Entropy-Aware Reshaping of Reinforcement Signals for Multi-Answer Reasoning

Zhi Li[1,2], Huidan Xu[1], Zhen Hu[1], Yali Du[2†], Ying Liu[1†]
[1]Tsinghua University, [2]King's College London
{li-z23,xhd24,huz25}@mails.tsinghua.edu.cn
yali.du@kcl.ac.uk,yingliu@tsinghua.edu.cn

Abstract

Reinforcement learning with verifiable rewards (RLVR) is a standard post-training paradigm for large language models (LLMs), typically relying on group-wise reward and advantage normalization for stability. In set-valued multi-answer tasks, where multiple outputs may be simultaneously correct, this normalization can over-amplify a small number of early high-reward samples, suppressing learning signals from other valid generations and leading to overly concentrated updates. We propose Entropy-Aware Reshaping of Reinforcement Signals (EARS), a framework that reshapes how learning signals are normalized and aggregated. EARS uses token-level predictive entropy as an uncertainty cue to compute entropy-weighted reward statistics for advantage normalization, encouraging broader exploration and more balanced learning-signal allocation early in training. An adaptive decay schedule then anneals uncertainty-aware reweighting back to standard group normalization to ensure stable convergence. EARS further incorporates a correctness-gated multi-head process reward that provides auxiliary supervision on reasoning traces while remaining aligned with verifiable correctness. Experiments on MC-TACO and MMLU-Multi using Qwen2.5-7B and Llama-3.1-8B-Instruct demonstrate consistent improvements in exact-set accuracy, training stability, and cross-dataset transfer performance on set-valued multi-answer reasoning.

1 Introduction

Recent advancements in large language models have yielded exceptional performance in structured reasoning domains (OpenAI et al., 2024; Team et al., 2025), most notably in mathematics (Shao et al., 2024) and code intelligence (Lozhkov et al., 2024; DeepSeek-AI et al., 2024) . However, moving beyond these well-defined domains toward general-purpose world perception requires models to operate under inherent ambiguity and diversity. In many real-world scenarios, an input does not map to a single uniquely correct response, but rather to a set of plausible outcomes that should be considered simultaneously. Effectively handling multi-answer tasks is therefore crucial; for example in event-order temporal questions where multiple continuations can be valid (Figure 1).

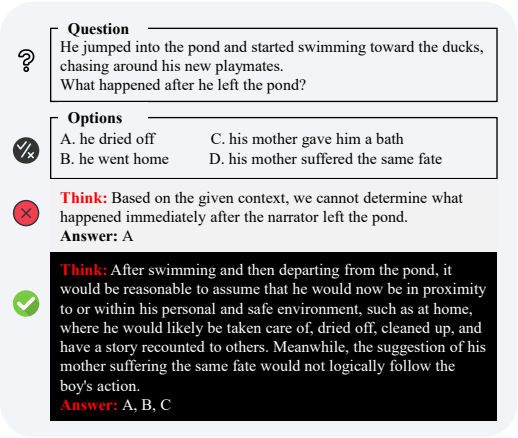

Figure 1: Illustration of the multi-answer reasoning setting. An input may admit multiple valid answers, while standard RL objectives tend to over-concentrate probability mass on a few dominant answers.

In parallel, reinforcement learning (RL) has become a core paradigm for the post-training of LLMs under sparse, outcome-based supervision. Preference-based alignment via reinforcement learning from human feedback (RLHF) and related pipelines has proven effective for instruction-following and safety alignment (Christiano et al., 2017; Ziegler et al., 2020; Stiennon et al., 2020; Ouyang et al., 2022; Bai et al., 2022). Beyond preference learning, RLVR is increasingly used when correctness can be programmatically evaluated, especially for reasoning-heavy tasks (Cobbe et al., 2021; Uesato et al., 2022; Lightman et al., 2024). Practically, policy-gradient methods

†Corresponding authors.

Findings of the Association for Computational Linguistics: ACL 2026, pages 40255–40268

such as proximal policy optimization (PPO) remain a standard choice in RL post-training (Schulman et al., 2017; Ouyang et al., 2022), which often stabilized by KL regularization to a reference policy. More recently, group-based optimization variants, including GRPO-style approaches, that avoid explicit value function estimation have been adopted for efficient RLVR in LLMs(Shao et al., 2024; Zhang et al., 2025). A common ingredient across these methods is batch/group-level normalization of rewards or advantages, which reduces variance and improves optimization stability (Schulman et al., 2018; Guo et al., 2025).

However, RL post-training becomes brittle when the underlying task is set-valued, admitting multiple valid outputs. Prior work on RLHF has documented that RL updates can reduce output diversity and lead to generative monoculture phenomena (Kirk et al., 2023; Wu et al., 2025). In multi-answer selection, we observe an additional mode: the policy may quickly overcommit to a narrow subset of early high-reward trajectories and stop exploring alternative solutions, leading to poor credit assignment and suboptimal convergence.

Concretely, when advantages are computed via group-level normalization within sampled outputs of each prompt, a small number of locally high-reward samples can be amplified early in training, suppressing gradients from other plausible generations before the model has sufficiently explored the space of valid answer sets. The model thus converges before sufficiently exploring alternative valid answer sets, which we refer to as premature confidence collapse.

While premature confidence collapse is conceptually related to entropy collapse (Feng et al., 2025; Wang et al., 2025a), they reflect different aspects of the learning process. Entropy collapse is a distribution-level phenomenon in which the conditional token distribution becomes overly peaked, resulting in reduced policy entropy. In contrast, premature confidence collapse is a credit-assignment failure driven by how normalized advantages allocate learning signals across samples. Consequently, directly encouraging higher entropy through common exploration techniques such as entropy regularization (Adamczyk et al., 2023) or higher-temperature sampling may increase the diversity of sampled trajectories, but it does not directly address the concentration of learning signals induced by group-wise normalization.

This raises a central question. How can we keep RL efficient and stable while preventing early overcommitment to a narrow subset of sampled solutions in set-valued tasks? In other words, the goal is to maintain the learnability of multiple valid answers during the early training phase without sacrificing accurate set alignment at convergence.

Instead of adding explicit entropy regularization or changing the underlying policy optimization algorithm, we focus on reconfiguring reinforcement signals to better control credit assignment throughout training. We propose EARS, a framework that mitigates premature confidence collapse by adjusting the normalization and aggregation of training signals. Specifically, EARS reshapes reinforcement signals via entropy-aware advantage normalization. Token-level predictive entropy is treated as an uncertainty cue, and uncertain yet informative generations receive relatively more credit when computing normalization statistics, counteracting early winner-takes-most dynamics. To preserve asymptotic behavior and avoid over-exploration, we further introduce an adaptive decay schedule that gradually anneals this uncertainty weighting back toward standard group-based normalization as training progresses, implementing a natural transition from exploration to exploitation. EARS also incorporates gated multi-head reasoning rewards to maintain reasoning quality during exploration, providing auxiliary supervision over reasoning traces while ensuring alignment with verifiable correctness constraints.

