	50.0	48.3	72.7
DFT	39.5	76.7	31.3	63.3	71.7	90.4	16.9	33.3	39.9	65.9
IW-SFT	54.6	**83.3**	41.0	73.3	80.8	94.0	20.0	50.0	49.1	75.1
PSFT	52.8	73.3	38.9	63.3	80.8	91.6	18.6	50.0	47.8	69.6
EKSFT (Ours)	**56.5**	**83.3**	**42.5**	**83.3**	**82.1**	**96.4**	**23.3**	50.0	**51.1**	**78.3**

Table 2: **Overall performance comparison (Stage 2: after RL).** We compared the pass@1 and pass@32 between different methods using the Qwen3-4B and Qwen3-8B models on four mathematical reasoning benchmarks.

+1.2% improvement in pass@32. A similar trend is observed for the **Qwen3-8B model**, where EKSFT reaches the highest average scores: **43.1%** in pass@1 and **71.8%** in pass@32. The best baselines for both pass@1 and pass@32 are IW-SFT and standard SFT. EKSFT still outperforms IW-SFT by **+2.1%** in pass@1 and outperforms standard SFT by **+1.4%** in pass@32.

We found that EKSFT gains more improvement on HMMT25, which is a comprehensive mathematical challenging benchmark, suggesting that avoiding token-level imitation on the high entropy and high KL divergence tokens is essential for supervised learning on the complex downstream tasks. Overall, the results indicate that EKSFT can enhance the reasoning performance and exploration diversity, showing that EKSFT provides an effective cold start for the subsequent RL phase. In addition, we also compared the parameters drift by calculating the changes in model parameters before and after training, shown in Appendix C.

Performance after the RL training. To verify the effectiveness of different methods as the cold start point of RL, we further continued training with RL on the remaining 43k dataset. As shown in Table 2, EKSFT+DAPO still achieves the best overall performance under both Qwen3-4B and Qwen3-8B, consistently leading on Avg pass@1 and Avg pass@32. For the **Qwen3-4B model**, EKSFT reaches **47.4%** in pass@1 and **75.8%** in pass@32, outperforming PSFT by **1.5%** in pass@1 and SFT by **5.6%** in pass@32. For the **Qwen3-8B model**, EKSFT reaches **51.1%** in pass@1 and **78.3%** in pass@32, outperforming IW-SFT by **2.0%** in pass@1 and by **3.2%** in pass@32. In detail, we observed a positive correlation between pass@1 and pass@32 during the RL training phase: models with higher pass@32 consistently achieve higher pass@1. For instance, when using Qwen3-4B as the base model, EKSFT+DAPO and SFT+DAPO achieve the best and second-best pass@32 scores, and correspondingly, they also attain the best and second-best pass@1 scores. The same trend holds for the Qwen3-8B model.

Therefore, as a cold start point of RL, a key challenge is to **enhance the model's exploratory ability** while maintaining its accuracy, which is crucial for improving the subsequent RL phase. By masking tokens with high uncertainty and those that significantly deviate from the model during supervised fine-tuning, our approach EKSFT simultaneously enhances the model's downstream performance and its exploratory capability. Consequently, it achieves superior results in the subsequent reinforcement learning stage compared to other baselines. Beyond mathematical reasoning, we validated EKSFT on the tool-use benchmark BFCL (Patil et al., 2025) after training on AceBench (Chen et al., 2025a), where EKSFT consistently outperforms SFT and DFT on both Qwen3-4B and Qwen3-8B; full results are in Appendix I.

Details pass@k analysis. To examine the exploration boundary of different supervised learning methods, we increased k in the pass@k metric. We chose AIME25 as the evaluation dataset and reported the trend between different approaches on Qwen3-8B by extending k up to 320.

As shown in Figure 2, EKSFT consistently achieved higher pass@k performance when k is greater than 32, indicating stronger exploration and reasoning diversity. In addition, we found

Model	AIME		AIME25		AMC		HMMT25	
	pass@1	pass@32	pass@1	pass@32	pass@1	pass@32	pass@1	pass@32
Qwen3-4B								
EKSFT	**45.7**	**73.3**	**33.4**	**60.0**	**68.9**	**90.4**	**18.5**	**50.0**
w/o Entropy Regularization	42.6	66.7	31.1	53.3	68.7	86.7	15.8	36.7
w/o KL Regularization	41.9	70.0	30.8	56.7	65.2	86.7	16.5	46.7
w/o Entropy & KL Regularization	41.7	70.0	31.6	50.0	67.8	86.7	16.4	43.3
Qwen3-8B								
EKSFT	**47.2**	**80.0**	**34.8**	**66.7**	**70.8**	**90.4**	**19.6**	**50.0**
w/o Entropy Regularization	45.4	76.7	32.1	53.3	69.2	88.0	16.3	40.0
w/o KL Regularization	42.2	73.3	32.3	56.7	67.6	86.7	18.7	50.0
w/o Entropy & KL Regularization	45.0	73.3	32.3	60.0	70.2	87.9	17.7	46.7

Table 3: Ablation study of different regularization.

that EKSFT exhibits a steeper performance curve, achieving higher pass@k faster, showing that our approach increases the sampling probability of the correct solution under constrained budgets ($k \leq 32$). In the high-k regime, EKSFT could reach a higher performance than all baselines, indicating a richer and more effective distribution of correct solutions. More results are shown in Appendix B.1.

