PEAR: PHASE ENTROPY AWARE REWARD FOR EFFICIENT REASONING

Chen Huang[1] **Wei Lu**[2] **Wenxuan Zhang**[1]
[1]Singapore University of Technology and Design [2]Nanyang Technological University
chen_huang@mymail.sutd.edu.sg wei.lu@ntu.edu.sg
wxzhang@sutd.edu.sg

ABSTRACT

Large Reasoning Models (LRMs) have achieved impressive performance on complex reasoning tasks by generating detailed chain-of-thought (CoT) explanations. However, these responses are often excessively long, containing redundant reasoning steps that inflate inference cost and reduce usability. Controlling the length of generated reasoning without sacrificing accuracy remains an open challenge. Through a systematic empirical analysis, we reveal a consistent positive correlation between model entropy and response length at different reasoning stages across diverse LRMs: the thinking phase exhibits higher entropy, reflecting exploratory behavior of longer responses, while the final answer phase shows lower entropy, indicating a more deterministic solution. This observation suggests that entropy at different reasoning stages can serve as a control knob for balancing conciseness and performance. Based on this insight, this paper introduces **P**hase **E**ntropy **A**ware **R**eward (PEAR), a reward mechanism that incorporates phase-dependent entropy into the reward design. Instead of treating all tokens uniformly, PEAR penalize excessive entropy during the thinking phase, allowing moderate exploration at the final answer phase, which encourages models to generate concise reasoning traces that retain sufficient flexibility to solve the task correctly. This enables adaptive control of response length without relying on explicit length targets or rigid truncation rules. Extensive experiments across six benchmarks demonstrate that PEAR consistently reduces response length while sustaining competitive accuracy across model scales. In addition, PEAR demonstrates strong out-of-distribution (OOD) robustness beyond the training distribution. Our code is available at: `https://github.com/iNLP-Lab/PEAR`.

1 INTRODUCTION

Large Language Models (LLMs) have demonstrated remarkable reasoning capabilities, particularly when employing techniques like Chain-of-Thought (COT) prompting (Wei et al., 2022). Building on this, recent Large Reasoning Models (LRMs) (Jaech et al., 2024; Guo et al., 2025; Yang et al., 2025a; Team et al., 2025; Team, 2025) encourage an explicit thinking phase via special tokens before generating the final answer, further improving models' complex problem-solving capability. However, LRMs tend to generate excessively long chain-of-thought responses (Chen et al., 2024; Yue et al., 2025), the models often produce redundant calculations or verbose explanations, which leads to bloated outputs and reduces inference efficiency (Hassid et al., 2025; Kuo et al., 2025). Consequently, a key challenge is to enable models to think less while preserving the performance.

Recent works have attempted to address this issue by enforcing efficiency through further training on filtered concise data (Yue et al., 2025; Qu et al., 2025; Sui et al., 2025). The common paradigm is to modify the training corpus so that the model is exposed primarily to shorter reasoning traces (Yuan et al., 2025; An et al., 2025; Zhao et al., 2025b). By strictly constraining the supervision signal, the model often struggles to adapt to novel reasoning styles or out-of-distribution (OOD) problems where the optimal length of reasoning may differ (Aggarwal & Welleck, 2025). Moreover, such methods risk discarding valuable intermediate reasoning that could improve accuracy. This motivates the need for a more adaptive and model-driven approach to efficient reasoning.

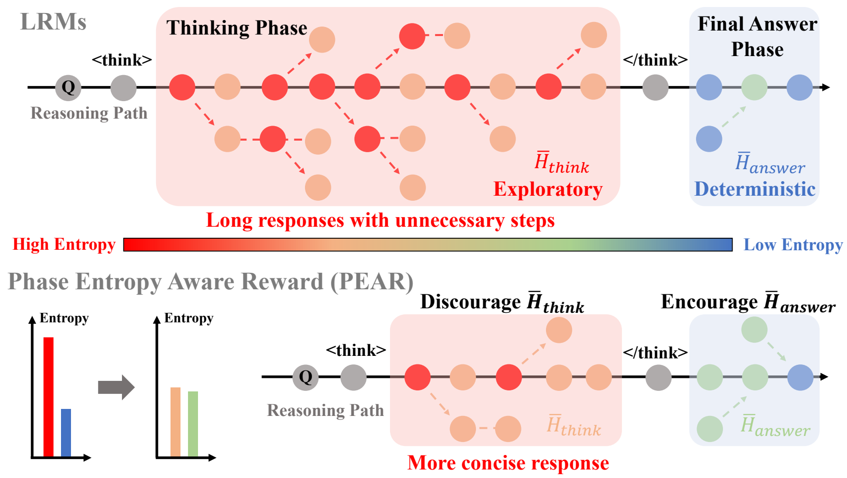

Figure 1: PEAR reduces the response length by penalizing excessive entropy during the thinking phase while allowing moderate exploration at the final answer phase.

Concurrently, there has been growing interest in understanding how token-level uncertainty, as measured by entropy, influences model behavior (Lei et al., 2025; Cheng et al., 2025a; Zhang et al., 2025b). Entropy captures the spread of the predictive distribution: high-entropy segments often correspond to exploratory reasoning steps where the model searches for a correct path, while low-entropy segments capture more deterministic computations or final answer generation (Wang et al., 2025c; Zhang et al., 2025f). Therefore, recent works have begun to exploit these signals for improving calibration or enhancing reasoning robustness (Zhang et al., 2025c; Wang et al., 2025b). However, the connection between entropy and efficient reasoning has been largely overlooked.

Intuitively, a model that operates at consistently high entropy may explore too broadly and thus produce unnecessarily long reasoning chains, while a model biased toward low entropy may commit earlier to a determined reasoning path with more concise outputs. Motivated by this hypothesis, we first conduct empirical analysis, and observe a consistent positive correlation between average token-level entropy and response length across model scales and benchmarks. Interestingly, this relationship is not uniform across reasoning stages: the "thinking" portion of the output exhibits substantially higher entropy than the "final answer" portion, highlighting distinct roles of exploration and commitment in different stages of reasoning. Moreover, when we filter out high-entropy tokens, models' performance will not be affected within a certain ratio, suggesting that excessive entropy can be pruned without harming reasoning quality. Based on these observations, we propose **P**hase **E**ntropy **A**ware **R**eward (**PEAR**), a reward mechanism that explicitly decomposes entropy into thinking and final answer phases and integrates both components into the training objective. As illustrated in Figure 1, by penalizing excessive entropy during the thinking phase while moderating entropy in the final answer phase, PEAR encourages models to produce more concise reasoning traces, providing a soft and adaptive mechanism for balancing exploration with efficiency.

We evaluate PEAR on six widely used benchmarks: GSM8K (Cobbe et al., 2021), MATH500 (Hendrycks et al., 2021b), AIME24 (Li et al., 2024), AMC23 (Li et al., 2024), GPQA Diamond (Rein et al., 2024) and MMLU (Hendrycks et al., 2021a). Across models of different scales, PEAR achieves substantial reductions in response length, ranging from 32.4% to 56.6%, while preserving accuracy with decreases of less than 1%. By incorporating both phases of a model's response into the reward calculation, PEAR eliminates the need for manual data curation and generalizes effectively to out-of-domain questions through its broadly applicable training objective.

To summarize, our work makes the following key contributions:

- We empirically establish and validate a positive correlation between model entropy and response length in LRMs, and show that the thinking phase exhibits substantially higher entropy than the final answer phase.
- We introduce Phase Entropy Aware Reward (PEAR), a reward mechanism that leverages this property to adaptively promote concise reasoning traces without depending on curated datasets or explicit length constraints.

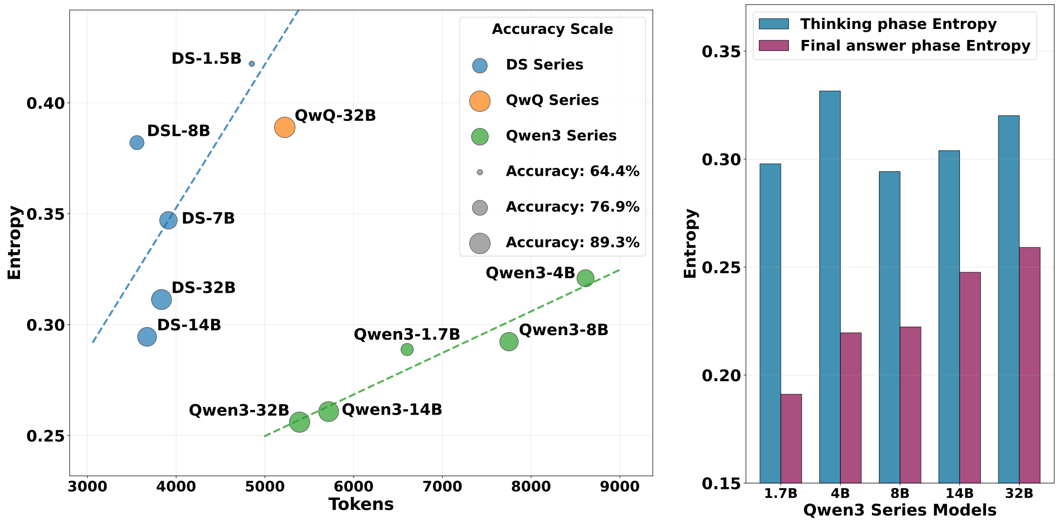

Figure 2: (a) Relationship between average entropy and response length across different models. The dot size indicates accuracy. DS(L) represents DeepSeek-R1-Distill-Qwen/Llama. (b) Comparison of average entropy between the thinking phase and the final answer phase.

