Reconsidering Overthinking: Penalizing Internal and External Redundancy in CoT Reasoning

Anonymous ACL submission

Abstract

Large Reasoning Models (LRMs) often suffer from overthinking, generating verbose reasoning traces that compromise both computational efficiency and interpretability. Unlike prior efforts that rely on global length-based rewards, we propose a semantic-aware decomposition of redundancy into two distinct forms: internal redundancy (informational stagnation within the reasoning process) and external redundancy (superfluous continuation after the final answer). We introduce a dual-penalty reinforcement learning framework that surgically targets these inefficiencies: a sliding-window semantic analysis is employed to penalize low-gain steps within the reasoning trajectory, while a normalized metric suppresses the post-answer tail. Extensive experiments demonstrate that our method significantly compresses Chain-of-Thought traces with minimal accuracy degradation, while maintaining strong generalization to out-of-domain tasks. Crucially, we reveal an asymmetry in redundancy: external redundancy can be safely eliminated without performance loss, whereas internal redundancy removal requires a calibrated trade-off to maintain reasoning fidelity. Our framework enables fine-grained, implicit control over reasoning length, paving the way for more concise and interpretable LRMs. Our code is here.

1 Introduction

Large reasoning models (LRMs), represented by OpenAI's o1 (Jaech et al., 2024), DeepSeek-R1 (Guo et al., 2025), and QwQ (Team, 2025), have achieved a paradigm shift in complex problem-solving. This success is primarily attributed to the emergence of dense Chain-of-Thought (CoT) sequences, which allow models to manifest intermediate reasoning steps and navigate complex logical landscapes. However, this increased reasoning depth comes at a significant cost: LRMs frequently exhibit pathological overthinking, the generation of redundant, repetitive, or superfluous reasoning traces that escalate computational overhead and obscure the core logical derivation (Chen et al., 2024; Sui et al., 2025).

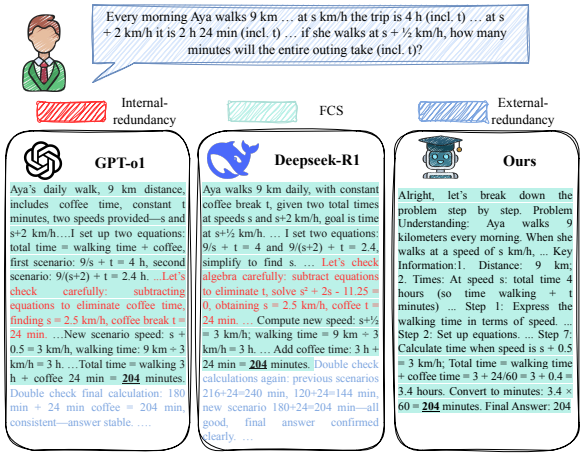


Figure 1: Response examples for AIME24. The underlined answer (e.g., "204") acts as a delimiter: content before it is the First Correct Solution (FCS) subject to internal redundancy; content after it is external redundancy. Our method generates a more efficient FCS (full content in Appendix D) while eliminating superfluous post-answer text compared to R1 and o1.

Recent research (Liu et al., 2025a; Sheng et al., 2025; Wang et al., 2025a,b) has begun to investigate CoT compression, primarily through reinforcement learning (RL) techniques that penalize total sequence length. However, these methods often treat reasoning as a one-dimensional token budget problem (Luo et al., 2025a; Arora and Zanette, 2025), overlooking the underlying semantic structure of redundancy. Such indiscriminate length-reduction strategies can inadvertently suppress essential reasoning steps, leading to a precarious trade-off between conciseness and accuracy. In this work, we propose that effectively mitigating overthinking requires a fine-grained, semantic-aware decomposition of the reasoning process.

We introduce a novel taxonomy that bisects CoT

redundancy based on the **First Correct Answer (FCA)**, the earliest point where the model attains the final answer (Chen et al., 2024). This framework distinguished between two fundamentally different types of inefficiency:

1. **Internal Redundancy:** This refers to informational stagnation within the First Correct Solution (FCS), where the model "cycles" through semantically similar content or reiterates premises without advancing the logical state.

2. **External Redundancy:** This denotes the "post-answer tail", the unnecessary continuation, re-derivation, or verification that occurs after the correct answer has already been reached.

As illustrated in Figure 1, mainstream LRMs exhibit significant density in both dimensions. Our analysis suggests that internal redundancy reflects a lack of **reasoning efficiency**, while external redundancy signifies a failure in **termination awareness**. By disentangling these components, we can apply targeted surgical penalties during RL training rather than relying on blunt global length constraints.

To operationalize this, we develop a **Dual-Redundancy Penalty** framework. For internal redundancy, we utilize a dynamic sliding-window semantic similarity metric to detect segments with low informational progression. Crucially, we implement an implicit threshold mechanism within this penalty to protect the model's essential reasoning structure, ensuring that only "stagnant" tokens are penalized. For external redundancy, we apply a normalized proportion-based penalty to encourage prompt termination upon reaching the FCA. Unlike prior length-constrained RL, our approach optimizes the information density of the reasoning trajectory, fostering a concise yet coherent logical flow.

Extensive experiments across multiple mathematical benchmarks demonstrate that our framework significantly compresses CoT sequences while preserving, or even refining, reasoning accuracy. Our ablation studies provide a critical insight: **external redundancy can be almost entirely eliminated with negligible impact on performance, whereas internal redundancy removal follows a more sensitive Pareto frontier.** This highlights the safety and necessity of our decoupled approach. In summary, our contributions are:

- We propose the first systematic decomposition of CoT overthinking into internal and external components, shifting the focus from sequence length to semantic efficiency.

- We design a semantic-aware RL reward featuring a sliding-window internal redundancy penalty with an implicit threshold, enabling precise control over reasoning fidelity and brevity.

- We demonstrate through extensive empirical results that our dual-penalty mechanism achieves a superior accuracy-efficiency trade-off compared to existing global-length baselines.

- We show that our findings generalize across different model scales and architectures, confirming that the "safety" of external redundancy removal is a robust property of large reasoning models.

2 Related Work

Chain-of-Thought Reasoning CoT prompting (Wei et al., 2022) and its subsequent refinements have established a paradigm for eliciting step-by-step reasoning in Large Language Models (LLMs), significantly enhancing performance on complex cognitive tasks (Qiao et al., 2022). However, as models scale and tasks grow in complexity, a pathological phenomenon known as overthinking has emerged (Chen et al., 2024; Team et al., 2025). Excessive verbosity in reasoning traces not only inflates computational overhead during inference but also obscures the transparency and interpretability of the model's logical derivation.