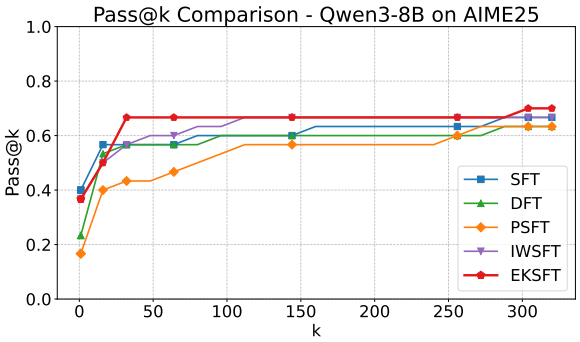

Figure 2: We compared the pass@k on AIME25 based on Qwen3-8B model.

5.3 Ablation Study

In this section, we performed an ablation study to examine the contributions of EKSFT components, including entropy regularization and KL-divergence regularization as summarized in Table 3. Overall, each regularization is important, but they contribute in different ways. **Removing entropy regularization causes the largest degradation on pass@32**, with the most dramatic drop appearing on HMMT25: **-13.3%** on Qwen3-4B and **-10.0%** on Qwen3-8B, which is consistent that the model may produce more repetitive but wrong path, reducing the chance that multiple samples explore diverse solution paths with the decline of entropy. In contrast, **removing KL-divergence degrades the pass@1**, especially on AIME: **-3.8%** on Qwen3-4B and **-5.0%** on Qwen3-8B, indicating that KL-divergence acts like an anchor term, which stabilizes training and preserves the useful reasoning patterns learned during supervised learning, improving pass@1 correctness. Interestingly, removing both regularization techniques is not uniformly the worst across all datasets, suggesting that entropy regularization and KL-divergence regularization are not simply additive, trading off exploration and constraint during training. In addition, we further compared EKSFT with a global-regularized SFT baseline and found that EKSFT consistently performs better across all benchmarks, highlighting the importance of the selective masking mechanism. Detailed results are provided in Appendix B.2. To further verify that the gain stems from *selectivity* rather than removing supervision, we included a random-masking baseline in Appendix H, and analyzed the sensitivity of EKSFT to the masking ratio ρ in Appendix G.

6 Conclusions

In this paper, we propose Entropy-KL Selective Fine-Tuning (EKSFT), a simple yet effective modification to supervised fine-tuning for the widely used SFT-then-RL post-training paradigm, which masks high entropy or high KL-divergence tokens from token-level imitation in the cross-entropy objective, mitigating entropy collapse or distributional drift during supervised learning. EKSFT also applies entropy and KL regularization on these masked tokens to maintain diversity and stay close to the pretrained distribution, while activating the task-relevant knowledge from the remaining tokens. Extensive experiments on mathematical reasoning benchmarks with Qwen3-4B/8B demonstrate that EKSFT consistently outperforms standard SFT and strong SFT variants, and also yields stronger post-RL performance when used as the initialization for the subsequent RL stage, suggesting that the masking mechanism in our approach provides a more robust and exploration-friendly initialization for reinforcement learning.

Limitations

Lack of attempts on more datasets. In our experiment setting, we only train the policy model on OpenR1-Math-46k-8192, lacking consideration for the impact of different training datasets, such as the DeepscaleR dataset.

Lack of attempts on more model scales. Due to high computational cost, our experiments are only trained on Qwen3-4B and Qwen-8B models. Future work should focus on different backbones, such as the LLaMA series or model scales larger than 32B parameters.

Ethics Statement

We affirm our commitment to the ethical conduct of this research and provide the following assurances:

- This research was conducted in strict adherence to the highest ethical standards, and all findings have been reported with integrity, ensuring clarity and accuracy throughout our communications.
- Our study strictly avoids the use of sensitive or confidential data, ensuring that all materials are appropriate for public dissemination.
- The datasets employed in our experiments are sourced from publicly accessible and peer-reviewed scientific resources, insuring transparency and reliability.
- We provided a detailed description of the dataset characteristics and the hyper-parameter settings used in our experiments to maintain transparency and consistency with our results.
- To promote transparency and facilitate further research, we commit to sharing our code on anonymous GitHub now and will open source after our paper is accepted.

References

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, and 12 others. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *CoRR*, abs/2204.05862.

Mislav Balunović, Jasper Dekoninck, Ivo Petrov, Nikola Jovanović, and Martin Vechev. 2025. Matharena: Evaluating llms on uncontaminated math competitions.

Bradley C. A. Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. 2024. Large language monkeys: Scaling inference compute with repeated sampling. *CoRR*, abs/2407.21787.

Chen Chen, Xinlong Hao, Weiwen Liu, Xu Huang, Xingshan Zeng, Shuai Yu, Dexun Li, Yuefeng Huang, Xiangcheng Liu, Wang Xinzhi, and Wu Liu. 2025a. ACEBench: A comprehensive evaluation of LLM tool usage. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 12970–12998, Suzhou, China. Association for Computational Linguistics.

Hardy Chen, Haoqin Tu, Fali Wang, Hui Liu, Xianfeng Tang, Xinya Du, Yuyin Zhou, and Cihang Xie. 2025b. SFT or rl? an early investigation into training r1-like reasoning large vision-language models. *Trans. Mach. Learn. Res.*, 2025.

Liang Chen, Xueting Han, Li Shen, Jing Bai, and Kam-Fai Wong. 2025c. Beyond two-stage training: Cooperative SFT and RL for LLM reasoning. *CoRR*, abs/2509.06948.

Zhipeng Chen, Xiaobo Qin, Youbin Wu, Yue Ling, Qinghao Ye, Wayne Xin Zhao, and Guang Shi. 2025d. Pass@ k training for adaptively balancing exploration and exploitation of large reasoning models. *arXiv preprint arXiv:2508.10751*.

