CATS: Category-Aware Token-level Steering for Training-Free Redundancy Reduction in Large Reasoning Models

Mengfei Zhang[1*†], **Zhenglin Wang**[2*]

[1]School of Software Technology, Zhejiang University
[2]School of Computer Science and Engineering, Southeast University
zmengfei@zju.edu.cn, zhenglin@seu.edu.cn

Abstract

While Large Reasoning Models (LRMs) exhibit remarkable capabilities in complex tasks, they often suffer from excessive redundancy in their chain-of-thought reasoning. This significantly reduces inference efficiency and increases computational costs. We identify that LRM redundancy is not uniformly homogeneous but can be taxonomized according to whether it is destructive to the final answer: destructive redundancy (e.g., logical drift, hallucination amplification) versus non-destructive redundancy (e.g., repetition, over-elaboration). Moreover, LRM's redundant and concise responses exhibit a significant distinction in their hidden layer representation spaces. Based on these insights, we propose CATS (Category-Aware Token-Level Steering), a training-free and lightweight method to reduce the redundancy phenomenon. CATS decomposes redundancy into six semantically interpretable characteristic dimensions. By flexibly weighting and combining the differential vectors corresponding to these dimensions, CATS synthesizes a composite intervention vector, enabling zero-parameter intervention in the hidden layers. Experiments across three LRM models and five mathematical reasoning datasets demonstrate that CATS reduces reasoning length by an average of 25% while maintaining or even slightly improving task accuracy. CATS offers a pluggable, training-free, and lightweight solution, making it particularly beneficial for users in low-resource environments.

Code — https://github.com/Zmfei/CATS.git

Introduction

Large Reasoning Models (LRMs) such as OpenAI's o1 (Jaech et al. 2024), Deepseek-R1 (Guo et al. 2025), by employing explicit Chain-of-Thought (CoT) mechanisms, mimic the human process of incrementally decomposing problems, iteratively verifying, and refining solutions. This explicit reasoning paradigm has led to unprecedented accuracy in complex tasks such as mathematical reasoning, scientific computation, and common-sense inference. Moreover, it not only boosts task performance but also offers a valuable window into the model's internal logic. However, enhanced capabilities often come with amplified costs: to ensure sufficiently reliable answers, LRMs tend to generate lengthy, redundant, or even unnecessary reasoning steps. We observe that for an identical mathematical problem, a model might produce a concise, correct path of merely 300 tokens, or a redundant yet correct path exceeding 800 tokens, with no difference in the final answer. This "overthinking" phenomenon significantly elevates inference latency and computational costs, posing substantial barriers to deployment, particularly in resource-constrained environments.

Existing work aimed at mitigating LRM redundancy primarily follows two technical routes. One involves training-based compression via supervised fine-tuning (SFT) (Munkhbat et al. 2025; Xia et al. 2025; Kang et al. 2025) or reinforcement learning (RL) (Luo et al. 2025; Hou et al. 2025). This typically involves constructing "verbose-concise" CoT data pairs or introducing length penalty terms, teaching the model to generate more concise reasoning paths during training. While effective, these methods heavily rely on manual annotation or custom reward functions, incurring high training costs. The second route is prompt engineering, which attempts to directly compress outputs during inference by designing instructions like "Be concise" or "skip obvious steps". However, these methods often lack fine-grained control, tending to indiscriminately compress necessary reasoning steps, leading to performance degradation (Renze and Guven 2024; Han et al. 2024; Xu et al. 2025). More recently, some research has explored using activation steering to improve model efficiency (Zhao et al. 2025). Yet, these works often treat redundancy as a single, homogeneous concept, overlooking its internal diversity, which limits the precision and controllability of intervention.

Redundancy is not a monolithic phenomenon but exhibits rich semantic hierarchies. We systematically categorize it into two main types: destructive redundancy and non-destructive redundancy (as shown in Figure 1). Destructive redundancy, including Logical Drift, Hallucination Amplification, and Internal Contradiction, not only wastes tokens but can also substantially compromise the correctness of the final answer. In contrast, non-destructive redundancy, which encompasses Repetition, Over-Elaboration, and Over-Cautiousness, while not affect-

*These authors contributed equally.
†Corresponding author

ing answer correctness, significantly increases unnecessary computational overhead. More crucially, by analyzing the hidden layer representation space, we discover significant divergences between concise paths and various redundant paths in the representation space (as shown in Figure 2). This finding provides our theoretical foundation: redundancy is not merely noise, but rather structured information that can be explicitly modeled and directionally compressed.