- We provide extensive experimental evidence on GSM8K, MATH500, AIME24, AMC23, GPQA and MMLU, showing that our method achieves substantial reductions in response length while preserving accuracy, with strong generalization capability to out-of-distribution tasks.

2 Preliminary Analysis

In this section, we present empirical observations that motivate our approach. We first examine the relationship between entropy and response length, showing how higher entropy is associated with longer reasoning traces. Next, we differentiate the roles of entropy in the thinking phase versus the final answer phase, highlighting distinct patterns across stages. Finally, we conduct entropy-filtering experiments to demonstrate the robustness of low-entropy reasoning traces. All analyses are performed on GSM8K, MATH500, AIME24, and AMC23, where we report average accuracy, response length (in tokens), and entropy.

2.1 Entropy and Response Length

We begin by analyzing the correlation between response entropy and length across a diverse set of LRMs. For each model, we measure the average entropy of the predictive distribution across all generated tokens and compare it against the total number of tokens produced during inference.

The entropy of the predictive distribution at each token position t is defined as

$$H_t = -\sum_{i=1}^{|V|} p_i^{(t)} \log p_i^{(t)}, \qquad \bar{H} = \frac{1}{T}\sum_{t=1}^{T} H_t \tag{1}$$

where $p_i^{(t)}$ denotes the predicted probability of token i at position t, $|V|$ is the vocabulary size, T is the total response length, and $\bar{H}$ is the average entropy across the entire response.

Figure 2(a) shows a consistent positive correlation between average entropy and response length across all examined model families and benchmarks. Responses with higher entropy are typically longer and more exploratory, while lower entropy corresponds to shorter and more concise traces. This pattern is especially evident within individual model series, where models of different scales exhibit a clear alignment between entropy levels and response characteristics.

These findings suggest that the entropy–length relationship is a fundamental property of large reasoning models. Longer responses naturally reflect higher uncertainty or diversity in token predictions, as captured by increased entropy. This makes entropy an interpretable internal signal for shaping model behavior. By integrating entropy into the reward design, we can provide models with a principled mechanism to balance thorough reasoning with efficient generation, enabling finer control over response length without relying on explicit constraints.

2.2 Phase-Dependent Entropy Analysis

To further investigate the role of entropy in model responses, we analyze how entropy is distributed across different stages of generation. As shown in Figure 2(b), a clear distinction emerges between the thinking phase (before the </think> token) and the final answer phase (after the </think> token). The thinking phase exhibits consistently higher entropy, reflecting exploratory behavior as the model searches through multiple potential reasoning paths and generates longer, more diverse traces. In contrast, the final answer phase shows much lower entropy, indicating a more confident and deterministic commitment to a specific solution. These results indicate that the two phases serve complementary functions of exploration versus conclusion and should therefore be optimized differently. Phase-specific reward mechanisms can leverage this distinction, reducing unnecessary exploration during reasoning while preserving confidence and clarity in final answers.

2.3 Entropy Filtering Experiments

To assess how high-entropy tokens influence model reasoning and whether pruning them impacts reasoning quality, we conduct a systematic filtering experiment, as shown in Figure 3. Our procedure consists of two stages: first, we generate complete reasoning traces and compute token-level entropy within the thinking phase. Second, we preserve only a specified percentage of tokens with the lowest entropy and discard the remaining high-entropy tokens, resulting in filtered reasoning traces consist of only low-entropy tokens. These filtered traces are then fed back to the model to produce final answers, enabling us to directly examine how entropy-based filtering influences both reasoning efficiency and task accuracy. Results for more models can be found at Appendix B.

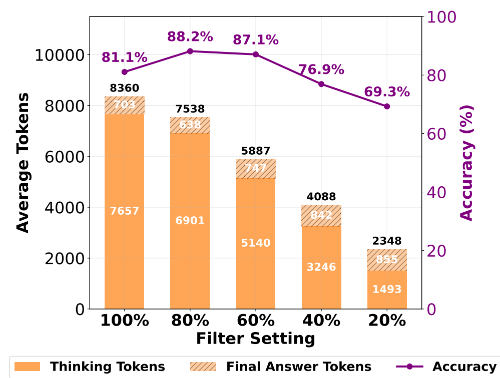

Figure 3: Accuracy and average response length in the entropy filtering experiments on Qwen3-4B.

When retaining 80% or 60% of low-entropy tokens, accuracy remains stable or even improves compared to the unfiltered baseline. This indicates that the high-entropy tokens being removed mainly drive excessive exploration rather than contributing to correct reasoning, and their absence reduces noise in the reasoning process. Performance degradation only emerges under more aggressive filtering: retaining 40% or fewer low-entropy tokens leads to a sharp drop in accuracy, showing that essential reasoning steps are lost when the trace is compressed too heavily. Notably, the length of the final answer phase remains relatively unchanged across filtering levels, reinforcing that redundancy is concentrated in the thinking phase, where high-entropy tokens leads to over-elaboration and inflates response length without improving outcomes.

3 Method

3.1 Group Relative Policy Optimization (GRPO)

We begin with a brief introduction to the Group Relative Policy Optimization (GRPO) algorithm (Shao et al., 2024). Unlike standard PPO (Schulman et al., 2017), GRPO eliminates the need for a critic model by estimating advantages through reward normalization across a group of sampled responses to the same prompt. Specifically, for a prompt q with G responses and corresponding rewards $\{r_i\}_{i=1}^{G}$, the group-normalized advantage is defined as:

$$\hat{A}_{i,t} = \frac{r_i - \mathrm{mean}(\{r_j\}_{j=1}^{G})}{\mathrm{std}(\{r_j\}_{j=1}^{G})}. \tag{2}$$

This normalization emphasizes the differences among candidate outputs for the same question, which improves the stability of the gradient signal even under sparse reward settings. GRPO also incorporates a KL divergence term that regularizes the learned policy against a reference policy. The overall surrogate objective can be written as:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{q\sim P(Q),\{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|q)}$$
$$\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\left\{\min\Big[r_{i,t}(\theta)\hat{A}_{i,t},\ \text{clip}\big(r_{i,t}(\theta),1-\epsilon,1+\epsilon\big)\hat{A}_{i,t}\Big] - \beta D_{\text{KL}}\big[\pi_\theta \,\|\, \pi_{\text{ref}}\big]\right\}. \tag{3}$$

where

$$r_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t} \mid q, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t} \mid q, o_{i,<t})}, \tag{4}$$

ϵ and β are hyperparameters, and D_{KL} denotes the KL divergence between the learned policy π_θ and a reference policy π_{ref}.

3.2 Phase Entropy Aware Reward (PEAR)

In the original GRPO algorithm, the reward r is typically defined in a rule-based manner, assigning a value of 1 to correct responses and 0 to incorrect ones. While simple and effective, this binary scheme overlooks richer characteristics of the response, such as the degree of exploration or reflection embedded in the reasoning trajectory. As a result, it provides no guidance on how the model should balance exploratory reasoning with concise and reliable answer generation.