Our contributions are threefold: (1) We propose EARS, an RL framework that reshapes reinforcement signals for multi-answer reasoning by incorporating entropy-aware advantage normalization with an adaptive decay schedule, alleviating the problem of premature confidence collapse in multi-answer RL post-training. (2) Within the EARS framework, we introduce a correctness-gated multi-head process reward that provides auxiliary supervision on inference quality while remaining aligned with answer-level correctness. (3) Through extensive experiments on set-valued multi-answer tasks, we demonstrate that these two mechanisms provide complementary benefits, leading to consistent improvements in accuracy, training stability, and cross-dataset transfer.

2 Method

To mitigate premature confidence collapse in reinforcement learning with large language models,

particularly in multi-answer multiple-choice settings, we propose the EARS framework that reshapes reinforcement signals of GRPO via entropy-aware advantage normalization with adaptive decay, and incorporates gated multi-head reasoning rewards as complementary reward signals. The framework is shown as Figure 2.

2.1 Entropy-Aware Advantage Reshaping

For each question q, the policy π_θ generates a group of N samples, where each sample $s = \{cot, a\}$ includes a reasoning process and a final answer. The inter-group advantages of samples will be then reshaped by token-level uncertainties.

Token-Level Uncertainty Estimation. For each generated sequence, we first measure token-level uncertainty using top-k restricted entropy. For the t-th token in $\{cot, a\}$, let $p_i^{(t)}$ denote the probability of the i-th most likely candidate token in the vocabulary. We define the renormalized distribution of top-k likely tokens as

$$\tilde{p}_i^{(t)} = \frac{p_i^{(t)}}{\sum_{j=1}^{k} p_j^{(t)}}, \tag{1}$$

and compute the corresponding entropy

$$H_{top-k}^{(t)} = -\sum_{i=1}^{k} \tilde{p}_i^{(t)} \log \tilde{p}_i^{(t)}. \tag{2}$$

Compared to full-vocabulary entropy, top-k restricted entropy focuses on competition among high-probability tokens and provides a more stable and decision-relevant uncertainty estimate.

The uncertainty score of the generated sequence is obtained by averaging token-wise entropy over all non-padding positions:

$$\mathrm{RSI}_{top-k} = \frac{1}{T} \sum_{t=1}^{T} H_{top-k}^{(t)} \cdot \mathbb{I}[\mathrm{token}_t \neq \mathrm{PAD}], \tag{3}$$

where T denotes the length of a sample $\{cot, a\}$.

Entropy-Weighted Advantage Reshaping. Given a group of N samples with uncertainty scores $\{RSI_i\}_{i=1}^{N}$ and corresponding rewards $\{r_i\}_{i=1}^{N}$, we rank samples in descending order of uncertainty and assign exponentially decaying weights

$$w_i = \frac{\exp(-\alpha \cdot \mathrm{rank}_i)}{\sum_{j=1}^{N} \exp(-\alpha \cdot \mathrm{rank}_j)}, \tag{4}$$

where $\alpha > 0$ controls the concentration of weights. Higher α values assign more weight to high-uncertainty samples. Notably, this procedure does not change reward values themselves, but only modulates their contribution to advantage normalization.

Then we compute weighted statistics over rewards

$$\mu_w = \textstyle\sum_{i=1}^{N} w_i r_i, \tag{5}$$

$$\sigma_w^2 = \textstyle\sum_{i=1}^{N} w_i (r_i - \mu_w)^2, \tag{6}$$

and define the reshaped advantage as

$$A_i = \frac{r_i - \mu_w}{\sigma_w + \epsilon}. \tag{7}$$

where ϵ is a small constant for numerical stability.

This entropy-aware normalization increases the relative gradient contribution of uncertain samples while preserving reward-based optimization objectives.

Adaptive Decay Schedule. To further improve training stability and maintain transferability across tasks, we introduce an adaptive decay schedule for the entropy-based reweighting coefficient α. This design enables stronger emphasis on uncertainty-aware learning signals during early training, while gradually recovering standard GRPO behavior as training progresses.

Specifically, we define the decay coefficient at training step s as

$$\alpha(s) = \alpha_0 \cdot (1 - \gamma s / S), \tag{8}$$

where S denotes the total number of training steps, α_0 and γ are the initial reweighting strength and the decay speed, respectively. In our experiments, we set $\alpha_0 = 0.5$ and $\gamma = 2$ by default.

The adaptive schedule reflects a natural transition from exploration to exploitation. When the policy is prone to premature overconfidence, a larger α amplifies learning signals from uncertain generations, encouraging exploration of alternative plausible solutions. As training proceeds, α smoothly decays toward zero, reducing the influence of entropy-based reweighting and yielding uniform weighting equivalent to standard GRPO advantage normalization.

2.2 Multi-Head Reasoning Reward Modeling

To provide auxiliary supervision on the quality of generated reasoning processes, we introduce a

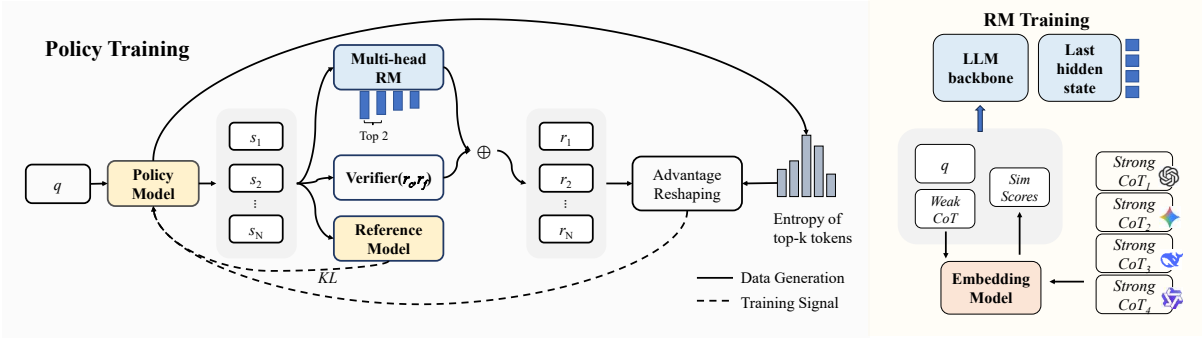

Figure 2: Overview of the EARS framework. We first train the RM by taking a weak reasoning process and reasoning processes generated by multiple strong models as input, and making the semantic similarity between them serve as the supervisory signal to train the scoring capabilities of the model's multiple output heads. Subsequently, we freeze the RM and train the policy model, performing advantage reshaping by calculating the entropy of the top-k tokens.

multi-head reasoning reward model (RM) that assigns dense and continuous scores to intermediate reasoning traces, complementing outcome-based rewards in RLVR.