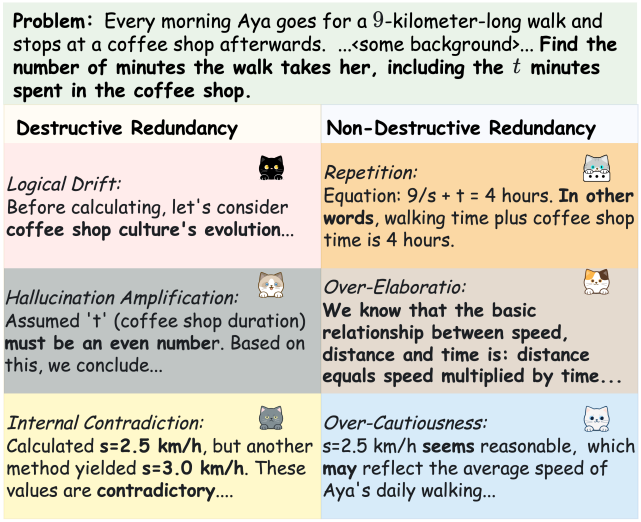

Figure 1: Examples of Redundancy Categorization. This figure illustrates our proposed six semantically interpretable redundancy categories using a mathematical reasoning problem.

Based on these insights, we propose CATS (Category-Aware Token-level Steering), a training-free, plug-and-play, and lightweight method (as shown in Figure 3). CATS first explicitly decomposes redundancy into six semantically interpretable characteristic dimensions. Subsequently, for each category, it computes a corresponding difference vector (the average difference between the hidden states of that redundant path and the most concise path). These vectors are then weighted and combined to generate a composite intervention vector. This vector is injected into a selected hidden layer during the inference stage in a single pass, enabling targeted compression of redundancy. The entire process requires no gradient updates and only involves a single forward pass, significantly reducing deployment barriers.

We conducted systematic experiments on three lightweight LRMs (DeepSeek-R1- Distill-Qwen-1.5B, DeepSeek-R1- Distill-Qwen-7B, and DeepSeek-R1- Distill-Llama-8B) across five mathematical reasoning datasets (MATH-500, AMC, AMC23, AIME2024, AIME2025). Results demonstrate that CATS, while compressing reasoning length by an average of 25%, maintains the original task accuracy and even achieves slight improvements in some scenarios. Furthermore, the incidence of destructive redundancy decreased by 21%, further validating the effectiveness of our category-aware intervention.

In summary, this paper's contributions are:

- Systematically decomposing LLM redundancy into six semantically interpretable characteristic dimensions for the first time, addressing the limitations of previous methods that broadly define redundancy and ensuring coverage and intervention for a wider range of redundancy types.
- Developing a training-free approach that extracts category-specific difference vectors and synthesizes them into a composite intervention vector via configurable weighting, enabling plug-and-play, efficient, and non-intrusive behavioral steering in the model's hidden layers.
- Extensive experimentation across three models and five mathematical reasoning datasets, achieving an average 25% token compression while maintaining or even improving accuracy, and significantly reducing the occurrence of destructive redundancy. This provides novel insights and tools for building efficient and cost-effective reasoning systems.

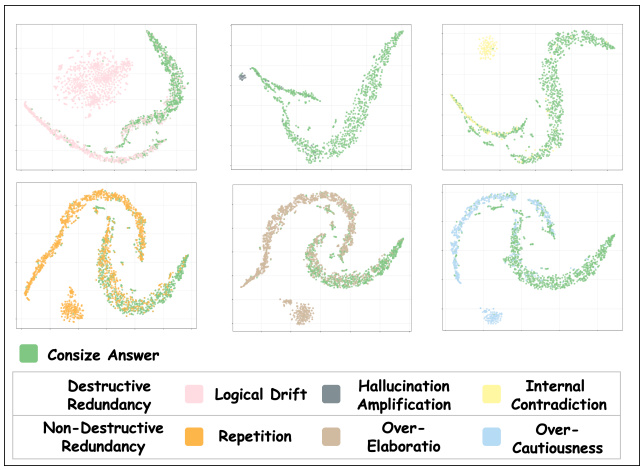

Figure 2: Hidden State Differences between Redundancy Categories and Concise Paths (t-SNE Visualization). This figure presents the t-SNE dimensionality reduction visualization of the hidden states from the 15th layer of the DeepSeek-R1- Distill-Qwen-7B model. It clearly shows a distinct separation of the six redundancy categories from concise paths in the representation space, validating the discriminability of our category-specific difference vectors. Similar patterns were observed in other models and layers.

Related Work

Large Reasoning Models (LRMs) Background

Large Reasoning Models (LRMs), such as OpenAI's o1 (Jaech et al. 2024) and DeepSeek-R1 (Guo et al. 2025), extend Large Language Models' reasoning ability by incorporating explicit Chain-of-Thought (CoT) mechanisms. This

simulates the human process of step-by-step problem decomposition, iterative verification, and idea refinement, significantly boosting model performance in mathematical, scientific, and common-sense reasoning tasks. Their typical inference process involves generating multi-step intermediate reasoning before outputting the final answer.

