# Rethinking Thinking Steps in Overthinking: Towards More Effective Reasoning

**Dezhi Zhao**[1,2], **Xin Liu**[2,*], **Xiaocheng Feng**[1,2,*], **Hui Wang**[2] and **Bing Qin**[1,2]
[1]Harbin Institute of Technology
[2]Pengcheng Laboratory
{dzzhao, xcfeng, qinb}@ir.hit.edu.cn, hit.liuxin@gmail.com, wangh06@pcl.ac.cn

## Abstract

Understanding overthinking in large reasoning models (LRMs) is crucial for interpretability as well as reasoning efficiency and effectiveness. However, existing approaches primarily adopt coarse-grained reasoning strategies, such as truncating Chains-of-Thought or switching reasoning modes, which reduce verbosity but are insufficient to actively guide reasoning toward more effective trajectories. To address these issues, we propose a training-free, interpretable framework that selects thinking words via attention heads to guide LRMs toward more effective reasoning. Specifically, we categorize thinking steps into effective and redundant states, identify the attention head that best discriminates between them as the Thinking Partition Head to construct an Effective Thinking Representation Prototype, and compute the Information Gain Ratio (IGR) between candidate thinking words and this prototype to select the word that steers reasoning toward a more effective direction. Extensive experiments on mathematical and scientific reasoning benchmarks, including AIME24, AMC23, MATH-500, GSM8K, and GPQA-D, show that our method consistently outperforms the strong baseline DEER, achieving average improvements of 1.2–1.3% in accuracy and 2.5–4.5% in compression rate. Compared to vanilla baselines, our approach yields larger gains of 2.6–6.3% in accuracy while reducing token usage by 22–43%.

## 1 Introduction

Large reasoning models (LRMs) [Jaech *et al.*, 2024; Guo *et al.*, 2025; Yang *et al.*, 2025a] have demonstrated extraordinary capabilities in tasks such as mathematics and code generation, establishing performance levels that even surpass human PhDs in scientific reasoning benchmarks [Rein *et al.*, 2024; Jaech *et al.*, 2024]. This success is largely attributed to the test-time scaling laws [Wu *et al.*, 2024; Muennighoff *et al.*, 2025], which allocate increased computational resources to allow models to think more deeply during inference. However, when reasoning is excessively prolonged or unguided, the phenomenon of overthinking [Chen *et al.*, 2024; Sui *et al.*, 2025; Yue *et al.*, 2025] arises. In such cases, models may become trapped in repetitive verification or erroneous attempts without knowing when to halt, which ultimately degrades both inference efficiency and reasoning effectiveness [Wu *et al.*, 2025; Yang *et al.*, 2025b].

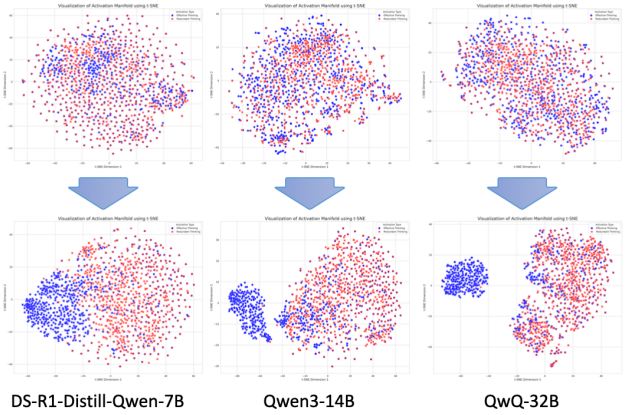


Figure 1: t-SNE visualization of thinking word representations across diverse model families. The upper row shows that thinking states are inextricably mixed within randomly selected attention heads. In contrast, the lower row demonstrates that under the identified Thinking Partition Heads (TPHeads), there is a clear, natural separation between effective (blue dots) and redundant (red dots) thinking steps within the internal representations of LRMs.

To mitigate overthinking, numerous studies have explored approaches ranging from post-training adjustments [Chung *et al.*, 2025; Fang *et al.*, 2025] to input-based [Ma *et al.*, 2025; Li *et al.*, 2025; Liang *et al.*, 2025; Wang *et al.*, 2025a] and output-based [Yang *et al.*, 2025b; Zhang *et al.*, 2025a] interventions. Despite these efforts, existing work predominantly focuses on rigid strategies such as truncating long Chains-of-Thought or switching thinking modes. Consequently, the active guidance of the thinking process remains under-explored, resulting in a deficiency in reasoning interpretability.

The rich information encoded in specific tokens offers a promising direction [Qian *et al.*, 2025; Dong *et al.*, 2025;

*Corresponding author

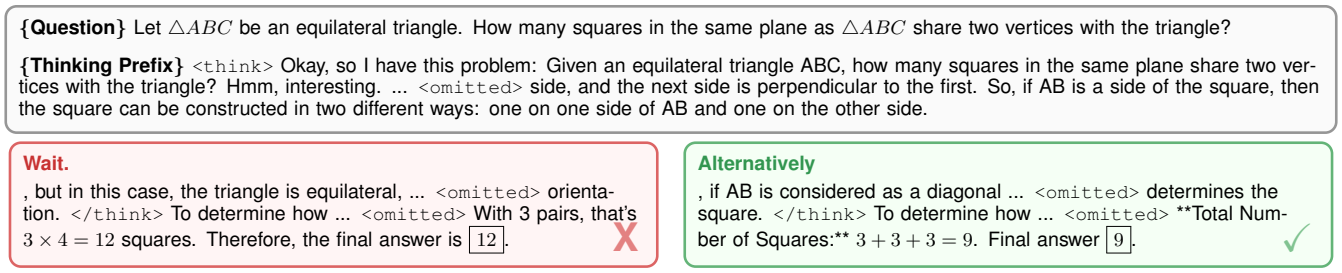


Figure 2: Impact of specific thinking words on reasoning outcomes. With the same context, the intervention of *Wait* versus *Alternatively* triggers different reasoning chains, causing a flip in the validity of the final answer.

Qi *et al.*, 2024]. Research on thinking tokens (e.g., Wait, Hmm, Therefore) indicates that the representations of thinking tokens exhibit a sudden and significant increase in Mutual Information (MI) with the golden answer, becoming highly informative about the correct outcome [Qian *et al.*, 2025]. Similarly, Dong *et al.* [2025] reveals that the model's first token often implicitly encodes information regarding multiple-choice answers at the end of the response. Given that a thinking word acts as the initial token of a thinking step, it likely plays a pivotal role in LRM inference. Our preliminary experiments confirm this, identifying a natural internal separation between effective and redundant thinking steps at the positions of thinking words (See the bottom line in Figure 1). By applying dimensionality reduction to the intermediate states of thinking words across diverse model families, such as DeepSeek-R1-Distill and Qwen3, we observe that distinct thinking states can be naturally and consistently distinguished. This cross-model consistency suggests a latent potential for the model to self-guide and self-interpret.

