Mitigating Overthinking in Large Reasoning Models via Manifold Steering

Yao Huang[1,2], **Huanran Chen**[2], **Shouwei Ruan**[1],
Yichi Zhang[2], **Xingxing Wei**[1,4], **Yinpeng Dong**[2,3*]
[1]Institute of Artificial Intelligence, Beihang University, Beijing 100191, China
[2]College of AI, Tsinghua University, Beijing 100084, China [3]Shanghai Qi Zhi Institute
[4]State Key Laboratory of Virtual Reality Technology and Systems, Beihang University
✉: {y_huang, xxwei}@buaa.edu.cn, dongyinpeng@mail.tsinghua.edu.cn

Abstract

Recent advances in Large Reasoning Models (LRMs) have demonstrated remarkable capabilities in solving complex tasks such as mathematics and coding. However, these models frequently exhibit a phenomenon known as *overthinking* during inference, characterized by excessive validation loops and redundant deliberation, leading to substantial computational overheads. In this paper, we aim to mitigate overthinking by investigating the underlying mechanisms from the perspective of mechanistic interpretability. We first showcase that the tendency of overthinking can be effectively captured by a single direction in the model's activation space and the issue can be eased by intervening the activations along this direction. However, this efficacy soon reaches a plateau and even deteriorates as the intervention strength increases. We therefore systematically explore the activation space and find that the overthinking phenomenon is actually tied to a low-dimensional manifold, which indicates that the limited effect stems from the noises introduced by the high-dimensional steering direction. Based on this insight, we propose **Manifold Steering**, a novel approach that elegantly projects the steering direction onto the low-dimensional activation manifold given the theoretical approximation of the interference noise. Extensive experiments on DeepSeek-R1 distilled models validate that our method reduces output tokens by up to 71% while maintaining and even improving the accuracy on several mathematical benchmarks. Our method also exhibits robust cross-domain transferability, delivering consistent token reduction performance in code generation and knowledge-based QA tasks. Code is available at: https://github.com/Aries-iai/Manifold_Steering.

1 Introduction

Building on the versatility of Large Language Models (LLMs) in text generation, particularly their emergent ability in chain-of-thought (CoT) reasoning [41], the field is now undergoing a transition toward Large Reasoning Models (LRMs). Exemplified by the OpenAI o-series [28] and the DeepSeek-R1 series [14], LRMs acquire internal capabilities for long-horizon reasoning through reinforcement learning with verifiable rewards. These models are able to explore diverse solution paths, reflect on potential errors, refine intermediate steps, and validate final outputs, mimicking the process of human problem-solving by scaling inference-time computations [28]. As a result, they excel in domains such as mathematics [1, 23, 39] and coding [27, 43, 44]. This makes them well-suited for tasks that demand deep logical analysis and paves the way for their applications in more complex scenarios, including web search [19] and research assistance [49].

*Corresponding Author

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

However, despite the remarkable reasoning capabilities, LRMs often suffer from a critical efficiency issue known as *overthinking* [35], where they generate excessive and unnecessary reasoning steps, even for simple questions. For example, when tasked with a straightforward calculation, like "$2 + 3$" [9], an LRM might redundantly validate its approach or explore irrelevant alternatives, significantly increasing the computation overloads. This overthinking not only impacts inference latencies, posing great challenges for time-critical applications, but also risks degrading performance by entangling the model in repetitive verification loops or unproductive reasoning paths [9, 16, 40]. To mitigate such overthinking in LRMs, several approaches [3, 8, 16, 22] have recently been proposed. They often utilize external mechanisms to regulate reasoning and prevent overthinking, which can incur additional computational overhead for probing [16] or be susceptible to performance degradation due to the reliance on external models [22]. While these methods address overthinking from external and behavioral perspectives – relying on human-designed workflows and interventions, the underlying mechanism remains underexplored, posing significant challenges to achieving intrinsic mitigation.