Despite the significant success of this strategy in improving accuracy, it also introduces substantial reasoning redundancy. To ensure answer reliability, LRMs tend to "overexplain for stability", leading to reasoning paths that can span hundreds to thousands of tokens for a single problem. Consequently, this results in issues such as increased inference latency and higher computational costs.

Existing redundancy mitigation strategies primarily fall into two categories: (1) training-based compression via Supervised Fine-Tuning (SFT) or Reinforcement Learning (RL), and (2) training-free compression through Prompt Engineering.

Supervised Fine-Tuning Strategies: Constructing "Verbose-Concise" Pairs for Compression

The core idea of these methods is to construct paired "verbose-concise" reasoning path data to train the model to generate more concise answers while maintaining reasoning validity. The specific process includes: collecting multiple reasoning paths for the same problem; selecting the shortest correct path as the gold standard; designing appropriate loss functions to guide the model to learn concise expressions during training.

For example, Self-training (Munkhbat et al. 2025) constructs a dataset with multiple solutions for the same problem and selects the shortest correct path for training to enhance the model's compression capability. The TokenSkip (Xia et al. 2025) method identifies and skips tokens that contribute minimally to the final answer, thereby compressing reasoning length while preserving semantic integrity. C3oT (Kang et al. 2025) designs a GPT-4-based compressor that generates shorter chain-of-thought paths by retaining critical reasoning steps.

Reinforcement Learning Strategies: Designing Dual-Objective Rewards for Length-Accuracy Trade-off

Reinforcement learning approaches focus on designing reward functions that guide the model to compress reasoning path length while ensuring output accuracy. This typically involves setting dual-objective rewards: a conciseness reward (penalizing redundant tokens) and an accuracy reward (ensuring the final answer's correctness).

For instance, O1-Pruner (Luo et al. 2025) uses length and accuracy as baselines for its reward function, encouraging the model to generate shorter reasoning paths without sacrificing precision. ThinkPrune (Hou et al. 2025) introduces a length-aware reward, requiring the model to complete correct reasoning within a given token budget, only receiving positive feedback when both objectives are met.

Prompt Engineering Strategies: Zero-Training Compression of Reasoning Paths

These methods do not rely on additional training. Instead, they guide the model to control reasoning length through prompt design, achieving immediate compression.

For example, CCoT (Renze and Guven 2024) explicitly prompts the model to "Be concise". Token-Budget (Han et al. 2024) sets a token usage limit in the prompt, guiding the model to complete reasoning tasks within the budget. Chain of Draft (Xu et al. 2025) requires the model to retain only core information in each reasoning step, even limiting the word count per step to reduce redundant descriptions.

In summary, while existing solutions can compress LRMs' lengthy reasoning to varying degrees, they each suffer from distinct shortcomings: SFT and RL require retraining, incurring high computational and human costs due to annotation or reward design; prompt engineering relies on manual prompt granularity, with effects drifting across tasks and lacking precise control; other hidden intervention methods (Zhao et al. 2025) use only a single directional vector, overlooking the semantic diversity of redundancy. In contrast, our proposed CATS requires no training, achieving 25% reasoning compression on different reasoning models by weighting and fusing six categories of differential vectors. It balances accuracy with interpretability, providing a plug-and-play, fine-grained, and cost-effective redundancy mitigation solution for resource-constrained scenarios.

Methodology

We propose CATS (Category-Aware Token-level Steering), a training-free and lightweight method. CATS constructs differential vectors based on redundancy categories and injects them into the model's hidden layers with configurable weights, thereby significantly compressing reasoning path length while maintaining output accuracy.

Problem Formulation

Given an input query x, a Large Reasoning Model (LRM) generates a multi-step reasoning output $y = \{t_1, t_2, \ldots, t_n\}$. Our objective is to minimize the length of the reasoning chain $|y|$ without compromising the accuracy of the final answer:

$$\min |y| \quad s.t. \quad acc(y) \geq acc(y_0) \tag{1}$$

where $acc(y_0)$ is the accuracy of the original model output. Let $\mathbf{h}_{l,i} \in R^d$ denote the hidden state of the input sequence' i-th token at the l-th layer, where d is the hidden dimension. We define the set of redundancy categories:

$$\mathcal{C} = \{c_1, c_2, \ldots, c_6\} \tag{2}$$

where each category c_k corresponds to a redundancy feature. During inference, we select a specific intervention layer l^* and only intervene on the hidden state of the last token in the input sequence, $\mathbf{h}_{l^*,-1}$:

$$\mathbf{h}'_{l^*,-1} = \mathbf{h}_{l^*,-1} + \sum_{k=1}^{6} w_k \cdot \mathbf{v}_k \tag{3}$$

where:

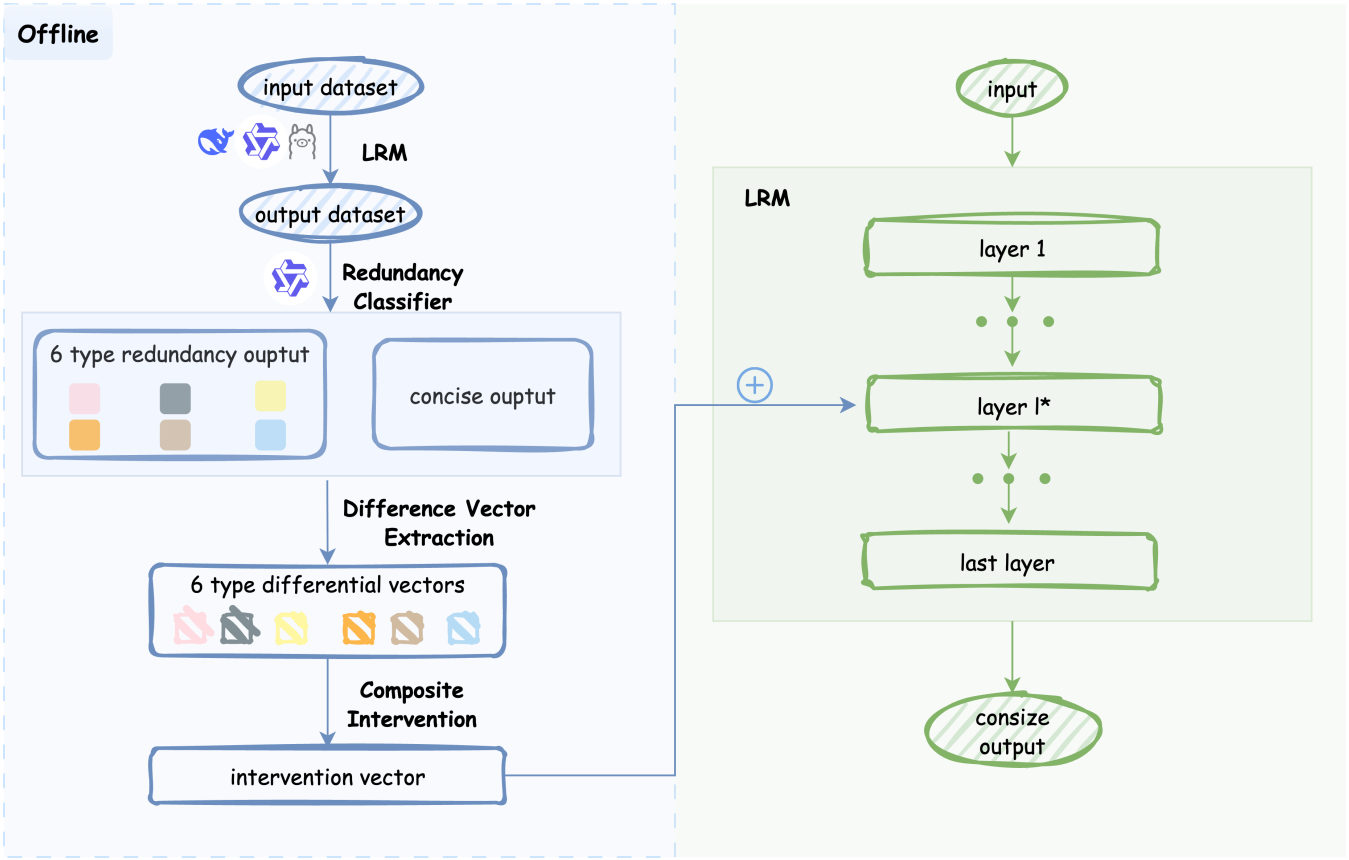

Figure 3: Overview of the CATS (Category-Aware Token-level Steering) Framework

- $\mathbf{v}_k$ is the difference vector for redundancy category c_k; (Its derivation is detailed in next section CATS Framework Overview, Stage 2.)
- w_k is a configurable weight, representing the suppression strength for redundancy of type c_k;
- The intervention occurs only during the forward pass, requiring no parameter updates.

CATS Framework Overview

As illustrated in Figure 3, CATS comprises three main stages:

Stage 1: Redundancy Classification
We categorize redundant content into two main types, encompassing six specific kinds of redundancy (as shown in Figure 1):
Destructive Redundancy
This type of redundancy interferes with the original reasoning path and reduces the correctness of the final answer.