CoT Compression via Reinforcement Learning Recent efforts have pivoted toward optimizing reasoning efficiency, primarily utilizing Reinforcement Learning (RL) to incentivize brevity. Most existing approaches employ global length-based rewards to penalize sequence length (Team et al., 2025; Arora and Zanette, 2025; Shen et al., 2025; Qu et al., 2025) or implement hard token-budget constraints during training (Hou et al., 2025). While effective at reducing total token counts, these methods treat redundancy as a monolithic attribute. By relying on coarse-grained length metrics, they risk inadvertently suppressing critical reasoning steps, leading to a fragile trade-off between conciseness and logical integrity.

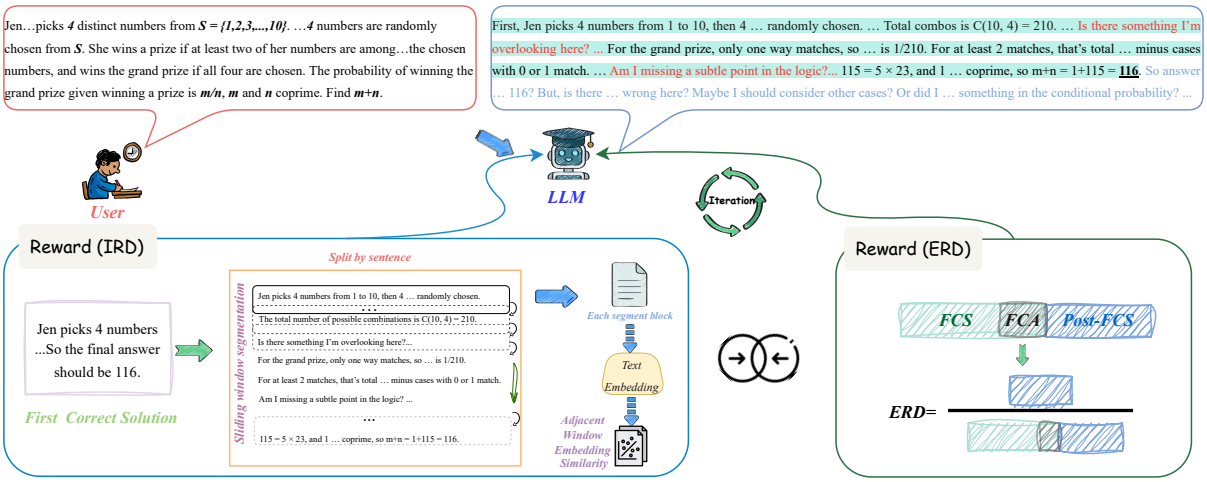


Figure 2: The Dual-redundancy Reward framework iteratively optimizes the LLM via complementary redundancy detection.

3 Redundancy Detection

Redundant reasoning in LRMs can occur at different stages of the CoT process (Han et al., 2024; Liu et al., 2024; Ma et al., 2025). As shown in Figure 1, we observe that redundancy clusters either before or after the first correct answer, motivating a segmentation-based analysis.

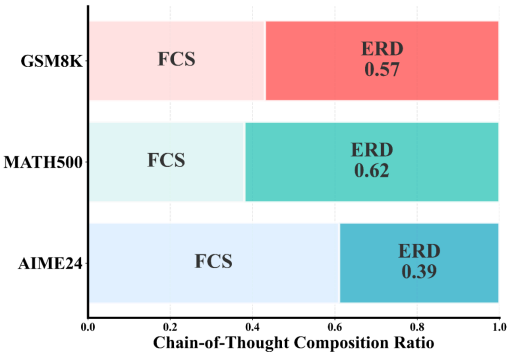


Figure 3: Analysis of ERD. Since human-written solutions inherently contain no external redundancy, we only report the ERD performance for the LRMs.

3.1 External Redundancy

Definition 1 (External Redundancy)**.** Considering a question Q, any content in a generated solution that appears after the first sentence containing the correct answer A is defined as external redundancy.

External redundancy comprises all tokens generated post-FCS. This content is generally superfluous to the derivation process, does not contribute to the final answer's correctness, and can be considered uninformative "overthinking". We emphasize that any trial-and-error reasoning that occurs *before* the FCA is not categorized as redundant.

To quantify the severity of this redundancy, we propose the External Redundancy Degree (ERD). The ERD measures the proportion of redundant content rather than its absolute length, which prevents bias against intrinsically longer CoT outputs. Specifically, for a given solution sequence:

$$\text{ERD} = 1 - \frac{T_{fcs}}{T_{total}} \tag{1}$$

where T_{fcs} is the number of tokens in the FCS, and T_{total} is the total length of the reasoning trace. As illustrated in Figure 3, a higher ERD value indicates a greater degree of post-hoc redundancy in the CoT process.

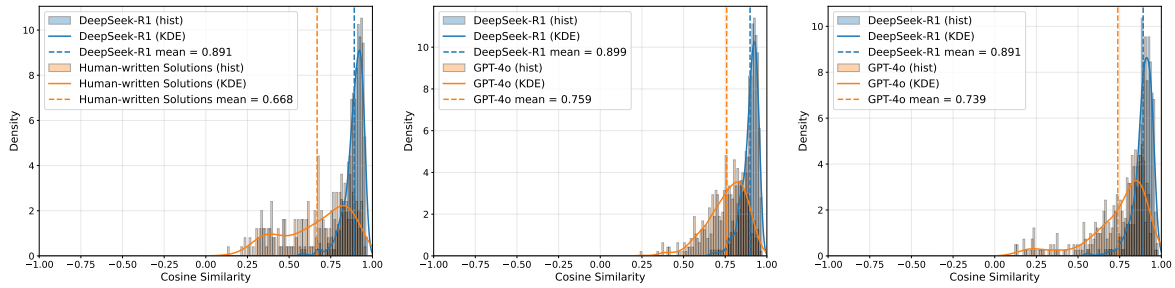
3.2 Internal Redundancy

3.2.1 Definition and Methodology

Unlike external redundancy, internal redundancy, occurring within the reasoning process prior to the FCA, is more abstract. To formalize this concept, we propose the following framework:

Assumption (Logical Trajectory)**.** For any solvable question Q, there exists an underlying minimal logical trajectory $L = (Q, l_1, l_2, \ldots, l_k, A)$, where each l_i represents a necessary and non-redundant logical transition.

In practice, a model's solution X serves as a natural language expansion of L. Internal redundancy arises when the expansion of a specific transition (e.g., from l_i to l_{i+1}) becomes excessively wordy without introducing new logical information.



