REFLECT LESS, REASON BETTER: ENTROPY-AWARE REFLECTION REWARDS FOR EFFICIENT LARGE REASONING MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large reasoning models make significant progress in solving complex problems, but their reasoning processes are still often affected by overthinking, which leads to high reasoning latency and computational costs without corresponding improvements in accuracy. Existing methods usually improve reasoning efficiency by directly penalizing reasoning length or the number of reflection words through reinforcement learning. However, these methods often ignore differences between problem difficulty and reflective reasoning structures, which may cause models to frequently switch reasoning paths and thus produce fragmented reflection. To address these issues, this paper proposes a reflection reward method that integrates answer entropy and difficulty awareness, based on analyses of the number of reflection words, continuous reasoning spans, and answer distributions. This method focuses on constraining inefficient reflection behaviors in two types of scenarios: high accuracy with low entropy, and low accuracy with high entropy. It penalizes fragmented and cyclic reflection, and favors reasoning trajectories with low-frequency reflection, long continuous reasoning spans, and continuous stability. Extensive experimental results show that, without reducing model performance, the proposed method effectively reduces token consumption in chain-of-thought reasoning processes, providing a more fine-grained and effective optimization approach for balancing efficiency and accuracy in large language model reasoning.

1 INTRODUCTION

Large Reasoning Models (LRMs), such as OpenAI o1 (Jaech et al., 2024), QwQ (Yang et al., 2025a), and DeepSeek-R1 (Guo et al., 2025), significantly enhance model performance on complex reasoning tasks by scaling computational cost during the inference stage. Such models typically generate longer and more structured chain-of-thought trajectories and demonstrate self-reflection during reasoning, thereby enabling systematic exploration of multiple solution paths. Their reasoning processes are usually explicitly marked by <think> and </think> tags to distinguish internal reasoning from the final output, providing observable clues for analyzing models' multi-step reasoning behavior (Guan et al., 2025; Ma et al., 2025). Reinforcement learning is regarded as a key mechanism behind the emergence of these capabilities. During training, models tend to produce increasingly longer and more complex outputs, while gradually acquiring advanced reasoning strategies such as problem decomposition, self-evaluation, self-verification, and self-correction (Wu et al., 2025).

Although extended reasoning chains offer many advantages, they often introduce inefficiencies due to unnecessary repetition and excessive elaboration, which is often referred to as "overthinking" (Hong et al., 2025). This problem significantly increases reasoning latency and computational cost, while the improvement in accuracy is often limited (Wan et al., 2026). At present, this problem is mainly addressed through two approaches. One is training-free and typically improves generation efficiency through prompt optimization or early stopping mechanisms (Hong et al., 2025; Xu et al., 2025). However, since it does not change the model's reasoning ability or output distribution, its effectiveness is relatively limited. The other trains LRMs through methods such as supervised fine-tuning or reinforcement learning to construct more efficient reasoning patterns (Lai & Nissim, 2026;

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

Parascandolo et al., 2026). However, while such methods compress reasoning length, they are often accompanied by a decline in reasoning performance. Therefore, how to achieve a dynamic trade-off between reasoning length and accuracy remains a key issue in research on efficient reasoning (Hu et al., 2026).

Existing studies indicate that the reflective behavior of LRMs is a key factor in improving models' reasoning abilities, but excessive reflection is also an important source of reasoning redundancy. Moreover, as problem complexity increases, excessive reflection in LRMs becomes more pronounced (Yu et al., 2026; Wang et al., 2025). We further analyze the reflective behavior of LRMs and find that as the number of reflections increases, the continuous reasoning span guided by each reflection steadily decreases, and the model's reasoning accuracy also declines. This suggests that excessive reflection does not necessarily lead to more sufficient reasoning. Instead, it may prompt the model to frequently switch its direction of thought, causing the reasoning process to become increasingly fragmented.

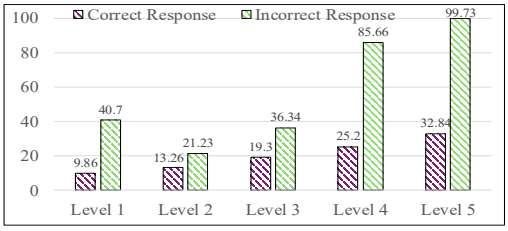
Difficulty awareness is a key issue in reward design. If problem difficulty cannot be accurately distinguished, the reward mechanism may compress the reasoning space required for difficult problems or tolerate redundant reflection on simple problems (Shen et al., 2025). Existing methods usually measure difficulty through sampling accuracy, but this metric mainly reflects the model's overall success rate across multiple samples and cannot capture distributional differences among incorrect answers (Xue et al., 2025). If the incorrect answers from multiple samples are relatively concentrated, this suggests that the model may have a stable knowledge or reasoning bias, in which case appropriate reflection can still help it escape an erroneous path. If the incorrect answers are highly dispersed, this indicates that the model lacks a stable basis for reasoning. The reasoning process is more like random exploration across different directions, and further increasing reflection is more likely to lead to ineffective search or cyclic reasoning.

Based on the above analysis, we propose **E**ntropy-**A**ware **R**eflection **R**eward (**EARR**), a reflection reward method that characterizes problem difficulty by combining accuracy and answer entropy based on the answer distributions obtained from multiple samples. EARR adaptively adjusts reflection rewards for different types of reasoning trajectories. For problems with high accuracy and low entropy, EARR suppresses redundant reflection on simple problems. For problems with low accuracy and high entropy, EARR penalizes ineffective search and cyclic reasoning that lack a clear direction. Meanwhile, EARR further incorporates reasoning length, the number of reflections, and continuous reasoning span, encouraging the model to conduct more sufficient and continuous reasoning after each reflection and to avoid frequent shifts in thinking direction, thereby reducing unnecessary reasoning redundancy while maintaining reasoning performance. In summary, our key contributions are as follows:

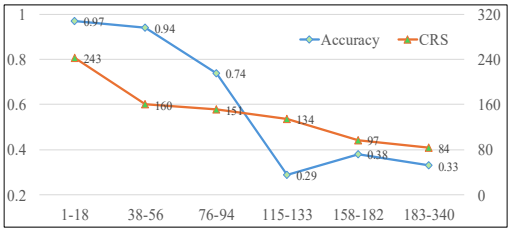
- We analyze the reflection behavior of LRMs in detail and find that too much reflection lowers accuracy and makes reasoning paths more fragmented.
- We use answer entropy to measure problem difficulty. It complements sampling accuracy by capturing uncertainty in the answer distribution, providing a finer basis for reward adjustment.
- We propose EARR, which adjusts reflection rewards based on accuracy and answer entropy. It also considers reasoning length, reflection count, and continuous reasoning span to reduce redundant reflection and improve reasoning continuity.