- **Logical Drift**: Deviation from the task objective during reasoning, producing reasoning steps in an unrelated direction.
- **Hallucination Amplification**: Continuously expanding reasoning based on false or unestablished premises, constructing seemingly plausible but actually incorrect paths.
- **Internal Contradiction**: The reasoning path contains logically inconsistent statements, where the subsequent reasoning attempts to reconcile or proceeds despite these inconsistencies.

Non-Destructive Redundancy
This type of redundancy does not affect the correctness of the final answer but significantly increases output length and reduces reasoning efficiency.

- **Repetition**: Repeatedly expressing existing information or rephrasing the same content with different wordings.
- **Over-Elaboration**: Introducing definitions, background, or redundant explanations unrelated to problem-solving, thereby extending the reasoning chain.
- **Over-Cautiousness**: Using vague expressions such as "might", "probably", or "seems like" to describe definitive information.

Classification Process: We utilize the Qwen3-235B-A22B model as a classifier to automatically label redundancy types in model outputs. For each category, 50 samples were randomly selected and manually reviewed (two independent annotators, majority voting for decisions, Kappa coefficient of 0.86) to ensure classification quality. Detailed rules

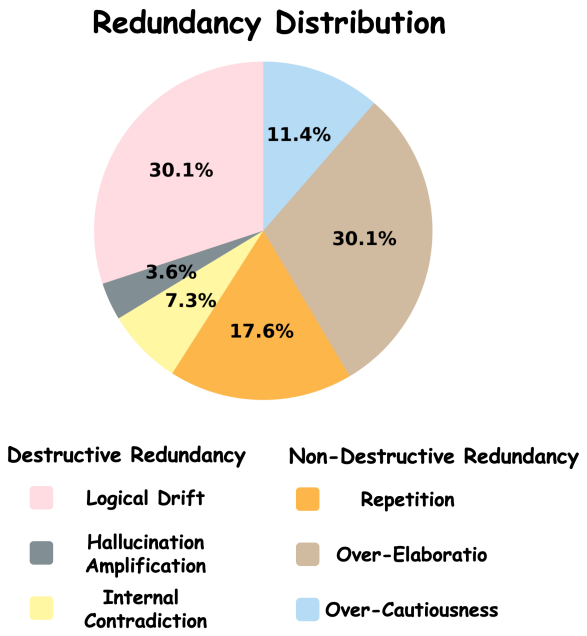

Figure 4: Distribution of Redundancy Categories in DeepSeek-R1- Distill-Qwen-7B Model Outputs

are provided in Appendix A. The distribution of each redundancy category in DeepSeek-R1- Distill-Qwen-7B model outputs is shown in Figure 4.

Stage 2: Difference Vector Extraction
To construct intervention directions for each redundancy category, we selected 1,000 mathematical problems from the GSM8K dataset (Cobbe et al. 2021) and prompted each LRM to generate 10 reasoning paths for each problem. Based on the reasoning results, two sample sets were constructed:

- **Concise Sample Set** S_c: For each problem, the shortest and ultimately correct reasoning path was selected.
- **Redundant Sample Set** $S_{r,k}$: The remaining paths were labeled by the redundancy classifier and assigned to their corresponding redundancy category k, where $k = 1, \ldots, 6$.

We extracted the hidden states of the last token of the input sequence at each hidden layer l for each sample, denoted as $\mathbf{h}^{i}_{c,-1}$ and $\mathbf{h}^{k,j}_{r,-1}$. Here, i indexes the concise samples, and j indexes the samples within the k-th redundancy category. We then calculated the mean hidden states for concise and redundant samples, respectively:

$$\mu^{l}_{S_c} = \frac{1}{|S_c|} \sum_{i} \mathbf{h}^{i}_{c,-1}, \quad \mu^{l}_{S_{r,k}} = \frac{1}{|S_{r,k}|} \sum_{j} \mathbf{h}^{k,j}_{r,-1} \tag{4}$$

Finally, the difference vector corresponding to category c_k is obtained:

$$\mathbf{v}^{l}_{k} = \mu^{l}_{S_c} - \mu^{l}_{S_{r,k}} \tag{5}$$

These difference vectors form interpretable redundancy suppression direction bases in the representation space.