Daixuan Cheng, Shaohan Huang, Xuekai Zhu, Bo Dai, Wayne Xin Zhao, Zhenliang Zhang, and Furu Wei. 2025. Reasoning with exploration: An entropy perspective. *CoRR*, abs/2506.14758.

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V. Le, Sergey Levine, and Yi Ma. 2025. SFT memorizes, RL generalizes: A comparative study of foundation model post-training. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*. OpenReview.net.

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, Zhiyuan Liu, Hao Peng, Lei Bai, Wanli Ouyang, Yu Cheng, Bowen Zhou, and Ning Ding. 2025. The entropy mechanism of reinforcement learning for reasoning language models. *CoRR*, abs/2505.22617.

Alan Dao and Dinh Bach Vu. 2025. Alphamaze: Enhancing large language models' spatial intelligence via grpo. *arXiv preprint arXiv:2502.14669*.

DeepSeek-AI. 2024. Deepseek-v3 technical report. *CoRR*, abs/2412.19437.

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. *CoRR*, abs/2506.19767.

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 incentivizes reasoning in llms through reinforcement learning. *Nat.*, 645(8081):633–638.

Lixuan He, Jie Feng, and Yong Li. 2025. Amft: Aligning llm reasoners by meta-learning the optimal imitation-exploration balance. *arXiv preprint arXiv:2508.06944*.

Hugging Face. 2025. Open r1: A fully open reproduction of deepseek-r1.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, and 4 others. 2024. Tülu 3: Pushing frontiers in open language model post-training. *CoRR*, abs/2411.15124.

Jungyup Lee, Jemin Kim, Sang Park, and SeungJae Lee. 2025. Making qwen3 think in korean with reinforcement learning. *CoRR*, abs/2508.10355.

Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Huang, Kashif Rasul, Longhui Yu, Albert Q Jiang, Ziju Shen, and 1 others. 2024. Numinamath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions. *Hugging Face repository*, 13(9):9.

Ziniu Li, Congliang Chen, Tian Xu, Zeyu Qin, Jiancong Xiao, Zhi-Quan Luo, and Ruoyu Sun. 2025. Preserving diversity in supervised fine-tuning of large language models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Mingyang Liu, Gabriele Farina, and Asuman E. Ozdaglar. 2025a. UFT: unifying supervised and reinforcement fine-tuning. *CoRR*, abs/2505.16984.

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. *CoRR*, abs/2506.13284.

Ilya Loshchilov and Frank Hutter. 2017. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Bo Pang, Yalu Ouyang, Hangfei Xu, Ziqi Jia, Panpan Li, Shengzhao Wen, Lu Wang, Shiyong Li, and Yanpeng Wang. 2025. Fevo: Financial knowledge expansion and reasoning evolution for large language models. *arXiv preprint arXiv:2507.06057*.

Shishir G Patil, Huanzhi Mao, Fanjia Yan, Charlie Cheng-Jie Ji, Vishnu Suresh, Ion Stoica, and Joseph E. Gonzalez. 2025. The berkeley function calling leaderboard (BFCL): From tool use to agentic evaluation of large language models. In *Forty-second International Conference on Machine Learning*.

Jake Poznanski, Luca Soldaini, and Kyle Lo. 2025. olmocr 2: Unit test rewards for document OCR. *CoRR*, abs/2510.19817.

Chongli Qin and Jost Tobias Springenberg. 2025. Supervised fine tuning on curated data is reinforcement learning (and can be improved). *arXiv preprint arXiv:2507.12856*.

Reuven Y Rubinstein and Dirk P Kroese. 2016. *Simulation and the Monte Carlo method*. John Wiley & Sons.

John Schulman, Sergey Levine, Pieter Abbeel, Michael I. Jordan, and Philipp Moritz. 2015. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 1889–1897. JMLR.org.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2025. Hybridflow: A flexible and efficient RLHF framework. In *Proceedings of the Twentieth European Conference on Computer Systems, EuroSys 2025, Rotterdam, The Netherlands, 30 March 2025 - 3 April 2025*, pages 1279–1297. ACM.

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. *CoRR*, abs/2508.05629.

Lechao Xiao. 2024. Rethinking conventional wisdom in machine learning: From generalization to scaling. *CoRR*, abs/2409.15156.

Chulin Xie, Yangsibo Huang, Chiyuan Zhang, Da Yu, Xinyun Chen, Bill Yuchen Lin, Bo Li, Badih Ghazi, and Ravi Kumar. 2024. On memorization of large language models in logical reasoning. *CoRR*, abs/2410.23123.

Derong Xu, Xinhang Li, Ziheng Zhang, Zhenxi Lin, Zhihong Zhu, Zhi Zheng, Xian Wu, Xiangyu Zhao, Tong Xu, and Enhong Chen. 2025a. Harnessing large language models for knowledge graph question answering via adaptive multi-aspect retrieval-augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 25570–25578.

Derong Xu, Yi Wen, Pengyue Jia, Yingyi Zhang, Wenlin Zhang, Yichao Wang, Huifeng Guo, Ruiming Tang, Xiangyu Zhao, Enhong Chen, and Tong Xu. 2026. From single to multi-granularity: Toward long-term memory association and selection of conversational agents. In *The Fourteenth International Conference on Learning Representations*.