2 Related Work

The overthinking phenomenon in LRMs prompts researchers to explore more concise and efficient reasoning mechanisms (Cheng et al., 2025; Xiang et al., 2025). Existing methods can be roughly divided into two categories. The first category consists of training-free methods, which mainly reduce redundant reasoning through prompt engineering or early-stopping mechanisms (Aytes et al., 2025; Yang et al., 2025b). However, since these methods do not change the model's intrinsic reasoning behavior, their improvement in reasoning efficiency is usually limited. The second category trains models by constructing supervision signals, including supervised fine-tuning and reinforcement learning (Ning et al., 2025; Hou et al., 2025). Related studies constrain reasoning length and improve reasoning efficiency from different perspectives. O1-Pruner designs a composite reward



(a) Correlation between problem difficulty and the number of reflections.



(b) Correlation of reflection count with reasoning accuracy and continuous reasoning span in LRMs.

Figure 1: Analysis of reflection behaviors in LRMs.

that combines accuracy and reasoning length, encouraging the model to shorten the reasoning process while maintaining performance (Luo et al., 2025); Thinkless is based on DeGRPO and enables LRMs to adaptively switch between thinking and non-thinking modes according to task complexity and their own capabilities (Fang et al., 2026); ConciseRL uses GPT-4.1 as an evaluator to score the conciseness of reasoning trajectories, thereby providing reward signals for reinforcement learning (Dumitru et al., 2025); SP introduces step-level rewards to improve reasoning efficiency while prioritizing answer correctness (Wu et al., 2025); AALC incorporates validation-set accuracy into the reward function and adopts a smooth dynamic length-penalty mechanism, so that length constraints take effect only after the target performance is achieved (Li et al., 2025); APR finds that models still perform a large amount of meaningless repeated verification after the answer becomes stable, and improves reasoning efficiency by penalizing this redundant part (Chang et al., 2026); ARLCP proposes an adaptive reflection and length co-penalty mechanism to achieve a better trade-off between reasoning efficiency and accuracy (Yu et al., 2026). However, existing methods mainly design rewards based on global signals such as reasoning length or reflection frequency, without considering the varying effectiveness of reflection across different reasoning scenarios. This limits their ability to adaptively regulate reflection behaviors according to problem difficulty.

3 Motivation

This section investigates the relationship among reflection behavior, reasoning length, and accuracy in LRMs. We conduct multiple sampling experiments on the MATH-500 dataset using the DeepSeek-R1-Distill-Qwen-1.5B model to analyze the behavioral patterns of the model across different reasoning processes (Guo et al., 2025; Hendrycks et al., 2021).

Observation 1: Excessive reflection may reduce reasoning efficiency. As shown in Figure 1a, the number of reflections in correct responses gradually increases as problem difficulty grows, indicating that more reflection is required for complex problems. However, incorrect responses usually involve more reflections, and this difference becomes more significant for difficult problems. This suggests that increasing the number of reflections does not necessarily lead to better results. Excessive reflection may instead cause inefficient reasoning and repeated revision processes, thereby reducing reasoning efficiency.

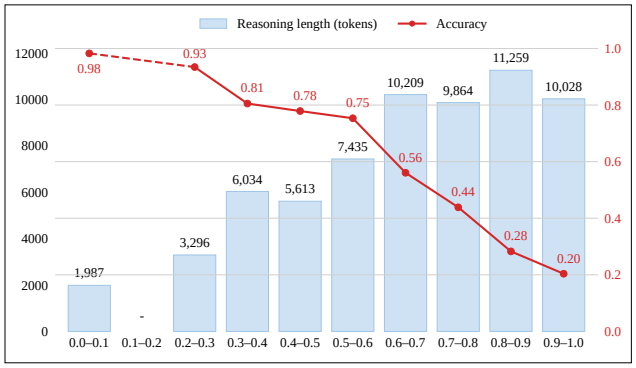


Figure 2: Reasoning length and accuracy across answer-entropy intervals.

Observation 2: Excessive reflection weakens reasoning continuity. Figure 1b illustrates the relationship among the number of reflections, accuracy, and Continuous Reasoning Span (CRS). CRS represents the average reasoning

length that the model can maintain within a continuous reasoning process and is used to measure reasoning continuity. As the number of reflections increases, both model accuracy and CRS show a declining trend, indicating that frequent reflections cause the model to switch reasoning directions more often, thereby reducing reasoning continuity and efficiency.

Observation 3: Answer entropy effectively characterizes reasoning uncertainty. As shown in Figure 2, we estimate the answer distribution based on multiple sampling results and calculate its entropy as an indicator of problem difficulty. The results show that as answer entropy increases, model accuracy gradually decreases while reasoning length increases. Low-entropy problems correspond to more concentrated answer distributions and more stable reasoning paths, whereas high-entropy problems exhibit greater uncertainty and more redundant reasoning. Therefore, answer entropy provides an effective basis for modeling problem difficulty and adjusting reasoning strategies.

4 METHOD

Given a question, EARR first samples multiple reasoning trajectories and profiles their reasoning behaviors in terms of reasoning length, reflection count, and continuous reasoning span. It then estimates question-level reasoning uncertainty from the sampled answer distribution using sampling accuracy and answer entropy. Based on these signals, EARR constructs a trajectory-level behavior penalty and modulates its strength using a difficulty-aware factor. The resulting reward is used as the outcome-level supervision signal for GRPO optimization. Figure 3 illustrates the overall framework.

4.1 REASONING BEHAVIOR PROFILING

Given a training set $Q = \{(q_j, y_j)\}_{j=1}^{M}$, where q_j denotes the j-th question and y_j denotes its corresponding ground-truth answer, we independently sample N reasoning trajectories from the current large reasoning model L_θ for each question q_j, denoted as

$$O_j = \{O_j^1, O_j^2, ..., O_j^N\},$$

where O_j^i denotes the i-th reasoning trajectory generated by the model for question q_j. Each reasoning trajectory corresponds to a final answer A_j^i. Therefore, the set of answers obtained from multiple samples can be written as

$$A_j = \{A_j^1, A_j^2, ..., A_j^N\}.$$

To characterize the efficiency differences and reflection behavior across reasoning trajectories, we profile each trajectory from three dimensions: reasoning length, reflection count, and continuous reasoning span. Specifically, for a reasoning trajectory O_j^i, we denote its reasoning length, reflection count, and continuous reasoning span as

$$L_j^i, C_j^i, S_j^i.$$

L_j^i denotes the number of tokens contained in the reasoning portion of a trajectory, which is used to measure the overall reasoning cost incurred by the model when solving the current problem. C_j^i denotes the number of reflection behaviors occurring in the reasoning trajectory. Following prior studies on reflective behavior in large reasoning models, we identify reflection behaviors through reflective words or phrases that indicate reconsideration, verification, self-correction, or shifts in reasoning direction, and count their occurrences in each trajectory (Yu et al., 2026; Chen et al., 2024). To further characterize the continuity of the reasoning process, we introduce the continuous reasoning span S_j^i. Specifically, we treat the continuous reasoning content between two adjacent reflection behaviors as a reasoning segment, and use the average length of these reasoning segments as the continuous reasoning span of the trajectory. Overall, L_j^i, C_j^i and S_j^i characterize the model's reasoning behavior from three perspectives: reasoning cost, reflection frequency, and reasoning continuity, providing a comprehensive view of the reasoning trajectory.