Multi-Head Reward Model. The reward model is trained via supervised fine-tuning on tuples of the form

$$\{\text{question, weak reasoning}, \mathbf{o}\}$$

where $\mathbf{o} = (o_1, o_2, o_3, o_4)$ denotes cosine similarity scores between the embedding of a weak reasoning process and the corresponding reasoning processes generated by four strong teacher models. Each head of the reward model is trained to regress one similarity score, resulting in a multi-head architecture that captures diverse-quality reasoning patterns.

Formally, given a generated reasoning trace ν, the reward model outputs

$$\mathbf{r}_{\text{cot}}(\nu) = \left(r_{\text{cot}}^{(1)}(\nu), \ldots, r_{\text{cot}}^{(4)}(\nu)\right), \tag{9}$$

where each head corresponds to a distinct teacher signal.

Aggregation with Correctness Gating. To mitigate bias toward any single teacher and reduce susceptibility to reward exploitation, we aggregate multi-head rewards using the mean of the top-2 scores. Additionally, we employ correctness gating to prevent reasoning rewards from dominating task objectives. Under this mechanism, reasoning rewards are only activated when the final answer satisfies task-level correctness constraints. The final reasoning reward is formulated as

$$r'_{\text{cot}}(\nu) = \frac{1}{2}\left(r_{\text{top}}^{(1)}(\nu) + r_{\text{top}}^{(2)}(\nu)\right) \cdot \mathbb{I}[r_c], \tag{10}$$

$$r_c(a, \mathcal{G}) = \begin{cases} 1, & \text{if } a = \mathcal{G}, \\ -0.5, & \text{if } a \subsetneq \mathcal{G}, \\ -1, & \text{otherwise}, \end{cases} \tag{11}$$

where $r_{\text{top}}^{(1)}$ and $r_{\text{top}}^{(2)}$ denote the two highest head outputs, $r_c \in \{-1, -0.5, 1\}$ represents the correctness reward in RLVR, and $\mathcal{G}$ is the gold answer. This design encourages high-quality reasoning aligned with correct outcomes, while preventing the model from optimizing for stylistic similarity alone.

Integration with RLVR Objective. The gated reasoning reward is incorporated into the RLVR objective as an auxiliary shaping term. The overall reward for a sample is defined as

$$r_{total} = r_c + r_f + \lambda_{\text{cot}} \cdot r'_{\text{cot}}. \tag{12}$$

Consistent with standard RLVR practices, $r_f \in \{-1, 1\}$ requires that the reasoning trace be encapsulated within `<think></think>` tags and the final answer within `<answer></answer>` tags.

The combined reward is then used to compute advantages under the entropy-aware and adaptively decayed framework described in previous sections. Following the standard GRPO paradigm, the objective function is formulated as

$$\mathcal{L}(\theta) = \mathbb{E}_{\substack{q\sim\mathcal{Q} \\ s\sim\pi_{\text{old}}(\cdot|q)}}\Big[\min\Big(r(\theta)A, \text{clip}\big(r(\theta), 1-\epsilon, 1+\epsilon\big)A\Big) - \beta\mathbb{D}_{KL}(\pi_\theta||\pi_{ref})\Big], \tag{13}$$

where $r(\theta) = \frac{\pi_\theta(s|q)}{\pi_{\theta_{\text{old}}}(s|q)}$ is the probability ratio.

3 Experiments

3.1 Datasets

We train and evaluate EARS on multi-answer multiple-choice reasoning tasks, where each question admits a *set* of correct options. We detail the selection and construction of the datasets.

MCTACO. (Zhou et al., 2019) We construct multiple-choice tasks from the original dataset by first removing questions with fewer than four options and filtering out questions for which all options are labeled as "no." We then ensure that each remaining question contains exactly four options and at least one correct answer. After filtering, we obtain 2,455 instances in a four-option multiple-choice format, which are randomly split into a training set and a test set with a 2,000:455 ratio.

The challenge lies in whether the model can understand the magnitude of time units and the constraints of the physical world while selecting all possible answers.

MMLU-MULTI. (Hendrycks et al., 2021) We use the publicly available dataset *mmlu-multi-answers*[1] hosted on Hugging Face, which is derived from the MMLU benchmark and contains multiple answers generated by GPT-4o-mini. We filtered out examples with more than four options and randomly sampled 3,000 instances from the remaining 3,079 examples, which were then split into a training set and a test set in a 2,000:1,000 ratio.

In addition to reading comprehension over long texts, MMLU-Multi also tests knowledge of some obscure topics without context, which poses challenges to the LLM's multiple-choice performance. Therefore, we adopt MMLU-Multi as a more difficult dataset.

3.2 Evaluation Metrics

Given a gold option set G_i and a predicted set P_i for instance i, we report (i) set exact match (EM) and (ii) example-level set F1 (Macro-F1):

$$\text{Macro-F1} = \mathbb{E}_i \left[\frac{2|P_i \cap G_i|}{|P_i| + |G_i|} \right]. \qquad (14)$$

We decode P_i from the <answer> span by extracting option letters (e.g., via a regex over {A,B,C,D}) and ignore intermediate reasoning traces at evaluation time. We treat invalid outputs as $\varnothing$ and report results on test split.

[1] https://huggingface.co/datasets/Obsismc/mmlu-multi_answers

3.3 Models and Training Setup

Base Models. We instantiate all methods on Qwen2.5-7B and Llama-3.1-8B-Instruct. We use the officially released versions and the same prompt templates across methods.

Supervised Fine-Tuning (SFT). SFT is trained on the training split using the standard cross-entropy loss for 4000 steps, with a learning rate of 2e-4 and a batch size of 16 (4 per device × 4 gradient accumulation steps).

RL Post-Training (RLVR with GRPO). We adopt an RLVR setting and optimize the policy using the GRPO-based objective. At each RL step, we sample $N = 8$ trajectories for one prompt with temperature $\tau = 0.9$, and a maximum of 512 generated tokens. For MCTACO, training is conducted for $S = 3,000$ steps using AdamW with a learning rate of $1e-6$ and a fixed KL regularization coefficient $\beta = 0.04$. For MMLU-Multi, we use the same training setup except for a smaller learning rate of $5e-7$ and a KL coefficient of $\beta = 0.02$. All experiments with additional components follow the same training protocol and hyperparameter settings as described above.

Reward Model. We construct the process reward model by adding four heads to the base model, each predicting a continuous-valued score for reasoning quality. Training is conducted following the mechanism described in Sec 3.2. Four external general models, GPT-5, Gemini-2.5-Flash (Comanici et al., 2025), DeepSeek-R1 (Guo et al., 2025), and Qwen3-Next-80B-A3B-Thinking (Yang et al., 2025), are employed to generate strong reasoning traces. Both weak and strong reasoning traces are encoded into vector representations using RoBERTa-large-v1. More details can be found in Appendix.