Jin Xu, Zhifang Guo, Hangrui Hu, Yunfei Chu, Xiong Wang, Jinzheng He, Yuxuan Wang, Xian Shi, Ting He, Xinfa Zhu, Yuanjun Lv, Yongqi Wang, Dake Guo, He Wang, Linhan Ma, Pei Zhang, Xinyu Zhang, Hongkun Hao, Zishan Guo, and 19 others. 2025b. Qwen3-omni technical report. *CoRR*, abs/2509.17765.

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. *CoRR*, abs/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 40 others. 2025. Qwen3 technical report. *CoRR*, abs/2505.09388.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, and 16 others. 2025. DAPO: an open-source LLM reinforcement learning system at scale. *CoRR*, abs/2503.14476.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. 2025. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *CoRR*, abs/2504.13837.

Kaiyan Zhang, Yuxin Zuo, Bingxiang He, Youbang Sun, Runze Liu, Che Jiang, Yuchen Fan, Kai Tian, Guoli Jia, Pengfei Li, Yu Fu, Xingtai Lv, Yuchen Zhang, Sihang Zeng, Shang Qu, Haozhan Li, Shijie Wang, Yuru Wang, Xinwei Long, and 20 others. 2025a. A survey of reinforcement learning for large reasoning models. *CoRR*, abs/2509.08827.

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

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

Xiaoyun Zhang, Xiaojian Yuan, Di Huang, Wang You, Chen Hu, Jingqing Ruan, Kejiang Chen, and Xing Hu. 2025d. Rediscovering entropy regularization: Adaptive coefficient unlocks its potential for LLM reinforcement learning. *CoRR*, abs/2510.10959.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025e. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *CoRR*, abs/2506.05176.

Haiquan Zhao, Chenhan Yuan, Fei Huang, Xiaomeng Hu, Yichang Zhang, An Yang, Bowen Yu, Dayiheng Liu, Jingren Zhou, Junyang Lin, Baosong Yang, Chen Cheng, Jialong Tang, Jiandong Jiang, Jianwei Zhang, Jijie Xu, Ming Yan, Minmin Sun, Pei Zhang, and 24 others. 2025. Qwen3guard technical report. *CoRR*, abs/2510.14276.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand. Association for Computational Linguistics.

Weihai Zhi, Jiayan Guo, and Shangyang Li. 2025. Medgr^2: Breaking the data barrier for medical reasoning via generative reward learning. *CoRR*, abs/2508.20549.

He Zhu, Junyou Su, Peng Lai, Ren Ma, Wenjia Zhang, Linyi Yang, and Guanhua Chen. 2025a. Anchored supervised fine-tuning. *CoRR*, abs/2509.23753.

Wenhong Zhu, Ruobing Xie, Rui Wang, Xingwu Sun, Di Wang, and Pengfei Liu. 2025b. Proximal supervised fine-tuning. *CoRR*, abs/2508.17784.

A Implementation Details

A.1 Training Hyperparameters

Supervise learning phase. For all compared methods, we kept the same hyperparameters. We set learning rate $\eta = 1\text{e}{-5}$, a maximum of 8 epochs, the optimizer AdamW (Loshchilov and Hutter, 2017) with $\beta_1 = 0.9$ and $\beta_2 = 0.95$ and the cutoff length of 20000 tokens. Table 5 shows the detailed information on method-specific hyperparameters for baselines.

Hyperparameter	DAPO
Learning Rate	1×10^{-6}
Sequence Length	8k
Train Batch Size	256
PPO Mini-batch Size	32
Total Steps	200
Rollout Numbers	16
Others	$c_l = 0.2,\ c_h = 0.28$

Table 4: Training hyperparameters for RL training.

Reinforcement learning phase. We selected the last checkpoint of all the supervised models as the cold start point for RL. For the subsequent RL phase, we employed the same hyperparameters and the same reinforcement learning algorithm *DAPO* (Yu et al., 2025). The details are shown in Table 4.

Training set up. We train all the methods with **8 NVIDIA H20 SXM GPUs**. For the supervise learning phase, we set *gradient accumulation steps* to 8, *per device train batch size* to 1, and *bfloat16* used. We employ the SFT and EKSFT algorithms based on LLaMA-Factory (Zheng et al., 2024). For DFT, PSFT and IW-SFT, we use the architectures each researcher provided (Wu et al., 2025; Zhu et al., 2025b; Qin and Springenberg, 2025). During the subsequent RL phase, we implement DAPO algorithms based on Verl (Sheng et al., 2025).

A.2 Inference Hyperparameters

During the evaluation, we adopt that the max response length is set to 8k, and *temperature* is set to 1.0. To avoid high variance in results and ensure fair comparisons, we report avg@32, i.e., pass@1, shown in our results, on four benchmarks by sampling 32 times.

Table 5: Training hyperparameters of different models

Method	Hyperparameters
SFT	Learning Rate: $1\text{e}{-5}$ Cutoff Length: 20000 Gradient Accumulation Steps: 8 Batch Size: 1 Epochs: 8 Others: -
DFT	Learning Rate: $1\text{e}{-5}$ Cutoff Length: 20000 Gradient Accumulation Steps: 8 Batch Size: 1 Epochs: 8 Others: -
IW-SFT	Learning Rate: $1\text{e}{-5}$ Cutoff Length: 20000 Gradient Accumulation Steps: 8 Batch Size: 1 Epochs: 8 Others: -
PSFT	Learning Rate: $1\text{e}{-5}$ Cutoff Length: 20000 Gradient Accumulation Steps: 8 Batch Size: 1 Epochs: 8 Others: $kl = 0.0, c_l = 0.2, c_h = 0.28$
EKSFT	Learning Rate: $1\text{e}{-5}$ Cutoff Length: 20000 Gradient Accumulation Steps: 8 Batch Size: 1 Epochs: 8 Others: $\lambda_H = 0.05, \lambda_{KL} = 0.05, \rho = 0.2$

B Results detail

B.1 Pass@k analysis on Qwen3-4B

As shown in Figure 3, we also compared the pass@k on AIME25 by extending the k up to 320 on the Qwen3-4B. EKSFT consistently achieved higher pass@k performance when k is greater than 250, indicating stronger exploration and reasoning diversity.