These observations drive us to further investigate thinking words. We find that different thinking words tend to steer the model toward divergent reasoning processes, which are associated with varying levels of final answer correctness. As shown in Figure 2, for the same question and identical reasoning prefix, replacing the thinking word "Wait" with "Alternatively" leads to a different reasoning continuation and a correct final answer. Inspired by this insight, we propose a training-free and highly interpretable framework that leverages attention head to select thinking words for guiding LRMs toward more effective reasoning trajectories, while characterizing their associations with underlying process distributions. Specifically, we first utilize the MATH-500 training set to construct a dataset of effective and redundant thinking steps. Second, we identify the attention head that best discriminates between these two states as the *Thinking Partition Head (TPHead)*, and then use the corresponding TP-Head representations to construct an *Effective Thinking Prototype (ETP)*. As shown in Figure 1, under the identified TP-Head, the hidden states at thinking word positions exhibit clear separability between effective and redundant reasoning states, in sharp contrast to the entangled thinking states observed within random attention heads. Third, to validate the correctness of this natural internal separation, we build upon the strong baseline DEER [Yang *et al.*, 2025b], the pioneering work on dynamic early exit for overthinking. In contrast, we extend the framework by computing the Information Gain Ratio (IGR) between candidate thinking words and the proposed Effective Thinking Prototype at each reasoning step. By selecting the thinking word with the highest IGR, our method steers the reasoning trajectory toward more effective direction. Finally, this principled guidance enables LRMs to achieve superior performance on both mathematical (AIME24, AMC23, MATH-500, GSM8K) and scientific (GPQA-D) reasoning benchmarks. Notably, our approach produces significantly fewer thinking tokens, achieving an average reduction of 2.8% compared to DEER, while further improving final accuracy by 1.3%. These results demonstrate that the identified effective/redundant separation is not only a consistent internal phenomenon, but also a structure that can be exploited to enhance both reasoning efficiency and effectiveness.

The main contributions of this paper are summarized as follows:

- We redefine and rethink the LRM thinking steps, providing a fine-grained categorization of effective versus redundant thinking. We uncover a natural internal separation of these states at the thinking word level, significantly reinforcing interpretability.
- Leveraging this natural separation (the ETP), we propose a novel early exit guidance strategy that steers LRMs toward more efficient reasoning trajectories and optimizes the reasoning path.
- Extensive empirical performance across multiple models and datasets validates both the effectiveness of our guidance strategy and the correctness of our finding regarding the internal effective/redundant separation.

## 2 Effective Thinking Prototype

### 2.1 Rethinking Thinking Steps

In this subsection, we define the thinking step and propose a quadrant-based taxonomy to distinguish between effective and redundant thinking. This classification hinges on comparing the correctness of intermediate answers generated by the current step against those from the preceding step, given their respective contexts. Based on these quadrants, we construct a dataset of effective and redundant thinking steps.

**Definition of Thinking Step.** Following DEER [Yang *et al.*, 2025b], we adopt the *Wait* token as a delimiter to define thinking steps. Specifically, we segment the CoT [Wei *et*

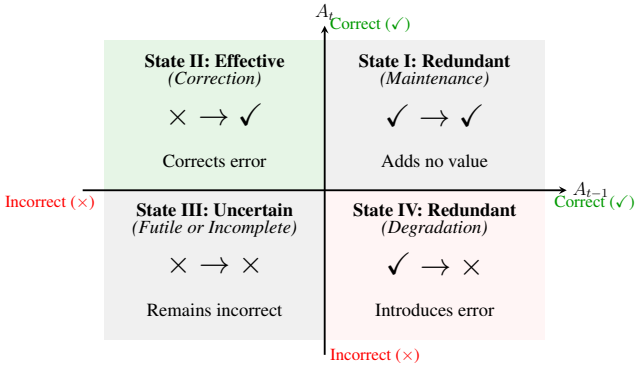


Figure 3: The Quadrants of Thinking Steps. $A_{t-1}$ and $A_t$ denote the correctness of intermediate answers derived from the preceding and current steps (along with their respective prefixes).

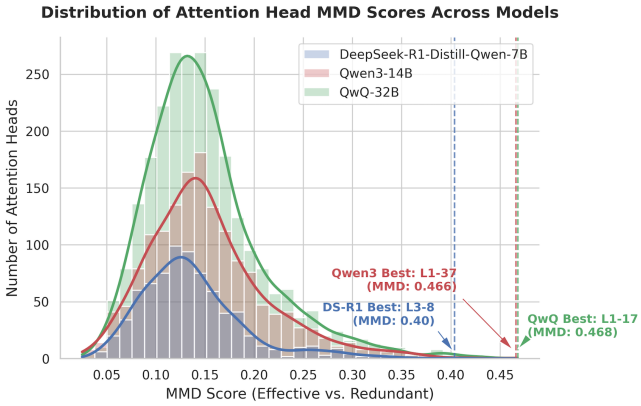


Figure 4: Distribution of attention head MMD scores. $L3$-8 refers to the $8^{th}$ head of Layer 3, with $L1$-37 and $L1$-17 defined analogously.

*al.*, 2022] generated by LRMs using occurrences of the *Wait* token, and treat each resulting segment as an individual thinking step.

**Thinking Step Quadrants.** Based on the segmented thinking steps, we introduce the *Thinking Step Quadrants* (Figure 3) to characterize the utility of each step. The X-axis represents the answer correctness after the preceding thinking step (positive for correct, negative for incorrect). The Y-axis represents the current thinking step, adhering to the same scale and definitions. The quadrants are defined as follows:

- Quadrant I (Redundant/Maintenance): The model maintains a correct answer. The step adds no value as the outcome was already correct.
- Quadrant II (Effective): The model transitions from incorrect to correct, indicating an effective thinking step that rectifies the answer.
- Quadrant III (Uncertain): The model remains incorrect. The step is ambiguous (either futile or incomplete) and labeled as uncertain.
- Quadrant IV (Redundant/Degradation): The model reverts from correct to incorrect, signifying a detrimental step that degrades performance.

**Data Construction for Effective and Redundant Thinking.** Based on the defined Thinking Step Quadrants, we construct a dataset of effective and redundant thinking steps using the MATH-500 training set. For each input problem, we first instruct the LRM to perform up to ten consecutive thinking steps, producing an intermediate answer after each step. Then, the correctness of each intermediate answer is automatically verified using standard evaluation tools [Ye *et al.*, 2025]. Finally, by analyzing correctness transitions between consecutive steps, we assign each thinking step to its corresponding quadrant and label it accordingly. We primarily select effective thinking steps from Quadrant II and redundant ones from Quadrant IV, as the distinction between these two states is the most unambiguous. The resulting dataset comprises 1,466 samples, evenly balanced (1:1) between effective and redundant thinking steps.