In this paper, we address the overthinking problem of LRMs through mechanistic interpretability [50], based on an in-depth analysis of their internal states. Specifically, we attribute this phenomenon to the distinctive activation patterns in the deeper layers of the model and identify a single, interpretable direction by comparing the differences in the activations between overthinking and concise reasoning. By manipulating the activations along this direction, we can effectively steer the model away from overthinking tendencies. However, this intervention is insufficient to fully resolve the problem. As in Fig. 2(a), the reduction in output tokens does not consistently scale with increasing intervention strength. This suggests that the computed steering direction is not accurate enough and introduces unintended *interference noise*.

To address this issue, we further analyze the model's activation patterns and find that the overthinking phenomenon is intrinsically tied to a low-dimensional manifold, which can be well approximated by a linear subspace. This result sheds light on why high-dimensional steering directions often introduce noises, as they fail to align with the underlying structure of model activations. To more effectively mitigate overthinking, we introduce a **Manifold Steering** method to align the steering direction with the reconstructed low-dimensional manifold. We first theoretically derive a linear approximation of the amplitudes of the interference noise and then project the steering direction by nullifying this approximated term. In this way, we can effectively purify the steering direction and better mitigate the overthinking issue with larger intervention strength, as depicted in Fig. 2(a).

Extensive experiments on multiple DeepSeek-R1 distilled models [14] of different sizes verify the effectiveness of our manifold steering method. We first test it on mathematical datasets of varying difficulty, including GSM8K [10], Math500 [20], AMC2023 [24], and AIME2024 [25]. Our method achieves up to 71% tokens reduction while consistently maintaining or improving accuracy. Moreover, it exhibits robust cross-domain transferability, delivering consistent mitigation effects in tasks such as LiveCodeBench [18] (code generation) and Diamond-GPQA [32] (knowledge-based QA), surpassing existing methods in both overthinking mitigation and accuracy preservation.

2 Related Work

Mechanistic Interpretability. Mechanistic interpretability [5, 11, 26, 29, 31, 34, 37, 50] seeks to reverse-engineer the internal computations of LLMs to uncover the causal mechanisms underlying their behavior, offering fine-grained insights into learned representations and decision processes. A key technique within this framework involves identifying *steering directions* [29, 34, 37]—linear vectors in the activation space that correspond to specific model behaviors. By manipulating these directions during inference, researchers can precisely control outputs, such as ablating refusal behaviors in safety-critical scenarios [2, 45]. Similarly, Cao et al. [7] proposed Bi-directional Preference Optimization (BiPO), leveraging steering vectors derived from contrasting human preference pairs to customize attributes like truthfulness and hallucination. These approaches highlight the versatility of steering directions in manipulating models' behaviors. Additionally, some efforts have explored dimensionality reduction in activation spaces: [6] use SVD-based spectral filtering to suppress noise in residual streams, while others derive steering directions from probabilistic classification of user history for multi-preference alignment [33]. Our work extends this paradigm to address *overthinking* in LLMs [9, 16, 35], a phenomenon characterized by redundant or divergent reasoning trajectories. By analyzing the latent space, we identify a steering direction that encapsulates overthinking and

further propose *manifold steering*, a novel method that projects this direction onto a low-dimensional manifold to mitigate interference noise, thereby improving its performance.

Overthinking Mitigation. Efforts [3, 8, 16, 22, 30, 42] to mitigate overthinking in LRMs have gained traction as a means to enhance inference efficiency and output quality. Among them, the training-based method [30] tend to modify the reward function for length control in reinforcement learning. However, these incur significant computational costs and are orthogonal to inference-time interventions like ours, thus warranting no direct comparison in this work. Existing training-free methods mainly rely on external mechanisms to regulate reasoning. For instance, Dynasor [16] employs periodic monitoring to detect and halt redundant reasoning, incurring computational overhead, while Thought Manipulation [22] uses auxiliary models to guide inference, limited by the external model's performance. These shortcomings suggest that a more fundamental solution lies in understanding and modifying the model's internal reasoning processes. Though some concurrent works [3, 8] have tried to leverage mechanistic interpretability for achieving it, they only partially reduce overthinking, quickly encountering bottlenecks due to interference noise in high-dimensional steering directions. In contrast, we propose manifold steering to project the steering direction onto a low-dimensional manifold, effectively eliminates interference noise, achieving superior overthinking mitigation and substantial token reductions across diverse tasks, as demonstrated in Sec. 5.2.