3.4 Baselines

We compare: **(i) Base**: pretrained model without task-specific training; **(ii) SFT**: supervised fine-tuning on the training split; **(iii) GRPO**: standard RLVR using GRPO with rewards $(r_c + r_f)$; **(iv) GRPO + Process Reward**: GRPO with additional gated process rewards $(r_c + r_f + \lambda_{\text{cot}} \cdot r'_{\text{cot}})$, as defined in Eq. 12, where the default value of λ_{cot}

Model	Method	MCTACO			MMLU-Multi		
		EM(%)	Macro-F1	EM-OOD	EM(%)	Macro-F1	EM-OOD
Qwen2.5-7B	Base	18.7	0.540	-	48.2	0.842	-
	+ SFT	71.4	0.899	46.7	58.4	0.869	43.1
	+ GRPO	82.2	0.938	48.0	59.2	0.874	63.5
	+ GRPO + Process Reward	82.9	0.944	49.1	58.7	0.875	65.3
	+ GRPO + Adaptive Decay	87.5	0.957	50.8	60.5	0.880	77.6
	+ EARS	**89.2**	**0.959**	**51.3**	**62.9**	**0.889**	**78.5**
Llama-3.1-8B -Instruct	Base	18.9	0.643	-	40.4	0.782	-
	+ SFT	61.3	0.832	46.5	47.8	0.745	20.4
	+ GRPO	64.0	0.851	44.5	49.2	0.797	33.0
	+ GRPO + Process Reward	65.1	0.856	48.4	49.9	0.803	**42.2**
	+ GRPO + Adaptive Decay	78.9	0.924	49.8	52.4	**0.848**	40.7
	+ EARS	**79.6**	**0.931**	**50.7**	**54.5**	0.846	41.3

Table 1: Main results on the datasets with different baselines. EM-OOD denotes the out-of-distribution Exact Match, training on one dataset (MCTACO/MMLU-Multi) and testing on another dataset (MMLU-Multi/MCTACO).

is 0.5; **(v) GRPO + Adaptive Decay**: GRPO with entropy-aware advantage reshaping and the adaptive decay schedule, but without process reward; **(vi) EARS**: full method with both components of **(iv)** and **(v)**.

3.5 Main Results

Table 1 shows the main results on MCTACO and MMLU-Multi using Qwen2.5-7B and Llama-3.1-8B-Instruct as base policies.

Across both datasets and base models, RL post-training substantially improves over SFT, and EARS consistently achieves the strongest performance. On MCTACO with Qwen2.5-7B, EARS reaches 89.2% EM, representing a clear improvement over GRPO and all intermediate variants. Similar gains are observed on MMLU-Multi, as well as when using Llama-3.1-8B-Instruct as the base policy, indicating that the effectiveness of EARS is robust across datasets and models.

Ablating individual components reveals that the two mechanisms in EARS provide complementary benefits. Adding gated process rewards yields modest but consistent improvements over GRPO, suggesting that auxiliary reasoning supervision can refine solutions when correctness constraints are satisfied. In contrast, entropy-aware advantage reshaping with an adaptive decay schedule yields larger and more consistent improvements across multi-answer datasets. When combined with gated process rewards, EARS achieves the strongest overall performance, suggesting that the two mechanisms provide complementary benefits in optimizing credit assignment and refining answer selection during reinforcement learning.

We further evaluate cross-dataset transfer using EM-OOD, where models are trained on one dataset and tested on the other. EARS consistently outperforms GRPO in both transfer directions. Notably, when trained on MMLU-Multi and evaluated on MCTACO, EARS improves EM-OOD from 63.5 to 78.5, highlighting its ability to learn more transferable decision boundaries for multi-answer selection.

Overall, these results show that EARS improves both in-domain performance and cross-dataset generalization by reshaping reinforcement signals in a way that promotes balanced exploration early in training while enabling precise answer alignment at convergence.

3.6 Analysis and Ablations

How Does EARS Affect Learning-Signal Dispersion and Exploration? To analyze how EARS influences training dynamics beyond aggregate performance, we examine learning-signal allocation and exploration behavior at the answer-set level. Specifically, we focus on two metrics: (i) answer-set learning-signal dispersion, quantified by the effective number of answer clusters receiving learning signals K_{eff}[2], and (ii) exploration coverage, measured by Coverage@8, i.e., the proportion of sampling trajectories that contain one of the correct answer, out of 8 sampling trajectories. Together, these metrics characterize whether learning signals are overly concentrated on a narrow subset of solutions and whether alternative plausible answers continue to be explored, which are two key aspects of premature confidence collapse in credit assignment.

[2]Details of the indicator calculation can be found in the appendix.

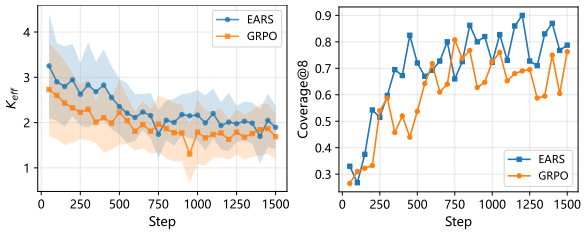

Figure 3: Answer-set-level training dynamics of EARS and GRPO in the first 1500 steps. Left: K_{eff}, the effective number of answer sets receiving positive learning signals, versus training steps. Right: Coverage@8 versus training steps.

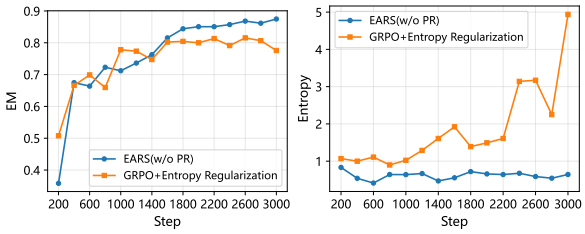

Figure 4: Training curves comparing EARS (w/o process reward) and GRPO with objective-level entropy regularization. Left: Exact Match on the test set versus training steps. Right: average policy entropy measured during training.

Figure 3 reports the effective number of answer sets receiving learning signals, measured by K_{eff}, throughout training. For both methods, K_{eff} generally decreases as training progresses, reflecting the transition from dispersed exploration toward more focused exploitation. EARS maintains a higher K_{eff} than GRPO throughout the first 1500 training steps, allowing a broader range of distinct answer sets to receive effective update signals. After 500 steps, the gap between the two methods gradually narrows, while EARS remains slightly higher overall. This behavior is consistent with the EARS framework, where entropy-aware reshaping focuses on initial credit assignment, allowing the adaptive decay schedule to gradually transition back to standard normalization.

The exploration metric Coverage@8 rises rapidly during early training and saturates in the middle-to-late stages. This trend indicates that as the policy improves, the predicted sets that can contain the correct answers under a limited sampling budget (8 samples) expands and gradually stabilizes. Compared to GRPO, EARS generally achieves higher Coverage@8 in the early stage, reflecting faster improvement in covering multiple feasible solutions under the same sampling budget. In the later stage, the Coverage@8 of both methods becomes comparable, suggesting that the benefit of EARS mainly lies in protecting exploration coverage during early training, without sacrificing the eventual convergence level of coverage.