4.2 ENTROPY-AWARE DIFFICULTY MODELING

Relying solely on sampling accuracy is insufficient to fully characterize problem difficulty, as it cannot capture differences in the distribution of sampled answers. Therefore, we further introduce answer entropy and combine it with sampling accuracy to provide a finer-grained characterization

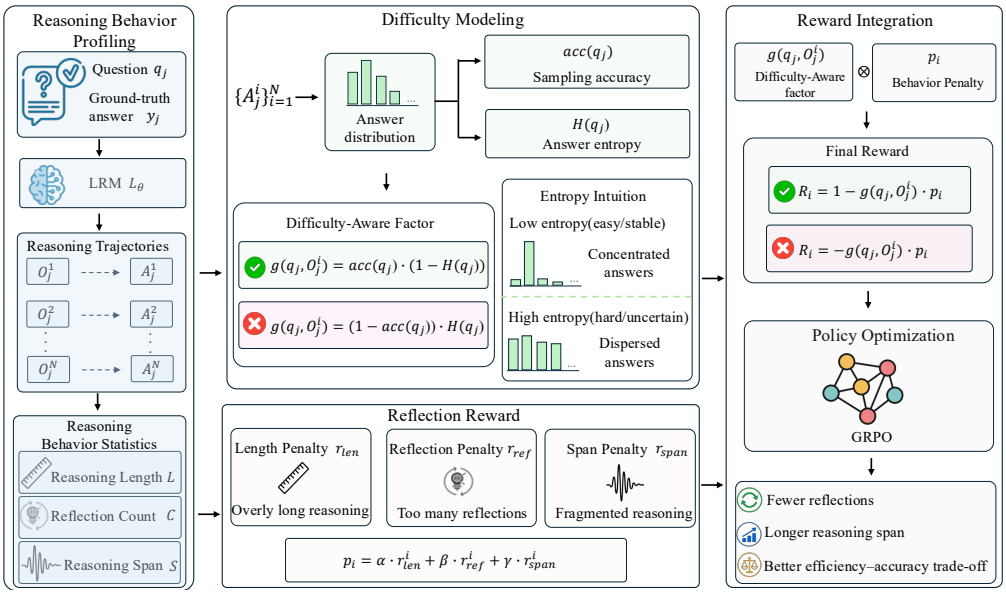


Figure 3: Overall framework of EARR. By combining answer-entropy-based difficulty awareness with trajectory-level behavior penalties, EARR adaptively regulates inefficient reflection and guides the model toward more continuous and efficient reasoning through GRPO optimization.

of the model's reasoning stability and uncertainty on the current problem. For a question q_j and any candidate answer a, we define its empirical probability over N sampled responses as

$$p_j(a) = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}(A_j^i = a),$$

where $\mathbf{1}(\cdot)$ is the indicator function. Based on this answer distribution, the answer entropy of q_j is defined as

$$H(q_j) = -\sum_{a \in V_j} p_j(a) log p_j(a),$$

where V_j denotes the set of distinct answers appearing among the N samples. To eliminate scale differences caused by the number of answer categories, we further normalize the answer entropy to the range [0,1].

Meanwhile, the sampling accuracy of question q_j is defined as

$$acc(q_j) = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}(A_j^i = y_j),$$

where y_j denotes the ground-truth answer of question q_j.

Combining the analysis in Section 3 with the statistical results in Figure 2, we focus on two scenarios in which inefficient reflection is more likely to occur. For high-accuracy, low-entropy problems, the model can already obtain the correct answer relatively stably, and additional reflection is therefore more likely to manifest as redundant verification. For low-accuracy, high-entropy problems, the model produces highly dispersed answers and lacks a stable solution direction, making continued reflection more likely to result in ineffective search or cyclic reasoning. Based on this observation, we construct a difficulty-aware modulation factor $g(q_j, O_j^i)$ to adaptively control the strength of the behavior penalty for different reasoning trajectories:

$$g(q_j, O_j^i) = \begin{cases} \text{acc}(q_j) \cdot (1 - H(q_j)), & A_j^i = y_j, \\ (1 - \text{acc}(q_j)) \cdot H(q_j), & A_j^i \neq y_j. \end{cases}$$

For correct trajectories, the factor takes a larger value under high-accuracy, low-entropy conditions, thereby imposing stronger constraints on redundant reasoning behaviors in simple and stable prob-

lems. For incorrect trajectories, the factor becomes larger under low-accuracy, high-entropy conditions, thereby more strongly suppressing ineffective search and cyclic reflection in highly uncertain scenarios. In this way, EARR adaptively adjusts the trajectory-level behavior penalty according to the question-level reasoning state.

4.3 Reward Construction

We construct a trajectory-level behavior penalty from three dimensions: reasoning length, reflection frequency, and reasoning continuity. For a reasoning trajectory O_j^i, we denote its behavioral statistic as

$$x_{j,k}^i \in \{L_j^i, C_j^i, S_j^i\}, \qquad k \in \{len, ref, span\}$$

To measure the deviation of the current trajectory from the reference reasoning behavior, we define a unified behavior penalty as

$$r_{\mathrm{k}}^i = sigmoid\left(d_k \frac{x_{j,k}^i - \mu_{j,k}^*}{s_{j,k}^* + \epsilon}\right),$$

where $\mu_{j,k}^*$ and $s_{j,k}^*$ denote the mean and standard deviation of the corresponding behavioral statistic in the sample set, respectively, and ϵ is a smoothing term. The direction coefficients are defined as

$$d_{len} = d_{ref} = 1, \qquad d_{span} = -1,$$

so that longer reasoning, more frequent reflection, and shorter continuous reasoning spans receive larger penalties.

We then combine the three behavioral dimensions into a unified trajectory-level behavior penalty:

$$p_i = \alpha r_{\mathrm{len}}^i + \beta r_{\mathrm{ref}}^i + \gamma r_{\mathrm{span}}^i,$$

where α, β and γ control the contributions of different reasoning behaviors. By further incorporating the difficulty-aware modulation factor $g(q_j, O_j^i)$ defined in Section 4.2, the final reward is formulated as

$$R(q_j, O_j^i) = \mathbf{1}(A_j^i = y_j) - g(q_j, O_j^i)p_i$$

The first term represents the correctness reward, while the second term adaptively modulates the trajectory-level behavior penalty according to the question-level reasoning state. In this way, EARR combines question-level difficulty modeling with trajectory-level reasoning behavior constraints, thereby suppressing redundant reflection on simple and stable problems as well as ineffective search and cyclic reasoning on highly uncertain problems.