3 Mechanistic Analysis of Overthinking

In this section, we investigate the phenomenon of overthinking within the activation space of Large Reasoning Models (LRMs) and identify a general mechanism by which ablating a single direction in the activation space can reduce redundant reasoning steps to some extent.

3.1 Background

Transformers. Decoder-only transformer language models [21, 38] map an input token sequence $\mathbf{x} = [x_1, \ldots, x_T]$ to a probability distribution over the vocabulary for next-token prediction. Each token x_i is associated with a sequence of residual stream activations $\mathbf{h}^{(l)}(x_i) \in \mathbb{R}^d$ across L layers, initialized by the token embedding $\mathbf{h}^{(0)}(x_i) = \text{Embed}(x_i)$. At each layer $l \in \{1, \ldots, L\}$, the residual stream $\mathbf{h}^{(l)}(x_i)$ is updated by combining the previous layer's activation $\mathbf{h}^{(l-1)}(x_i)$ with two components: (i) a multi-head self-attention mechanism, which computes $\mathbf{a}^{(l)}(x_{1:i})$ by attending to prior tokens $\{x_j : j \leq i\}$ using a causal mask to enforce autoregressive context flow; and (ii) a multi-layer perceptron (MLP), which applies non-linear transformations to the post-attention state $\mathbf{h}^{(l-1)}(x_i) + \mathbf{a}^{(l)}(x_{1:i})$ and produces $\mathbf{m}^{(l)}(x_i)$. The whole update is expressed as follows:

$$\mathbf{h}^{(l)}(x_i) = \mathbf{h}^{(l-1)}(x_i) + \mathbf{a}^{(l)}(x_{1:i}) + \mathbf{m}^{(l)}(x_i), \quad \mathbf{m}^{(l)}(x_i) = \text{MLP}(\mathbf{h}^{(l-1)}(x_i) + \mathbf{a}^{(l)}(x_{1:i})). \quad (1)$$

Through autoregressive aggregation, each $\mathbf{h}^{(l)}(x_i)$ aggregates context from prior tokens, with the final token's residual stream $\mathbf{h}^{(l)}(x) :\rightarrow \mathbf{h}^{(l)}(x_T)$ encapsulating the entire input's context.

Large Reasoning Models. LRMs are tailored for complex problem-solving and instruction-following, which leverage structured templates to handle user inputs:

```
<|begin_of_sentence|><|User|>{instruction}<|Assistant|><think>\n
```

where the content following `<think>\n` comprises the model's reasoning process and final answer, separated by `</think>`. Despite the excellent reasoning capabilities of these models, they often exhibit the overthinking phenomenon [9, 12] during the reasoning process, characterized by repetitive validation or redundant deliberation. As a high-level cognitive phenomenon, overthinking may manifest in the model's residual stream activations, similar to other abstract concepts such as safety [5] and honesty [50], as widely studied from the perspective of mechanistic interpretability. This suggests that overthinking and concise reasoning exhibit distinct activation patterns. In the next section, we systematically examine this hypothesis and investigate whether these activation differences are sufficient to identify a specific direction that characterizes overthinking – one that, if isolated, could be ablated to improve reasoning efficiency.