Building on the observed correlation between model entropy and response length in Section 2, we introduce **Phase Entropy Aware Reward (PEAR)** that leverages entropy as guidance to train models to reason more efficiently. Let a sampled response be the token sequence $y = (y_1, \ldots, y_T)$ that contains a thinking segment between <think> and </think> followed by the final answer. Let k denote the index of the closing token </think> in y. We compute token entropies with respect to the old policy $\pi_{\theta_{\text{old}}}$:

$$H_t \;=\; -\sum_{v\in\mathcal{V}} \pi_{\theta_{\text{old}}}(v \mid y_{<t}) \log \pi_{\theta_{\text{old}}}(v \mid y_{<t}), \qquad t = 1, \ldots, T. \tag{5}$$

We then average entropies for the thinking phase and final answer phase (excluding the </think> token itself):

$$\bar{H}_{\text{think}} = \frac{1}{k-1}\sum_{t=1}^{k-1} H_t, \qquad\qquad \bar{H}_{\text{answer}} = \frac{1}{T-k}\sum_{t=k+1}^{T} H_t. \tag{6}$$

The phase reward $\mathcal{P}$ integrates entropy from both the thinking and final answer phases, defined as:

$$\mathcal{P}(y) \;=\; \max\!\big(0,\ \bar{H}_{\text{think}} - \alpha\,\bar{H}_{\text{answer}}\big). \tag{7}$$

The $\max(0, \cdot)$ ensures that the entropy-based term never becomes negative, keeping the combined reward within the canonical $[0, 1]$ range used in GRPO and avoiding pathological scaling of advantages. The policy update remains the same clipped-surrogate objective as in GRPO, PEAR only changes the per-sample scalar reward. Thus, the optimum under PEAR differs from standard GRPO only in preferring correct trajectories with lower excessive thinking-phase entropy where correctness remains the dominant signal, and the entropy term acts as a second-order preference among correct responses. The coefficient α is a tunable hyperparameter that adjusts the contribution of the final answer phase entropy, enabling flexible control over the balance between reasoning exploration and final answer confidence.

As discussed in Section 2.2, the reasoning process exhibits distinct entropy patterns: the thinking phase is characterized by higher entropy with exploratory behavior, while the final answer phase reflects lower entropy associated with deterministic solutions. To promote more efficient reasoning, we therefore aim to reduce entropy during the thinking phase to mitigate unnecessary exploration. Although the final answer phase already exhibits low entropy, the role of the $\alpha \bar{H}_{\text{answer}}$ term is not to encourage uncertainty in the answer. Instead, it makes the entropy penalty phase-relative rather than absolute. Subtracting final-answer entropy normalizes the thinking-phase entropy against the natural entropy level required to articulate a correct final answer. This prevents degenerate reward gaming through indiscriminate entropy collapse and ensures that the penalty activates primarily when thinking-phase entropy is disproportionately large.

Given a base score $s \in (0,1]$ for a correct final answer and a format score $r_{\text{fmt}} \in [0,1)$ for malformed/incorrect answers, the phase-aware entropy-inclusive reward for response y is:

$$r(y) \;=\; \begin{cases} \min\bigl(1,\; s - \mathcal{P}(y)\bigr), & \text{if the extracted answer equals the ground truth,} \\ r_{\text{fmt}}, & \text{otherwise.} \end{cases} \tag{8}$$

Finally, we replace r_i in Eq. (2) by $r(y_i)$ and keep the same GRPO advantage normalization:

$$A_i \;=\; \frac{r(y_i) - \text{mean}\bigl(\{r(y_j)\}_{j=1}^{G}\bigr)}{\text{std}\bigl(\{r(y_j)\}_{j=1}^{G}\bigr)}. \tag{9}$$

Edge cases. If `</think>` token is absent we set $k = T$ and use $\bar{H}_{\text{post}} = 0$ (i.e., only thinking phase entropy contributes); if the answer cannot be parsed, we assign $r(y) = r_{\text{fmt}}$. This fallback does not cause such responses to systematically receive higher rewards. PEAR applies entropy-based shaping only when the extracted final answer is correct. In practice, responses missing a closing `</think>` token are overwhelmingly malformed or incomplete and therefore fall into the incorrect category, receiving the fixed reward r_{fmt} independent of entropy.

With PEAR, the model is guided not only by final answer correctness but also by the quality of its reasoning behavior. The component for the thinking phase discourages excessive exploration, as high-entropy reasoning yields lower reward, thereby encouraging the model to generate more focused and efficient reasoning traces. Meanwhile, the component for the final answer phase helps stabilize and structure the concluding steps, ensuring that the model produces complete and coherent answers without sacrificing accuracy.

4 Results

4.1 Experiment Setting

Baseline Methods. GRPO (Group Relative Policy Optimization) (Shao et al., 2024) is a reinforcement learning framework that eliminates the need for a critic model by estimating advantages through reward normalization within a group of responses to the same prompt. **Step Entropy** (Li et al., 2025) adopts a two-stage training strategy that enables LLMs to generate compressed chain-of-thought (CoT) reasoning at inference time by strategically inserting [SKIP] tokens. **LCPO** (Length-Controlled Policy Optimization) (Aggarwal & Welleck, 2025) is a reinforcement learning method designed to jointly optimize for accuracy and compliance with user-specified length constraints.

Baseline Models. We evaluate our method on widely used LRMs, including DeepSeek-R1-Distill-Qwen-1.5B (Guo et al., 2025), Qwen3-4B, and Qwen3-8B (Yang et al., 2025a), which are commonly adopted in prior works. For fair comparison, we also report results on these baseline models across different model scales. Detailed implementation settings for all baseline methods are provided in Appendix C.

Training and Evaluation Setup. We conduct training using the open-source `verl` framework (Sheng et al., 2025), with 7,473 samples from GSM8K (Cobbe et al., 2021) as the training dataset for all models. The dataset consists of grade school math word problems, which are designed to evaluate question answering on basic mathematics that requires multi-step reasoning. The training configuration uses a batch size of 128 and a learning rate of 1×10^{-6}. We set the coefficient α

Table 1: Acc@1 results on mathematical reasoning benchmarks across LRMs. ↓ indicates the relative change with respect to the *Original* row of each model. PEAR consistently achieves the largest reduction in token usage across model scales, while maintaining comparable accuracy.

Method	GSM8K		MATH500		AIME24		AMC23		GPQA		MMLU		Average	
	Acc	Tok	Acc	Tok	Acc	Tok	Acc	Tok	Acc	Tok	Acc	Tok	Acc	Tok
DeepSeek-R1-Distill-Qwen-1.5B														
Original	85.97	1496	75.00	3620	26.66	8843	70.00	5253	36.36	7878	40.38	1739	55.73	4805
GRPO	87.86	1493	76.80	3132	33.33	7839	67.50	4899	40.90	8288	41.09	1868	57.91	4587 (↓ 4.5%)
Step Entropy	85.59	1629	76.80	3298	26.66	5640	70.00	4911	38.88	7095	40.67	1649	56.43	4037 (↓ 16.0%)
LCPO	87.11	2149	76.00	2895	26.66	5358	70.00	3324	37.87	6371	38.55	1450	56.03	3591 (↓ 25.3%)
PEAR	87.94	624	77.20	2358	23.33	5379	70.00	3705	38.38	6193	41.87	1242	56.45	3250 (↓ 32.4%)
Qwen3-4B														
Original	94.69	2634	85.40	5795	56.66	16792	87.50	9234	52.52	8184	72.32	1930	74.85	7428
GRPO	94.38	2321	84.80	5434	63.33	14061	90.00	8568	46.96	8244	73.01	1919	75.41	6758 (↓ 9.0%)
Step Entropy	94.84	2261	85.40	4704	60.00	9467	87.50	7317	53.53	7527	71.71	1889	75.50	5528 (↓ 25.6%)
LCPO	93.47	1846	84.20	3569	63.33	8528	85.00	6518	47.97	5073	73.94	1373	74.65	4485 (↓ 39.6%)
PEAR	94.01	1439	84.00	2695	56.66	5685	87.50	4173	49.49	4177	73.96	1154	74.27	3221 (↓ 56.6%)
Qwen3-8B														
Original	96.13	2335	86.60	5532	63.33	14977	90.00	8161	55.55	8572	73.28	1495	77.48	6845
GRPO	95.83	1999	85.20	5375	66.66	13195	90.00	7881	52.02	8745	75.54	1368	77.54	6427 (↓ 6.1%)
Step Entropy	95.14	2087	86.00	4658	60.00	6816	90.00	7352	57.07	6063	72.89	1454	76.85	4738 (↓ 30.8%)
LCPO	94.54	1645	85.00	4234	63.33	7173	82.50	6961	46.46	5229	73.23	1165	74.18	4401 (↓ 35.7%)
PEAR	94.54	1092	85.40	2664	60.00	6104	92.50	4045	54.04	4333	78.86	961	77.56	3200 (↓ 53.3%)

for the final answer phase reward calculation as 1. To evaluate the effectiveness and generalizability of our compression method, we conduct experiments on four standard mathematical reasoning benchmarks: GSM8K test set (Cobbe et al., 2021), MATH500 (Hendrycks et al., 2021b), AIME24 (Li et al., 2024), and AMC23 (Li et al., 2024). We also evaluate on two knowledge benchmarks: GPQA Diamond (Rein et al., 2024) and MMLU (Hendrycks et al., 2021a) to further demonstrate the out-of-domain capability of our method beyond mathematical reasoning. Descriptions of these benchmarks are provided in Appendix D.