Comparison to Objective-Level Entropy Regularization. A common alternative to our signal-reshaping approach is to directly add an entropy regularization term to the reward or objective (Adamczyk et al., 2023)

$$\mathcal{L}_{\text{GRPO-en}} = \mathcal{L}_{\text{GRPO}} - \mathbb{E}_t\big[\psi \cdot \mathcal{H}\big(\pi_\theta(\cdot \mid s_t)\big)\big]. \quad (15)$$

We implement GRPO with Entropy Regularization and compare it with EARS, while disabling process rewards in EARS for a fair comparison.

Figure 4 reports both the EM and the average policy entropy during training. EARS achieves higher EM after 1400 steps and converges around 2000 steps with noticeably smoother training, whereas GRPO with entropy regularization exhibits a less stable improvement trajectory and minor oscillations after convergence.

Importantly, the entropy curves reveal a key difference in behavior. Entropy regularization steadily pushes the policy entropy upward and becomes increasingly volatile in later training, with pronounced spikes near the end of training. Despite this substantially higher entropy, the EM does not improve accordingly and even shows instability, suggesting that simply maximizing entropy can lead to over-exploration or brittle optimization in multi-answer generation.

In contrast, EARS maintains a relatively low and stable entropy throughout training while achieving better EM. This suggests that the entropy-aware reshaping signal improves credit assignment without explicitly driving the policy toward high-entropy behavior.

Component Ablations. We further ablate (i) the decay speed $\gamma \in \{0, 0.5, 1, 2, \infty\}$ of the adaptive reshaping schedule, (ii) correctness gating $I_g \in \{0, 1\}$ for process rewards , and (iii) the coefficient of the CoT reward $\lambda_{\text{cot}} \in \{0.5, 1, 2\}$. With the default calibration settings $\gamma = 2$, $I_g = 1$, $\lambda_{\text{cot}} = 0.5$, we conduct ablation experiments by varying one factor at a time and report EM and Macro-F1 in Table 2.

The results for $\gamma = 2$ are superior to those of $\gamma = 1$ and 0.5, yielding the optimal performance. It indicates that keeping strong uncertainty reweight-

ing active for too long can undermine exact-set alignment near convergence. The cases where $\gamma = 0$ and ∞ are equivalent to keeping the reshaping weight constant and omitting the reshaping weight, respectively; both configurations lead to a degradation in performance.

Toggling the correctness gating switch I_g has a pronounced effect on EM. Transitioning I_g from 1 to 0 results in a 4.5% decrease in EM. This indicates that the gating mechanism of process rewards relative to verifiable correctness can materially impact the selection of exact answer sets, even while overlap-based metrics like Macro-F1 remain constant.

In the ablation study of the process reward coefficient, we find that a smaller coefficient $\lambda_{\text{cot}} = 0.5$ is optimal, whereas increasing the coefficient significantly impairs performance. This supports treating the process reward as an auxiliary shaping signal, as over-weighting reasoning similarity can distract optimization away from the primary verifiable objective needed for exact multi-answer selection.

Aggregation Rule of Multi-Head Process Reward. We further study how to aggregate the multi-head process reward. Since different teacher heads may capture complementary reasoning patterns, an appropriate aggregation rule should balance robustness and informativeness. As shown in Table 3, the top-2 mean achieves the best performance. Using top-1 is less stable because the reward can be dominated by a single high-scoring head, while increasing w to 3 or 4 introduces noisier signals and degrades performance. Median reward also performs worse than top-2 mean. These results suggest that the top-2 mean provides a good trade-off between leveraging agreement among strong teachers and reducing sensitivity to noisy or biased heads, which supports our default design.

Component	Value	EM	Macro-F1
Decay Rate γ	0	82.9	0.944
	0.5	86.2	0.951
	1	84.8	0.952
	2	**89.2**	**0.959**
	∞	82.2	0.938
Correctness Gating I_g	0	84.6	0.958
	1	**89.2**	**0.959**
Reward Coefficient λ_{cot}	**0.5**	**89.2**	**0.959**
	1	82.2	0.943
	2	77.6	0.940

Table 2: Ablation studies on the impact of decay rate, correctness gating, and reward coefficient.

4 Related Work

RLVR for LLM Reasoning. Reinforcement learning with verifiable rewards has rapidly become a mainstream post-training route for enhancing the reasoning capabilities of large language models(Trung et al., 2024; Lambert et al., 2025; Zelikman et al., 2022; Le et al., 2022; Gou et al., 2024). This approach is particularly effective in domains where the accuracy of outputs can be assessed through automated and deterministic verification mechanisms. Meanwhile, a growing direction analyzes RLVR's optimization behavior and practical pitfalls, including GRPO dynamics and implicit objective interpretations (Mroueh, 2025), entropy performance and clipping trade-offs during RLVR training (Park et al., 2025; Jin et al., 2025), and empirical studies debating whether RLVR expands reasoning capacity versus primarily improving sampling efficiency under fixed model priors (Yue et al., 2025; Wen et al., 2025). Additionally, complementary directions examine data and compute efficiency in RLVR, demonstrating that credit assignment and training dynamics play a more decisive role in performance than reward availability alone, as evidenced by recent studies on one-example RLVR and off-policy rollout schemes(Wang et al., 2025b).

Learning with Set-Valued Signals and Process Feedback. Multi-answer prediction has been extensively studied in supervised learning under set-valued labels, including multi-label classification and learning with multiple valid targets (Tsoumakas and Katakis, 2007; Zhang and Zhou, 2014). In natural language process, set-valued or ambiguity-aware supervision arises in tasks where multiple outputs are acceptable, such as ambiguous open-domain QA (Min et al., 2020) and temporal commonsense reasoning datasets where several op-

Aggregation rule	Value	EM	Macro-F1
Top-w mean aggregation	$w = 1$	88.4	0.953
	$w = 2$	89.2	0.959
	$w = 3$	88.6	0.951
	$w = 4$	86.6	0.949
Other aggregation rule	median reward only	87.3	0.947

Table 3: Ablation study on process-reward aggregation rules.

tions can be plausible in the same context (Zhou et al., 2019); more broadly, permutation-invariant set modeling provides principled architectures and objectives for set prediction (Zaheer et al., 2017).

Complementing outcome-level supervision, process-level feedback has emerged as a key driver for robust reasoning (Uesato et al., 2022; Wu et al., 2023; Rafailov et al., 2024; Cui et al., 2025; Yoon et al., 2024). Step-wise verification and process reward models have proven effective in shaping intermediate trajectories (Lightman et al., 2024; Cobbe et al., 2021), yet they typically operate under single-answer correctness assumptions. Our gated multi-head reasoning rewards align with these insights by providing auxiliary process guidance while keeping optimization grounded in verifiable set correctness.