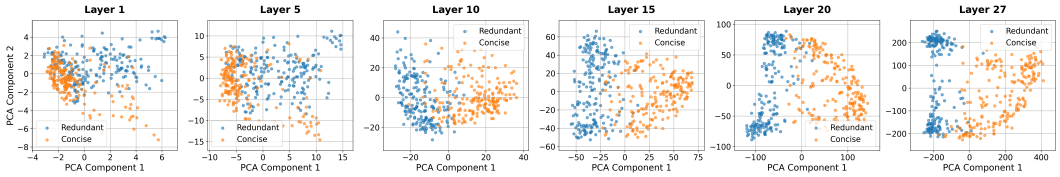


Figure 1: Visualization of residual stream activations $\mathbf{h}^{(l)}(x)$ for $D_{\text{redundant}}$ and D_{concise} across different layers of DeepSeek-R1-Distill-Qwen-7B (R1-7B). Early layers show considerable overlap between redundant and concise data, while middle-to-late layers exhibit distinct separation.

3.2 Extracting and Ablating an Overthinking Direction

Before extracting an overthinking direction, we first investigate whether the residual stream activations corresponding to redundant and concise reasoning are separable in the model's activation space, as this is a necessary condition for identifying a meaningful and controllable direction. Drawing on [16], we also focus on mathematical problems, where the overthinking phenomenon is particularly pronounced. To construct representative data, we first randomly sample questions from the OpenMathInstruct-2 training set [36]. For each model, five responses per question are independently generated. Based on these responses, we construct two model-specific datasets[2] as follows:

- **Redundant set** $D_{\text{redundant}}$: consists of questions for which all five responses exceed 16k tokens and contain hesitation keywords (e.g., "*wait*", "*alternatively*", etc.) surpassing a specified number.
- **Concise set** D_{concise}: consists of questions for which all five responses are under 1k tokens and contain none of the hesitation keywords.

As demonstrated in Fig. 1, we visualize the distribution of residual stream activations $\mathbf{h}^{(l)}(x)$ for both $D_{\text{redundant}}$ and D_{concise} across different layers of R1-7B. We observe that, while early layers exhibit substantial overlap between the two distributions, the middle-to-late layers display clear separation. This separation indicates that the overthinking phenomenon is more prominent in specific layers and provides empirical support for identifying a meaningful overthinking direction.

We use the *difference-in-means* technique [4] for extracting the steering direction, which computes the mean difference in residual stream activations between the redundant and concise data for each layer l. The overthinking direction $\mathbf{r}^{(l)}$ is then defined as:

$$\mathbf{r}^{(l)} = \frac{1}{|D_{\text{redundant}}|} \sum_{x \in D_{\text{redundant}}} \mathbf{h}^{(l)}(x) - \frac{1}{|D_{\text{concise}}|} \sum_{x \in D_{\text{concise}}} \mathbf{h}^{(l)}(x), \tag{2}$$

where $\mathbf{h}^{(l)}(x)$ denotes the residual stream activation of the final token of input x at layer l, with x being the prompt concatenated with the response for $x \in \mathcal{D}_{\text{redundant}}$ but only the prompt for $x \in \mathcal{D}_{\text{concise}}$. The direction $\mathbf{r}^{(l)}$ is normalized to unit length, i.e., $\mathbf{r}^{(l)} = \mathbf{r}^{(l)}/\|\mathbf{r}^{(l)}\|_2$. Following [2], we also select the single most effective direction $\mathbf{r}^{(l^*)}$ and apply it for intervention across all layers.

Finally, to further explore the role of the overthinking direction $\mathbf{r}^{(l^*)}$ in the model's computations, we ablate the component aligned with $\mathbf{r}^{(l^*)}$ each residual stream activation $\mathbf{h}$. Specifically, the modified activation $\mathbf{h}'$ is computed as:

$$\mathbf{h}' = \mathbf{h} - \alpha \times \mathbf{r}^{(l^*)}(\mathbf{r}^{(l^*)})^\top \mathbf{h}, \tag{3}$$

where α controls the intervention strength. We apply this ablation to every activation $\mathbf{h}^{(l)}(x_i)$, across all layers l and token positions i. The parameter α allows adapting the extent of overthinking mitigation, balancing the reduction of redundant reasoning with the problem-solving accuracy.