$R(q_j, O_j^i)$ is used as the outcome reward for GRPO-based policy optimization (Guo et al., 2025). For multiple reasoning trajectories sampled from the same question, GRPO updates the policy according to their relative rewards, increasing the probability of generating high-quality reasoning trajectories while gradually suppressing inefficient and redundant reasoning behaviors.

5 Experiments

5.1 Experimental Setup

LRMs: We use two publicly available reasoning models as the base models: DeepSeek-R1-Distill-Qwen-1.5B (Guo et al., 2025) and Qwen3-4B (Yang et al., 2025a).

Datasets: This paper uses the MATH training set as the source of training data, which contains 7,500 mathematical problem samples (Hendrycks et al., 2021). To systematically evaluate the mathematical reasoning ability of the models, we conduct experiments on six widely recognized mathematical benchmark datasets, including MATH500 (Hendrycks et al., 2021), AIME 2024 (AIME24, 2024), AIME 2025 (AIME25, 2025), AMC 2023 (AMC23, 2023), OlympiadBench (He et al., 2024), and MinervaMath (Lewkowycz et al., 2022). We use pass@1 accuracy (Acc) and response length (Length) as the main evaluation metrics, and further report the average accuracy change (ΔAcc) and the average relative length change (ΔLength) across all test datasets.

Implementation Details. We conduct all experiments within the Verl framework. During training, we set the context length to 16K tokens, the batch size to 128, and the learning rate to 2e-6. We train on all data for 3 epochs and sample 10 responses for each data instance. The EARR hyperparameters α, β, and γ are set to 0.15, 0.05, and 0.05, respectively.

Table 1: Comparison of the accuracy and response length of different methods across six datasets. The results show that our method substantially reduces response length while maintaining or even improving overall accuracy, achieving a better trade-off between accuracy and reasoning conciseness than other methods.

Method	AIME24		AIME25		AMC23		MATH500		Minervamath		Olympiad		Overall	
	Acc	Length	Acc	Length	Acc	Length	Acc	Length	Acc	Length	Acc	Length	ΔAcc	ΔLength
DeepSeek-R1-Distill-Qwen-1.5B														
Vanilla	26.3	12340	21.3	11861	65.6	7724	81.1	4675	25.1	6212	42.1	8686	-	-
CoD	26.1	11477	21.5	11215	68.5	6648	80.4	4043	22.5	4723	42.5	7815	+0.0	-12.3%
SFT	26.6	11512	21.4	11252	67.8	6930	80.4	4196	23.9	5521	41.4	8316	+0.0	-8.0%
DPO	29.0	10851	20.2	10877	70.3	6576	82.1	3882	26.1	5508	44.7	7752	+1.8	-12.4%
O1-Pruner	24.0	8655	21.3	8987	71.9	6306	81.1	4181	25.2	3433	44.1	5725	+1.0	-27.0%
AdaptThink	**32.1**	9546	22.9	9552	67.0	4181	81.4	2134	**26.7**	2597	45.0	6058	+2.3	-38.5%
ARLCP	30.4	7857	**23.2**	7420	73.3	3774	82.8	2138	**26.7**	2585	46.0	4816	+3.5	–47.0%
EARR	31.7	**6652**	22.5	**6505**	**75.2**	**3460**	**83.1**	**1931**	26.6	**2247**	**46.7**	**4206**	**+4.0**	**-53.4%**
Qwen3-4B														
Vanilla	64.0	12705	53.1	14632	91.6	7892	92.1	4963	38.0	6676	62.6	9487	-	-
CoD	39.3	9296	36.8	10933	79.2	4640	75.4	3272	32.7	4829	55.2	5582	-13.8	-32.7%
SFT	**65.2**	12617	55.4	14334	91.1	7448	92.6	4770	39.4	6476	62.8	9355	+0.8	-2.7%
DPO	63.5	10630	54.2	12438	93.0	5476	91.8	3042	37.8	3814	62.8	6936	+0.3	-28.4%
O1-Pruner	57.9	9875	52.5	10995	**93.9**	5895	92.1	3810	39.2	4242	62.6	7130	-0.5	-26.2%
AdaptThink	60.4	9240	51.1	10350	90.3	4373	91.8	2555	38.0	2944	62.0	5720	-1.3	-40.9%
ARLCP	62.9	**7352**	54.3	**8433**	91.7	3258	**93.2**	1953	38.1	2115	62.4	4405	+0.2	-54.3%
EARR	**65.2**	7875	**56.5**	8757	91.9	**2971**	92.5	**1511**	**40.1**	**1379**	**63.4**	**4316**	**+1.4**	**-57.3%**

5.2 Baselines

We compare EARR with several representative methods for efficient reasoning. CoD uses prompting to encourage concise responses (Yu et al., 2026); SFT is trained on the shortest correct solutions (Zhang et al., 2026), and DPO prefers shorter correct solutions over longer ones (Rafailov et al., 2023). O1-Pruner reduces reasoning length under an accuracy constraint (Luo et al., 2025), while AdaptThink adaptively selects reasoning modes according to problem difficulty (Zhang et al., 2025). ARLCP regulates reflection and reasoning length during training to improve reasoning efficiency (Yu et al., 2026).

5.3 Main Results

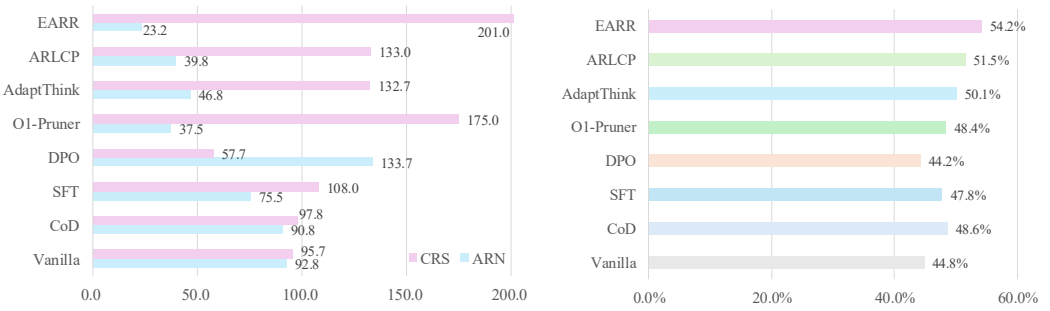
Table 1 presents the performance of different methods on mathematical reasoning tasks. Compared with the original DeepSeek-R1-Distill-Qwen-1.5B and Qwen3-4B models, EARR significantly reduces response length while improving average accuracy, achieving length reductions of 53.4% and 57.3%, along with average accuracy improvements of 4.0 and 1.4 percentage points, respectively. Overall, although some methods can shorten the reasoning process, they often suffer from accuracy degradation or limited performance gains. In contrast, EARR achieves a better trade-off between reasoning efficiency and accuracy on both models.