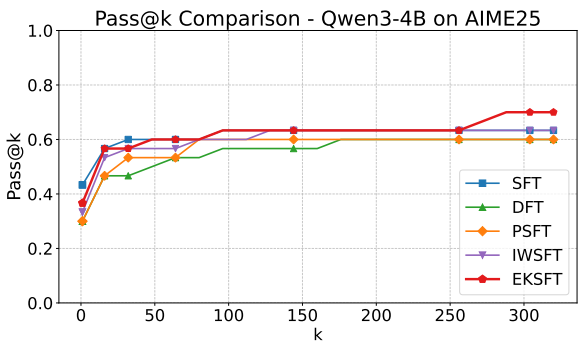

Figure 3: We compared the pass@k on AIME25 based on Qwen3-4B model.

Model	AIME		AIME25		AMC		HMMT25	
	pass@1	pass@32	pass@1	pass@32	pass@1	pass@32	pass@1	pass@32
Qwen3-4B								
EKSFT	**45.7**	**73.3**	**33.4**	**60.0**	**68.9**	**90.4**	**18.5**	**50.0**
SFT w/ Entropy & KL Regularization	41.7	70.0	30.2	53.3	67.7	87.9	17.1	43.3
Qwen3-8B								
EKSFT	**47.2**	**80.0**	**34.8**	**66.7**	**70.8**	**90.4**	**19.6**	**50.0**
SFT w/ Entropy & KL Regularization	44.6	76.7	32.4	56.7	67.5	87.9	17.6	43.3

Table 6: Ablation study of masking mechanism.

B.2 Ablation on the masking mechanism

As shown in Table 6, we compared EKSFT with a global-regularized SFT baseline that applies entropy and KL regularization to all tokens. Across both Qwen3-4B and Qwen3-8B, EKSFT consistently outperforms this baseline on all benchmarks in terms of both pass@1 and pass@32. For the Qwen3-4B, EKSFT achieves **+6.7%** improvement on AIME25 and HMMT25 pass@32, while for the Qwen3-8B, EKSFT still shows notable improvements on AIME25 pass@32 (**+10.0%**) and HMMT25 pass@32 (**+6.7%**). These results suggest that the benefits of EKSFT are not solely due to entropy/KL regularization, but critically, from the selective exclusion of high-entropy or high-KL tokens during token-level imitation, which is crucial for improving both accuracy and exploration.

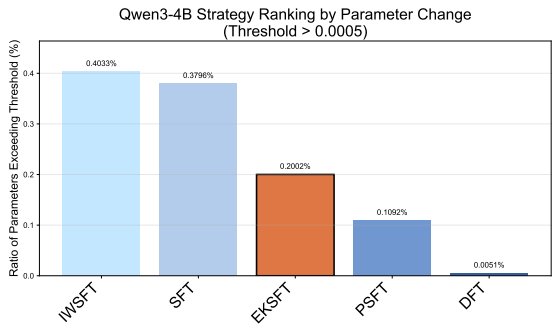

Figure 4: We reported the relative changes of the trained models' parameters, compared to the base model on Qwen3-4B.

C Parameters Drift

Combing Table 1 with Figure 4, we observe that standard SFT and IW-SFT introduce the largest parameter changes, and while they improve over the base model, their gains come with substantial parameters drift that may impair the pretrained distribution and exploration capacity. In contrast, EKSFT achieves the best overall performance on Qwen3-4B, while exhibiting much smaller parameter drift than SFT and IW-SFT. This suggests that EKSFT attains stronger exploration during the subsequent RL phase, as shown in Table 2.

Interestingly, PSFT and DFT yield even smaller parameter changes than EKSFT, yet their performance is worse than EKSFT, indicating that minimizing drift alone is insufficient, since overly conservative updates will fail to elicit task-relevant behaviors. Taken together these four methods, these results support the notion of **effective drift**. Specifically, EKSFT employs the masking mechanism that excludes the high entropy and high KL-divergence tokens from imitation and introducing the entropy regularization and KL regularization for the masked tokens can make task-targeted updates and lead to minimal but purposeful parameter movement.

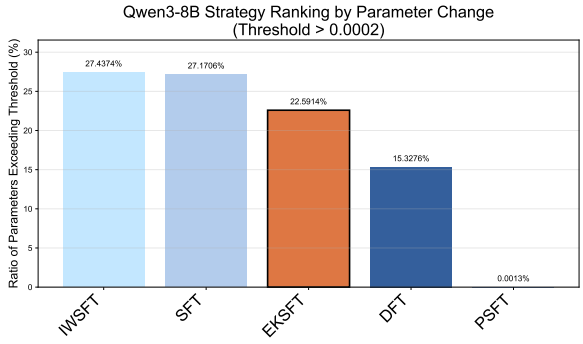

Figure 5: We reported the relative changes of the trained models' parameters, compared to the base model on Qwen3-8B.