Stage 3: Composite Intervention
During the model inference phase, we select an intervention layer l^* and apply a weighted sum of difference vectors to the hidden state of the last token in the input sequence:

$$\mathbf{h}'_{l^*,-1} = \mathbf{h}_{l^*,-1} + \sum_{k=1}^{6} w_k \cdot \mathbf{v}^{l^*}_{k} \tag{6}$$

Where:

- **Weights** (w_k): These weights w_k are values determined by the user's need to suppress specific redundancy types. This provides CATS with flexible "composability", allowing users to combine and adjust the intervention strength for different redundancy directions based on experience, achieving highly customized intervention effects to meet the specific demands of various application scenarios.
- **Intervention Position**: The specific hidden layer number for intervention needs to be selected. Our preliminary experiments indicate that intervention at intermediate layers of the model yields the best results.

CATS compresses reasoning through only a single forward pass without requiring any training samples or parameter fine-tuning. This offers excellent deployment flexibility and interpretability, making it particularly suitable for resource-constrained scenarios.

Experiments

This section details the experimental configuration used to validate the effectiveness of the CATS method, including selected models, datasets, evaluation metrics, and baselines.

Model and Dataset

Models To evaluate the performance of CATS, we selected the following three representative reasoning-optimized models from the DeepSeek-R1- Distill series as experimental subjects:

- DeepSeek-R1- Distill-Qwen-1.5B
- DeepSeek-R1- Distill-Qwen-7B
- DeepSeek-R1- Distill-Llama-8B

These models are representative in their reasoning capabilities and encompass different underlying architectures, which helps to investigate CATS's effectiveness under various conditions.
Datasets Experiments were primarily validated on the following five classic mathematical reasoning datasets:MATH-500 (Lightman et al. 2023), AMC, AMC23, AIME 2024, AIME 2025 (Balunović et al. 2025). These datasets all require complex chain-of-thought reasoning, and the generated thought processes often contain a rich variety of redundancy types, making them ideal scenarios for observing, classifying, and quantifying redundancy phenomena.

	MATH-500		AMC		AMC23		AIME 2024		AIME 2025	
	len(↓)	acc(↑)	len(↓)	acc(↑)	len(↓)	acc(↑)	len(↓)	acc(↑)	len(↓)	acc(↑)
DeepSeek-R1- Distill-Qwen-1.5B	4528.31	83.21	8982.86	58.52	8697.42	55.42	11758.16	27.54	11709.96	25.43
+ Generic Intervention	3935.10	79.55	6902.43	57.12	7338.45	54.25	10637.61	26.54	10503.83	24.80
+ CATS	**3577.36**	82.01	**6108.34**	**60.13**	**6523.07**	**56.22**	**9759.27**	**27.94**	**9133.77**	**25.83**
DeepSeek-R1- Distill-Qwen-7B	3817.48	92.96	6572.61	79.07	6204.70	81.03	10220.63	51.64	10051.13	36.85
+ Generic Intervention	3169.55	91.34	4814.84	76.20	5425.86	78.15	9390.91	48.66	8661.53	34.55
+ CATS	**2855.45**	**93.20**	**4115.25**	**79.37**	**4559.55**	**82.26**	**7825.76**	**52.32**	**7278.60**	**37.15**
DeepSeek-R1- Distill-Llama-8B	3863.48	89.73	6878.65	77.80	6639.62	79.21	11404.50	44.80	11579.90	30.84
+ Generic Intervention	3256.30	87.81	5175.50	76.65	5969.68	78.81	10318.79	43.83	9944.82	32.17
+ CATS	**2936.24**	**90.06**	**4539.91**	**78.21**	**5378.09**	**80.01**	**8895.51**	**46.14**	**8800.72**	**34.22**
avg. Δlen = -25%, avg. Δacc = 1.2%										

Table 1: Overview of Main Results of CATS across Different Models and Datasets

Evaluation Metrics

We comprehensively evaluated model performance from two dimensions: core task performance and redundancy.

Core Task Performance Metric

- **Accuracy**: Measures the correctness of the model's final answer, used to assess whether CATS intervention degrades the model's original reasoning capability.

For generation, we set the maximum generation length (including both reasoning trace and final answer) for all models to 16384 tokens. For each test question, we sample 4 to 16 outputs with a temperature of 0.7.

Redundancy Metrics

- **Average Token Length:** The most direct indicator of reasoning chain conciseness.
- **Redundancy Occurrence Rate per Category:** Calculates the occurrence rate of each redundancy category in model responses before and after intervention, evaluating the model's effectiveness in mitigating each redundancy category. We use the Qwen3-235B-A22B model as a classifier to automatically annotate redundancy types in the model's output.

Baselines

We conducted comparative experiments with the following baseline methods:

- **Original LLM:** The original model without any intervention.
- **Generic Intervention Baseline:** To highlight the value of our fine-grained redundancy classification, we constructed a comparative baseline. This baseline does not classify redundant samples; instead, it directly selects the longest answer for each problem as a generic redundant sample, computes a general "longest-concise" differential vector, and applies it for intervention. This baseline aims to simulate unidirectional intervention methods that treat all redundancy as homogeneous and undifferentiated, serving to prove the effectiveness of our multi-category fine classification.