(a) AIME2024 (b) LCB (c) GPQA

Figure 4: IRD Analysis on different datasets. The local similarity of DeepSeek-R1 is significantly higher than that of human answers and GPT-4o.

Definition 2 (Internal Redundancy)**.** Given a logical trajectory L, internal redundancy is defined as any linguistic expansion within the First Correct Solution that does not facilitate a transition to a subsequent logical step l_i.

Internal redundancy manifests as the excessive use of tokens to reiterate or over-explain existing logical states without advancing the reasoning process. This form of redundancy represents a degradation in reasoning efficiency, where the model's output becomes dense in tokens but sparse in new information. Based on this, we formalize its detection via the following hypothesis:

Hypothesis (Similarity-Redundancy Correlation)**.** For two solutions X_a and X_b derived from the same trajectory L, if X_a is more redundant than X_b, it will exhibit higher local semantic similarity.

To quantify this phenomenon, we propose the **Internal Redundancy Degree (IRD)**, which employs a dynamic sliding-window approach to measure local semantic density.

Methodology Given a solution X partitioned into N sentences $\{s_1, \ldots, s_N\}$, we define a window size $w = \lfloor \alpha N \rfloor$ and a stride $t = \lfloor \beta N \rfloor$ (with $\alpha = 0.1, \beta = 0.05$). For each window W_i, we compute its embedding $v_i = f_{\text{embed}}(W_i)$. The IRD is defined as the average cosine similarity between adjacent windows:

$$\text{IRD} = \frac{1}{M-1} \sum_{i=1}^{M-1} \cos(v_i, v_{i+1}) \qquad (2)$$

where M is the total number of windows. A high IRD signals that adjacent segments are semantically stagnant, indicating that the model is "cycling" through similar concepts without sufficient logical advancement.

To justify the design of this metric, we conducted extensive comparative experiments across various window scales, specifically evaluating the efficacy of absolute sentence counts versus dynamic ratios. Our results demonstrate that the ratio-based approach provides superior resolution in distinguishing local similarity across sequences of varying lengths, confirming its robustness over fixed-length alternatives (detailed experiments are provided in Appendix A.1 and A.2).

Property We emphasize that IRD is a *relative metric*. Our objective is not to reduce similarity to zero, as a baseline level of redundancy is essential for maintaining semantic coherence, but to identify the optimal efficiency balance. Furthermore, by utilizing a proportional window design, IRD effectively measures informational density per unit of progress. This allows for consistent comparisons across varying problem complexities and divergent logical trajectories.

3.2.2 Empirical Validation

To validate the IRD metric, we compare the reasoning traces of DeepSeek-R1 against high-quality, concise benchmarks, including human-written solutions and GPT-4o outputs. We randomly sampled 20 instances from each of the three datasets: AIME24, GPQA (Rein et al., 2024), and LiveCodeBench (Jain et al., 2024). As illustrated in Figure 4, the semantic similarity distributions reveal a stark contrast in reasoning patterns. The IRD values for DeepSeek-R1 are densely clustered in the high-similarity region (typically above 0.75), indicating a high degree of informational stagnation. In contrast, the distributions for human references and GPT-4o are notably more dispersed and uniform across a lower similarity range. Furthermore, the mean IRD of DeepSeek-R1 is significantly higher

than that of both human and GPT-4o benchmarks across all evaluated domains. These empirical results confirm that diminished reasoning efficiency consistently manifests as elevated local semantic similarity, validating the IRD as a robust metric for quantifying internal redundancy.

4 Dual-Redundancy Penalty

To mitigate both internal and external redundancy, we augment the reinforcement learning objective by incorporating two distinct penalty terms into the accuracy-based reward function $R_{total} = R_{acc} \cdot p_{int} \cdot p_{ext}$. These penalties, derived from our defined redundancy degrees, incentivize the model to generate concise reasoning trajectories while eliminating unnecessary repetition or post-answer continuation.

Internal Redundancy Penalty In section 6.1, our ablations reveal that excessive compression of the internal reasoning process can lead to significant accuracy degradation. To safeguard the model's reasoning integrity, we design the internal redundancy penalty with an implicit threshold mechanism using a sharpened sigmoid function:

$$p_{\text{int}} = 1 - \sigma(\text{IRD}) \tag{3}$$

$$\sigma(x) = \frac{1}{1 + e^{-k(x-c)}} \tag{4}$$

where $\sigma(x)$ denotes a sigmoid function with steepness k and center c. In this work, we set $k = 20$ and $c = 0.7$. This configuration ensures that the penalty remains negligible when IRD is below a safe threshold (approx. 0.5), but escalates rapidly once redundancy exceeds this limit. By tuning k and c, we can precisely calibrate the tolerance threshold and the severity of the penalty. Unlike global length-based rewards that treat all tokens equally, this design enables fine-grained control over internal redundancy, balancing brevity with reasoning fidelity.

External Redundancy Penalty To discourage post-answer verbosity, we apply a normalized linear penalty based on the ERD:

$$p_{\text{ext}} = 1 - \text{ERD} \tag{5}$$

5 Experiment

5.1 Training Setup

We adopt verl (Sheng et al., 2024), a high-throughput, scalable reinforcement learning library optimized for LLMs. All experiments are conducted using the Group Relative Policy Optimization (GRPO) (Shao et al., 2024) algorithm across a cluster of 64 NVIDIA A800 GPUs. Our training is performed with a maximum response length of 16k tokens. We employ a sampling temperature of 0.6 and a top-p of 1.0. During the GRPO process, we set the group size to 8 samples per prompt and a global batch size of 128.

We fine-tune *DeepSeek-R1-Distill-Qwen-1.5B* and *7B* on the DeepScaleR dataset (Luo et al., 2025b). To ensure robust identification and extraction of the First Correct Solution from reasoning trajectories, we specifically filter for problems with numeric answers containing at least two digits, thereby minimizing extraction noise. The comprehensive training procedure is formalized in Appendix B.

5.2 Baselines

We benchmark our method against state-of-the-art RL-based compression approaches, all of which rely on global length-based objectives.

ThinkPrune (Hou et al., 2025): Implements a hard token-budget truncation during RL, compelling the model to condense reasoning within a fixed length.

LC-R1 (Cheng et al., 2025): Utilizes an auxiliary LLM to provide external supervision, rewarding the model based on the compression ratio between the original and teacher-distilled responses.

Laser-DE (Liu et al., 2025b): Employs a soft-margin reward mechanism, incentivizing correct outputs that fall below a predefined target length within a large context window.

Training (Arora and Zanette, 2025): Exploits intra-sample competition during RL, assigning higher rewards to shorter trajectories among multiple correct completions.