## 2.2 Localization of Thinking Partition Heads

We utilize T-distributed Stochastic Neighbor Embedding (t-SNE) [Maaten and Hinton, 2008] to visualize the representations of effective and redundant thinkings across each attention head of the model. As illustrated in Figure 1, specific attention heads exhibit a clear demarcation between these two states. To identify the attention head that optimally discriminates between the two states, we employ the Maximum Mean Discrepancy (MMD) [Gretton *et al.*, 2012] metric to quantify the degree of independence between the distributions of effective ($X$) and redundant ($Y$) thinkings:

$$\mathrm{MMD}(X,Y) = \left\| \frac{1}{N_E}\sum_{i=1}^{N_E}\phi(x_i) - \frac{1}{N_R}\sum_{j=1}^{N_R}\phi(y_j) \right\|_{\mathcal{H}} \quad (1)$$

A higher MMD value indicates greater independence between the two distributions, thereby signifying maximal discriminability. Upon sequentially computing the MMD for all heads, we select the head with the highest MMD score (Top-1) to serve as the effective Thinking Partition Head for the target model.

The specific distribution of MMD scores across attention heads is depicted in Figure 4. It can be observed that as the model parameter size increases, the model's internal capability to distinguish between effective and redundant states, characterized by higher MMD peaks and a rightward shift in the overall distribution, exhibits an upward trend. This implies that more powerful models possess more pronounced internal mechanisms for metacognition or self-monitoring.

Furthermore, models with varying architectures consistently exhibit a similar long-tail distribution. The vast majority of heads yield relatively low MMD scores (0.10 - 0.20), while only a minute fraction (Top 1%) function effectively as Thinking Partition Heads. Although the specific location of the Top-1 head varies across models ($L3$-8 vs. $L1$-37 vs. $L1$-17), their consistent existence underscores the necessity of the localization process.

## 2.3 Effective Thinking Standard Vector

Finally, we construct an effective thinking prototype utilizing solely the intermediate states of effective thinking data

within the identified discriminating head. Specifically, we compute the average of all effective thinking representations to derive the *Effective Thinking Standard Vector*, denoted as $\mathbf{H}_E$. Crucially, we do not average raw hidden states. The average is strictly computed within the TPHead subspace, identified via MMD to maximally separate effective from redundant states. It acts as a *meta-cognitive filter* that performs a low-dimensional projection, stripping away problem-specific semantics from hidden states while preserving the core *reasoning validity* signal. Analogous to Prototypical Networks [Snell *et al.*, 2017], $\mathbf{H}_E$ is the empirical centroid (prototype) of the effective thinking class. This standard vector serves as the effective thinking prototype in the subsequent thinking word selection phase.

## 3 Validation on Early Exit for LRMs

To further validate the correctness of the intrinsic partition between effective and redundant thinking discovered within LRMs, we conducted experiments on the early exit task to mitigate the overthinking problem. Our core hypothesis posits that if this partition is valid, selecting thinking words guided by the identified TPHead that align more closely with the ETP should steer the model toward more effective reasoning paths. This would improve reasoning performance while reducing token consumption. The remainder of this section is organized as follows: Section 3.1 outlines the overall framework, Section 3.2 details the experimental setup, and Section 3.3 introduces the baselines.

### 3.1 Validation Framework

The overall validation framework is illustrated in Figure 5 and comprises three main phases: (1) construction of the effective thinking prototype, (2) early exit determination at pause points, and (3) selection of the optimal thinking word to guide subsequent reasoning.

**Step 1: Effective Thinking Prototype Construction** We first locate the *Thinking Partition Head (TPHead)* by leveraging the method described in Section 2.2, and subsequently construct the *Effective Thinking Standard Vector ($\mathbf{H}_E$)* following Section 2.3. Both TPHead and $\mathbf{H}_E$ are utilized in the subsequent thinking word selection process.

**Step 2: Early Exit Determination** Following DEER, we pause the generation process whenever the LRM encounters `Wait` and then utilize the prompt $\mathbf{c}$ = `\n**Final Answer**\n \boxed` to elicit a direct intermediate answer $\mathbf{a}$ to assess whether an early exit is warranted. Specifically, after generating $\mathbf{a}$, we calculate the average log-probability of its tokens as shown in Equation 2:

$$C = \frac{1}{|\mathbf{a}|} \sum_{i=1}^{|\mathbf{a}|} \log P(\mathbf{a}_i | \mathbf{q}, \mathbf{o}_{<t}, \mathbf{c}, \mathbf{a}_{<i}), \tag{2}$$

where $|\mathbf{a}|$ is the number of tokens in the intermediate answer. If $C \geq \tau$ (where $\tau = 0.95$), the model triggers an early exit. In this case, a `</think>` tag is appended to the current sequence, and the model proceeds to generate the final answer. If $C < \tau$, the process enters the Thinking Word Selection phase (Step 3).

**Step 3: Thinking Word Selection** Prior work has shown that thinking words in LRMs exhibit an information peak with respect to the golden answer, indicating that representations at thinking word positions encode rich information relevant to correct reasoning [Qian *et al.*, 2025]. This relationship can be quantified using Mutual Information (MI) between the hidden representations of thinking words ($h_{think}$) and the final answer representations ($h_{ans}$):

$$MI(h_{think}; h_{ans}) = H(h_{think}) - H(h_{think} \mid h_{ans}). \tag{3}$$

$H(\cdot)$ denotes the Shannon Entropy [Shannon, 1948], representing the uncertainty of a variable. $H(h_{think} \mid h_{ans})$ is the conditional entropy, measuring the remaining uncertainty of the thinking process given the final answer. Qian *et al.* [2025] provide theoretical guarantees that higher MI between the representations and the golden answer reduces prediction error.

However, during inference, the golden answer representation $h_{ans}$ is not available. To address this, we leverage the constructed *Effective Thinking Standard Vector* $\mathbf{H}_E$ as a proxy for *golden thinking*. Specifically, we extract the TPHead representations of candidate thinking words $h_{cand}$ and compute their MI with $\mathbf{H}_E$:

$$MI(h_{cand}; \mathbf{H}_E) = H(h_{cand}) - H(h_{cand} \mid \mathbf{H}_E). \tag{4}$$

We estimate MI using the Hilbert-Schmidt Independence Criterion (HSIC) [Gretton *et al.*, 2005; Poole *et al.*, 2019]. Since MI is an absolute measure and different candidate words may have varying intrinsic entropy, directly comparing MI values can be biased. To mitigate this issue, we introduce the *Information Gain Ratio (IGR)* by normalizing MI with respect to the entropy of each candidate word:

$$IGR(h_{cand}) = \frac{MI(h_{cand}; \mathbf{H}_E)}{H(h_{cand})}. \tag{5}$$

This ratio reflects the proportion of information in a candidate thinking word that aligns with effective thinking. We compute the IGR for each candidate in the thinking word list and select the thinking word with the highest IGR to guide the subsequent reasoning process.