Results

This section presents the experimental results of the CATS method on different models and datasets and provides an in-depth analysis of the value of redundancy classification and its contribution to model performance.

Main Results: CATS Performance in Redundancy Compression and Performance Maintenance

As shown in Table 1 , CATS demonstrates significant reasoning length compression capabilities across all tested models and datasets, achieving an average token length compression rate of 25%. Encouragingly, while substantially compressing reasoning length, the model's accuracy not only remains unimpaired but even slightly improves in some scenarios. This strongly validates the effectiveness of the CATS method in reducing redundancy and improving efficiency, proving its potential to optimize model output without sacrificing core performance.

From Table 2, CATS intervention effectively reduced the occurrence rate of all redundancy categories. Specifically, the average occurrence rate of destructive redundancy decreased by 21%, while that of non-destructive redundancy decreased by 27%, further enhancing the conciseness and reliability of model outputs.

Ablation Study: Value and Contribution of Redundancy Classification

To deeply investigate the effectiveness and necessity of fine-grained redundancy classification, we conducted the following two sets of ablation experiments:

Generic Intervention vs. CATS

As shown in Table 1, the comparative experimental results show that CATS outperforms the generic intervention baseline in both length compression rate and accuracy. This strongly demonstrates the critical role of fine-grained redundancy classification in improving intervention quality. While generic intervention can compress length to some extent, by not differentiating redundancy types, it may inadvertently harm necessary reasoning steps while eliminating non-destructive redundancy, or fail to effectively suppress destructive redundancy, leading to decreased accuracy or unresolved destructive redundancy issues. CATS, in contrast, can target different redundancy types more

	Destructive Redundancy			Non-Destructive Redundancy		
	Log. Drift	Hall. Amp.	Int. Contr.	Rep.	Over-Elab.	Over-Caut.
DeepSeek-R1- Distill-Qwen-1.5B	39.09	3.20	4.90	15.81	13.65	13.70
+ CATS	**31.27**	**2.62**	**3.43**	**11.38**	**10.51**	**10.41**
DeepSeek-R1- Distill-Qwen-7B	31.50	3.60	4.40	15.94	17.82	16.57
+ CATS	**26.15**	**2.81**	**3.34**	**11.64**	**12.47**	**12.10**
DeepSeek-R1- Distill-Llama-8B	32.20	2.70	5.40	16.72	14.57	17.37
+ CATS	**25.44**	**1.89**	**3.24**	**11.70**	**11.22**	**12.16**
avg. ΔOccurrence Rate of Destructive Redundancy = -21%						
avg. ΔOccurrence Rate of Non-Destructive Redundancy = -27%						

Table 2: Occurrence Rates of Redundancy Categories in Model Responses Before and After CATS Intervention

precisely, achieving a more intelligent and safe efficiency optimization.

Per-Category Intervention Ablation

Within the CATS framework, we further individually removed the intervention vector for specific redundancy categories (i.e., setting their w_k to 0), observing the impact on the average change in accuracy and length, to reveal the unique contribution of each category's intervention, as shown in Figure 5.

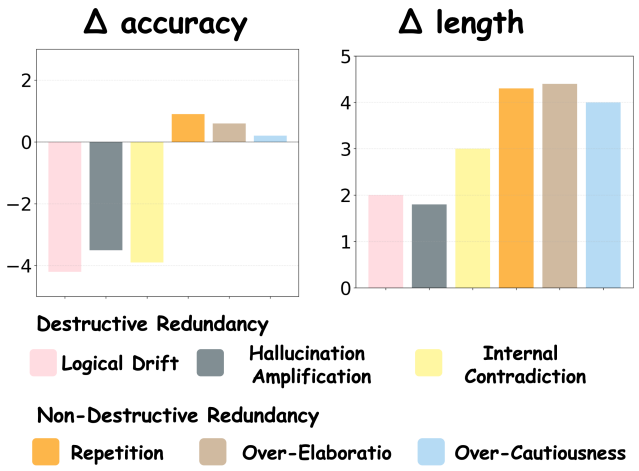

Figure 5: Impact of Removing Specific Redundancy Category Interventions on Core Metrics (Accuracy and Length)

Example Analysis:

- **Removing "Logical Drift" category intervention**: As seen in Figure 5, when intervention for the "Logical Drift" category is removed, the model's average accuracy decreases by 4.2%, and the average length only increases by 2.0%. This strongly proves the critical role of the "Logical Drift" category and its corresponding intervention vector in ensuring model output quality and suppressing destructive redundancy. It alerts us that not all redundancy is harmless, and identifying and intervening in specific destructive redundancies is crucial.
- **Removing "Repetition" category intervention**: Conversely, when we remove intervention for the "Repetition" category, the average length increases by 4.3%, but accuracy remains almost unchanged (change less than 1.0%). This indicates that intervention for non-destructive redundancies like "Repetition" primarily focuses on text conciseness and contributes significantly to length compression without affecting core performance.