5.4 Ablation Study

To analyze the contribution of each component, we conduct ablation experiments on DeepSeek-R1-Distill-Qwen-1.5B. The results in Table 2 show that different components of EARR play complementary roles in improving reasoning efficiency while maintaining accuracy. Removing the length penalty r_{len} significantly reduces the length reduction performance, indicating that it mainly controls reasoning length. Removing the reflection penalty r_{ref} or span penalty r_{span} leads to declines in both accuracy and compression effectiveness, demonstrating that suppressing excessive reflection and fragmented reasoning helps improve reasoning quality and efficiency. Removing the difficulty-aware factor g also weakens the length reduction capability, suggesting that it enables adaptive ad-

Table 2: Ablation study on DeepSeek-R1-Distill-Qwen-1.5B across six mathematical reasoning benchmarks.

Method	AIME24		AIME25		AMC23		MATH500		Minervamath		Olympiad		Overall	
	Acc	Length	Acc	Length	Acc	Length	Acc	Length	Acc	Length	Acc	Length	ΔAcc	ΔLength
Vanilla	26.3	12340	21.3	11861	65.6	7724	81.1	4675	25.1	6212	42.1	8686	-	-
w/o r_{len}	**32.1**	9550	**24.8**	8895	**75.5**	5118	**85.4**	2981	27.0	3955	46.9	6277	**+5.0**	-30.3%
w/o r_{ref}	27.9	7966	24.2	7304	72.0	4043	83.1	2221	26.2	2547	46.0	4805	+3.0	-46.3%
w/o r_{span}	29.6	7573	24.4	6942	72.5	3881	83.8	2160	26.7	2447	46.4	4754	+3.6	-48.3%
w/o g	30.2	8580	23.3	8350	**75.5**	4869	**85.4**	2700	**27.2**	4059	**48.1**	5692	+4.7	-34.7%
EARR	31.7	**6652**	22.5	**6505**	75.2	**3460**	83.1	**1931**	26.6	**2247**	46.7	**4206**	+4.0	**-53.4%**



(a) Reflection frequency and reasoning continuity.

(b) Comparison of FCAR across different methods.

Figure 4: Analysis of reasoning behaviors across different methods.

justment of penalty strategies based on problem difficulty. Overall, these components work together to allow EARR to achieve a better trade-off between reasoning efficiency and accuracy.

5.5 Generalization

To further evaluate the cross-task generalization ability of EARR, we conduct experiments on MMLU (Hendrycks et al., 2020). Unlike mathematical reasoning benchmarks, MMLU covers diverse domains and task types, providing a more comprehensive evaluation of model capabilities. As shown in Table 3, EARR achieves a better accuracy–efficiency trade-off across different backbones, improving performance while significantly reducing response length. These results demonstrate that EARR can effectively compress redundant reasoning while maintaining strong generalization ability.

5.6 Reasoning Behavior Analysis

This section further analyzes the impact of EARR on model reasoning behaviors. The experiments are conducted on DeepSeek-R1-Distill-Qwen-1.5B across six mathematical reasoning datasets. The reported results are averaged over all six datasets, while detailed results for each individual dataset are provided in the Appendix.

5.6.1 Reflection Pattern

To analyze the impact of different methods on model reflection behaviors, we measure the average reflection number (ARN) and CRS, which characterize the reflection frequency and reasoning continuity, respectively. As shown in Figure 4a, EARR achieves the lowest ARN and the highest CRS. This indicates that EARR effectively reduces redundant reflections while maintaining longer continuous reasoning processes. Notably, EARR does not simply reduce the number of reflections; instead, it constrains inefficient reflection behaviors, enabling the model to perform fewer but more effective reflections and thereby improving reasoning continuity and efficiency.

Table 4: Effect of the sampling number N during training on accuracy, reasoning length, and training time across six mathematical reasoning benchmarks.

Number	AIME24		AIME25		AMC23		MATH500		Minervamath		Olympiad		Training Time
	Acc	Length	Acc	Length	Acc	Length	Acc	Length	Acc	Length	Acc	Length	
$N = 6$	31.7	7954	22.9	7280	73.3	3757	83.0	2215	26.6	2460	45.5	4560	26h
$N = 10$	31.7	6652	22.5	6505	**75.2**	3460	83.1	1931	26.6	2247	46.7	4206	31h
$N = 16$	30.1	6903	22.9	6684	74.3	**3301**	83.4	2077	27.3	2321	45.8	4313	37h
$N = 32$	**31.8**	**6498**	**24.6**	**5994**	74.1	3339	**84.4**	**1781**	**28.8**	**2046**	**48.5**	**4132**	61h

5.6.2 First Correct Answer Ratio

We define the first correct answer ratio (FCAR) as the relative position in a reasoning trajectory at which the model first explicitly produces a correct answer, normalized by the total length of the trajectory. Specifically, we use a prompt-based approach developed with GPT-5.0 (Singh et al., 2025) to analyze each reasoning trajectory, identify the first explicit occurrence of a correct answer, and calculate its relative position within the entire reasoning process. Any incorrect answers appearing earlier in the trajectory are not considered when determining this position. As shown in Figure 4b, EARR achieves the highest FCAR, indicating that the model performs a larger proportion of its reasoning before arriving at the correct answer, while spending less of the subsequent reasoning on repeated verification and redundant reflection after the correct answer has emerged.

5.7 Sensitivity to Sampling Number

To analyze the impact of the sampling number N during training on model performance and training cost, we conduct experiments on DeepSeek-R1-Distill-Qwen-1.5B with $N = \{6, 10, 16, 32\}$ using 8 A100 GPUs for training. As shown in Table 4, increasing the sampling number generally improves the overall accuracy while reducing reasoning length, indicating that more samples provide richer group-wise comparison signals and facilitate policy optimization. However, larger N also introduces substantial training overhead, with training time increasing from 26 hours for $N = 6$ to 61 hours for $N = 32$. Considering the trade-off between performance improvement and training cost, we finally adopt $N = 10$, which achieves a better balance between accuracy, reasoning efficiency, and computational cost, requiring only approximately 31 hours of training.

Table 3: Performance of different methods on MMLU.

Method	*DS-1.5B*		*Qwen3-4B*	
	Acc	Length	Acc	Length
Vanilla	34.9	2183	79.6	1951
CoD	33.9	1400	78.6	913
SFT	34.7	2841	79.4	1966
DPO	30.2	1700	79.2	1254
O1-Pruner	37.8	1100	79.8	1766
AdaptThink	**43.0**	1176	79.2	1132
ARLCP	41.9	877	79.2	916
EARR	42.6	**772**	**80.1**	**608**

6 Conclusion

This paper proposes EARR, an entropy-aware reflection reward method, to improve the reasoning efficiency of large reasoning models. EARR models problem difficulty by combining answer entropy and sampling accuracy, and constrains inefficient reasoning behavior from three aspects: reasoning length, reflection count, and continuous reasoning span. Experimental results show that EARR significantly reduces reasoning token consumption while maintaining or even improving model accuracy. Further analysis of reflection behavior and generalization experiments also show that EARR effectively reduces redundant reflection, enhances the continuity of the reasoning process, and maintains good generalization ability.