### 3.2 Experimental Setup

**Datasets and Metrics.** To ensure a fair comparison with published SOTA methods, we evaluated our method on four mathematical reasoning datasets: AIME 2024 [MAA Committees, 2024], AMC 2023 [AI-MO, 2023], MATH-500 [Hendrycks *et al.*, 2021], GSM8K [Cobbe *et al.*, 2021], and one scientific reasoning dataset: GPQA Diamond [Rein *et al.*, 2024]. We follow the same experimental settings as DEER, a strong baseline method. Due to the limited sample size of AIME 2024 and AMC 2023, we performed four independent runs for our method on these datasets and averaged the results to ensure stability and reliability. Following DEER, we employ Accuracy (Acc), Token Number (Tok), and Compression Rate (CR) as evaluation metrics. Acc denotes the accuracy of the final answer, Tok represents the average generation length across all test samples, and CR is the ratio of the average generation length to that of the original model. Lower values for Tok and CR indicate lower computational cost.

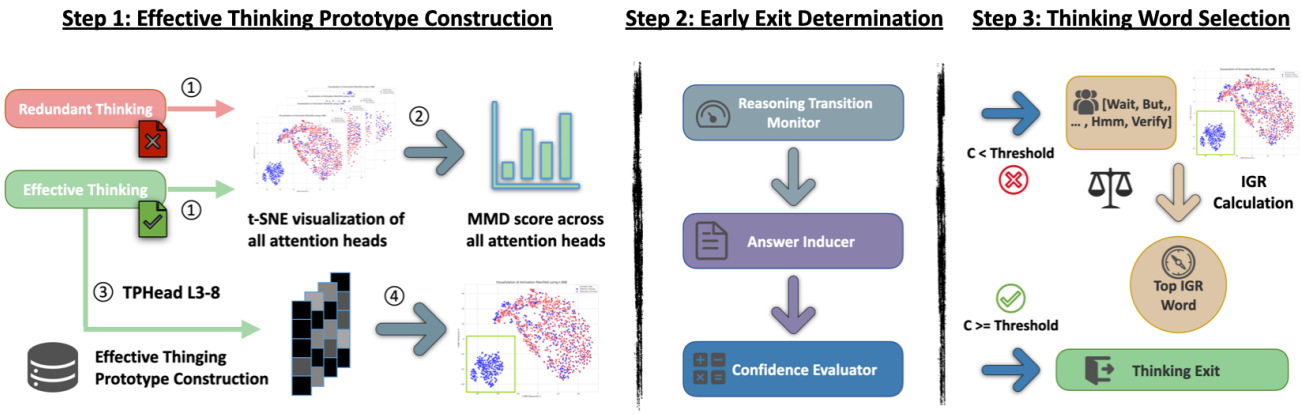


Figure 5: Overview of the validation framework for the early exit task in LRMs. ①–④ are executed in sequence. In Step 1, we take the TPHead $L3$-8 (the $8^{th}$ head of Layer 3) of the DeepSeek-R1-Distill-Qwen-7B model as an example.

**Models.** We use the same model backbones as the DEER main experiments: DeepSeek-R1-Distill-Qwen-7B, Qwen3-14B, and QwQ-32B.

**Implementation Details.** We adopted a zero-shot setting using the standard CoT prompt: *"Please reason step by step, and put your final answer within \boxed{}."* All experiments utilized a greedy decoding strategy with a maximum generation length of 16,384 tokens. The early exit threshold was set to 0.95, consistent with DEER. Accuracy evaluation was conducted using the LIMO tool [Ye *et al.*, 2025]. All experiments were conducted on two NVIDIA A100 GPUs.

### 3.3 Baselines

In addition to the primary baseline DEER, we compared our method against other baselines evaluated in the same work: (1) Vanilla: The original LRM without any intervention, serving as the benchmark for compression rate calculations. (2) Prompt-based Methods: Token-Conditional Control (TCC) [Muennighoff *et al.*, 2025]: Prompts the model to answer within a fixed token budget. NoThinking [Ma *et al.*, 2025]: Prompts the model to skip the thinking process and generate the final answer directly. Dynasor-CoT [Fu *et al.*, 2025]: Prompts the model to generate an intermediate answer at fixed token intervals, early exit is triggered when three consecutive intermediate answers are consistent. All baseline results are directly cited from the published paper [Yang *et al.*, 2025b].

## 4 Experimental Results and Analysis

In Section 4.1, we present the main experimental results and analyze the scalability of our approach. Subsequently, Section 4.2 investigates the impact of employing different attention heads as the TPHead. Section 4.3 examines the sensitivity of our method to the candidate list of thinking words. Finally, we provide ablation studies in Section 4.4.

### 4.1 Main results

As demonstrated in Table 1, our method achieves consistent improvements across three models of varying scales. Specifically, on the DeepSeek-R1-Distill-Qwen7B model, our method outperforms DEER with an average accuracy improvement of 1.3% and a reduction in compression rate by 4.5%. Compared to the Vanilla baseline, accuracy improves by 6.3% while the compression rate decreases by 43%.

Breaking down performance by dataset, our method exhibits superior performance on more challenging benchmarks (e.g., AIME 2024), achieving a 4.1% increase in accuracy. We attribute this to the fact that harder datasets typically require more reasoning steps, thereby providing a larger search space for thinking word selection.

Notably, although the effective thinking prototype used in our experiments was constructed solely from the MATH-500 training set, the improvements generalize not only to the MATH-500 test set but also to other mathematical reasoning tasks. Furthermore, when extending the scope to scientific reasoning tasks (GPQA-D), we observe an accuracy improvement of 0.5%–2.6% while maintaining comparable token consumption. This further validates the scalability of our constructed effective thinking space, suggesting that the internal mechanism of LRMs for distinguishing between effective and redundant thinking steps is not dependent on specific task types or domain knowledge, but rather represents an inherent capability to delineate validity.

### 4.2 Impact of Different Thinking Partition Heads

In our main experiments, we employed the attention head with the highest MMD value (Top-1) as the TPHead. However, visualization results (detailed in Appendix A) indicate that numerous other attention heads can also effectively discriminate between effective and redundant thinking steps. Therefore, in this subsection, we explore the impact of selecting different attention heads for this role, with results detailed in Table 2.