5.3 Evaluation Setup

We conduct evaluations across three mathematical reasoning benchmarks of varying difficulty: GSM8K (Cobbe et al., 2021), MATH500 (Hendrycks et al., 2021), and AIME24. Reasoning accuracy is measured using the **Pass@1** metric. During inference, we set the maximum generation length to 16k tokens. For GSM8K and MATH500, we sample $n = 4$ responses per problem at a temperature of 0.6; for AIME24, we increase the sample size to $n = 64$ to account for its smaller problem set.

To ensure a fair comparison across CoT compression methods, some of which may truncate or omit the final conclusion, we exclude the terminal answer statement from our token count statistics. This refinement ensures that our length measurements precisely reflect the efficiency of the reasoning trajectory itself rather than the final formatting.

To evaluate the trade-off between reasoning performance and computational cost, We adopt the Accuracy-Efficiency Score (AES) as proposed by (Luo et al., 2025a). The AES provides a holistic metric to assess model efficiency, defined as:

$$\text{AES} = \begin{cases} \alpha \cdot \Delta\text{Len} + \beta \cdot |\Delta\text{Acc}|, & \text{if } \Delta\text{Acc} \geq 0 \\ \alpha \cdot \Delta\text{Len} - \gamma \cdot |\Delta\text{Acc}|, & \text{if } \Delta\text{Acc} < 0 \end{cases} \tag{6}$$

where the relative changes in length (ΔLen) and accuracy (ΔAcc) are calculated relative to the baseline:

$$\Delta\text{Len} = \frac{\text{Len}_{\text{base}} - \text{Len}_{\text{model}}}{\text{Len}_{\text{base}}} \tag{7}$$

$$\Delta\text{Acc} = \frac{\text{Acc}_{\text{model}} - \text{Acc}_{\text{base}}}{\text{Acc}_{\text{base}}} \tag{8}$$

Following the original implementation, we set the hyperparameters to $\alpha = 1$, $\beta = 3$, and $\gamma = 5$. This configuration prioritizes accuracy preservation by assigning it higher weights (β, γ) relative to length reduction, while the larger value of γ imposes a more stringent penalty for any degradation in reasoning accuracy.

5.4 Main Results

As summarized in Table 1, our method achieves state-of-the-art (SOTA) results in terms of AES across both the *DeepSeek-R1-Distill-Qwen-1.5B* and *7B* models. The performance gain is particularly pronounced on the more challenging AIME24 benchmarks, where our approach significantly outperforms all baseline methods.

A qualitative analysis reveals that despite the overall length reduction in existing baselines, many still harbor substantial internal and external redundancy in their reasoning traces. This highlight a critical limitation of global length-based rewards: they may reduce volume without necessarily improving informational density. In contrast, our method acts as a precision tool for CoT refinement.

We further observe a strong correlation between high AES scores and low redundancy degrees across all baselines. This convergent trend suggests that effective CoT compression is inherently driven by the minimization of internal and external redundancy, an observation that provides strong empirical validation that our metrics successfully capture the fundamental nature of overthinking in LRMs.

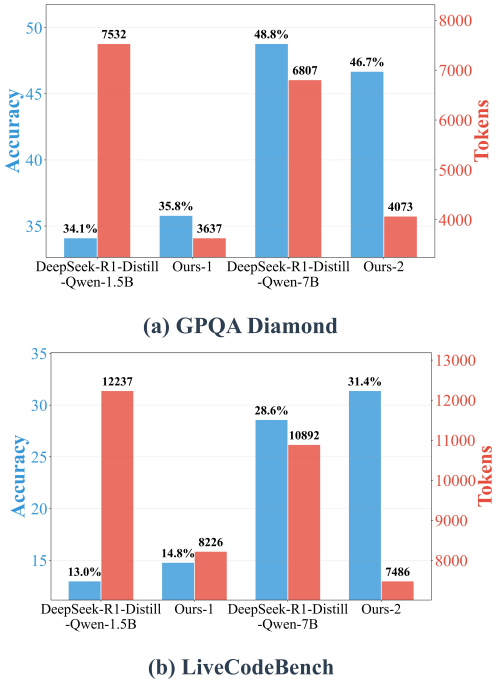


Figure 5: Performance on GPQA and LiveCodeBench. Our method generalizes well to out-of-domain reasoning tasks.

5.5 Cross-Domain Generalization

To assess the generation of our reinforcement learning framework, we evaluate whether redundancy compression patterns learned from mathematical tasks transfer to non-mathematical domains. We test on two out-of-domain (OOD) benchmarks: GPQA diamond and LiveCodeBench.

As illustrated in Figure 5, our method consistently compresses CoT traces across both benchmarks while maintaining reasoning integrity. These results indicate that the model has internalized a domain-agnostic paradigm for concise reasoning rather than merely overfitting to the training distribution. This underscores the robustness and broader transferability of our dual-penalty framework to diverse reasoning-intensive tasks.

6 Ablations

6.1 Internal vs. External Redundancy

We conduct an ablation study to investigate the individual and joint effects of internal and external

Model	GSM8K					MATH500					AIME24					Overall
	Acc	Tokens	IRD	ERD	AES	Acc	Tokens	IRD	ERD	AES	Acc	Tokens	IRD	ERD	AES	AES
DeepSeek-R1-Distill-Qwen-1.5B																
Baseline	84.1	1555	73.7	43.0	/	82.2	3549	77.5	55.2	/	28.5	8681	71.4	28.6	/	/
ThinkPrune-4k	86.1	910	77.9	40.1	0.49	83.7	2101	73.2	39.8	0.46	28.6	6431	75.2	21.0	0.27	1.22
LC-R1	82.5	507	67.2	19.3	0.58	79.6	1673	75.8	22.5	0.37	24.2	5075	79.6	20.4	-0.34	0.61
Laser-DE	86.4	971	74.3	37.5	0.46	83.6	2282	78.0	36.3	0.41	32.7	7268	73.5	22.2	0.60	1.47
Training	81.0	292	61.6	7.8	0.63	82.8	1543	65.5	14.5	0.59	28.5	7049	73.2	17.4	0.21	1.13
Ours	84.9	513	49.6	5.7	**0.70**	83.8	1505	51.0	7.9	**0.63**	34.0	6077	72.5	10.9	**0.88**	**2.21**
DeepSeek-R1-Distill-Qwen-7B																
Baseline	91.1	844	70.0	36.0	/	91.2	2836	78.1	51.6	/	52.3	7241	77.8	31.1	/	/
ThinkPrune-4k	92.8	716	70.5	36.0	0.21	89.7	1683	77.9	36.1	0.32	50.4	5723	79.2	14.6	0.03	0.56
LC-R1	87.5	152	61.8	4.9	**0.62**	87.5	1201	65.8	7.0	0.37	52.7	6087	79.1	10.2	0.18	1.17
Laser-DE	93.3	637	68.2	31.1	0.32	92.1	1402	77.0	30.1	**0.54**	52.7	5061	80.5	11.8	0.32	1.18
Training	91.2	387	65.1	14.6	0.54	91.0	2090	76.3	38.3	0.25	50.8	6669	78.8	23.1	-0.06	0.73
Ours	90.9	318	51.8	6.5	0.61	89.8	1200	58.7	6.1	0.50	53.2	5025	77.4	3.7	**0.36**	**1.47**