These ablation results collectively demonstrate the rationality and effectiveness of our six-category redundancy classification system, as well as each category's unique contribution to achieving the overall optimization goals.

Conclusion

In this paper, we introduced CATS (Category-Aware Token-level Steering), a novel training-free and lightweight framework designed to mitigate excessive reasoning redundancy prevalent in Large Reasoning Models (LRMs). CATS systematically decomposes LRM redundancy into six semantically interpretable characteristic dimensions and, through flexibly weighted composite intervention vectors, achieves an average 25% reasoning length compression. This is accomplished while maintaining or even improving task accuracy and effectively suppressing the occurrence of destructive redundancy across multiple LRM models and mathematical reasoning datasets. A core contribution of CATS lies in its ability to achieve fine-grained, differentiated control over various types of redundancy—particularly by distinguishing and handling both destructive and non-destructive forms. This, coupled with its excellent interpretability and broad transferability, makes CATS a plug-and-play, lightweight solution especially friendly to users in low-resource environments. While CATS currently relies on static weights and has primarily been validated on mid-sized models in mathematical reasoning, future work will explore adaptive weight optimization and extend its application to larger-scale models and broader multimodal reasoning scenarios, laying the foundation for more efficient and robust LRM deployments.

References

Balunović, M.; Dekoninck, J.; Petrov, I.; Jovanović, N.; and Vechev, M. 2025. Matharena: Evaluating llms on uncontaminated math competitions. *arXiv preprint arXiv:2505.23281*.

Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Guo, D.; Yang, D.; Zhang, H.; Song, J.; Zhang, R.; Xu, R.; Zhu, Q.; Ma, S.; Wang, P.; Bi, X.; et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Han, T.; Wang, Z.; Fang, C.; Zhao, S.; Ma, S.; and Chen, Z. 2024. Token-budget-aware llm reasoning. *arXiv preprint arXiv:2412.18547*.

Hou, B.; Zhang, Y.; Ji, J.; Liu, Y.; Qian, K.; Andreas, J.; and Chang, S. 2025. Thinkprune: Pruning long chain-of-thought of llms via reinforcement learning. *arXiv preprint arXiv:2504.01296*.

Jaech, A.; Kalai, A.; Lerer, A.; Richardson, A.; El-Kishky, A.; Low, A.; Helyar, A.; Madry, A.; Beutel, A.; Carney, A.; et al. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.

Kang, Y.; Sun, X.; Chen, L.; and Zou, W. 2025. C3ot: Generating shorter chain-of-thought without compromising effectiveness. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 24312–24320.

Lightman, H.; Kosaraju, V.; Burda, Y.; Edwards, H.; Baker, B.; Lee, T.; Leike, J.; Schulman, J.; Sutskever, I.; and Cobbe, K. 2023. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*.

Luo, H.; Shen, L.; He, H.; Wang, Y.; Liu, S.; Li, W.; Tan, N.; Cao, X.; and Tao, D. 2025. O1-pruner: Length-harmonizing fine-tuning for o1-like reasoning pruning. *arXiv preprint arXiv:2501.12570*.

Munkhbat, T.; Ho, N.; Kim, S. H.; Yang, Y.; Kim, Y.; and Yun, S.-Y. 2025. Self-training elicits concise reasoning in large language models. *arXiv preprint arXiv:2502.20122*.

Renze, M.; and Guven, E. 2024. The benefits of a concise chain of thought on problem-solving in large language models. In *2024 2nd International Conference on Foundation and Large Language Models (FLLM)*, 476–483. IEEE.

Xia, H.; Leong, C. T.; Wang, W.; Li, Y.; and Li, W. 2025. Tokenskip: Controllable chain-of-thought compression in llms. *arXiv preprint arXiv:2502.12067*.

Xu, S.; Xie, W.; Zhao, L.; and He, P. 2025. Chain of draft: Thinking faster by writing less. *arXiv preprint arXiv:2502.18600*.

Zhao, W.; Guo, J.; Deng, Y.; Sui, X.; Hu, Y.; Zhao, Y.; Che, W.; Qin, B.; Chua, T.-S.; and Liu, T. 2025. Exploring and Exploiting the Inherent Efficiency within Large Reasoning Models for Self-Guided Efficiency Enhancement. *arXiv preprint arXiv:2506.15647*.