Based on the analysis in Section 4.1, our method shows more pronounced effects on harder datasets requiring extensive reasoning steps. Consequently, we conducted this analysis on the AIME24 and AMC23 datasets, comparing results obtained using Top-10 attention head versus random attention head. As shown in Table 2, when utilizing the Top-10

| Method | MATH | | | | | | | | | | | | SCIENCE | | | Overall | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | AIME24 | | | AMC23 | | | MATH-500 | | | GSM8K | | | GPQA-D | | | | |
| | Acc↑ | Tok↓ | CR↓ | Acc↑ | Tok↓ | CR↓ | Acc↑ | Tok↓ | CR↓ | Acc↑ | Tok↓ | CR↓ | Acc↑ | Tok↓ | CR↓ | Acc↑ | CR↓ |
| ***DeepSeek-R1-Distill-Qwen-7B*** | | | | | | | | | | | | | | | | | |
| Vanilla | 41.7 | 13,765 | 100% | 78.8 | 6,792 | 100% | 87.4 | 3,858 | 100% | 89.6 | 1,484 | 100% | 23.7 | 10,247 | 100% | 64.2 | 100% |
| TCC | 48.4 | 10,603 | 77.0% | 82.5 | 6,491 | 95.6% | 89.2 | 3,864 | 100.2% | 88.0 | 892 | 60.1% | 27.3 | 8,442 | 82.4% | 67.1 | 83.0% |
| NoThinking | 26.7 | 4,427 | 32.2% | 65.0 | 1,911 | 28.1% | 80.6 | 834 | 21.6% | 87.1 | 284 | 19.1% | 29.8 | 724 | 7.1% | 57.8 | **21.6%** |
| Dynasor-CoT | 46.7 | 12,695 | 92.2% | 85.0 | 5,980 | 88.0% | 89.0 | 2,971 | 77.0% | 89.6 | 1,285 | 86.6% | 30.5 | 7,639 | 74.5% | 68.2 | 83.7% |
| DEER | 49.2 | 9,839 | 71.5% | 85.0 | 4,451 | 65.5% | 89.8 | 2,143 | 55.5% | 90.6 | 917 | 61.8% | 31.3 | 5,469 | 53.4% | **69.2** | 61.5% |
| Ours | 53.3 | 9494 | 69.0% | 87.5 | 4,245 | 62.5% | 90.6 | 2,129 | 55.2% | 89.9 | 664 | 44.7% | 31.3 | 5,505 | 53.7% | **70.5** | **57.0%** |
| ***Qwen3-14B*** | | | | | | | | | | | | | | | | | |
| Vanilla | 70.0 | 10,859 | 100% | 95.0 | 7,139 | 100% | 93.8 | 4,508 | 100% | 95.1 | 2,047 | 100% | 60.1 | 7,339 | 100% | 82.8 | 100% |
| TCC | 70.8 | 11,573 | 106.6% | 95.0 | 7,261 | 101.7% | 94.6 | 4,484 | 99.5% | 95.7 | 1,241 | 60.6% | 60.1 | 7,138 | 97.3% | 83.3 | 93.1% |
| NoThinking | 26.7 | 7,337 | 67.6% | 77.5 | 2,133 | 29.9% | 85.0 | 1,228 | 27.2% | 94.8 | 286 | 14.0% | 50.5 | 2,320 | 31.6% | 66.9 | **34.1%** |
| Dynasor-CoT | 73.3 | 10,369 | 95.5% | 95.6 | 6,582 | 92.2% | 93.8 | 4,063 | 90.1% | 95.6 | 1,483 | 72.4% | 59.6 | 5,968 | 81.3% | 83.6 | 86.3% |
| DEER | 76.7 | 7,619 | 70.2% | 95.0 | 4,763 | 66.7% | 94.0 | 3,074 | 68.2% | 95.3 | 840 | 41.0% | 57.6 | 2,898 | 39.5% | **83.7** | 57.1% |
| Ours | 73.3 | 8,213 | 75.6% | 96.3 | 4,397 | 61.6% | 94.8 | 2,723 | 60.4% | 96.2 | 852 | 41.6% | 58.1 | 2,939 | 40.0% | **83.7** | **55.8%** |
| ***QwQ-32B*** | | | | | | | | | | | | | | | | | |
| Vanilla | 66.7 | 10,821 | 100% | 92.5 | 6,792 | 100% | 93.8 | 4,508 | 100% | 96.7 | 1,427 | 100% | 63.1 | 7,320 | 100% | 82.6 | 100% |
| TCC | 60.0 | 11,263 | 104.1% | 90.0 | 6,818 | 100.4% | 94.4 | 4,315 | 95.7% | 95.8 | 1,348 | 94.5% | 61.6 | 7,593 | 103.7% | 80.4 | 99.7% |
| NoThinking | 66.7 | 10,859 | 100.4% | 87.5 | 6,908 | 101.7% | 94.8 | 3,930 | 87.2% | 96.2 | 1,113 | 78.0% | 63.6 | 7,668 | 104.8% | 81.8 | 94.4% |
| Dynasor-CoT | 63.3 | 11,156 | 103.1% | 93.8 | 6,544 | 96.3% | 94.2 | 4,176 | 92.6% | 95.2 | 1,095 | 76.7% | 64.1 | 7,024 | 96.0% | 82.1 | 93.0% |
| DEER | 70.0 | 10,097 | 93.3% | 95.0 | 5,782 | 85.1% | 94.6 | 3,316 | 73.6% | 96.3 | 977 | 68.5% | 64.1 | 6,163 | 84.2% | **84.0** | **80.9%** |
| Ours | 73.3 | 9664 | 89.3% | 95.0 | 5,863 | 86.3% | 94.8 | 3,226 | 71.6% | 96.1 | 983 | 68.9% | 66.7 | 5,552 | 75.8% | **85.2** | **78.4%** |

Table 1: Experimental results across different series of LRMs. *Acc* denotes accuracy, *Tok* denotes token count, and *CR* denotes compression rate. ↑ indicates that higher values are better, while ↓ indicates that lower values are better. The top-2 best results are highlighted in **bold.**

| Method | MATH | | | | | | Overall | |
|---|---|---|---|---|---|---|---|---|
| | AIME24 | | | AMC23 | | | | |
| | Acc↑ | Tok↓ | CR↓ | Acc↑ | Tok↓ | CR↓ | Acc↑ | CR↓ |
| DeepSeek-R1-Distill-Qwen-7B | | | | | | | | |
| Vanilla | 41.7 | 13,765 | 100% | 78.8 | 6,792 | 100% | 60.3 | 100% |
| DEER | 49.2 | 9,839 | 71.5% | 85.0 | 4,451 | 65.5% | 67.1 | 68.5% |
| Random head | 46.7 | 9,450 | 68.7% | 85.0 | 4,276 | 63.0% | 65.9 | 65.9% |
| Top-10 head | 52.5 | **9,080** | **66.0%** | **87.5** | 4,433 | 65.3% | 70.0 | **65.7%** |
| **Top-1 head** | **53.3** | 9,494 | 69.0% | **87.5** | **4,245** | **62.5%** | **70.4** | **65.7%** |

Table 2: Performance comparison of different TPHeads.