Table 1: Performance comparison across CoT compression baselines. *Acc* refers to Pass@1; *Tokens* represents the average reasoning length. Our method achieves the optimal trade-off between reasoning fidelity and token efficiency. For improved legibility, IRD and ERD are scaled by a factor of 100.

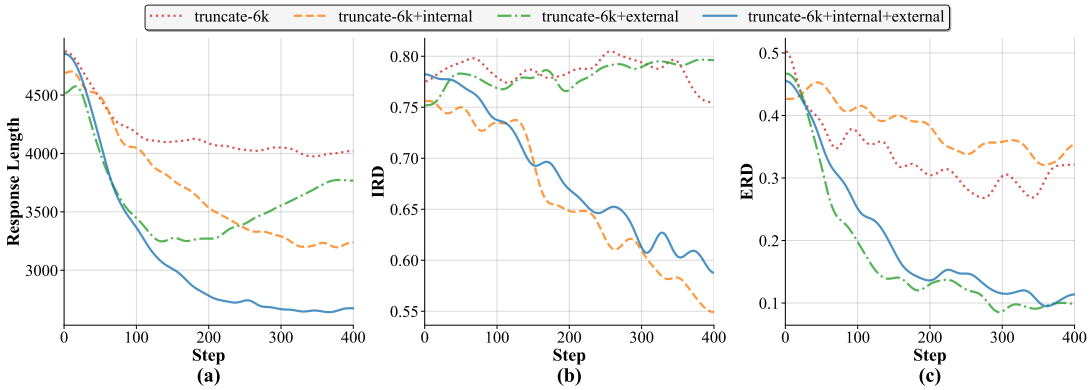


Figure 6: Impact of internal and external redundancy penalties on CoT compression. These two penalties operate independently with minimal interference, yet their combined use enhances compression efficiency beyond individual applications.

redundancy penalties on CoT compression. Figure 6 illustrates the training dynamics across four experimental configurations. Compared to a baseline utilizing only a length-based truncation penalty, incorporating either internal or external redundancy penalties yields a more pronounced reduction in converged response length. Notably, the simultaneous application of both penalties achieves the most significant compression, demonstrating a clear synergistic effect.

As shown in Figures 6b and 6c, we track the evolution of IRD and ERD throughout the training process. Our analysis reveals two key insights:

1. **Inefficacy of Global Truncation:** Global truncation penalties fail to effectively target either internal or external redundancy, explaining their inferior compression performance.

2. **Orthogonality of Redundancy Types:** The internal and external redundancy penalty appear to operate independently. We attribute this independence to the spatial isolation of these two redundancy domains within the solution sequence (i.e., before and after the first correct answer). Despite the fact that the GRPO reward signal is distributed across all tokens, the model successfully identifies the specific sources of redundancy and optimizes accordingly.

In addition, we observe a non-monotonic trend in response length when only the external redundancy penalty is applied: it initially decreases but subsequently rebounds. This suggests that the model attempts to circumvent the penalty by increasing the total response length to lower the relative proportion of external redundancy. Notably, this side effect is effectively neutralized by the simultaneous application of the internal redundancy penalty. The disappearance of this trend underscores the robustness of our dual redundancy penalty mechanism. Similar empirical results for the 7B model are provided in Appendix C.1.

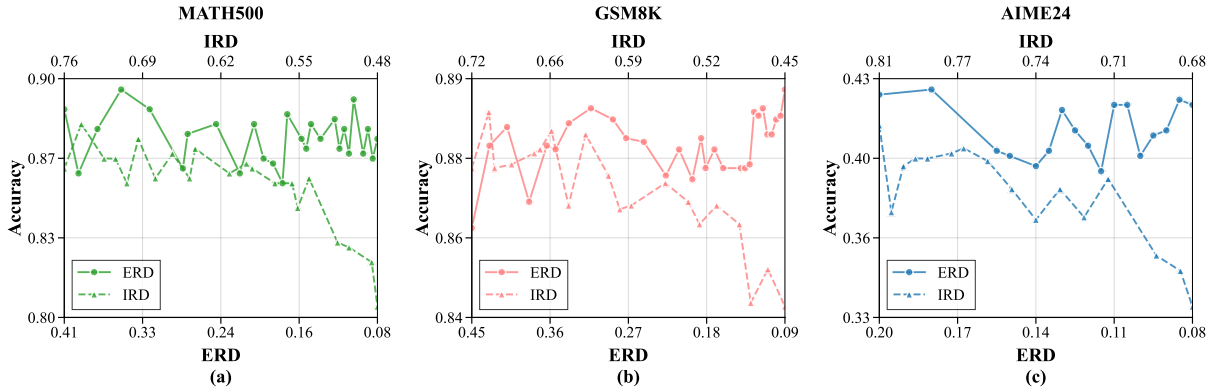


Figure 7: Impact of IRD and ERD reduction on accuracy. Reducing IRD consistently lowers accuracy, whereas penalizing external redundancy does not harm performance.

6.2 Analysis of Accuracy Drop

To isolate the causes of accuracy degradation, we conduct an ablation study on *DeepScaleR-1.5B-Preview*. Unlike models that may still gain performance during RL, this performance-saturated model provides a stable testbed. Its plateaued reasoning capabilities allow us to observe the intrinsic impact of redundancy compression, decoupling it from the interference of reward-driven improvements.

Experimental Protocol The ablation is conducted in two sequential stages.

- **Stage I (External):** We first apply the external redundancy penalty alone during RL training with a 16k maximum response length. After 160 steps, as the average response length converges around 4100 tokens, we conclude this stage.
- **Stage II (Internal):** Starting from the Stage I checkpoint, we proceed to a second phase where the internal redundancy penalty is added.