Performance is measured along two dimensions: Accuracy (Acc) and the number of Generated Tokens (Tok), with a generation length cap of 16,384 tokens. Following the evaluation protocol of Guo et al. (2025), we adopt sampling with temperature set to 0.6 and top-p set to 0.95. Answer extraction and verification are carried out following the methodology of Yang et al. (2024).

4.2 Effectiveness of PEAR

As shown in Table 1, PEAR achieves the most substantial reduction in response length across all benchmarks and evaluated models, while maintaining accuracy at a level comparable to original models. Compared to original reasoning models, PEAR achieves an average response length reduction of 32.4% to 56.6%, while preserving the same performance with the decrease of only 0.58% in accuracy. This indicates that encouraging models to lower entropy level at the thinking phase during training provides an effective mechanism for eliminating redundant reasoning steps, thereby producing more concise outputs without compromising correctness.

Compared to the 1.5B model, the results for the 4B and 8B models suggest that larger models, which are prone to verbose reasoning, benefit more from PEAR by achieving over 50% reduction in response length. This supports the intuition that bigger models tend to "over-explain", creating greater opportunities for efficiency gains. Moreover, PEAR delivers a superior efficiency-accuracy trade-off on larger models relative to other baselines. In the case of Qwen3-8B, while Step Entropy and LCPO enforce shorter responses, they incur larger accuracy drops of 3.3%. In contrast, PEAR achieves even greater compression while remain the performance unchanged. This underscores PEAR's adaptive nature, enabling it to compress reasoning traces aggressively without compromising accuracy.

In addition, the benefits of PEAR extend beyond the training distribution, demonstrating strong OOD robustness. Although trained solely on the GSM8k training set, our method yields consistent efficiency improvements across all six benchmarks. For example, on Qwen3-4B, PEAR matches the

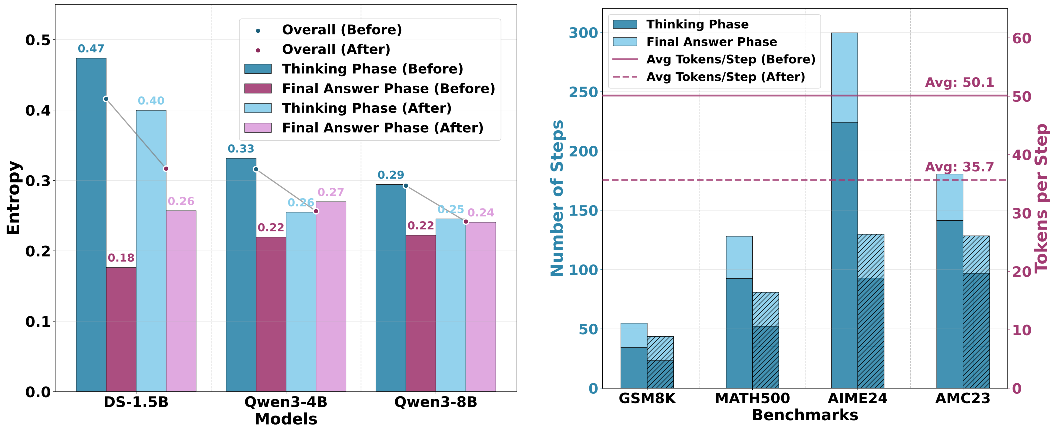

Figure 4: (a) Entropy changes before and after training with PEAR across thinking and final answer phases. (b) Changes in the number of reasoning steps and average tokens per step for Qwen3-4B. PEAR reduces both the number of reasoning steps and the average tokens per step.

vanilla model's accuracy on AIME24 and AMC23 while consuming only 34% and 45% of the original reasoning budget, respectively. Furthermore, on non-math reasoning benchmarks such as GPQA and MMLU, PEAR still achieves high efficiency without performance degradation, further demonstrating OOD robustness across tasks and domains. These results highlight that phase-dependent entropy serves as a universal, domain-agnostic signal for controlling reasoning efficiency, enabling our approach to generalize effectively across diverse tasks.

Overall, these results validate the central hypothesis of our work: incorporating phase-dependent entropy into the reward design enables LRMs to generate shorter and more efficient reasoning trajectories, while preserving accuracy and demonstrating strong generalization across domains. Extended experiments on additional model series and scales are in Appendix E.

4.3 How PEAR affects reasoning

We further analyze how PEAR influences model reasoning across different phases, focusing on changes in entropy, number of reasoning steps, and average tokens per step after training with PEAR.

As shown in Figure 4(a), PEAR consistently reduces the overall entropy across all evaluated models. Crucially, the largest reduction occurs in the thinking phase, where excessive exploration had previously contributed to unnecessarily long reasoning traces. This demonstrates that our reward effectively steers models toward more confident and focused reasoning, eliminating redundant exploratory steps in the thinking process. In contrast, the final answer phase shows a slight increase in entropy, indicating that the model retains flexibility when articulating its conclusions. Such phase-specific adjustments highlight PEAR's ability to suppress over-exploration during reasoning while still supporting diversity and completeness in the final answer through the control towards entropy.

Figure 4(b) illustrates the changes in the number of reasoning steps and tokens per step for the Qwen3-4B model across all benchmarks before and after applying PEAR. The results show that PEAR not only reduces the total number of reasoning steps but also decreases the average tokens per step, reflecting a shift toward more deterministic and efficient reasoning. Importantly, the reduction is concentrated in the thinking phase, consistent with PEAR's objective of discouraging excessive exploration while maintaining entropy in the final answer phase. This effect is especially pronounced on more challenging datasets such as AIME24, where the number of thinking steps is reduced by more than half. These results further validate the effectiveness of PEAR in producing concise reasoning trajectories without compromising solution quality.

Crucially, these findings explain why PEAR achieves substantial reductions in response length without sacrificing accuracy, highlighting phase-dependent entropy as a powerful control signal for balancing efficiency and performance in large reasoning models. We also include a case study in Appendix F to provide a qualitative comparison.

4.4 Hyperparameter study

A central hyperparameter in our reward design is the coefficient α for the final answer phase's entropy. This parameter directly controls the extent to which the model is encouraged for higher entropy in the final answer phase. Figure 5 illustrates the impact of the hyperparameter α on Qwen3-4B across four benchmarks. By default, α is set to a positive value in order to avoid "reward gaming", where the model drives entropy down indiscriminately to maximize reward, which often leads to degraded performance. Conceptually, although PEAR does not introduce a separate coefficient explicitly multiplying $\bar{H}_{\text{think}}$, the hyperparameter α implicitly controls the effective strength of the thinking-phase entropy penalty. According to Eq. (7), decreasing α increases the relative weight of $\bar{H}_{\text{think}}$ by enlarging the gap $\bar{H}_{\text{think}} - \alpha\bar{H}_{\text{answer}}$, while increasing α offsets the thinking-phase penalty through the final-answer entropy term. Thus, a single hyperparameter α jointly regulates the entropy penalties on both phases and determines how aggressively exploratory reasoning is suppressed.

The experiments confirm this hypothesis. When $\alpha = 0$, post-thinking entropy is ignored, and the model is optimized solely to minimize entropy in the thinking phase. While efficient, this strict reduction harms accuracy, as the model loses the flexibility needed in the answer phase to refine or adjust its predictions. The problem becomes even more pronounced when $\alpha = -1$, where both the reasoning and answer phases are simultaneously penalized for entropy. In this setting, the model is overly constrained, producing shorter but less reliable responses and further degrading performance.

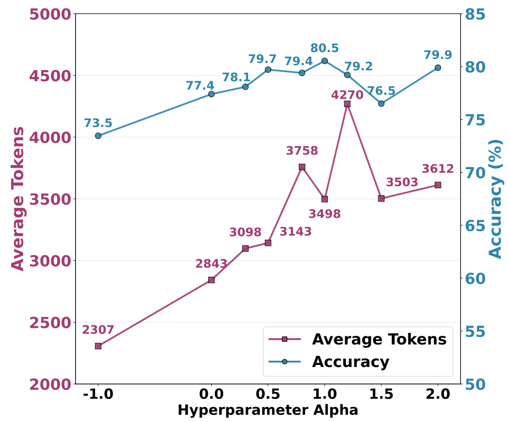

Figure 5: Average accuracy and response length of Qwen3-4B trained with different α.