7 AI USE STATEMENT

Generative AI tools are used in this work to assist with the prompt design and analysis procedure for identifying the position at which a model first explicitly produces an answer in a reasoning trajectory, as described in Section 5.6.2. The AI-assisted analysis is used to compute the FCAR reported in this paper. Generative AI tools are not used to determine the main conclusions of this work. The authors take full responsibility for the content of this work and for all AI-assisted outputs used in the research process.

REFERENCES

AIME24. https://huggingface.co/datasets/HuggingFaceH4/aime_2024, 2024.

AIME25. https://huggingface.co/datasets/opencompass/AIME2025, 2025.

AMC23. https://huggingface.co/datasets/zwhe99/amc23, 2023.

Simon A Aytes, Jinheon Baek, and Sung Ju Hwang. Sketch-of-thought: Efficient llm reasoning with adaptive cognitive-inspired sketching. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 24307–24331, 2025.

Kaiyan Chang, Chenwei Zhu, Yingfeng Luo, Yifu Huo, Chenglong Wang, Xiaoqian Liu, Qiaozhi He, Tong Xiao, Zhengtao Yu, and Jingbo Zhu. Apr: Penalizing structural redundancy in large reasoning models via anchor-based process rewards. *arXiv preprint arXiv:2602.00760*, 2026.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, et al. Do not think that much for 2+ 3=? on the overthinking of o1-like llms. *arXiv preprint arXiv:2412.21187*, 2024.

Zhengxiang Cheng, Dongping Chen, Mingyang Fu, and Tianyi Zhou. Optimizing length compression in large reasoning models. *arXiv preprint arXiv:2506.14755*, 2025.

Razvan-Gabriel Dumitru, Darius Peteleaza, Vikas Yadav, and Liangming Pan. Conciserl: Conciseness-guided reinforcement learning for efficient reasoning models. *arXiv preprint arXiv:2505.17250*, 4, 2025.

Gongfan Fang, Xinyin Ma, and Xinchao Wang. Thinkless: Llm learns when to think. *Advances in neural information processing systems*, 38:151268–151295, 2026.

Xinyu Guan, Li Lyna Zhang, Yifei Liu, Ning Shang, Youran Sun, Yi Zhu, Fan Yang, and Mao Yang. Rstar-math: Small llms can master math reasoning with self-evolved deep thinking. *arXiv preprint arXiv:2501.04519*, 2025.

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

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

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

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

Jialiang Hong, Taihang Zhen, Kai Chen, Jiaheng Liu, Junlan Feng, Wenpeng Zhu, Jing Huo, Yang Gao, Depeng Wang, Haitao Wan, et al. Reconsidering overthinking: Penalizing internal and external redundancy in cot reasoning. *arXiv preprint arXiv:2508.02178*, 2025.

Bairu Hou, Yang Zhang, Jiabao Ji, Yujian Liu, Kaizhi Qian, Jacob Andreas, and Shiyu Chang. Thinkprune: Pruning long chain-of-thought of llms via reinforcement learning. *arXiv preprint arXiv:2504.01296*, 2025.

Chenzhi Hu, Qinzhe Hu, Yuhang Xu, Junyi Chen, Ruijie Wang, Shengzhong Liu, Jianxin Li, Fan Wu, and Guihai Chen. Smartthinker: Progressive chain-of-thought length calibration for efficient large language model reasoning. *arXiv preprint arXiv:2603.08000*, 2026.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

Huiyuan Lai and Malvina Nissim. Tacler: Tailored curriculum reinforcement learning for efficient reasoning. *arXiv preprint arXiv:2601.21711*, 2026.

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

Ruosen Li, Ziming Luo, Quan Zhang, Ruochen Li, Ben Zhou, Ali Payani, and Xinya Du. Aalc: Large language model efficient reasoning via adaptive accuracy-length control. *arXiv preprint arXiv:2506.20160*, 2025.

Haotian Luo, Li Shen, Haiying He, Yibo Wang, Shiwei Liu, Wei Li, Naiqiang Tan, Xiaochun Cao, and Dacheng Tao. O1-pruner: Length-harmonizing fine-tuning for o1-like reasoning pruning. *arXiv preprint arXiv:2501.12570*, 2025.

Ruotian Ma, Peisong Wang, Cheng Liu, Xingyan Liu, Jiaqi Chen, Bang Zhang, Xin Zhou, Nan Du, and Jia Li. S2r: Teaching llms to self-verify and self-correct via reinforcement learning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 22632–22654, 2025.

Yansong Ning, Wei Li, Jun Fang, Naiqiang Tan, and Hao Liu. Not all thoughts are generated equal: Efficient llm reasoning via multi-turn reinforcement learning. *arXiv preprint arXiv:2505.11827*, 2025.

Fiorenzo Parascandolo, Wenhui Tan, Enver Sangineto, Ruihua Song, and Rita Cucchiara. Bfs-po: Best-first search for large reasoning models. *arXiv preprint arXiv:2602.14917*, 2026.

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

Yi Shen, Jian Zhang, Jieyun Huang, Shuming Shi, Wenjing Zhang, Jiangze Yan, Ning Wang, Kai Wang, Zhaoxiang Liu, and Shiguo Lian. Dast: Difficulty-adaptive slow-thinking for large reasoning models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pp. 2322–2331, 2025.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*, 2025.

Qian Wan, Ziao Xu, Luona Wei, Xiaoxuan Shen, and Jianwen Sun. Mitigating overthinking in large reasoning models via difficulty-aware reinforcement learning. *arXiv preprint arXiv:2601.21418*, 2026.

Yue Wang, Qiuzhi Liu, Jiahao Xu, Tian Liang, Xingyu Chen, Zhiwei He, Linfeng Song, Dian Yu, Juntao Li, Zhuosheng Zhang, et al. Thoughts are all over the place: On the underthinking of o1-like llms. *arXiv preprint arXiv:2501.18585*, 2025.

Canhui Wu, Qiong Cao, Chang Li, Zhenfang Wang, Chao Xue, Yuwei Fan, Wei Xi, and Xiaodong He. Beyond token length: Step pruner for efficient and accurate reasoning in large language models. *arXiv preprint arXiv:2510.03805*, 2025.

Violet Xiang, Chase Blagden, Rafael Rafailov, Nathan Lile, Sang Truong, Chelsea Finn, and Nick Haber. Just enough thinking: Efficient reasoning with adaptive length penalties reinforcement learning. *arXiv preprint arXiv:2506.05256*, 2025.