For Qwen3-8B, Figure 5 shows a similar drift pattern. Standard SFT and IW-SFT produce the largest parameter changes, whereas EKSFT substantially reduces the fraction of parameters exceeding the change threshold, indicating better preservation of the pretrained distribution. When combined with the results in Table 1, EKSFT also achieves the strongest overall performance on Qwen3-8B, outperforming SFT and IW-SFT despite incurring less drift, validating the effectiveness of EKSFT across both Qwen3-4B and Qwen3-8B.

Overall, combining Table 1, Table 2, Figure 4

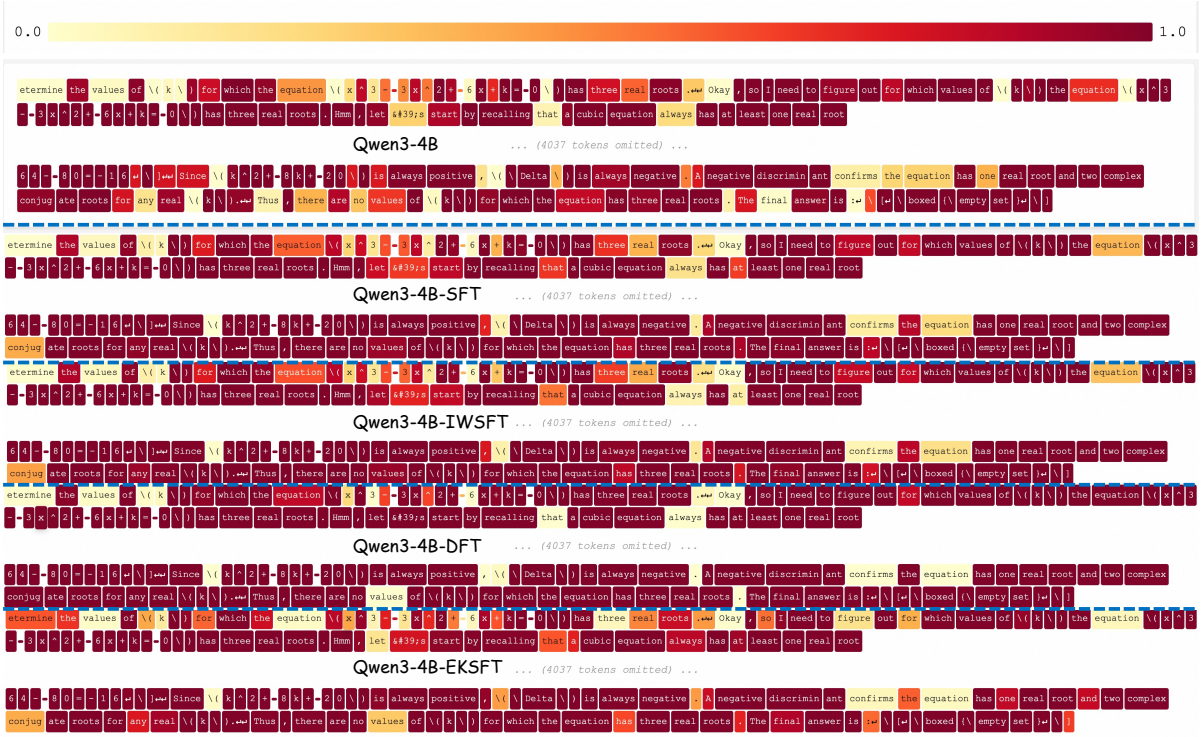

Figure 6: We reported the token probabilities of different methods on the same question with its solution based on Qwen3-4B.

and Figure 5, we find that EKSFT consistently delivers the best or near-best performance in both the supervised learning stage and the subsequent RL stage, while inducing substantially less parameter drift than standard SFT-style training. Meanwhile, PSFT and DFT can reduce drift further but do not improve performance, indicating that minimizing drift alone is insufficient. These results support our motivation that selectively excluding high-entropy and high-KL tokens from imitation enables effective and activates task-relevant knowledge without reducing diversity.

D Case Study

In this section, we reported the token probabilities of different methods when using the same question and its ground truth. Figure 6 illustrates the next-token probability assigned to the ground-truth token along the reasoning trace, where darker red indicates higher probability. Standard SFT (along with IWSFT and DFT) results in a uniform increase in saturation across the sequence, causing most tokens to become darker red, indicating that the model indiscriminately assigns high probability to the demonstrated tokens. This probability overconfidence reflects a sharpened distribution and reduces sampling diversity in the subsequent usage.

In contrast, EKSFT exhibits a structured probability reallocation, which excludes the high entropy and KL-divergence tokens from imitation, avoiding the dark red saturation observed in SFT, preventing mode collapse and maintaining distribution. This aligns with our motivation to activate the task-relevant knowledge in favor of reliable reasoning, ultimately preserving the exploration capability essential for subsequent RL.

E Theoretical Analysis of Selective Masking

In this section we provide a token-level formal analysis that justifies the design of EKSFT: (i) why standard cross-entropy (CE) imitation is most harmful precisely on high-entropy and high-KL tokens, (ii) why replacing CE on the masked set with our entropy/KL regularization yields a label-free, low-variance update, and (iii) why selective regularization on $\mathcal{M}$ is preferable to applying the same regularization globally.