As α increases, the penalty on post-thinking entropy becomes stronger. This relaxes the restrictive effect on the answer phase, allowing the model to preserve higher entropy where needed and thereby improving accuracy. At moderate values of α (e.g., 1), we observe a favorable balance: the model reduces redundancy in its reasoning while maintaining strong performance. However, when α is set too high, the penalty effect becomes negligible, and the model's behavior converges toward the baseline, producing longer responses and diminishing the efficiency gains. Overall, these results reveal a clear behavioral transition governed by α. Small α values lead to overly aggressive suppression of thinking-phase entropy, causing premature commitment and accuracy degradation. Large α values weaken the entropy penalty and recover baseline behavior with limited efficiency gains. The intermediate regime around $\alpha \approx 1$ consistently emerges as a stable operating point across benchmarks and model sizes, where PEAR selectively suppresses excessive, redundant exploration while preserving the core reasoning required for correctness. This explains why $\alpha \approx 1$ serves as a robust default choice in practice.

5 Related Work

5.1 Efficient Reasoning

A growing body of research has focused on improving the efficiency of LRMs. Early exit stops model dynamically once certain criteria has been reached (Liao et al., 2025). Typical methods include designing stopping rules based on internal reasoning state (Yang et al., 2025b; Qiao et al., 2025; Zhu et al., 2025; Xu et al., 2025), generation behavior (Wang et al., 2025a;d; Liu & Wang, 2025), or without relying on pre-defined triggers (Dai et al., 2025). Another complementary research direction focuses on compressing chain-of-thought reasoning traces, such as parallel thinking compression (Munkhbat et al., 2025; Ghosal et al., 2025), filtering or summarizing intermediate reasoning tokens and steps (Yu et al., 2025; Luo et al., 2025a; Yuan et al., 2025; Xia et al., 2025; Zhao et al., 2025a), and compression reward mechanisms (Cheng et al., 2025b; Zeng et al., 2025). Notably, Li et al. (2025) introduce step entropy for quantifying the informational contribution of each reasoning step within CoT trajectories, enabling selective removal of low-entropy steps. Besides, adaptive reasoning methods attempt to dynamically adjust the depth or length of reasoning depending on the difficulty of the input, this includes carefully designed reward (Jiang et al., 2025; Wang et al., 2025e; Luo et al., 2025b) and reasoning mode switching (Zhang et al., 2025d; Huang et al., 2025; Zhang et al., 2025a). For example, LCPO (Aggarwal & Welleck, 2025) include user-specified length constraint into the training reward to guide the model toward answering within the constraint. However, such methods discard valuable intermediate reasoning that could improve accuracy. In contrast, our method utilizes the intrinsic phase-dependent entropy as reward signal, making it an adaptive and model-driven approach to helps the model reason more efficiently.

5.2 Reasoning Through Entropy Control

With the increasing research focus on Reinforcement Learning with Verifiable Rewards (RLVR), model entropy (Shannon, 1948) has emerged as a powerful internal signal for shaping reasoning behaviors in large language models. Recent work has investigated how policy entropy evolves during reinforcement learning-based post-training of reasoning models. Zhang et al. (2025f) reveal the correlation between entropy collapse and performance saturation as well as subsequent degradation. Cui et al. (2025) further shows how high-probability/high-advantage updates systematically reduce entropy. Another complementary direction treats entropy minimization itself as supervision by directly minimizing token-level entropy via finetuning or using negative entropy as the sole reward in RL (Agarwal et al., 2025; Prabhudesai et al., 2025). Besides, recent work has explored augmenting reinforcement learning approaches by incorporating entropy-based mechanisms to encourage exploration in reasoning chains (Zhang et al., 2025e; Cheng et al., 2025a). Furthermore, Wang et al. (2025c) reveal that the effectiveness of RLVR stems primarily from optimizing high-entropy tokens that determine critical reasoning directions. Selectively targeting these high-entropy minority tokens during optimization can substantially enhance reasoning capabilities while improving computational efficiency. While most existing studies leverage entropy to improve reasoning capability, our approach uses entropy as a control signal for efficiency, enabling adaptive length control without explicit token budgets while preserving accuracy. This reframes entropy not only as a tool for capability shaping but also as a principled knob for controlling the cost of reasoning.

6 Conclusion

In this work, we conduct empirical analysis and observed the consistent positive relation between entropy and response length across reasoning stages: the thinking phase exhibits higher entropy, reflecting exploratory behavior of longer responses, while the final answer phase shows lower entropy, indicating a more deterministic solution. Based on this finding, we address the challenge of efficient reasoning by introducing Phase Entropy Aware Reward (PEAR), a reward mechanism that distinguishes entropy between the thinking phase and the final answer phase during training. By discouraging entropy in the thinking phase while preserving flexibility in the final answer phase, PEAR enables adaptive control of response length without requiring explicit length targets or rigid truncation rules. Extensive experiments across six benchmarks have demonstrated that PEAR reduces token redundancy by a large percentage of 32.4% to 56.6% while preserving accuracy with less than 1% degradation. Additionally, PEAR demonstrates strong generalization to out-of-distribution tasks on both math reasoning and general knowledge benchmarks.

Acknowledgements

We would like to thank the anonymous reviewers and area chairs for their constructive and helpful comments on this work. This research/project is supported by the National Research Foundation, Singapore under its National Large Language Models Funding Initiative. (AISG Award No: AISG-NMLP-2024-005).

Ethics Statement

This work adheres to the ICLR Code of Ethics.[1] Our research focuses on improving the efficiency of Large Reasoning Models (LRMs) through phase-dependent entropy reward design. No human subjects, personally identifiable information, or sensitive user data were used in this study. All datasets employed (GSM8K, MATH500, AIME24, AMC23, GPQA and MMLU) are publicly available benchmarks designed for evaluating mathematical reasoning tasks and general tasks. The methods proposed in this paper aim to reduce computational overhead by shortening reasoning traces, which contributes to lowering energy consumption and improving the sustainability of large-scale model deployment. We do not anticipate direct harmful applications; however, as with all advances in language modeling, there exists a risk of misuse in generating misleading or harmful reasoning traces. We encourage responsible use and recommend that future work continue to consider fairness, transparency, and accountability in the deployment of reasoning models. No conflicts of interest or external sponsorships influenced this work.

Reproducibility Statement

We have made extensive efforts to ensure the reproducibility of our work. All datasets used in our experiments (GSM8K, MATH500, AIME24, AMC23, GPQA and MMLU) are publicly available and referenced in Section 4.1 and Appendix D. Detailed descriptions of our training setup, hyperparameters, and evaluation protocol are provided in Section 4.1 and Appendix C. For baselines, we follow official implementations and cite the corresponding repositories to ensure faithful comparison in Appendix C. Our method is implemented using the open-sourced `verl` framework (Sheng et al., 2025), and we will release the complete source code and training scripts in the future to facilitate replication of results. Together, these resources provide a clear pathway for reproducing both the training process and reported results.

References

Shivam Agarwal, Zimin Zhang, Lifan Yuan, Jiawei Han, and Hao Peng. The unreasonable effectiveness of entropy minimization in llm reasoning. *arXiv preprint arXiv:2505.15134*, 2025. URL https://arxiv.org/abs/2505.15134.

Pranjal Aggarwal and Sean Welleck. L1: Controlling how long a reasoning model thinks with reinforcement learning. *arXiv preprint arXiv:2503.04697*, 2025. URL https://arxiv.org/abs/2503.04697.

Sohyun An, Ruochen Wang, Tianyi Zhou, and Cho-Jui Hsieh. Don't think longer, think wisely: Optimizing thinking dynamics for large reasoning models. *arXiv preprint arXiv:2505.21765*, 2025. URL https://arxiv.org/abs/2505.21765.

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.

Daixuan Cheng, Shaohan Huang, Xuekai Zhu, Bo Dai, Wayne Xin Zhao, Zhenliang Zhang, and Furu Wei. Reasoning with exploration: An entropy perspective. *arXiv preprint arXiv:2506.14758*, 2025a. URL https://arxiv.org/abs/2506.14758.

Zhengxiang Cheng, Dongping Chen, Mingyang Fu, and Tianyi Zhou. Optimizing length compression in large reasoning models. *arXiv preprint arXiv:2506.14755*, 2025b. URL https://arxiv.org/abs/2506.14755.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. URL https://arxiv.org/abs/2110.14168.

[1] https://iclr.cc/public/CodeOfEthics

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, et al. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617*, 2025. URL `https://arxiv.org/abs/2505.22617`.

Muzhi Dai, Chenxu Yang, and Qingyi Si. S-grpo: Early exit via reinforcement learning in reasoning models. *arXiv preprint arXiv:2505.07686*, 2025. URL `https://arxiv.org/abs/2505.07686`.