Silei Xu, Wenhao Xie, Lingxiao Zhao, and Pengcheng He. Chain of draft: Thinking faster by writing less. *arXiv preprint arXiv:2502.18600*, 2025.

Boyang Xue, Qi Zhu, Hongru Wang, Rui Wang, Sheng Wang, Hongling Xu, Fei Mi, Yasheng Wang, Lifeng Shang, Qun Liu, et al. Dast: Difficulty-aware self-training on large language models. *arXiv preprint arXiv:2503.09029*, 2025.

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

Chenxu Yang, Qingyi Si, Yongjie Duan, Zheliang Zhu, Chenyu Zhu, Qiaowei Li, Minghui Chen, Zheng Lin, and Weiping Wang. Dynamic early exit in reasoning models. *arXiv preprint arXiv:2504.15895*, 2025b.

Zewei Yu, Lirong Gao, Yuke Zhu, Bo Zheng, Junbo Zhao, Sheng Guo, and Haobo Wang. Stop unnecessary reflection: Training lrms for efficient reasoning with adaptive reflection and length coordinated penalty. *arXiv preprint arXiv:2602.12113*, 2026.

Jiajie Zhang, Nianyi Lin, Lei Hou, Ling Feng, and Juanzi Li. Adaptthink: Reasoning models can learn when to think. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 3716–3730, 2025.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Guoyin Wang, et al. Instruction tuning for large language models: A survey. *ACM Computing Surveys*, 58(7):1–36, 2026.

Table 5: Detailed reflection behavior results across six mathematical reasoning benchmarks.

Method	AIME24		AIME25		AMC23		MATH500		Minervamath		Olympiad	
	ARN	CRS	ARN	CRS	ARN	CRS	ARN	CRS	ARN	CRS	ARN	CRS
Vanilla	140	88	142	83	74	104	40	116	65	94	96	89
CoD	124	92	131	85	131	85	35	115	39	119	85	91
SFT	107	107	120	93	56	122	33	126	54	101	83	99
DPO	193	55	205	53	112	58	59	65	87	62	146	53
O1-Pruner	59	146	59	151	29	215	16	252	23	143	39	143
AdaptThink	82	115	92	103	26	156	12	165	14	149	55	108
ARLCP	70	110	74	99	24	153	12	173	17	151	42	112
EARR	**42**	**158**	**41**	**155**	**14**	**239**	**7**	**272**	**10**	**220**	**25**	**162**

Table 6: Detailed FCAR results across six mathematical reasoning benchmarks.

Method	AIME24	AIME25	AMC23	MATH500	Minervamath	Olympiad
Vanilla	53.5%	59.0%	46.6%	36.2%	33.4%	39.8%
CoD	57.6%	63.5%	48.5%	42.4%	38.0%	41.8%
SFT	59.1%	59.8%	50.1%	39.2%	39.2%	39.4%
DPO	53.4%	51.5%	46.5%	38.4%	34.6%	40.6%
O1-Pruner	57.8%	62.4%	48.3%	41.1%	38.7%	41.8%
AdaptThink	58.4%	62.3%	52.3%	43.2%	38.7%	45.9%
ARLCP	**61.2%**	64.2%	54.2%	42.1%	38.9%	48.1%
EARR	60.2%	**68.6%**	**57.5%**	**46.9%**	**40.7%**	**51.5%**

APPENDIX

A DETAILED REASONING BEHAVIOR RESULTS

This section provides the detailed per-dataset results corresponding to the reasoning behavior analysis in Section 5.6.

A.1 DETAILED REFLECTION PATTERN RESULTS

Table 5 reports the detailed reflection pattern results on each of the six mathematical reasoning benchmarks. We report the ARN and CRS for all compared methods. These per-dataset results correspond to the averaged results presented in Section 5.6.1.

A.2 DETAILED FCAR RESULTS

Table 6 presents the detailed FCAR results on each of the six mathematical reasoning benchmarks. These results correspond to the averaged results reported in Section 5.6.2 and provide a more fine-grained comparison across datasets.

B EVALUATION ON A LARGER MODEL

To further evaluate the scalability of EARR on larger models, we conduct experiments on Qwen3-8B. We follow the same experimental setting as the main experiments and evaluate both accuracy and reasoning length on six mathematical reasoning benchmarks. As shown in Table 7, EARR effectively reduces reasoning overhead on Qwen3-8B, achieving a 48.8% reduction in average reasoning length while improving the average accuracy by 1.3 points. These results demonstrate that EARR is not limited to smaller models; it remains effective in reducing redundant reasoning and maintaining strong reasoning performance as model size increases.

Table 7: Comparison of Vanilla and EARR on accuracy and reasoning length across six mathematical reasoning benchmarks.

Method	AIME24		AIME25		AMC23		MATH500		Minervamath		Olympiad		Overall	
	Acc	Length	Acc	Length	Acc	Length	Acc	Length	Acc	Length	Acc	Length	ΔAcc	ΔLength
Vanilla	65.5	12097	55.0	13404	91.7	8243	93.0	5343	**41.7**	7016	65.4	9537	–	–
EARR	**68.7**	**8187**	**58.1**	**9136**	**92.7**	**3801**	**94.0**	**2139**	40.7	**1958**	**66.4**	**5479**	**+1.3**	**-48.8%**

Table 8: Sensitivity analysis of the reward weight coefficients α, β, and γ on DeepSeek-R1-Distill-Qwen-1.5B across six mathematical reasoning benchmarks.

Method	AIME24		AIME25		AMC23		MATH500		Minervamath		Olympiad		Overall	
	Acc	Length	Acc	Length	Acc	Length	Acc	Length	Acc	Length	Acc	Length	ΔAcc	ΔLength
Vanilla	26.3	12340	21.3	11861	65.6	7724	81.1	4675	25.1	6212	42.1	8686	–	–
$\alpha = 0.05$	30.0	8878	**23.1**	8539	73.3	4671	**84.4**	2751	**28.0**	3387	**48.1**	5792	**+4.2**	-35.9%
$\alpha = 0.30$	27.5	6681	22.1	**6191**	72.7	**3154**	80.8	**1677**	26.1	**1959**	44.5	**4055**	+2.0	**-56.5%**
$\beta = 0.01$	**32.1**	8704	22.7	8485	74.1	4291	83.6	2419	27.6	3131	46.7	5479	**+4.2**	-39.5%
$\beta = 0.10$	29.0	6933	22.7	6237	73.0	3429	82.0	1834	26.5	2115	46.4	4050	+3.0	-54.5%
$\gamma = 0.01$	31.3	8024	**23.1**	7321	70.6	3909	83.9	2216	**28.0**	2590	45.8	5006	+3.5	-46.0%
$\gamma = 0.10$	27.5	6897	22.7	6443	71.7	3492	82.7	2004	24.3	2234	46.9	4110	+2.4	-53.1%
EARR	31.7	**6652**	22.5	6505	**75.2**	3460	83.1	1931	26.6	2247	46.7	4206	+4.0	-53.4%

C Reward Weight Coefficients

In our main experiments, we set the reward weight coefficients to $\alpha = 0.15$, $\beta = 0.05$, and $\gamma = 0.05$, which control the contributions of reasoning length, reflection frequency, and continuous reasoning span, respectively. To investigate the sensitivity of EARR to these coefficients, we vary α, β, and γ individually while keeping the other coefficients fixed at their default values. We conduct experiments on DeepSeek-R1-Distill-Qwen-1.5B and evaluate different coefficient settings across six mathematical reasoning benchmarks. The results are reported in Table 8.