5 Conclusion

In this paper, we reveal that standard RL post-training can suffer from premature confidence collapse in multi-answer selection tasks. We introduce EARS, a novel approach that mitigates this by leveraging predictive entropy to reweight advantage signals and integrating auxiliary process supervision. Empirical results across datasets show that EARS not only stabilizes training but also achieves superior cross-dataset generalization compared to standard methods like GRPO.

Limitations

While our work demonstrates the effectiveness of entropy-aware reshaping and auxiliary reasoning supervision for improving reinforcement learning in multi-answer settings, EARS still has certain limitations. EARS uses token-level predictive entropy as a proxy for uncertainty and applies rank-based exponential weighting when computing normalization statistics. However, high entropy does not always correspond to informative uncertainty. It can also arise from genuine confusion or low-quality generations, which may introduce noise during the early stages of training. The process reward is defined over explicit reasoning traces and is activated only under correctness gating. This design assumes access to structured intermediate reasoning text, and it is not obvious that the same benefits hold when reasoning traces are unavailable, truncated, or discouraged at deployment time. Future work may explore alternative uncertainty signals beyond token-level entropy and identify surrogates for process rewards in RL settings where explicit reasoning supervision is absent.

Acknowledgements

This work was supported by High Performance Computing Center, Tsinghua University. We also thank the anonymous reviewers for their constructive feedback.

References

Jacob Adamczyk, Argenis Arriojas, Stas Tiomkin, and Rahul V. Kulkarni. 2023. Utilizing Prior Solutions for Reward Shaping and Composition in Entropy-Regularized Reinforcement Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(6):6658–6665.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, et al. 2022. Constitutional AI: Harmlessness from AI Feedback. *arXiv preprint*. ArXiv:2212.08073.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep Reinforcement Learning from Human Preferences. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Karl Cobbe, Vineet Kosaraju, Mo Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, et al. 2021. Training Verifiers to Solve Math Word Problems. *ArXiv*. ArXiv:2110.14168.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, et al. 2025. Gemini 2.5: Pushing the Frontier with Advanced Reasoning, Multimodality, Long Context, and Next Generation Agentic Capabilities. *arXiv preprint*. ArXiv:2507.06261.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, et al. 2025. Process Reinforcement through Implicit Rewards. *arXiv preprint*. ArXiv:2502.01456.

DeepSeek-AI, Qihao Zhu, Daya Guo, Zhihong Shao, Dejian Yang, Peiyi Wang, Runxin Xu, Y. Wu, Yukun Li, et al. 2024. DeepSeek-Coder-V2: Breaking the Barrier of Closed-Source Models in Code Intelligence. *arXiv preprint*. ArXiv:2406.11931.

Weitao Feng, Lixu Wang, Tianyi Wei, Jie Zhang, Chongyang Gao, Sinong Zhan, Peizhuo Lv, and Wei Dong. 2025. Token Buncher: Shielding LLMs from Harmful Reinforcement Learning Fine-Tuning. *arXiv preprint*. ArXiv:2508.20697.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Minlie Huang, Nan Duan, and Weizhu Chen. 2024. ToRA: A Tool-Integrated Reasoning Agent for Mathematical Problem Solving. *International Conference on Representation Learning*, 2024:48362–48395.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, et al. 2025. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645(8081):633–638. Publisher: Nature Publishing Group.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring Massive Multitask Language Understanding. *arXiv preprint*. ArXiv:2009.03300.

Renren Jin, Pengzhi Gao, Yuqi Ren, Zhuowen Han, Tongxuan Zhang, Wuwei Huang, Wei Liu, Jian Luan, and Deyi Xiong. 2025. Revisiting Entropy in Reinforcement Learning for Large Reasoning Models. *arXiv preprint*. ArXiv:2511.05993.

Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. 2023. Understanding the Effects of RLHF on LLM Generalisation and Diversity. ArXiv.2310.06452.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, et al. 2025. Tulu 3: Pushing Frontiers in Open Language Model Post-Training. *arXiv preprint*. ArXiv:2411.15124.

Hung Le, Yue Wang, Akhilesh Deepak Gotmare, Silvio Savarese, and Steven Chu Hong Hoi. 2022. CodeRL: Mastering Code Generation through Pretrained Models and Deep Reinforcement Learning. *Advances in Neural Information Processing Systems*, 35:21314–21328.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let's Verify Step by Step. *International Conference on Representation Learning*, 2024:39578–39601.

Anton Lozhkov, Raymond Li, Loubna Ben Allal, Federico Cassano, Joel Lamy-Poirier, Nouamane Tazi, Ao Tang, Dmytro Pykhtar, Jiawei Liu, et al. 2024. StarCoder 2 and The Stack v2: The Next Generation. *arXiv preprint*. ArXiv:2402.19173.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. AmbigQA: Answering Ambiguous Open-domain Questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5783–5797, Online. Association for Computational Linguistics.

Youssef Mroueh. 2025. Reinforcement Learning with Verifiable Rewards: GRPO's Effective Loss, Dynamics, and Success Amplification. *arXiv preprint*. ArXiv:2503.06639.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, et al. 2024. GPT-4 Technical Report. *arXiv preprint*. ArXiv:2303.08774.

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

Jaesung R. Park, Junsu Kim, Gyeongman Kim, Jinyoung Jo, Sean Choi, Jaewoong Cho, and Ernest K. Ryu. 2025. Clip-Low Increases Entropy and Clip-High Decreases Entropy in Reinforcement Learning of Large Language Models. *arXiv preprint*. ArXiv:2509.26114.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2024. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. *arXiv preprint*. ArXiv:2305.18290.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. 2018. High-Dimensional Continuous Control Using Generalized Advantage Estimation. *arXiv preprint*. ArXiv:1506.02438.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. *arXiv preprint*. ArXiv:1707.06347.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, et al. 2024. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. *arXiv preprint*. ArXiv:2402.03300.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems*, volume 33, pages 3008–3021. Curran Associates, Inc.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, et al. 2025. Gemini: A Family of Highly Capable Multimodal Models. *arXiv preprint*. ArXiv:2312.11805.

Luong Trung, Xinbo Zhang, Zhanming Jie, Peng Sun, Xiaoran Jin, and Hang Li. 2024. ReFT: Reasoning with Reinforced Fine-Tuning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*,

pages 7601–7614, Bangkok, Thailand. Association for Computational Linguistics.

Grigorios Tsoumakas and Ioannis Katakis. 2007. Multi-Label Classification: An Overview. *International Journal of Data Warehousing and Mining (IJDWM)*, 3(3):1–13. Publisher: IGI Global Scientific Publishing.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. 2022. Solving math word problems with process- and outcome-based feedback. *arXiv preprint*. ArXiv:2211.14275.

Jiakang Wang, Runze Liu, Fuzheng Zhang, Xiu Li, and Guorui Zhou. 2025a. Stabilizing Knowledge, Promoting Reasoning: Dual-Token Constraints for RLVR. *arXiv preprint*. ArXiv:2507.15778.