Soumya Suvra Ghosal, Souradip Chakraborty, Avinash Reddy, Yifu Lu, Mengdi Wang, Dinesh Manocha, Furong Huang, Mohammad Ghavamzadeh, and Amrit Singh Bedi. Does thinking more always help? understanding test-time scaling in reasoning models. *arXiv preprint arXiv:2506.04210*, 2025. URL `https://arxiv.org/abs/2506.04210`.

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

Michael Hassid, Gabriel Synnaeve, Yossi Adi, and Roy Schwartz. Don't overthink it. preferring shorter thinking chains for improved llm reasoning. *arXiv preprint arXiv:2505.17813*, 2025. URL `https://arxiv.org/abs/2505.17813`.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021a.

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. In *Advances in Neural Information Processing Systems (NeurIPS), Track on Datasets and Benchmarks*, 2021b.

Shijue Huang, Hongru Wang, Wanjun Zhong, Zhaochen Su, Jiazhan Feng, Bowen Cao, and Yi R Fung. Adactrl: Towards adaptive and controllable reasoning via difficulty-aware budgeting. *arXiv preprint arXiv:2505.18822*, 2025. URL `https://arxiv.org/abs/2505.18822`.

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. URL `https://arxiv.org/abs/2412.16720`.

Lingjie Jiang, Xun Wu, Shaohan Huang, Qingxiu Dong, Zewen Chi, Li Dong, Xingxing Zhang, Tengchao Lv, Lei Cui, and Furu Wei. Think only when you need with large hybrid-reasoning models. *arXiv preprint arXiv:2505.14631*, 2025. URL `https://arxiv.org/abs/2505.14631`.

Martin Kuo, Jianyi Zhang, Aolin Ding, Qinsi Wang, Louis DiValentin, Yujia Bao, Wei Wei, Hai Li, and Yiran Chen. H-cot: Hijacking the chain-of-thought safety reasoning mechanism to jailbreak large reasoning models, including openai o1/o3, deepseek-r1, and gemini 2.0 flash thinking. *arXiv preprint arXiv:2502.12893*, 2025. URL `https://arxiv.org/abs/2502.12893`.

Shiye Lei, Zhihao Cheng, Kai Jia, and Dacheng Tao. Revisiting llm reasoning via information bottleneck. *arXiv preprint arXiv:2507.18391*, 2025. URL `https://arxiv.org/abs/2507.18391`.

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

Zeju Li, Jianyuan Zhong, Ziyang Zheng, Xiangyu Wen, Zhijian Xu, Yingying Cheng, Fan Zhang, and Qiang Xu. Compressing chain-of-thought in llms via step entropy. *arXiv preprint arXiv:2508.03346*, 2025. URL `https://arxiv.org/abs/2508.03346`.

Baohao Liao, Hanze Dong, Yuhui Xu, Doyen Sahoo, Christof Monz, Junnan Li, and Caiming Xiong. Fractured chain-of-thought reasoning. *arXiv preprint arXiv:2505.12992*, 2025. URL https://arxiv.org/abs/2505.12992.

Xin Liu and Lu Wang. Answer convergence as a signal for early stopping in reasoning. *arXiv preprint arXiv:2506.02536*, 2025. URL https://arxiv.org/abs/2506.02536.

Feng Luo, Yu-Neng Chuang, Guanchu Wang, Hoang Anh Duy Le, Shaochen Zhong, Hongyi Liu, Jiayi Yuan, Yang Sui, Vladimir Braverman, Vipin Chaudhary, et al. Autol2s: Auto long-short reasoning for efficient large language models. *arXiv preprint arXiv:2505.22662*, 2025a. URL https://arxiv.org/abs/2505.22662.

Haotian Luo, Haiying He, Yibo Wang, Jinluan Yang, Rui Liu, Naiqiang Tan, Xiaochun Cao, Dacheng Tao, and Li Shen. Adar1: From long-cot to hybrid-cot via bi-level adaptive reasoning optimization. *arXiv preprint arXiv:2504.21659*, 2025b. URL https://arxiv.org/abs/2504.21659.

Tergel Munkhbat, Namgyu Ho, Seo Hyun Kim, Yongjin Yang, Yujin Kim, and Se-Young Yun. Self-training elicits concise reasoning in large language models. *arXiv preprint arXiv:2502.20122*, 2025. URL https://arxiv.org/abs/2502.20122.

Mihir Prabhudesai, Lili Chen, Alex Ippoliti, Katerina Fragkiadaki, Hao Liu, and Deepak Pathak. Maximizing confidence alone improves reasoning. *arXiv preprint arXiv:2505.22660*, 2025. URL https://arxiv.org/abs/2505.22660.

Ziqing Qiao, Yongheng Deng, Jiali Zeng, Dong Wang, Lai Wei, Fandong Meng, Jie Zhou, Ju Ren, and Yaoxue Zhang. Concise: Confidence-guided compression in step-by-step efficient reasoning. *arXiv preprint arXiv:2505.04881*, 2025. URL https://arxiv.org/abs/2505.04881.

Xiaoye Qu, Yafu Li, Zhaochen Su, Weigao Sun, Jianhao Yan, Dongrui Liu, Ganqu Cui, Daizong Liu, Shuxian Liang, Junxian He, et al. A survey of efficient reasoning for large reasoning models: Language, multimodality, and beyond. *arXiv preprint arXiv:2503.21614*, 2025. URL https://arxiv.org/abs/2503.21614.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=Ti67584b98.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. URL https://arxiv.org/abs/1707.06347.

Claude E Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. *arXiv preprint arXiv:2402.03300*, 2(3):5, 2024. URL https://arxiv.org/abs/2402.03300.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pp. 1279–1297, 2025.

Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Hanjie Chen, et al. Stop overthinking: A survey on efficient reasoning for large language models. *arXiv preprint arXiv:2503.16419*, 2025. URL https://arxiv.org/abs/2503.16419.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*, 2025. URL https://arxiv.org/abs/2501.12599.

Qwen Team. Qwq-32b: Embracing the power of reinforcement learning, March 2025. URL https://qwenlm.github.io/blog/qwq-32b/.

Chenlong Wang, Yuanning Feng, Dongping Chen, Zhaoyang Chu, Ranjay Krishna, and Tianyi Zhou. Wait, we don't need to" wait"! removing thinking tokens improves reasoning efficiency. *arXiv preprint arXiv:2506.08343*, 2025a. URL https://arxiv.org/abs/2506.08343.

Jiakang Wang, Runze Liu, Fuzheng Zhang, Xiu Li, and Guorui Zhou. Stabilizing knowledge, promoting reasoning: Dual-token constraints for rlvr. *arXiv preprint arXiv:2507.15778*, 2025b. URL https://arxiv.org/abs/2507.15778.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, et al. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *arXiv preprint arXiv:2506.01939*, 2025c. URL https://arxiv.org/abs/2506.01939.

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*, 2025d. URL https://arxiv.org/abs/2501.18585.

Yunhao Wang, Yuhao Zhang, Tinghao Yu, Can Xu, Feng Zhang, and Fengzong Lian. Adaptive deep reasoning: Triggering deep thinking when needed. *arXiv preprint arXiv:2505.20101*, 2025e. URL https://arxiv.org/abs/2505.20101.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Heming Xia, Chak Tou Leong, Wenjie Wang, Yongqi Li, and Wenjie Li. Tokenskip: Controllable chain-of-thought compression in llms. *arXiv preprint arXiv:2502.12067*, 2025. URL https://arxiv.org/abs/2502.12067.

Yuhui Xu, Hanze Dong, Lei Wang, Doyen Sahoo, Junnan Li, and Caiming Xiong. Scalable chain of thoughts via elastic reasoning. *arXiv preprint arXiv:2505.05315*, 2025. URL https://arxiv.org/abs/2505.05315.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024. URL https://arxiv.org/abs/2507.10671.

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. URL https://arxiv.org/abs/2505.09388.

Chenxu Yang, Qingyi Si, Yongjie Duan, Zheliang Zhu, Chenyu Zhu, Qiaowei Li, Zheng Lin, Li Cao, and Weiping Wang. Dynamic early exit in reasoning models. *arXiv preprint arXiv:2504.15895*, 2025b. URL https://arxiv.org/abs/2504.15895.

Bin Yu, Hang Yuan, Haotian Li, Xueyin Xu, Yuliang Wei, Bailing Wang, Weizhen Qi, and Kai Chen. Long-short chain-of-thought mixture supervised fine-tuning eliciting efficient reasoning in large language models. *arXiv preprint arXiv:2505.03469*, 2025. URL https://arxiv.org/abs/2505.03469.