4 Manifold Steering for Robust Intervention

Following our mechanistic analysis in Sec. 3, which identifies a single direction capturing overthinking in the model's activation space, we proceed to explore whether increasing the intervention strength α further reduces redundant reasoning, as expected.

[2]Details on data selection and dataset composition are provided in the Appendix A.

4.1 Low-Dimensional Manifold Analysis

To rigorously evaluate the effect of increasing intervention strength α for the direction $\mathbf{r}^{(l^*)}$, derived via Eq. (2), we test R1-7B's performance on the Math500 dataset [20], a diverse mathematical test set, different from the anchor dataset used in Sec. 3.2.

Formulation of Interference Noise. As illustrated in Fig. 2(a), increasing the intervention strength α initially reduces the token count. However, beyond a certain threshold, the token count ceases to decrease, and as α continues to increase beyond 1.5, it rebounds, even nearly returning to levels observed without intervention. This suggests that the intervention direction $\mathbf{r}^{(l^*)}$ may be imprecise and introduces unintended noise in the model's activation space, which is defined as *interference noise*. Meanwhile, we confirm the model collapse caused by interference noise, as below:

```
Instruction:  What power of 4 is equal to 8?  Express your answer.
Response:  10000000000000000000 ···
```

Thus, we can hypothesize that the $\mathbf{r}^{(l^*)}$, computed via the difference-in-means method in $\mathbb{R}^d$, comprises both the overthinking direction $\mathbf{r}_{overthinking}$ and an orthogonal[3] interference component $\mathbf{r}_{other}$, such that $\mathbf{r}^{(l^*)} = \mathbf{r}_{overthinking} + \mathbf{r}_{other}$. The Eq. (3) actually modifies the activation as follows:

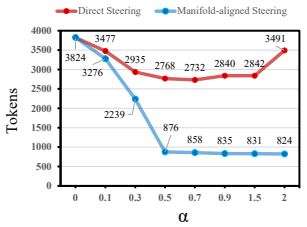
$$\mathbf{h}'^{(l)}(x_i) = \mathbf{h}^{(l)}(x_i) - \alpha \left[\underbrace{(\mathbf{r}_{overthinking})^\top \mathbf{h}^{(l)}(x_i)\mathbf{r}_{overthinking}}_{\text{overthinking component}} + \underbrace{(\mathbf{r}_{other})^\top \mathbf{h}^{(l)}(x_i)\mathbf{r}_{other}}_{\text{interference component}} \right]. \quad (4)$$

The $\mathbf{r}_{other}$ term perturbs $\mathbf{h}'^{(l)}(x_i)$, potentially disrupting unrelated capabilities such as normal expression, especially for large α. This interference explains the token count rebound beyond $\alpha = 1.5$, as the intervention affects dimensions irrelevant to overthinking.

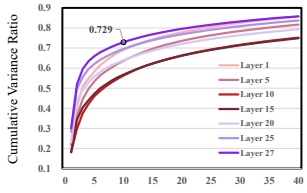
Linear Low-Dimensional Manifold Verification. As shown above, the direct computation of difference in high-dimensional activation space leads to noisy estimation due to the existence of the interference part $\mathbf{r}_{other}$. A straightforward solution is to estimate the amplitude of $\mathbf{r}_{other}$ and remove its influence from the steering direction $\mathbf{r}^{(l^*)}$. However, it is orthogonal to the overthinking direction $\mathbf{r}_{overthinking}$ and is decided by the space of $\mathbf{r}_{overthinking}$. Inspired by prior work [15] that the activations in LLMs reside on a low-dimensional manifold $\mathcal{M} \subset \mathbb{R}^d$, it is reasonable to assume that $\mathbf{r}_{overthinking}$, representing the shift between activations of redundant and concise reasoning paths, also falls into this manifold. To verify this, we employ a simple linear method – Principal Component Analysis (PCA), on the activations from the complete reasoning dataset $D_{\text{reasoning}} = D_{\text{redundant}} \bigcup D_{\text{concise}}$ at layer l. Let $\mathbf{A}^{(l)} = [\mathbf{h}^{(l)}(x_1), \ldots, \mathbf{h}^{(l)}(x_N)] \in \mathbb{R}^{d \times N}$ denote the matrix of activation vectors $\mathbf{h}^{(l)}(x_i) \in \mathbb{R}^d$ for inputs $x_i \in D_{\text{reasoning}}$. We compute the covariance matrix and its eigendecomposition as:

$$\mathbf{C}^{(l)} = \frac{1}{N-1}(\mathbf{A}^{(l)} - \bar{\mathbf{A}}^{(l)})(\mathbf{A}^{(l)} - \bar{\mathbf{A}}^{(l)})^\top = \mathbf{U}^{(l)}\mathbf{\Lambda}^{(l)}(\mathbf{U}^{(l)})^\top, \quad (5)$$

where $\bar{\mathbf{A}}^{(l)} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{h}^{(l)}(x_i)$, $\mathbf{U}^{(l)} \in \mathbb{R}^{d \times d}$ contains the principal components, $\mathbf{\Lambda}^{(l)} = \text{diag}(\lambda_1^{(l)}, \ldots, \lambda_d^{(l)})$, and $\text{VR}(k) = \frac{\sum_{i=1}^{k} \lambda_i^{(l)}}{\sum_{i=1}^{d} \lambda_i^{(l)}}$ is the variance ratio. As Fig. 2(b) shows, the top $k = 10$ components account for over 70% of the variance, indicating that the effective dimension of $\mathcal{M}$, denoted d_{eff}, is significantly smaller than the ambient dimension d. This confirms the low-dimensional structure of $\mathcal{M}$. This also suggests that the linear manifold composed by the orthogonal basis effectively captures the activations of reasoning trajectories and therefore overthinking direction $\mathbf{r}_{overthinking}$



(a) Performance on Math500



(b) Cumulative Variance Ratio of Activation Space

Figure 2: (a) Performance of R1-7B with varying α for direct and manifold steering on Math500. (b) Cumulative variance ratio of R1-7B's activation space on $D_{\text{redundant}}$ across different hidden layers.

[3]The orthogonality of $\mathbf{r}_{overthinking}$ and $\mathbf{r}_{other}$ is a property arising from the principles of PCA.

can be estimated using simple linear dimensionality reduction in this subspace. Eventually, this finding supports our earlier hypothesis (Eq. (2)) that the steering direction $\mathbf{r}^{(l^*)} = \mathbf{r}_{overthinking} + \mathbf{r}_{other}$ includes an orthogonal interference component $\mathbf{r}_{other}$, which falls into $\mathcal{M}^{\perp}$.

4.2 Theoretical Analysis of Interference Noise

As discussed in Sec. 4.1, the overthinking phenomenon is tied to the low-dimensional manifold structure of the activation space. The overthinking direction $\mathbf{r}^{(l^*)}$, computed via Eq. (2), introduces an orthogonal interference component $\mathbf{r}_{other}$ due to computation in high-dimensional spaces. When the activation dimension d far exceeds the sample size N ($d \gg N$), interference noise even accumulates in $\mathcal{M}^{\perp}$, inflating $\mathbf{r}_{other}$ and disrupting the model's other normal abilities. To further clarify the potential effects, we quantify this interference noise in the following theorem.