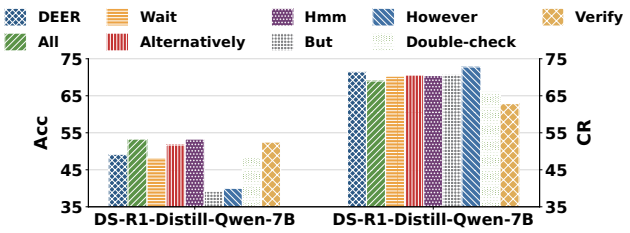


Figure 6: Thinking words sensitivity on Acc (left) and CR (right).

attention head as TPHead, the final performance decreases slightly compared to the Top-1 head but remains significantly higher than DEER. This aligns with the observation that Top-10 head maintains a relatively strong capability to distinguish effective from redundant thinking steps (Appendix A).

Conversely, when employing random attention head (with MMD value of approximately 0.13, corresponding to the most densely populated interval in Figure 4), performance drops significantly, falling below that of DEER. This is likely due to their inability to distinguish effective thinking states. These results further corroborate the validity of the internal thinking state division within LRMs: such distinctiveness exists in a minority of attention heads (the long-tail distribution in Figure 4), while the vast majority (the peak area of the distribution) lacks this discriminatory power.

## 4.3 Thinking Words Analysis

The candidate list of thinking words used in our method is derived from statistical findings in recent studies [Wang *et al.*, 2025b; Zhang *et al.*, 2025a]. Specifically, we initialize the list with the top-5 most frequent thinking words reported by Wang *et al.* [2025b] (*Wait, Alternatively, Hmm, But, However*). To prevent critical omissions, we then augment this list with the overlapping words found in both Wang *et al.* [2025b] and Zhang *et al.* [2025a] (*Double-check, Verify*), resulting in a final vocabulary of seven words. Given that this curation process may inherently introduces a degree of subjectivity, we investigate the sensitivity of our method to specific thinking words in this section to address scalability concerns. Specifically, we performed an analysis where we iteratively removed one word from the list and conducted experiments using the remaining six words, as shown in Figure 6.

As observed in the figure, in three out of the seven experimental groups, the results remained superior to the DEER baseline. This indicates that our method possesses considerable stability and is not heavily reliant on a subjectively constructed list of words. More specifically, removing *Hmm* or *Verify* resulted in accuracy comparable to the full seven-words list. Interestingly, removing *Verify* yielded better results in terms of CR, suggesting that *Verify* and *Hmm* are

| Method | MATH | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | AIME24 | | | AMC23 | | | Overall | |
| | Acc↑ | Tok↓ | CR↓ | Acc↑ | Tok↓ | CR↓ | Acc↑ | CR↓ |
| DeepSeek-R1-Distill-Qwen-7B | | | | | | | | |
| Vanilla | 41.7 | 13,765 | 100% | 78.8 | 6,792 | 100% | 60.3 | 100% |
| **Ours** | **53.3** | 9,494 | 69.0% | 87.5 | 4,245 | 62.5% | **70.4** | 65.7% |
| **w/o ETP** | 46.7 | 9,450 | 68.7% | 85.0 | 4,276 | 63.0% | 65.9 | 65.9% |
| **w/o EE** | 45.8 | 9,999 | 72.6% | **88.1** | 5,876 | 86.5% | 67.0 | 79.6% |
| **w/o TWS** | 46.7 | 9,372 | 68.1% | 87.5 | 4,657 | 68.6% | 67.1 | 68.4% |
| **w/o IGR** | 48.3 | 9,344 | 67.9% | 84.4 | 3,990 | 58.7% | 66.4 | 63.3% |
| **w/ MSW** | 51.7 | **9,246** | **67.2%** | 83.8 | 4,398 | 64.8% | 67.8 | 66.0% |
| **w/ LSW** | 46.7 | 9,815 | 71.3% | 84.4 | **3,595** | **52.9%** | 65.5 | **62.1%** |

Table 3: Ablation studies on DeepSeek-R1-Distill-Qwen-7B. Abbreviations: ETP (Effective Thinking Prototype), EE (Early Exit), TWS (Thinking Word Selection), IGR (Information Gain Ratio), MSW (Most Sensitive Word), LSW (Least Sensitive Word).

likely the least sensitive thinking words. Conversely, when removing *But* or *However* led to a drastic decline in accuracy. This identifies *But* and *However* as the most sensitive thinking words, implying that reflection and transition markers are more critical for complex reasoning processes than verification markers (*Hmm*, *Verify*).

### 4.4 Ablation Studies

In this section, we evaluate the contribution of each module. The configurations are detailed below:

**w/o ETP:** We eliminate the influence of the effective thinking prototype. Instead of using it for thinking word selection, we employ an indistinguishable random attention head representation (similar to the random head in Section 4.2) as the baseline for the thinking prototype.

**w/o EE:** We remove the early exit mechanism by setting the threshold to an extremely large value, forcing the model to continue reasoning until it naturally generates `</think>`.

**w/o TWS:** We bypass the selection process and directly append the original thinking word to resume reasoning.

Table 3 shows that removing any single module results in a significant performance drop, demonstrating the necessity of each component. Furthermore, we conducted three fine-grained experiments specifically regarding the Thinking Word Selection module:

**w/o IGR:** Instead of calculating information gain ratio, we randomly select a word from the candidate list to guide subsequent thinking, testing the rationality of the metric.

**w/ MSW:** We consistently append the most sensitive word (*But*) identified in Section 4.3 at every pause.

**w/ LSW:** We consistently append the least sensitive word (*Hmm*) identified in Section 4.3 at every pause.

The results in Table 3 indicate clear declines across all three settings, further validating the rationality of using IGR to select the most appropriate thinking words. It is worth noting that even when using a fixed, highly sensitive word (*But*), the performance surpasses the *w/o TWS* baseline (where no replacement occurs). This further underscores the necessity of the thinking word selection intervention itself.