Hang Yuan, Bin Yu, Haotian Li, Shijun Yang, Christina Dan Wang, Zhou Yu, Xueyin Xu, Weizhen Qi, and Kai Chen. Not all tokens are what you need in thinking. *arXiv preprint arXiv:2505.17827*, 2025. URL https://arxiv.org/abs/2505.17827.

Linan Yue, Yichao Du, Yizhi Wang, Weibo Gao, Fangzhou Yao, Li Wang, Ye Liu, Ziyu Xu, Qi Liu, Shimin Di, et al. Don't overthink it: A survey of efficient r1-style large reasoning models. *arXiv preprint arXiv:2508.02120*, 2025. URL https://arxiv.org/abs/2508.02120.

Zihao Zeng, Xuyao Huang, Boxiu Li, Hao Zhang, and Zhijie Deng. Done is better than perfect: Unlocking efficient reasoning by structured multi-turn decomposition. *arXiv preprint arXiv:2505.19788*, 2025. URL https://arxiv.org/abs/2505.19788.

Jiajie Zhang, Nianyi Lin, Lei Hou, Ling Feng, and Juanzi Li. Adaptthink: Reasoning models can learn when to think. *arXiv preprint arXiv:2505.13417*, 2025a. URL https://arxiv.org/abs/2505.13417.

Jinghan Zhang, Xiting Wang, Fengran Mo, Yeyang Zhou, Wanfu Gao, and Kunpeng Liu. Entropy-based exploration conduction for multi-step reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 3895–3906, 2025b. doi: 10.18653/v1/2025.findings-acl.201. URL https://aclanthology.org/2025.findings-acl.201/.

Qingyang Zhang, Haitao Wu, Changqing Zhang, Peilin Zhao, and Yatao Bian. Right question is already half the answer: Fully unsupervised llm reasoning incentivization. *arXiv preprint arXiv:2504.05812*, 2025c. URL https://arxiv.org/abs/2504.05812.

Shengjia Zhang, Junjie Wu, Jiawei Chen, Changwang Zhang, Xingyu Lou, Wangchunshu Zhou, Sheng Zhou, Can Wang, and Jun Wang. Othink-r1: Intrinsic fast/slow thinking mode switching for over-reasoning mitigation. *arXiv preprint arXiv:2506.02397*, 2025d. URL https://arxiv.org/abs/2506.02397.

Xingjian Zhang, Siwei Wen, Wenjun Wu, and Lei Huang. Edge-grpo: Entropy-driven grpo with guided error correction for advantage diversity. *arXiv preprint arXiv:2507.21848*, 2025e. URL https://arxiv.org/abs/2507.21848.

Yanzhi Zhang, Zhaoxi Zhang, Haoxiang Guan, Yilin Cheng, Yitong Duan, Chen Wang, Yue Wang, Shuxin Zheng, and Jiyan He. No free lunch: Rethinking internal feedback for llm reasoning. *arXiv preprint arXiv:2506.17219*, 2025f. URL https://arxiv.org/abs/2506.17219.

Haoran Zhao, Yuchen Yan, Yongliang Shen, Haolei Xu, Wenqi Zhang, Kaitao Song, Jian Shao, Weiming Lu, Jun Xiao, and Yueting Zhuang. Let llms break free from overthinking via self-braking tuning. *arXiv preprint arXiv:2505.14604*, 2025a. URL https://arxiv.org/abs/2505.14604.

Shangziqi Zhao, Jiahao Yuan, Guisong Yang, and Usman Naseem. Can pruning improve reasoning? revisiting long-cot compression with capability in mind for better reasoning. *arXiv preprint arXiv:2505.14582*, 2025b. URL https://arxiv.org/abs/2505.14582.

Zihao Zhu, Hongbao Zhang, Ruotong Wang, Ke Xu, Siwei Lyu, and Baoyuan Wu. To think or not to think: Exploring the unthinking vulnerability in large reasoning models. *arXiv preprint arXiv:2502.12202*, 2025. URL https://arxiv.org/abs/2502.12202.

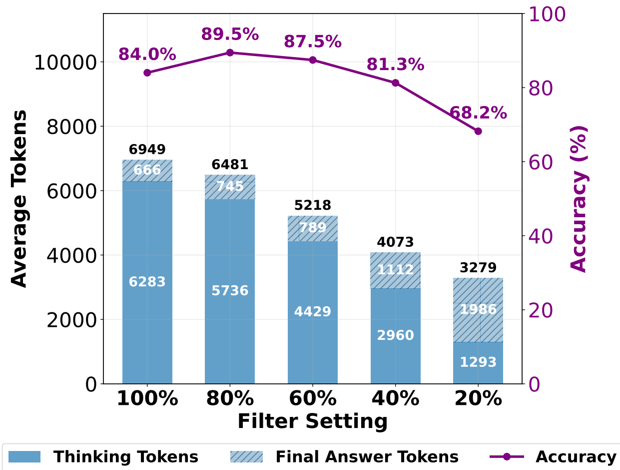

Figure 6: Accuracy and average response length in the entropy filtering experiments on Qwen3-8B.

A The Use of Large Language Models (LLMs)

Large Language Models (LLMs) were used solely for polishing the writing and improving the clarity of presentation. They were **NOT** involved in research ideation, methodology design, experiments, analysis, or any other substantive aspect of this work. All scientific contributions, results, and conclusions are the sole responsibility of the authors.

B Entropy Filtering Experiments for Qwen3-8B

Figure 6 demonstrates the entropy filtering experiment result on Qwen3-8B. The results reveal a similar trend as Qwen3-4B discussed in Section 2.3. When retaining 80% or 60% of low-entropy tokens, accuracy remains stable or even improves compared to the unfiltered baseline. Performance degradation only emerges under more aggressive filtering: retaining 40% or fewer low-entropy tokens leads to a sharp drop in accuracy, showing that essential reasoning steps are lost when the trace is compressed too heavily. Notably, the length of the final answer phase also remains relatively unchanged across filtering levels, reinforcing that redundancy is concentrated in the thinking phase.

This result further supports the conclusion that the high-entropy tokens being removed mainly drive excessive exploration rather than contributing to correct reasoning, and their absence reduces noise in the reasoning process.

C Experiment details for baseline methods

We evaluate three baseline methods: **GRPO** (Group Relative Policy Optimization) (Shao et al., 2024), **Step Entropy** (Li et al., 2025), and **LCPO** (Length-Controlled Policy Optimization) (Aggarwal & Welleck, 2025) using the GSM8K training set (Cobbe et al., 2021). Experiments are conducted across three model sizes: DeepSeek-R1-Distill-Qwen-1.5B (Guo et al., 2025), Qwen3-4B, and Qwen3-8B (Yang et al., 2025a). The implementation details for each baseline are provided below.

For GRPO (Shao et al., 2024), we use the open-source `verl` framework (Sheng et al., 2025)[2] with the original rule-based reward, which assigns a reward of 1 for correct answers and 0 otherwise. We set the rollout number to 8 and the KL penalty coefficient to 1×10^{-3}.

For Step Entropy (Li et al., 2025), we use the official implementation provided by the authors[3]. The method follows a two-stage training strategy: Supervised Fine-Tuning (SFT) with pruned CoT data,

[2] `https://github.com/volcengine/verl`
[3] `https://github.com/staymylove/COT_Compression_via_Step_entropy`

followed by Reinforcement Learning (RL) with GRPO. During the SFT stage, training is performed with mixed precision (FP16), a learning rate of 2×10^{-5}, and a weight decay of 0.01. In the RL stage, the learning rate is set to 1×10^{-5} and the KL penalty is fixed at 0.1.

For LCPO (Aggarwal & Welleck, 2025), we use the official codebase provided by the authors[4] and follow the L1-Exact setup. Training is performed with GRPO under length control and a maximum length constraint. We set the learning rate to 1×10^{-6} with a batch size of 64, and restrict the context length to 4K tokens during training. Rollout number is fixed at 8 with a sampling temperature of 0.6, and the KL penalty coefficient is set to 1×10^{-3}.

D Evaluation Benchmarks

To evaluate the effectiveness and generalizability of our compression method, we benchmark on four standard mathematical reasoning datasets and two knowledge benchmarks.