Core Findings Our results reveal a stark contrast between the two types of redundancy. As shown in Figure 7, when the ERD is reduced to 0.09, the model maintains nearly identical accuracy across all three benchmarks (GSM8K, MATH500, and AIME24) compared to its initial state. In contrast, progressive reduction of the IRD triggers a substantial drop in performance, particularly on the challenging AIME24 dataset. This disparity suggests that **accuracy degradation during CoT compression is primarily attributable to the removal of internal redundancy**. We hypothesize that aggressive IRD compression forces the model to bypass essential intermediate steps, widening the semantic gap between adjacent reasoning segments. This disruption of local coherence leads to "reasoning leaps" that exceed the model's inherent inference capacity, aligning with observations in (Xu et al., 2025).

Identical trends are observed in *DeepSeek-R1-Distill-Llama-8B* (see Appendix C.2), confirming that the asymmetric impact of redundancy holds across different architectures and scales.

7 Conclusion

In this paper, we introduced a novel perspective on overthinking in LRMs by decomposing it into internal and external redundancy. We developed a dual-penalty reinforcement learning framework that utilizes fine-grained semantic rewards to surgically mitigate both inefficiencies. Our results demonstrate that this approach significantly compresses reasoning traces while strictly preserving accuracy across multiple benchmarks. A key insight from our study is that external redundancy can be safely eliminated without performance loss, whereas internal redundancy requires a calibrated trade-off to maintain reasoning fidelity. These findings provide a robust foundation for developing more efficient, interpretable, and human-aligned reasoning models.

Limitations

Despite the effectiveness of our dual-redundancy penalty framework, several limitations warrant further investigation.

- Our methodology relies on the identification of the First Correct Answer as a pivot for redundancy decomposition. While this is

straightforward in objective reasoning tasks such as mathematics and programming, extending this framework to open-ended generation or tasks without a singular objective answer remains challenging. In such scenarios, defining the boundary between essential elaboration and redundancy becomes inherently subjective.

- The IRD metric, based on sliding-window semantic similarity, primarily captures linguistic verbosity and informational stagnation. However, it may struggle to detect higher-level logical redundancies, such as cyclic reasoning that utilizes diverse vocabulary. The sensitivity of the IRD is also coupled with the performance of the underlying embedding model, which may exhibit biases in specific specialized domains.

- While we demonstrated strong out-of-domain generalization, our study predominantly focuses on reasoning-intensive tasks. The impact of internal redundancy compression on tasks requiring divergent thinking or creative exploration has not yet been fully explored, where a certain level of redundancy might be beneficial for maintaining contextual nuance.

References

Daman Arora and Andrea Zanette. 2025. Training language models to reason efficiently. *arXiv preprint arXiv:2502.04463*.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, and 1 others. 2024. Do not think that much for 2+ 3=? on the overthinking of o1-like llms. *arXiv preprint arXiv:2412.21187*.

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

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Tingxu Han, Zhenting Wang, Chunrong Fang, Shiyu Zhao, Shiqing Ma, and Zhenyu Chen. 2024. Token-budget-aware llm reasoning. *arXiv preprint arXiv:2412.18547*.

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

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

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, and 1 others. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. 2024. Livecodebench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*.

Kaiyuan Liu, Chen Shen, Zhanwei Zhang, Junjie Liu, Xiaosong Yuan, and Jieping ye. 2025a. Efficient reasoning through suppression of self-affirmation reflections in large reasoning models. *Preprint*, arXiv:2506.12353.

Tengxiao Liu, Qipeng Guo, Xiangkun Hu, Cheng Jiayang, Yue Zhang, Xipeng Qiu, and Zheng Zhang. 2024. Can language models learn to skip steps? *arXiv preprint arXiv:2411.01855*.

Wei Liu, Ruochen Zhou, Yiyun Deng, Yuzhen Huang, Junteng Liu, Yuntian Deng, Yizhe Zhang, and Junxian He. 2025b. Learn to reason efficiently with adaptive length-based reward shaping. *arXiv preprint arXiv:2505.15612*.

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

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Tianjun Zhang, Li Erran Li, and 1 others. 2025b. Deepscaler: Surpassing o1-preview with a 1.5 b model by scaling rl. *Notion Blog*.

Xinyin Ma, Guangnian Wan, Runpeng Yu, Gongfan Fang, and Xinchao Wang. 2025. Cot-valve: Length-compressible chain-of-thought tuning. *arXiv preprint arXiv:2502.09601*.

Shuofei Qiao, Yixin Ou, Ningyu Zhang, Xiang Chen, Yunzhi Yao, Shumin Deng, Chuanqi Tan, Fei Huang, and Huajun Chen. 2022. Reasoning with language model prompting: A survey. *arXiv preprint arXiv:2212.09597*.

Yuxiao Qu, Matthew YR Yang, Amrith Setlur, Lewis Tunstall, Edward Emanuel Beeching, Ruslan Salakhutdinov, and Aviral Kumar. 2025. Optimizing test-time compute via meta reinforcement fine-tuning. *arXiv preprint arXiv:2503.07572*.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Yi Shen, Jian Zhang, Jieyun Huang, Shuming Shi, Wenjing Zhang, Jiangze Yan, Ning Wang, Kai Wang, Zhaoxiang Liu, and Shiguo Lian. 2025. Dast: Difficulty-adaptive slow-thinking for large reasoning models. *arXiv preprint arXiv:2503.04472*.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2024. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*.

Leheng Sheng, An Zhang, Zijian Wu, Weixiang Zhao, Changshuo Shen, Yi Zhang, Xiang Wang, and Tat-Seng Chua. 2025. On reasoning strength planning in large reasoning models. *arXiv preprint arXiv:2506.08390*.

Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Hanjie Chen, and 1 others. 2025. Stop overthinking: A survey on efficient reasoning for large language models. *arXiv preprint arXiv:2503.16419*.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, and 1 others. 2025. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*.

Qwen Team. 2025. Qwq-32b: Embracing the power of reinforcement learning.

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

Yue Wang, Qiuzhi Liu, Jiahao Xu, Tian Liang, Xingyu Chen, Zhiwei He, Linfeng Song, Dian Yu, Juntao Li, Zhuosheng Zhang, and 1 others. 2025b. Thoughts are all over the place: On the underthinking of o1-like llms. *arXiv preprint arXiv:2501.18585*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Haolei Xu, Yuchen Yan, Yongliang Shen, Wenqi Zhang, Guiyang Hou, Shengpei Jiang, Kaitao Song, Weiming Lu, Jun Xiao, and Yueting Zhuang. 2025. Mind the gap: Bridging thought leap for improved chain-of-thought tuning. *arXiv preprint arXiv:2505.14684*.