Theorem 4.1. *(Proof in Appendix B) Let $\mathbf{P}_{\mathcal{M}} = \mathbf{U}^{(l)}[:, 1:k](\mathbf{U}^{(l)}[:, 1:k])^{\top}$ be the projection matrix onto the low-dimensional manifold $\mathcal{M}$, where $\mathbf{U}^{(l)}[:, 1:k]$ contains top-k principal components of the activation covariance $\mathbf{C}^{(l)}$ for $D_{redundant}$ and $D_{concise}$. As the sample sizes grow sufficiently large, the expected noise norm of $\mathbf{r}_{other}$ is:*

$$\mathbb{E}[\|\mathbf{r}_{other}\|_2^2] = tr\left((\mathbf{I} - \mathbf{P}_{\mathcal{M}})\mathbf{\Sigma}_{noise}^{(l)}\right), \quad \mathbf{\Sigma}_{noise}^{(l)} = \frac{\mathbf{C}^{(l)}}{|D_{redundant}|} + \frac{\mathbf{C}^{(l)}}{|D_{concise}|}. \tag{6}$$

The trace is significant, indicating that the interference noise is substantial and is greatly likely to disrupt the model's normal abilities.

The significant noise norm of $\mathbf{r}_{other}$, as established in Theorem 4.1, suggests that interventions using $\mathbf{r}^{(l^*)}$ introduce considerable perturbations in $\mathcal{M}^{\perp}$. Moreover, these perturbations can propagate through layers, amplified by attention mechanisms, non-linear activations, and residual connections, leading to more substantial shifts in the activation distribution. To understand this, we further analyze the mean activation shift and its layer-wise amplification in the following theorem.

Theorem 4.2. *(Proof in Appendix B) Let $\mathbf{r}^{(l^*)}$ and $\mathbf{r}_{other}$ be as in Theorem 4.1, and let the intervention be applied as in Eq. (4). The mean activation shift at layer l and its amplification through layers are:*

$$\Delta\boldsymbol{\mu}^{(l)} = -\alpha\frac{1}{N}\sum_{i=1}^{N}[(\mathbf{r}^{(l^*)})^{\top}\mathbf{h}^{(l)}(x_i)]\mathbf{r}^{(l^*)}, \quad \|\Delta\boldsymbol{\mu}^{(l)}\|_2 \propto \alpha\|\mathbf{r}_{other}\|_2, \tag{7}$$

$$\|\Delta\boldsymbol{\mu}^{(l+1)}\|_2 \geq \gamma\|\Delta\boldsymbol{\mu}^{(l)}\|_2 + \alpha\gamma_{attn}\gamma_{\sigma}\sigma_{min}(\mathbf{W}^{(l+1)})|(\mathbf{r}_{other})^{\top}\mathbf{h}^{(l)}(x_i)|\|\mathbf{r}_{other}\|_2, \tag{8}$$

where α is the intervention strength, $\mathbf{h}^{(l)}(x_i)$ is the activation at layer l, $\mathbf{W}^{(l+1)}$ denotes the combined MLP and attention weights, γ_{attn} and γ_{σ} are the minimum amplification factors of the attention softmax and GeLU non-linearities, $\sigma_{min}(\mathbf{W}^{(l+1)})$ is the minimum singular value of the weights, and $\gamma > 1$ is the layer-wise amplification factor.

The shift in $\mathcal{M}^{\perp}$ is significant and grows through layer-wise propagation, driven by attention and non-linear transformations, severely disrupting the model's other normal abilities.

4.3 Manifold Steering

The interference direction $\mathbf{r}_{other}$, quantified in Theorem 4.1, causes activation shifts that amplify through transformer layers and disrupt reasoning (Theorem 4.2). To eliminate this interference, a simple but effective approach is to set Eq. (6) to $\mathbf{0}$. Based on this insight, we propose **Manifold Steering**, which projects the direction $\mathbf{r}^{(l^*)}$ onto $\mathcal{M}$ to mitigate $\mathbf{r}_{other}$.