GSM8K test set (Cobbe et al., 2021) is a carefully designed benchmark comprising 1,319 grade-school mathematics word problems. Each question typically requires two to eight sequential reasoning steps, primarily involving basic arithmetic operations applied across multiple intermediate stages. **MATH500** (Hendrycks et al., 2021b) contains a subset of 500 problems drawn from high school mathematics competitions. We follow the evaluation setup of OpenAI by adopting the same curated subset. **AIME24** (Li et al., 2024) features 30 problems from the 2024 American Invitational Mathematics Examination (AIME). As one of the most prestigious secondary-level competitions, AIME problems demand sophisticated reasoning across diverse topics, including algebra, combinatorics, geometry, number theory, and probability. **AMC23** (Li et al., 2024) consists of 40 problems taken from the 2023 American Mathematics Competition (AMC). The dataset covers core high school mathematics domains such as algebra, geometry, combinatorics, and number theory, providing a broad yet rigorous evaluation of mathematical reasoning ability. **GPQA** (Rein et al., 2024) is a carefully constructed benchmark of graduate-level questions spanning physics, chemistry, and biology. In our experiments we use the **GPQA Diamond** subset, which comprises 198 questions selected for highest quality. **MMLU** (Hendrycks et al., 2021a) is a large multitask benchmark of multiple-choice questions drawn from 57 subjects across the humanities, social sciences, STEM, and other areas (e.g., elementary mathematics, US history, computer science, and law). Strong performance on MMLU requires broad world knowledge and problem-solving ability.

E Extended experiments

To validate that PEAR achieves robust efficiency across recent models, different base architectures, and model scales, we conducted additional experiments presented in Table 2.

Recent models. Following the release of DeepSeek R1 0528 and its improved distilled variants, we evaluated PEAR using **DeepSeek-R1-0528-Qwen3-8B**, which reflects the updated reasoning style and stronger chain-of-thought capabilities of newer LRMs. Across all five benchmarks, PEAR continues to demonstrate its effectiveness on this more advanced distilled model: it achieves a 41.3% reduction in reasoning length on average while maintaining or slightly improving accuracy compared with both the base model and the GRPO-trained model. Notably, PEAR improves performance on AIME24 and GPQA Diamond, two challenging benchmarks that typically benefit from deeper reasoning, indicating that reducing redundant high-entropy exploration does not harm, and may even enhance, the model's reasoning robustness. These results confirm that PEAR generalizes well to recent models with different thinking patterns and that the phase-dependent entropy reward remains effective for models with evolving reasoning styles.

Different architectures. To address whether PEAR generalizes beyond Qwen-based models, we further evaluated it on **DeepSeek-R1-Distill-Llama3.1-8B**, a model from the Llama family trained with DeepSeek-R1 reasoning traces. We used the same four math benchmarks as in the main paper and additionally GPQA Diamond. Across all benchmarks, PEAR again achieves substantial reductions in reasoning length while maintaining competitive accuracy relative to the base model. These results indicate that PEAR is model-agnostic and can be applied effectively to different LLM fam-

[4] https://github.com/cmu-l3/l1

Table 2: Acc@1 results on five benchmarks across three LRMs in different size and model series. ↓ / ↑ indicates the relative change in average token usage with respect to the *Original* row of each model.

Method	GSM8K		MATH500		AIME24		AMC23		GPQA		Average	
	Acc	Tok	Acc	Tok	Acc	Tok	Acc	Tok	Acc	Tok	Acc	Tok
DeepSeek-R1-0528-Qwen3-8B												
Original	93.55	1969	80.6	5560	53.33	11768	87.50	8287	52.02	5260	73.40	6569
GRPO	95.75	2371	81.20	5854	60.00	13354	92.50	8414	57.57	5628	77.40	7124 (↑ 8.45%)
PEAR	95.07	689	81.00	2560	66.67	8052	92.50	5318	59.09	2669	78.87	3858 (↓ 41.28%)
DeepSeek-R1-Distill-Llama3.1-8B												
Original	83.32	1272	80.60	3303	46.66	5673	90.00	4586	50.00	4255	70.12	3818
GRPO	91.58	1586	79.00	3530	43.33	5585	87.50	4568	48.98	4384	70.08	3931 (↑ 2.96%)
PEAR	88.70	1056	77.60	2419	40.00	4270	87.50	4356	47.97	2868	68.35	2994 (↓ 21.58%)
Qwen3-14B												
Original	96.05	1721	85.80	4279	73.33	10181	95.00	6693	55.55	3946	81.15	5364
GRPO	96.13	1700	83.80	4023	73.33	9677	97.50	6491	56.56	3618	81.46	5101 (↓ 4.89%)
PEAR	96.66	1196	84.40	2772	76.66	7650	97.50	4150	54.54	2772	81.95	3708 (↓ 30.87%)

ilies, including Llama-based reasoning models, without relying on model-specific inductive biases or training pipelines.

Larger scale. To demonstrate that PEAR is model-size agnostic, we evaluated it on **Qwen3-14B**, extending our analysis beyond the 1.5B–8B models reported in the main paper. The results show that PEAR continues to deliver substantial reductions in reasoning length (over 30% on average) while preserving or slightly improving accuracy across all benchmarks. This consistent pattern at 14B confirms that PEAR scales effectively with model capacity and maintains its efficiency–accuracy benefits as the underlying LRM becomes larger. These results are integrated into the evaluation section and discussed in the context of scalability, demonstrating PEAR's robustness across model sizes.

F Case Study: Qualitative Comparison of Reasoning Traces

We present a qualitative case study comparing the complete reasoning processes of the original Qwen3-8B model and the PEAR-trained Qwen3-8B model ($\alpha = 1$) on the same mathematical problem in Figure 7. This example is representative of a pattern consistently observed across benchmarks. Both models arrive at the correct solution and employ the same core deductive steps: recalling the definition of polar coordinates, computing the radius, determining the angle, and verifying the quadrant. However, their reasoning traces differ substantially in structure and efficiency. The baseline model repeatedly revisits the same checks, including angle verification and formula validation, leading to long reflection segments that do not contribute new information. In contrast, the PEAR-trained model follows the same logical path but avoids repeated self-verification and meta-cognitive loops. As a result, the reasoning trace becomes shorter and more focused.

Importantly, the reduction in length does not degrade reasoning correctness. Across qualitative inspection of multiple tasks, we do not observe cases where PEAR increases step-level errors for problems whose final answers are correct. This behavior directly follows from PEAR's reward design: entropy-based shaping is applied only to already-correct responses, so the model is not incentivized to shorten incorrect or incomplete reasoning. Thus, PEAR improves reasoning efficiency by removing redundant exploration while preserving the essential logical structure of the solution.

Example Qwen3-8B: Baseline vs. PEAR

Question

Convert the point $(0, 3)$ in rectangular coordinates to polar coordinates.

Baseline Model (3195 tokens)

`<think>` Okay, so I need to convert the rectangular coordinates $(0, 3)$ to polar coordinates. Hmm, let me recall how to do this. From what I remember, polar coordinates are represented as (r, θ), where r is the distance from the origin to the point, and θ is the angle measured from the positive x-axis.
First, I think I need to find r.
[450 tokens]
However, in this case, x is 0 and y is 3. Let me think.
[506 tokens]
But wait, let me make sure there isn't another angle that could also work.
[434 tokens]
Wait, let me verify this with the formula.
[309 tokens]
Wait, let me make sure there isn't a mistake here.
[425 tokens]
Therefore, I think my answer is correct. The polar coordinates for $(0, 3)$ are $(3, \pi/2)$.
But just to be thorough, let me check with another method.
[342 tokens]
Therefore, the final answer is $(3, \pi/2)$.
I think that's all. I considered different approaches, checked for quadrant issues, and verified using multiple methods. Seems solid. `</think>`

To convert the rectangular coordinates $(0, 3)$ to polar coordinates, we use the standard formulas for converting from rectangular to polar coordinates:
[467 tokens]

Final Answer. Thus, the polar coordinates of the point $(0, 3)$ are: $\boxed{(3,\ \pi/2)}$

PEAR-Trained Model $\alpha = 1$ (2317 tokens)

`<think>` Okay, so I need to convert the rectangular coordinates $(0, 3)$ to polar coordinates. Hmm, let me remember how that works. Polar coordinates are given as (r, θ), where r is the distance from the origin to the point, and θ is the angle made with the positive x-axis.
First, let me recall the formulas for converting from rectangular (x, y) to polar (r, θ).
[368 tokens]
But let me make sure. Let me think about the coordinates.
[286 tokens]
Wait, let me check again. Let me visualize the point $(0, 3)$. It's 3 units up the y-axis.
[439 tokens]
I don't see any mistakes here. Let me check with another method.
[302 tokens]
Therefore, the polar coordinates are $(3, \pi/2)$. `</think>`
[408 tokens]
Final Answer. Putting it all together, the polar coordinates corresponding to the rectangular point $(0, 3)$ are: $\boxed{(3,\ \pi/2)}$

Figure 7: Qualitative comparison of reasoning traces produced by the original Qwen3-8B model and the PEAR-trained model on the same problem. PEAR significantly reduces redundant reflective reasoning while preserving correctness.