A Window Size Analysis

A.1 Dynamic Window Size

To evaluate the robustness of the Internal Redundancy Degree metric, we investigate the impact of the window size ratio (α) on the detection of local semantic similarity. In this experiment, we vary α across a range of scales while maintaining the stride β at half the window size ($\beta = \alpha/2$). The analysis is conducted on solutions from both *DeepSeek-R1* and human references within the MATH500 dataset. As illustrated in Figure 8, the disparity in semantic similarity between *DeepSeek-R1* and human-authored solutions narrows as the window size ratio increases. We hypothesize that larger window sizes encompass excessively broad reasoning segments, which tends to average out the semantic signals and obscure the fine-grained, local informational progression. Based on these empirical results, we selected a window size of $\alpha = 0.1$ for our primary experiments. This configuration provides a sufficiently high resolution to capture local sentence-level redundancies.

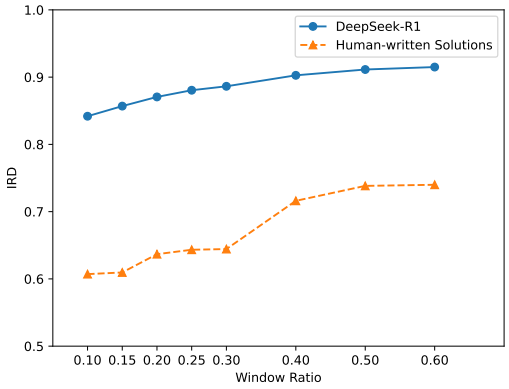


Figure 8: Impact of window size ratio α on local semantic similarity detection.

A.2 Absolute-Sentence Window Size

To further justify the necessity of our dynamic window size design, we investigate an alternative approach using a fixed number of sentences n as the window size, rather than a relative proportion α. In this setup, the sliding stride is consistently maintained at half the window size ($n/2$). We compare the Internal Redundancy Degree of *DeepSeek-R1* and human references across varying values of n on the MATH500 dataset. As illustrated in Figure 9, the results demonstrate two critical limitations of the absolute window approach. First, the IRD of *DeepSeek-R1* does not consistently maintain a significantly higher level compared to human references, failing to reliably distinguish between redundant and concise reasoning. Second, the maximum margin between the two groups is approximately 0.1, which is markedly smaller than the margin observed in our ratio-based configuration. This lack of sensitivity suggests that absolute windows fail to account for the inherent variations in response length across different problems. Consequently, we confirm that the ratio-based dynamic window is a more robust and stable metric for quantifying internal redundancy in LRMs.

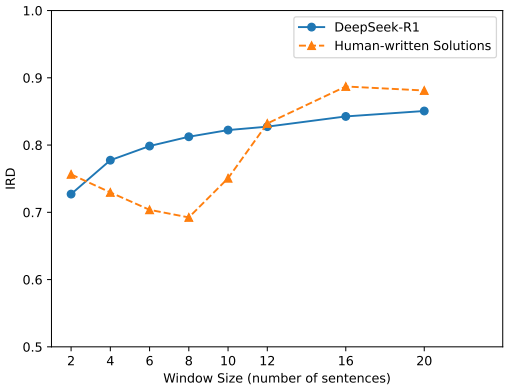


Figure 9: Comparison of IRD using absolute sentence window sizes.

B Training Algorithm

Algorithm 1 Dual-Redundancy Penalty Training

Require: Model $\mathcal{M}$, Dataset $\mathcal{D}$, Reward function $\mathcal{R}$, Optimizer $\mathcal{O}$, Window size w

for each batch of prompts $\{x_i\}_{i=1}^{N}$ from $\mathcal{D}$ **do**
  Sample K responses $\{\hat{y}_i^{(k)}\}_{k=1}^{K}$ for each x_i
  for each response $\hat{y}_i^{(k)}$ **do**
    if final answer is incorrect **then**
      Assign reward $r_i^{(k)} \leftarrow 0$
    else
      Locate the FCS in $\hat{y}_i^{(k)}$
      Split into [FCS, post-FCS]
      Compute IR penalty: p_{int}
      Compute ER penalty: p_{ext}
      Assign reward: $r_i^{(k)} \leftarrow r_i^{(k)} \cdot p_{\text{int}} \cdot p_{\text{ext}}$
    end if
  end for
  Compute GRPO policy loss with $\{r_i^{(k)}\}$
  Update model $\mathcal{M}$ via optimizer $\mathcal{O}$
end for

C Supplement Experiments

C.1 Ablation 1

As discussed in Section 6.1, we observed a non-monotonic trend in response length when only the external redundancy penalty is applied. Figure 10 provides a detailed longitudinal analysis of this phenomenon across both the *DeepSeek-R1-Distill-Qwen-1.5B* and *7B* models.

C.2 Ablation 2

We conduct accuracy drop analysis on *DeepSeek-R1-Distill-Llama-8B*. As illustrated in Figure 11, same conclusion derived from the experiment.

D Examples of compressed solutions

A response sample from our proposed method after applying the dual redundancy penalties is shown in Figure 12. It is evident that content after the answer has been successfully eliminated, resulting in a concise output. Furthermore, compared to baseline models, the internal logic within the FCS is significantly more compact. The overall response exhibits a clear, highly interpretable logical structure, improving the clarity and readability of the reasoning process.

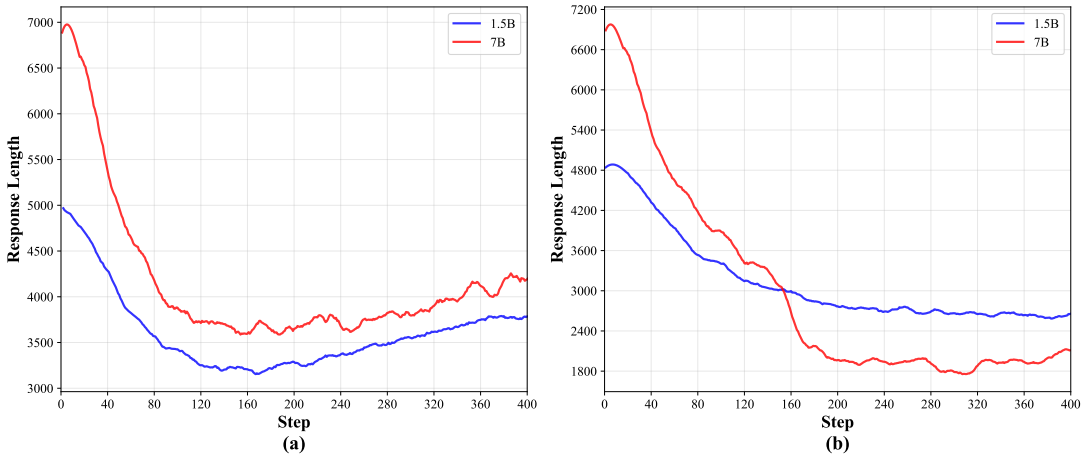


Figure 10: (a) Only external redundancy penalty applied. (b) Both internal and external redundancy penalties applied.