Yiping Wang, Qing Yang, Zhiyuan Zeng, Liliang Ren, Liyuan Liu, Baolin Peng, Hao Cheng, Xuehai He, Kuan Wang, et al. 2025b. Reinforcement Learning for Reasoning in Large Language Models with One Training Example. *arXiv preprint*. ArXiv:2504.20571.

Xumeng Wen, Zihan Liu, Shun Zheng, Shengyu Ye, Zhirong Wu, Yang Wang, Zhijian Xu, Xiao Liang, Junjie Li, et al. 2025. Reinforcement Learning with Verifiable Rewards Implicitly Incentivizes Correct Reasoning in Base LLMs. *arXiv preprint*. ArXiv:2506.14245.

Fan Wu, Emily Black, and Varun Chandrasekaran. 2025. Generative Monoculture in Large Language Models. *International Conference on Representation Learning*, 2025:33068–33107.

Zeqiu Wu, Yushi Hu, Weijia Shi, Nouha Dziri, Alane Suhr, Prithviraj Ammanabrolu, Noah A. Smith, Mari Ostendorf, and Hannaneh Hajishirzi. 2023. Fine-Grained Human Feedback Gives Better Rewards for Language Model Training. *arXiv preprint*. ArXiv:2306.01693.

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

Eunseop Yoon, Hee Suk Yoon, SooHwan Eom, Gunsoo Han, Daniel Nam, Daejin Jo, Kyoung-Woon On, Mark Hasegawa-Johnson, Sungwoong Kim, and Chang Yoo. 2024. TLCR: Token-Level Continuous Reward for Fine-grained Reinforcement Learning from Human Feedback. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14969–14981, Bangkok, Thailand. Association for Computational Linguistics.

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? *arXiv preprint*. ArXiv:2504.13837.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. 2017. Deep Sets. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. 2022. STaR: Bootstrapping Reasoning With Reasoning. *arXiv preprint*. ArXiv:2203.14465.

Min-Ling Zhang and Zhi-Hua Zhou. 2014. A Review on Multi-Label Learning Algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 26(8):1819–1837.

Wenhao Zhang, Yuexiang Xie, Yuchang Sun, Yanxi Chen, Guoyin Wang, Yaliang Li, Bolin Ding, and Jingren Zhou. 2025. On-Policy RL Meets Off-Policy Experts: Harmonizing Supervised Fine-Tuning and Reinforcement Learning via Dynamic Weighting. ArXiv:2508.11408.

Ben Zhou, Daniel Khashabi, Qiang Ning, and Dan Roth. 2019. "Going on a vacation" takes longer than "Going for a walk": A Study of Temporal Commonsense Understanding. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3363–3369, Hong Kong, China. Association for Computational Linguistics.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2020. Fine-Tuning Language Models from Human Preferences. *arXiv preprint*. ArXiv:1909.08593.

A Answer-Set Learning-Signal Diversity via Absolute-Advantage Mass

To quantify how learning signals distribute across distinct predicted answer sets, we define an answer-set-level diversity metric based on the *absolute* advantage mass. For a prompt-level group with sampled trajectories indexed by $j \in \{1, \ldots, N\}$, let P_j denote the predicted answer set decoded from trajectory j, and let A_j denote its (reshaped) advantage.

Absolute-advantage mass. For each answer-set cluster s (i.e., a unique predicted set), we define

$$m_s^{\text{abs}} = \sum_{j:\, P_j = s} |A_j|,$$

$$q_s^{\text{abs}} = \frac{m_s^{\text{abs}}}{\sum_{s'} m_{s'}^{\text{abs}} + \epsilon}, \tag{16}$$

where ϵ is a small constant for numerical stability.

Entropy and effective number of answer sets. We then compute the entropy over answer-set clusters and its corresponding effective number

$$H_{\text{ans}}^{\text{abs}} = -\sum_s q_s^{\text{abs}} \log q_s^{\text{abs}},$$

$$K_{\text{eff}}^{\text{abs}} = \exp\big(H_{\text{ans}}^{\text{abs}}\big). \tag{17}$$

For simplicity, we refer to $K_{\text{eff}}^{\text{abs}}$ as K_{eff} in the main text.

Interpretation. $|A_j|$ serves as a direction-agnostic proxy for the *update strength* contributed (or incurred) by trajectory j, so m_s^{abs} measures how much overall update "energy" is associated with answer-set cluster s. If *premature confidence collapse* occurs, update energy tends to concentrate on a very small number of clusters (especially in early-stage winner-take-all regimes), leading to a smaller $K_{\text{eff}}^{\text{abs}}$. Conversely, if EARS successfully preserves informative exploration, its effect is often more visible in $K_{\text{eff}}^{\text{abs}}$—capturing a more balanced distribution of update energy across alternative answer sets—than in coarse sign-based statistics such as the positive-advantage ratio.

$|A|$ vs. A^2. One may alternatively replace $|A_j|$ with A_j^2 in Eq. (16). Using $|A|$ is typically more robust (less sensitive to extreme outliers), while A^2 is more head-sensitive and may better highlight collapse patterns dominated by a few trajectories with exceptionally large advantages.

B Training of Reward Model

We construct the process reward model by adding four regression heads to the base model, where each head predicts a continuous-valued score for reasoning quality. The model is trained using two types of reasoning signals: weak reasoning traces collected from preliminary training with standard GRPO, and strong reasoning traces generated by external high-capacity teacher models.

Specifically, for each question q, we first collect the reasoning trace produced by the preliminary policy and treat it as weak reasoning, denoted by y_{weak}. We then query several strong general-purpose teacher models to generate corresponding strong reasoning traces for the same question, denoted by $\{y_{\text{strong}}^{(k)}\}_{k=1}^{4}$. For MCTACO, the prompt used to generate strong reasoning traces is shown in Table 4; for MMLU-Multi, we use the same template but remove the context field.

Both weak and strong reasoning traces are encoded into dense vector representations using RoBERTa-large-v1. Let

$$\mathbf{v}_{\text{weak}} = E(y_{\text{weak}}),$$

$$\mathbf{v}^{(k)} = E\Big(y_{\text{strong}}^{(k)}\Big), \tag{18}$$

where $E(\cdot)$ denotes the embedding model. We then define the supervision target for the k-th head as the cosine similarity between the weak reasoning trace and the k-th teacher reasoning trace:

$$o_k = \text{CosSim}\Big(\mathbf{v}_{\text{weak}}, \mathbf{v}^{(k)}\Big) = \frac{\mathbf{v}_{\text{weak}} \cdot \mathbf{v}^{(k)}}{\|\mathbf{v}_{\text{weak}}\|_2 \, \|\mathbf{v}^{(k)}\|_2}. \tag{19}$$

Accordingly, each training instance is associated with a four-dimensional target vector

$$\mathbf{o} = (o_1, o_2, o_3, o_4), \tag{20}$$

which measures the semantic similarity between the weak reasoning trace and the strong reasoning traces from multiple teachers.