## 5 Related Work

**Information encoded in specific tokens.** LLMs encode a wealth of latent information within their internal representations, which has been extensively leveraged across diverse research domains, including data selection, hallucination mitigation, safety alignment, content planning, and logical reasoning [Li *et al.*, 2023; Zhou *et al.*, 2024; Zhao *et al.*, 2025; Dong *et al.*, 2025; Qian *et al.*, 2025]. Investigating the representations of specific tokens has emerged as a prominent research direction. Regarding thinking tokens, Qian *et al.* [2025] demonstrates that the representations of these tokens in LRMs exhibit a sudden and significant increase in Mutual Information (MI) with the golden answer. Concerning the first token, Dong *et al.* [2025] reveals that the model's initial token often implicitly encodes information regarding the final multiple-choice answers. In the context of safety tokens, Qi *et al.* [2024] show that simply forcing an unaligned LLM to begin its responses with specific safe tokens can significantly enhance model safety. These findings underscore the untapped potential within the intermediate states of LLMs, motivating us to further investigate the internal information encapsulated in thinking tokens that specifically as the initial token of a reasoning step and explore their utility.

**Mitigation of overthinking.** A substantial body of work addresses the overthinking problem through post-training methods [Hou *et al.*, 2025; Fatemi *et al.*, 2025; Dong and Fan, 2025; Zhang *et al.*, 2025b; Chung *et al.*, 2025; Chen *et al.*, 2025a]. These approaches primarily utilize data with token budget controls for Supervised Fine-Tuning (SFT) or design reinforcement learning rewards that incorporate length penalties to encourage more efficient reasoning. However, these methods fall outside the scope of our training-free framework.

Additionally, output-based methods [Zhang *et al.*, 2025a; Wang *et al.*, 2025b; Huang *et al.*, 2025; Chen *et al.*, 2025b; Yang *et al.*, 2025b] have been proposed. For instance, Zhang *et al.* [2025a] employs a probe classifier trained on correct/incorrect QA pairs to identify early exit points at chunk answer positions. Nevertheless, this approach still necessitates the training of an external verifier model. Other approaches focus on suppressing reflection during the output process [Wang *et al.*, 2025b; Huang *et al.*, 2025; Chen *et al.*, 2025b]. These methods achieve token reduction by either suppressing the generation probability of reflection tokens during inference or applying vector intervention to inhibit reflection. However, these techniques often compromise the model's inherent reflective capabilities, while they achieve significant token reduction, it frequently comes at the cost of substantial accuracy degradation.

Most closely related to our work is DEER [Yang *et al.*, 2025b], which proposes dynamic early exit for overthinking. DEER triggers early exit by dynamically generating an intermediate answer during inference and evaluating its generation probability. However, DEER primarily performs truncation of long CoT without considering the guidance or transformation of the specific thinking process. Consequently, it fails to address scenarios where the model persists in errors and lacks interpretability. In contrast, our method integrates an effective thinking space partition strategy. By utilizing more

appropriate thinking words, we guide the reasoning process toward more effective directions, altering the reasoning trajectory to escape local error loops. This approach not only improves both reasoning efficiency and performance but also enhances interpretability.

## 6 Conclusion

Inspired by the insight that specific tokens encode rich latent information, our analysis reveals that LRMs inherently possess a latent space that naturally partitions effective thinking from redundancy. Building on this discovery, we propose a novel thinking-word-guided strategy for dynamic early exit to mitigate overthinking. Our approach not only significantly enhances both reasoning efficiency and performance but also improves system interpretability. Furthermore, these results empirically validate the existence of this intrinsic effective thinking partition within LRMs and the precision of our localization method. In future work, we aim to further explore how to leverage this natural partition to steer the reasoning process, such as facilitating immediate exit for intractable problems or implementing more granular optimization of reasoning paths.

## A Appendix

To further illustrate the internal distribution of the Thinking Partition Heads within the model, we present additional t-SNE visualizations of attention heads across different model families in this section.

### A.1 t-SNE Visualization of Attention Heads Across Different Model Families

Specifically, this section provides t-SNE visualizations of the thinking word representations extracted from the Top-1, Top-10, and random attention heads across DeepSeek-R1-Distill-Qwen-7B, Qwen3-14B, and QwQ-32B.

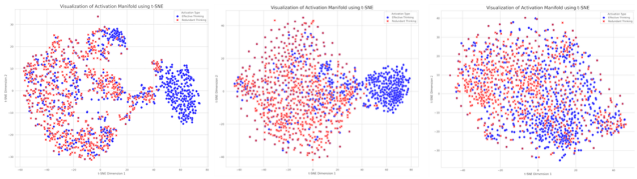

Figure 7: t-SNE visualization of thinking word representations for DeepSeek-R1-Distill-Qwen-7B. Left to right: Top-1, Top-10, and random attention heads.

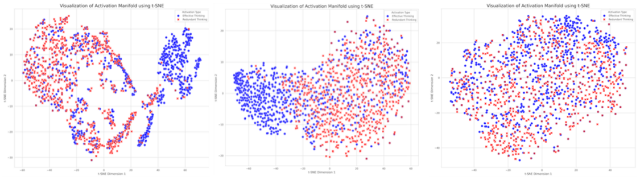

Figure 8: t-SNE visualization of thinking word representations for Qwen3-14B. Left to right: Top-1, Top-10, and random attention heads.

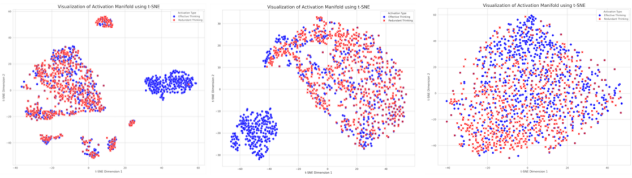

Figure 9: t-SNE visualization of thinking word representations for QwQ-32B. Left to right: Top-1, Top-10, and random attention heads.

## Acknowledgements

Xiaocheng Feng and Xin Liu are the co-corresponding author of this work. We thank the anonymous reviewers for their insightful comments. This work was supported by the Major Key Project of PCL via grant No. PCL2025A11, the National Natural Science Foundation of China (NSFC) (grant 62522603, 62276078), the Key R&D Program of Heilongjiang via grant 2022ZX01A32, and the Fundamental Research Funds for the Central Universities ( XNJKKGYDJ2024013 ). Thanks for the support provided by OpenI Community (https://openi.pcl.ac.cn).

## References

[AI-MO, 2023] AI-MO. Amc 2023. https://huggingface.co/datasets/AI-MO/aimo-validation-amc, 2023.

[Chen *et al.*, 2024] 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.

[Chen *et al.*, 2025a] Qiguang Chen, Dengyun Peng, Jinhao Liu, HuiKang Su, Jiannan Guan, Libo Qin, and Wanxiang Che. Aware first, think less: Dynamic boundary self-awareness drives extreme reasoning efficiency in large language models. *arXiv preprint arXiv:2508.11582*, 2025.

[Chen *et al.*, 2025b] Runjin Chen, Zhenyu Zhang, Junyuan Hong, Souvik Kundu, and Zhangyang Wang. Seal: Steerable reasoning calibration of large language models for free. *arXiv preprint arXiv:2504.07986*, 2025.