E.1 High-entropy tokens dominate the SFT gradient

For a token position t, standard SFT optimizes

$$\ell_t^{\mathrm{SFT}}(\theta) = -\log \pi_\theta(y_t \mid x_{<t}). \tag{16}$$

Let z_t denote the pre-softmax logits and $\pi_\theta = \mathrm{softmax}(z_t)$. The gradient with respect to the logits is

$$\frac{\partial \ell_t^{\mathrm{SFT}}}{\partial z_t} = \pi_\theta(\cdot \mid x_{<t}) - e_{y_t}, \tag{17}$$

where e_{y_t} is the one-hot indicator at the gold token. Therefore

$$\Big\| \frac{\partial \ell_t^{\mathrm{SFT}}}{\partial z_t} \Big\|_2^2 = \sum_{v \in \mathcal{V}} \pi_\theta(v)^2 + 1 - 2\pi_\theta(y_t). \tag{18}$$

Using $\sum_v \pi_\theta(v)^2 \le 1$, we obtain

$$\Big\| \frac{\partial \ell_t^{\mathrm{SFT}}}{\partial z_t} \Big\|_2^2 \le 2\big(1 - \pi_\theta(y_t)\big). \tag{19}$$

At a high-entropy position, π_θ is close to uniform, so $\pi_\theta(y_t)$ is small and the residual $\|\pi_\theta - e_{y_t}\|$ stays at an $O(1)$ scale. In the uniform limit $\pi_\theta(v) \approx 1/|\mathcal{V}|$,

$$\Big\| \frac{\partial \ell_t^{\mathrm{SFT}}}{\partial z_t} \Big\|_2^2 \approx 1 - \frac{1}{|\mathcal{V}|}. \tag{20}$$

By contrast, at a low-entropy, high-confidence position with $\pi_\theta(y_t) = 1 - \varepsilon$,

$$\|\pi_\theta - e_{y_t}\|_2^2 \approx 2\varepsilon^2, \tag{21}$$

which is much smaller. High-entropy tokens therefore generate disproportionately large CE gradients and tend to dominate parameter updates in the low-data SFT regime, accelerating distribution sharpening and parameter drift. A symmetric argument applies to high-KL tokens: such positions are exactly where π_θ already deviates from π_{ref}, so further label-driven imitation pushes the policy further away from the pretrained distribution. Masking these tokens removes the high-energy, unstable component of the gradient and reallocates the limited update budget to more stable, transferable positions.

E.2 Label-free regularization on the masked set

On the masked set $\mathcal{M}$, instead of $-\log \pi_\theta(y_t)$ we optimize

$$\mathcal{L}_{\mathcal{M}}^{\mathrm{reg}}(\theta) = \sum_{t \in \mathcal{M}} \Big[\lambda_{\mathrm{KL}}\, \mathrm{KL}(\pi_\theta \,\|\, \pi_{\mathrm{ref}}) - \lambda_H\, H(\pi_\theta) \Big]. \tag{22}$$

Expanding $\mathrm{KL}(\pi_\theta \| \pi_{\mathrm{ref}}) = -H(\pi_\theta) + H(\pi_\theta, \pi_{\mathrm{ref}})$ with $H(\pi_\theta, \pi_{\mathrm{ref}}) = -\sum_v \pi_\theta(v) \log \pi_{\mathrm{ref}}(v)$, we obtain

$$\begin{aligned} \mathcal{L}_{\mathcal{M}}^{\mathrm{reg}}(\theta) = \sum_{t \in \mathcal{M}} \Big[& -(\lambda_{\mathrm{KL}} + \lambda_H)\, H(\pi_\theta) \\ & + \lambda_{\mathrm{KL}}\, H(\pi_\theta, \pi_{\mathrm{ref}}) \Big]. \end{aligned} \tag{23}$$

Letting $J_v = \nabla \log \pi_\theta(v)$, the standard SFT gradient at token t can be written as

$$\nabla \ell_t^{\mathrm{SFT}} = \sum_{v \in \mathcal{V}} \pi_\theta(v)\, J_v - J_{y_t}, \tag{24}$$

which contains the explicit one-hot forcing term $-J_{y_t}$. In contrast, the gradient of $\mathcal{L}_{\mathcal{M}}^{\mathrm{reg}}$ at any masked token has the form

$$\nabla R_t = \sum_{v \in \mathcal{V}} a_v(\theta)\, J_v, \tag{25}$$

where the coefficients $a_v(\theta)$ depend only on π_θ, $\log \pi_\theta$, and $\log \pi_{\mathrm{ref}}$. Crucially, no term of the form $-J_{y_t}$ appears: the update is *label-free*. Hence on the most uncertain or already-drifted positions our update no longer contains the high-variance, "exogenous forcing" direction that drives entropy collapse and parameter drift. Instead, it balances two soft forces—staying close to π_{ref} and maintaining higher entropy—scaled linearly by λ_{KL} and λ_H. This is consistent with the smaller drift we observe empirically (Appendix C).

E.3 Why selective regularization beats global regularization

A natural alternative is to apply entropy/KL regularization *globally*, on every token, alongside full CE supervision. We argue this is suboptimal in our setting and our experiments confirm it (Appendix B.2). Decompose the per-token update under global regularization into two parts: a CE part proportional to $\pi_\theta - e_{y_t}$ and a regularization part proportional to ∇R_t. On low-entropy, low-KL tokens, the CE part is small (Appendix E.1) and reliably injects task-relevant signal; adding regularization here mostly *cancels* useful gradient and makes updates overly conservative, particularly damaging in the low-data regime. On high-entropy, high-KL tokens, both parts are large and partially counteract each other, leaving training dynamics dominated by the high-variance CE component.

Selective masking instead concentrates the constraint budget where it is most needed. On the masked set $\mathcal{M}$, the CE term is *removed entirely* (not merely offset), so the strong forcing direction $-J_{y_t}$ is eliminated and only the soft, label-free regularizer remains. On the complement $\overline{\mathcal{M}}$, the clean CE signal is left intact, free of regularization-induced cancellation. This split—hard control where risk is high, undisturbed learning where signal is clean—is what enables EKSFT to simultaneously activate task-relevant knowledge and preserve exploration capacity.