The results show that EARR remains effective across different coefficient settings, consistently reducing reasoning length while maintaining or improving overall accuracy. Different coefficient values exhibit different trade-offs between reasoning accuracy and efficiency. In particular, increasing α imposes a stronger constraint on reasoning length, leading to more aggressive length reduction, whereas a smaller α preserves more reasoning and tends to achieve higher accuracy. Similar trade-offs can also be observed when varying β and γ, indicating that the relative strengths of different behavioral penalties affect the balance between reasoning performance and efficiency. Overall, EARR exhibits relatively stable performance across different reward weight settings. We therefore adopt $\alpha = 0.15$, $\beta = 0.05$, and $\gamma = 0.05$ as the default setting to achieve a balanced trade-off between reasoning accuracy and efficiency.

D Reflection words

We regard reflective words or phrases as observable evidence of reflection behavior and count their occurrences in a reasoning trajectory as the number of reflections. In our main experiments, we use the following set of reflection words:

{`wait`, `alternatively`, `hold on`, `another thought`, `verify`, `think again`, `but`, `however`, `alternative`, `check`, `double-check`, `oh`, `hmm`}.

Since different models may express reflection using different lexical patterns, a potential concern is that some models may not frequently produce such a diverse set of reflection words. To examine whether EARR depends on a comprehensive reflection-word set, we progressively vary the words used to identify reflection behavior. As shown in Table 9, using only `wait` as the reflection indicator

```
You need to determine the earliest step where the model response first fully provides the standard answer.

The input includes:
query: the problem
answer: the standard answer
model response: the model response split into step0, step1, step2, ...

Judging rules:
-Scan the steps from step0 upward and stop at the first step that already gives a complete answer equivalent to the standard answer.
-Do NOT wait for a boxed answer, a "Final Answer" section, or a repeated summary if an earlier step already contains the complete answer.
-The answer does not need to appear in exactly the same format as the standard answer. Equivalent forms count as complete answers.
-Only count a step if it provides the full final quantity asked by the problem. Intermediate results or partial results do not count.

For example:
-If the problem asks for total time including coffee-shop time, a step giving only walking time does not count.
-If the problem asks for a point (r, theta), a step giving only r or only theta does not count.
-If a step only gives an intermediate variable such as s or t, it does not count.
-If a step states the complete final answer in a different unit or equivalent expression, count that step.
-If a later step converts, verifies, boxes, repeats, or summarizes the same answer, prefer the earlier complete step.
-If no step fully provides a complete answer equivalent to the standard answer, output None.

Output only the step number, in the exact format: stepn, or None.
Do not output any explanation, punctuation, or extra text.
Now judge based on the following content:
query: {query}
answer: {answer}
model response: {model_response}
```

Figure 5: Prompt for identifying the first correct answer step

Table 9: Effect of different reflection-word sets on reasoning accuracy and efficiency. W denotes `wait`; WAV denotes `wait`, `alternatively`, and `verify`; WAVHC further includes `however` and `check`.

Method	AIME24		AIME25		AMC23		MATH500		Minervamath		Olympiad		Overall	
	Acc	Length	Acc	Length	Acc	Length	Acc	Length	Acc	Length	Acc	Length	ΔAcc	ΔLength
Vanilla	26.3	12340	21.3	11861	65.6	7724	81.1	4675	25.1	6212	42.1	8686	–	–
W	**34.6**	8001	23.1	7688	75.0	4056	**84.3**	2343	**27.9**	2860	**47.5**	5284	**+5.1**	-43.5%
WAV	30.4	7830	23.1	7797	74.1	3872	84.2	2260	27.1	2649	44.6	5132	+3.7	-45.1%
WAVHC	30.4	7289	**24.8**	6822	72.3	3787	82.8	2225	25.8	2535	45.7	4698	+3.4	-48.6%
EARR	31.7	**6652**	22.5	**6505**	**75.2**	**3460**	83.1	**1931**	26.6	**2247**	46.7	**4206**	+4.0	**-53.4%**

already achieves a strong performance, improving average accuracy by 5.1 points while reducing response length by 43.5%. This result suggests that EARR does not require a large or exhaustive reflection lexicon to be effective, and even a single explicit reflection indicator can provide a useful signal for regulating redundant reasoning.

Expanding the reflection-word set further improves reasoning efficiency, but introduces a moderate decrease in accuracy. Specifically, using `wait`, `alternatively`, and `verify` achieves a 45.1% length reduction with a 3.7-point accuracy improvement, while further including `however` and `check` increases the length reduction to 48.6%, with an accuracy improvement of 3.4 points. One possible explanation is that `wait` is a relatively explicit indicator of reconsideration, whereas some other words are more context-dependent. For example, `however` or `but` may simply serve as discourse connectives, while `check` or `verify` may describe a local reasoning operation without indicating an actual change in reasoning direction. Consequently, expanding the lexicon increases the coverage of potential reflection behaviors but may also introduce false-positive detections, causing some normal reasoning steps to be counted as reflections and penalized during training. This may explain the stronger reduction in response length together with the moderate decrease in accuracy.

Using the full reflection-word set, EARR achieves a 53.4% reduction in response length while still improving average accuracy by 4.0 points. Overall, these results indicate that reflection words serve as a lightweight proxy for observable reflection behavior rather than requiring an exact or exhaustive lexical definition. EARR remains effective even when only a small number of explicit reflection indicators are available, while a broader reflection-word set provides stronger control over reasoning redundancy at the cost of potentially introducing additional detection noise.

E PROMPTS

To identify the position at which the model first obtains the correct answer during reasoning, we first split the original response using double newline characters (`\n\n`) and index the resulting segments sequentially as `step0`, `step1`, ..., `stepN`. We then use an LLM-based evaluator to scan the steps from the beginning and identify the earliest step that provides a complete answer equivalent to the ground-truth answer.

We use step-level localization to obtain a consistent and easily parsable positional reference, avoiding ambiguities caused by text spans or natural-language descriptions returned by the evaluator. The final position is still measured at the token level: if the correct answer first appears in `stepk`, its position is defined as the cumulative number of tokens from `step0` through `stepk`. This position is then used to compute the FCAR.