In our implementation, we employ four external general models, GPT-5, Gemini-2.5-Flash, DeepSeek-R1, and Qwen3-Next-80B-A3B-Thinking, to generate strong reasoning traces, resulting in four cosine-similarity targets for each sample. The reward model is then trained by supervised regression. Given an input pair (q, y^{weak}), the reward model predicts

$$\hat{\mathbf{o}} = (\hat{o}_1, \hat{o}_2, \hat{o}_3, \hat{o}_4), \tag{21}$$

and is optimized with a sum-of-squared-errors objective:

$$\mathcal{L}_{\mathrm{RM}} = \sum_{k=1}^{4} \|\hat{o}_k - o_k\|_2^2 . \tag{22}$$

After training, the reward model is frozen and used to provide auxiliary multi-head process-reward signals during policy optimization.

C Pseudocode of Training

Algorithms 1 and 2 demonstrate the training process for the reward model and the policy model, respectively.

Algorithm 1 Training of Multi-Head Reasoning Reward Model

Require: Dataset $\mathcal{D}$, reward model RM_ϕ, teacher models $\mathcal{T} = \{T_1, \ldots, T_4\}$, embedding model E(e.g., RoBERTa)
Ensure: Trained reward model RM_ϕ

Initialize parameters ϕ
Initialize training set $\mathcal{D}_{\text{train}} \leftarrow \emptyset$
for each question $q \in \mathcal{D}$ **do**
  Generate weak reasoning trace y_{weak} from the preliminary policy
  Generate strong reasoning traces $\{y_{\text{strong}}^{(k)}\}_{k=1}^{4}$ using teacher models $\mathcal{T}$
  Compute weak-trace embedding: $\mathbf{v}_{\text{weak}} \leftarrow E(y_{\text{weak}})$
  for $k = 1$ to 4 **do**
    Compute teacher-trace embedding: $\mathbf{v}^{(k)} \leftarrow E\left(y_{\text{strong}}^{(k)}\right)$
    Compute cosine-similarity target: $o_k \leftarrow \mathrm{CosSim}\left(\mathbf{v}_{\text{weak}}, \mathbf{v}^{(k)}\right)$
  end for
  Form target vector $\mathbf{o} \leftarrow (o_1, o_2, o_3, o_4)$
  Add $(q, y_{\text{weak}}, \mathbf{o})$ to $\mathcal{D}_{\text{train}}$
end for
while not converged **do**
  for each batch $(q, y_{\text{weak}}, \mathbf{o})$ in $\mathcal{D}_{\text{train}}$ **do**
    Predict head outputs: $\hat{\mathbf{o}} \leftarrow RM_\phi(q, y_{\text{weak}})$
    Compute regression loss: $\mathcal{L}_{\text{RM}} \leftarrow \sum_{k=1}^{4} \|\hat{o}_k - o_k\|_2^2$
    Update $\phi \leftarrow \phi - \eta \nabla_\phi \mathcal{L}$
  end for
end while

Algorithm 2 Policy Training with EARS Framework

Require: Policy π_θ, Reward Model RM_ϕ, Reference π_{ref}, Group size N, Decay params α_0, γ

for $step = 1$ to S_{total} **do**
  Sample batch of prompts q
  Generate N trajectories $\{s_i\}_{i=1}^{N}$ per prompt, where $s_i = (cot_i, a_i)$
  Compute Rewards:
  for $i \leftarrow 1$ to N **do**
    Calculate outcome reward r_c
    Process scores $\mathbf{h} \leftarrow RM_\phi(q, cot_i)$
    Gated CoT reward $r'_{cot} \leftarrow \mathrm{Mean}(\mathrm{Top2}(\mathbf{h})) \cdot \mathbb{I}[r_c]$
    Total reward $r_i \leftarrow r_c + r_f + \lambda_{cot} \cdot r'_{cot}$
  end for
  Entropy-Aware Reshaping:
  for $i \leftarrow 1$ to N **do**
    Calculate uncertainty RSI_i using Top-k entropy
  end for
  Update decay $\alpha \leftarrow \alpha_0 \cdot (1 - \gamma \frac{step}{S_{total}})$
  Compute weights $w_i \propto \exp(-\alpha \cdot \mathrm{Rank}(RSI_i))$
  Compute weighted mean μ_w and std σ_w of rewards
  Reshaped Advantage $A_i \leftarrow \frac{r_i - \mu_w}{\sigma_w + \epsilon}$
  Update Policy:
  Optimize $\mathcal{L}_{GRPO}(\theta)$ using reshaped advantages A_i
end for

D RSI-Quality Correlation in Early Training

To further examine whether RSI in Eq. 3 captures informative uncertainty rather than random noise, we analyze the correlation between sequence-level entropy and prediction quality during early training. Specifically, we compute the sequence-level uncertainty score (RSI) for each sampled trajectory within the first 200 training steps and divide samples into five quantile bins according to RSI. For each bin, we report the proportions of exact matches ($r_c = 1$), partial matches ($r_c = -0.5$), and incorrect predictions ($r_c = -1$).

As shown in table 5, higher-entropy bins do not simply correspond to a surge of incorrect outputs. Instead, they contain a higher proportion of partially correct predictions ($r_c = -0.5$), suggesting that elevated entropy often captures informative

```
You are an expert in generating high-quality Chain of Thought (CoT) data
for multi-select reasoning tasks. The existing data contains a Context, a
Question, multiple Options, and the Ground Truth Answer (which may contain
multiple correct options).
Your task is to create a concise, logical Chain of Thought to supplement
this data. Adhere to the following rules:
Step-by-Step Evaluation: Systematically evaluate each option against the
provided context. Briefly explain why correct options are valid and why
incorrect options are flawed.
Forward Reasoning: Ensure the reasoning logically deduces the correct
options based on the context, rather than just reverse-engineering the
provided Ground Truth.
Format & Masking: Enclose your entire reasoning process within
<think></think> tags. Do not explicitly mention the ground truth answer
string (e.g., "The correct answer is A") inside the tags.
Language: Match the language of your reasoning process to the language of
the Question.
Context: {context}
Question: {question}
Options: A. {option_A}
B. {option_B}
C. {option_C}
D. {option_D}
Ground Truth Answer: {ground_truth}
```

Table 4: The CoT generating prompt applied to the MCTACO data.

RSI quantile	$p^{(1)}$	$p^{(-0.5)}$	$p^{(-1)}$
$[80\% - 100\%]$ (highest)	0.29	0.51	0.20
$[60\% - 80\%)$	0.31	0.43	0.26
$[40\% - 60\%)$	0.30	0.33	0.37
$[20\% - 40\%)$	0.26	0.23	0.51
$[0\%, 20\%)$ (lowest)	0.22	0.14	0.64

Table 5: Distribution of correctness labels across entropy (RSI) quantile bins.

uncertainty rather than random noise. This supports the use of entropy as a proxy for exploration-relevant uncertainty in early training.