[Chung *et al.*, 2025] Stephen Chung, Wenyu Du, and Jie Fu. Thinker: Learning to think fast and slow. *arXiv preprint arXiv:2505.21097*, 2025.

[Cobbe *et al.*, 2021] 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.

[Dong and Fan, 2025] Yubo Dong and Hehe Fan. Enhancing large language models through structured reasoning. *arXiv preprint arXiv:2506.20241*, 2025.

[Dong *et al.*, 2025] Zhichen Dong, Zhanhui Zhou, Zhixuan Liu, Chao Yang, and Chaochao Lu. Emergent response planning in llms. *arXiv preprint arXiv:2502.06258*, 2025.

[Fang *et al.*, 2025] Gongfan Fang, Xinyin Ma, and Xinchao Wang. Thinkless: Llm learns when to think. *arXiv preprint arXiv:2505.13379*, 2025.

[Fatemi *et al.*, 2025] Mehdi Fatemi, Banafsheh Rafiee, Mingjie Tang, and Kartik Talamadupula. Concise reasoning via reinforcement learning. *arXiv preprint arXiv:2504.05185*, 2025.

[Fu *et al.*, 2025] Yichao Fu, Junda Chen, Yonghao Zhuang, Zheyu Fu, Ion Stoica, and Hao Zhang. Reasoning without self-doubt: More efficient chain-of-thought through certainty probing. In *ICLR 2025 Workshop on Foundation Models in the Wild*, 2025.

[Gretton *et al.*, 2005] Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf. Measuring statistical dependence with hilbert-schmidt norms. In *International conference on algorithmic learning theory*, pages 63–77. Springer, 2005.

[Gretton *et al.*, 2012] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The journal of machine learning research*, 13(1):723–773, 2012.

[Guo *et al.*, 2025] 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.

[Hendrycks *et al.*, 2021] 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.

[Hou *et al.*, 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, 2025. *URL https://arxiv.org/abs/2504.01296*, 2025.

[Huang *et al.*, 2025] Jiameng Huang, Baijiong Lin, Guhao Feng, Jierun Chen, Di He, and Lu Hou. Efficient reasoning for large reasoning language models via certainty-guided reflection suppression. *arXiv preprint arXiv:2508.05337*, 2025.

[Jaech *et al.*, 2024] 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.

[Li *et al.*, 2023] Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, 36:41451–41530, 2023.

[Li *et al.*, 2025] Wei Li, Yanbin Wei, Qiushi Huang, Jiangyue Yan, Yang Chen, James T Kwok, and Yu Zhang. Dynamicmind: A tri-mode thinking system for large language models. *arXiv preprint arXiv:2506.05936*, 2025.

[Liang *et al.*, 2025] Guosheng Liang, Longguang Zhong, Ziyi Yang, and Xiaojun Quan. Thinkswitcher: When to think hard, when to think fast. *arXiv preprint arXiv:2505.14183*, 2025.

[Ma *et al.*, 2025] Wenjie Ma, Jingxuan He, Charlie Snell, Tyler Griggs, Sewon Min, and Matei Zaharia. Reasoning models can be effective without thinking. *arXiv preprint arXiv:2504.09858*, 2025.

[MAA Committees, 2024] MAA Committees. Aime problems and solutions. https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions, 2024.

[Maaten and Hinton, 2008] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[Muennighoff *et al.*, 2025] Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori B Hashimoto. s1: Simple test-time scaling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 20286–20332, 2025.

[Poole *et al.*, 2019] Ben Poole, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker. On variational bounds of mutual information. In *International conference on machine learning*, pages 5171–5180. PMLR, 2019.

[Qi *et al.*, 2024] Xiangyu Qi, Ashwinee Panda, Kaifeng Lyu, Xiao Ma, Subhrajit Roy, Ahmad Beirami, Prateek Mittal, and Peter Henderson. Safety alignment should be made more than just a few tokens deep. *arXiv preprint arXiv:2406.05946*, 2024.

[Qian *et al.*, 2025] Chen Qian, Dongrui Liu, Haochen Wen, Zhen Bai, Yong Liu, and Jing Shao. Demystifying reasoning dynamics with mutual information: Thinking tokens are information peaks in llm reasoning. *arXiv preprint arXiv:2506.02867*, 2025.

[Rein *et al.*, 2024] 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.

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

[Snell *et al.*, 2017] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017.

[Sui *et al.*, 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, 2025. *URL https://arxiv.org/abs/2503.16419*, 2025.

[Wang *et al.*, 2025a] Chen Wang, Xunzhuo Liu, Yuhan Liu, Yue Zhu, Xiangxi Mo, Junchen Jiang, and Huamin Chen.

When to reason: Semantic router for vllm. *arXiv preprint arXiv:2510.08731*, 2025.

[Wang *et al.*, 2025b] 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*, 2025.

[Wei *et al.*, 2022] 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.

[Wu *et al.*, 2024] Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. Scaling inference computation: Compute-optimal inference for problem-solving with language models. In *The 4th Workshop on Mathematical Reasoning and AI at NeurIPS*, volume 24, 2024.

[Wu *et al.*, 2025] Yuyang Wu, Yifei Wang, Ziyu Ye, Tianqi Du, Stefanie Jegelka, and Yisen Wang. When more is less: Understanding chain-of-thought length in llms. *arXiv preprint arXiv:2502.07266*, 2025.

[Yang *et al.*, 2025a] 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*, 2025.

[Yang *et al.*, 2025b] 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*, 2025.

[Ye *et al.*, 2025] Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. Limo: Less is more for reasoning. *arXiv preprint arXiv:2502.03387*, 2025.

[Yue *et al.*, 2025] 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.

[Zhang *et al.*, 2025a] Anqi Zhang, Yulin Chen, Jane Pan, Chen Zhao, Aurojit Panda, Jinyang Li, and He He. Reasoning models know when they're right: Probing hidden states for self-verification. *arXiv preprint arXiv:2504.05419*, 2025.

[Zhang *et al.*, 2025b] Jiajie Zhang, Nianyi Lin, Lei Hou, Ling Feng, and Juanzi Li. Adaptthink: Reasoning models can learn when to think, 2025. *URL https://arxiv.org/abs/2505.13417*, 2025.

[Zhao *et al.*, 2025] Dezhi Zhao, Xin Liu, Xiaocheng Feng, Hui Wang, and Bing Qin. Probing and boosting large language models capabilities via attention heads. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 28518–28532, 2025.

[Zhou *et al.*, 2024] Zhenhong Zhou, Haiyang Yu, Xinghua Zhang, Rongwu Xu, Fei Huang, and Yongbin Li. How alignment and jailbreak work: Explain llm safety through intermediate hidden states. *arXiv preprint arXiv:2406.05644*, 2024.