F Complementarity of Entropy and KL Token Sets

A natural concern is whether the high-entropy and high-KL token sets are largely the same, in which case using both signals would be redundant. To test this, during training we logged per-token entropy and per-token KL for every step, formed the two top-ρ sets $\mathcal{M}_H$ and $\mathcal{M}_{\mathrm{KL}}$ (with $\rho = 0.2$), and computed the Intersection-over-Union $\mathrm{IoU} = |\mathcal{M}_H \cap \mathcal{M}_{\mathrm{KL}}|/|\mathcal{M}_H \cup \mathcal{M}_{\mathrm{KL}}|$ at each step.

Statistic	IoU
Max	0.59
Min	0.09
Avg	0.50

Table 7: IoU between the high-entropy token set $\mathcal{M}_H$ and the high-KL token set $\mathcal{M}_{\mathrm{KL}}$ over training steps.

With an average IoU of about 0.50, only roughly half of the tokens overlap between the two sets. In other words, positions where the model is uncertain (high entropy) are not necessarily the positions that deviate most from the reference distribution (high KL). The two signals capture different aspects of training dynamics, which is why combining them via union (Equation 11) yields strictly more coverage than either single signal alone.

G Sensitivity to Masking Ratio ρ

We sweep the masking ratio $\rho \in \{0.0, 0.1, 0.2, 0.3, 0.4\}$ on Qwen3-4B and report pass@1 and pass@32 on AIME, AIME25, and AMC.

The results show a clear sweet spot around $\rho = 0.2$: small ρ behaves close to standard SFT and provides little benefit, while $\rho = 0.4$ collapses performance across all three benchmarks (e.g., a -11.3% pass@1 drop on AIME relative to $\rho = 0.0$). This is expected, since excessive masking removes too much supervision signal and prevents the model from acquiring task-relevant patterns. A moderate ratio (around 0.2) strikes the right balance between preserving pretrained capabilities and providing adequate supervision for capability activation, which is why we adopt $\rho = 0.2$ throughout the paper.

ρ	AIME		AIME25		AMC	
	pass@1	pass@32	pass@1	pass@32	pass@1	pass@32
0.0	46.3	73.3	32.1	56.7	68.7	86.7
0.1	46.0	73.3	32.3	56.7	68.7	86.7
0.2	**45.7**	**73.3**	**33.4**	**60.0**	**68.9**	**90.4**
0.3	45.2	70.0	32.5	60.0	65.6	85.5
0.4	35.0	60.0	24.0	46.7	58.7	80.1

Table 8: Sensitivity of EKSFT to the masking ratio ρ on Qwen3-4B. Best per column in bold (excluding the ρ=0.0 row, which corresponds to no masking).

H Comparison with Random Masking

To verify that the gains of EKSFT come from *which* tokens are masked rather than from masking per se, we compare EKSFT with a Random Masking baseline that uniformly drops 10% of tokens from CE supervision and applies the same entropy/KL regularization on the dropped tokens. All other settings are kept identical.

Method	AIME		AIME25		AMC	
	pass@1	pass@32	pass@1	pass@32	pass@1	pass@32
EKSFT (Ours)	**45.7**	**73.3**	**33.4**	**60.0**	**68.9**	**90.4**
Random Masking (10%)	35.0	60.0	26.5	56.7	66.0	85.5

Table 9: Comparison with a Random Masking baseline on Qwen3-4B.

Random Masking lags EKSFT on every benchmark and metric, with particularly large gaps on AIME pass@1 (-10.7%) and AIME pass@32 (-13.3%). Indiscriminate masking discards task-relevant evidence and introduces unstructured noise, while EKSFT concentrates masking on positions that are demonstrably risky for entropy collapse and parameter drift. This confirms that the benefit of EKSFT stems from the entropy/KL-guided selection criterion, not merely from reducing the number of supervised tokens.

I Generalization to Tool-Use Benchmarks

To assess whether the benefits of EKSFT generalize beyond mathematical reasoning, we additionally evaluate it on the tool-use setting. We train on AceBench (Chen et al., 2025a) under the same

training configuration as the main experiments, and evaluate on the widely used BFCL (Patil et al., 2025) *live* and *Irrelevance Detection* subsets, which respectively measure overall tool-use performance and robustness to irrelevant requests.

Method (Qwen3-4B)	**BFCL live**	**Irrelevance Det.**
Base	82.4	75.9
SFT	84.4	89.7
DFT	70.5	93.5
EKSFT (Ours)	**86.2**	**100.0**

Table 10: BFCL results on Qwen3-4B trained on AceBench.

Method (Qwen3-8B)	**BFCL live**	**Irrelevance Det.**
Base	84.5	76.9
SFT	86.6	99.5
DFT	82.5	**100.0**
EKSFT (Ours)	**88.0**	**100.0**

Table 11: BFCL results on Qwen3-8B trained on AceBench.

Across both model sizes, EKSFT achieves the best BFCL live score and saturates Irrelevance Detection at 100.0. Notably, DFT improves Irrelevance Detection but degrades on BFCL live (especially on Qwen3-4B, -13.9% versus base), indicating that aggressive reweighting can harm in-task tool-use accuracy. EKSFT avoids this trade-off and improves both metrics simultaneously, demonstrating that its benefits transfer to unseen tasks and datasets beyond mathematical reasoning.