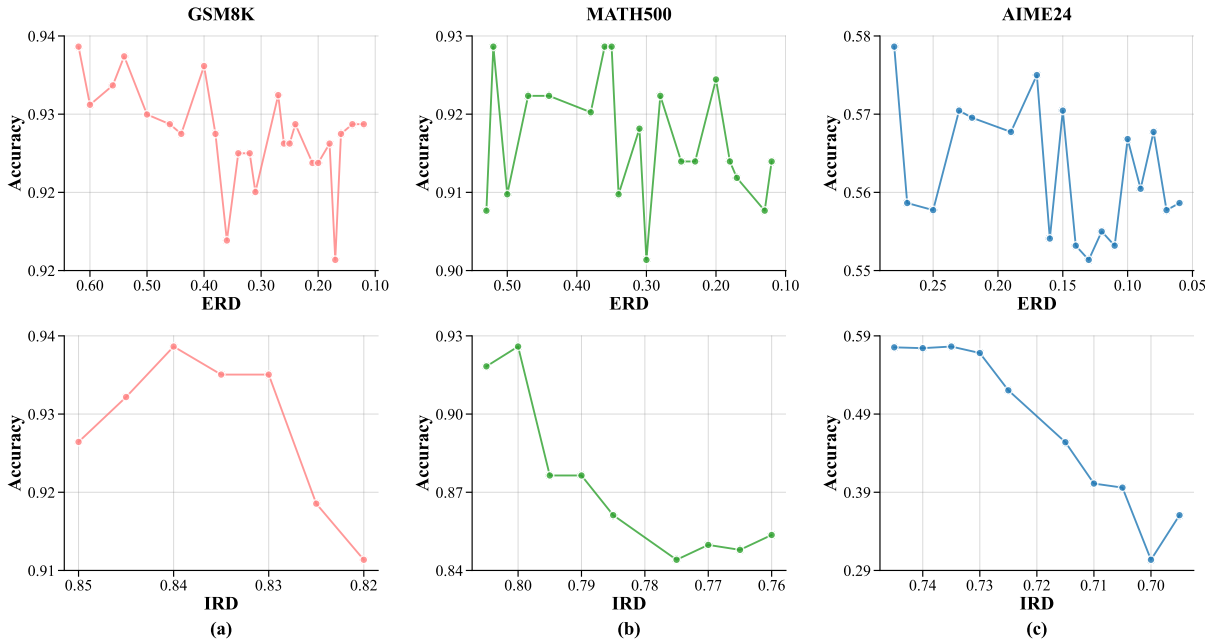


Figure 11: Panels (a)-(c) illustrate the accuracy dynamics during ERD and IRD reduction.

Alright, let's break down the problem step by step.

Problem Understanding: Aya walks 9 kilometers every morning. When she walks at a speed of s km/h, it takes her 4 hours (including t minutes in the coffee shop). When her speed increases to $s+2$ km/h, it takes her 2 hours and 24 minutes (including t minutes in the shop). We need to find the time it takes when her speed is $s+\frac{1}{2}$ km/h, in minutes.

Key Information: 1. Distance: 9 km 2. Times: - At speed s: total time 4 hours (so time walking + t minutes) - At speed $s+2$: total time 2 hours 24 minutes (which is $2+\frac{24}{60}=2.4$ hours) Our goal is to find the time when her speed is $s+\frac{1}{2}$.

Step 1: Express the walking time in terms of speed. Let's denote: - t = time spent in the coffee shop (constant) - w = walking time (in hours) So for the first case: Total time (walking + coffee) = $w+\frac{t}{60}=4$ hours. Walking time $w=\frac{9}{s}$ hours. Similarly, for the second case: Total time = $w+\frac{t}{60}=2.4$ hours. Walking time $w=\frac{9}{s+2}$ hours.

Step 2: Set up equations. From the first case: $\frac{9}{s}+\frac{t}{60}=4$ --- (Equation 1) From the second case: $\frac{9}{s+2}+\frac{t}{60}=2.4$ --- (Equation 2)

Step 3: Subtract Equation (1) from Equation (2) to eliminate $\frac{t}{60}$. Equation (2) - Equation (1): $\left(\frac{9}{s+2}-\frac{9}{s}\right)+\left(\frac{t}{60}-\frac{t}{60}\right)=2.4-4$ Simplify: $9\left(\frac{1}{s+2}-\frac{1}{s}\right)=-1.6$ Calculate the difference: $9\left(\frac{s-(s+2)}{s(s+2)}\right)=-1.6$ Simplify numerator: $s-s-2=-2$ Thus: $9\left(\frac{-2}{s(s+2)}\right)=-1.6$ Multiply: $\frac{-18}{s(s+2)}=-1.6$ Multiply both sides by -1: $\frac{18}{s(s+2)}=1.6$

Step 4: Solve for $s(s+2)$. Multiply both sides by $s(s+2)$: $18=1.6s(s+2)$ Divide both sides by 1.6: $\frac{18}{1.6}=s(s+2)$ Calculate $\frac{18}{1.6}=\frac{180}{16}=\frac{45}{4}=11.25$ Hence: $s^2+2s=11.25$ Bring all terms to one side: $s^2+2s-11.25=0$

Step 5: Solve the quadratic equation for s. Using quadratic formula: $s=\frac{-2\pm\sqrt{(2)^2-4(1)(-11.25)}}{2(1)}$ Calculate discriminant: $4+45=49$ Thus: $s=\frac{-2\pm7}{2}$ We take the positive solution: $s=\frac{5}{2}=2.5$ km/h

Step 6: Find time t using Equation (1). From Equation (1): $\frac{9}{2.5}+\frac{t}{60}=4$ Calculate: $\frac{9}{2.5}=3.6$ hours So: $3.6+\frac{t}{60}=4$ Subtract: $\frac{t}{60}=0.4$ Multiply: $t=0.4\times60=24$ minutes

Step 7: Calculate time when speed is $s+0.5=3$ km/h Walking time $w=\frac{9}{3}=3$ hours. Total time = walking time + coffee time = $3+\frac{24}{60}=3+0.4=3.4$ hours Convert to minutes: $3.4\times60=204$ minutes

Final Answer: \boxed{204}

Figure 12: Response sample showing a concise reasoning trajectory for a distance-speed problem.