Formally, let $\mathbf{U}_{\text{eff}}^{(l)} \in \mathbb{R}^{d\times k}$ denote the top-k principal components of the activation covariance in Eq. (5), spanning $\mathcal{M}$. The manifold direction is obtained by:

$$\mathbf{r}_{overthinking}^{(l^*)} = \mathbf{P}_{\mathcal{M}}\mathbf{r}^{(l^*)} = \mathbf{U}_{\text{eff}}^{(l)}(\mathbf{U}_{\text{eff}}^{(l)})^{\top}\mathbf{r}^{(l^*)}, \quad \mathbf{r}_{overthinking}^{(l)} = \frac{\mathbf{r}_{overthinking}^{(l)}}{\|\mathbf{r}_{overthinking}^{(l)}\|_2}, \tag{9}$$

where $\mathbf{P}_{\mathcal{M}} = \mathbf{U}_{\text{eff}}^{(l)}(\mathbf{U}_{\text{eff}}^{(l)})^{\top}$ is the projection matrix onto $\mathcal{M}$. The reason why the interference norm $\mathbb{E}[\|\mathbf{r}_{other}\|_2^2] = \text{tr}\left((\mathbf{I} - \mathbf{P}_{\mathcal{M}})\mathbf{\Sigma}_{\text{noise}}^{(l)}\right) = 0$ holds is because $\mathbf{\Sigma}_{\text{noise}}^{(l)}$ is now primarily supported in $\mathcal{M}$,

Table 1: Performance of Manifold Steering compared to Vanilla, Dynasor, and SEAL on GSM8K, MATH500, AMC2023, and AIME2024 for varied LRMs. Metrics include **Pass@1** (↑) and **#Tokens** (↓). Changes relative to Vanilla are shown in yellow for Pass@1 and blue for #Tokens.

Model	Methods	GSM8k		MATH500		AMC2023		AIME2024	
		Pass@1 (↑, %)	#Tokens (↓)	Pass@1 (↑, %)	#Tokens (↓)	Pass@1 (↑, %)	#Tokens (↓)	Pass@1 (↑, %)	#Tokens (↓)
R1-1.5B	Vanilla	76.7	2035	76.4	4762	70.0	7089	26.7	11352
	Dynasor	77.1 (+0.4)	1035 (-49%)	77.2 (+0.8)	3694 (-22%)	72.5 (+2.5)	6505 (-8%)	26.7 (+0.0)	10564 (-7%)
	SEAL	76.9 (+0.2)	1076 (-47%)	77.8 (+1.4)	3721 (-22%)	70.0 (+0.0)	6418 (-10%)	26.7 (+0.0)	10437 (-8%)
	Ours	**77.2** (+0.5)	**593** (-71%)	**78.6** (+2.2)	**3458** (-27%)	**72.5** (+2.5)	**6236** (-12%)	**30.0** (+3.3)	**10134** (-11%)
R1-7B	Vanilla	87.5	1143	88.2	3824	87.5	5871	50.0	10784
	Dynasor	87.6 (+0.1)	732 (-36%)	88.2 (+0.0)	2723 (-29%)	85.0 (-2.5)	5121 (-13%)	46.7 (-3.3)	9864 (-9%)
	SEAL	87.7 (+0.2)	829 (-32%)	87.8 (-0.4)	2651 (-34%)	85.0 (-0.0)	4750 (-19%)	46.7 (-3.3)	9394 (-13%)
	Ours	**87.6** (+0.1)	**440** (-62%)	**88.4** (+0.2)	**2239** (-42%)	**87.5** (+0.0)	**4440** (-24%)	**53.3** (+3.3)	**8457** (-22%)
R1-8B	Vanilla	82.7	1217	87.8	4009	85.0	5723	33.3	11278
	Dynasor	82.9 (+0.2)	826 (-32%)	88.0 (+0.2)	3171 (-21%)	82.5 (-2.5)	5019 (-12%)	46.7 (+13.4)	9901 (-12%)
	SEAL	82.7 (+0.0)	749 (-38%)	87.4 (-0.4)	3091 (-23%)	85.0 (+0.0)	4731 (-17%)	46.7 (+13.4)	9789 (-13%)
	Ours	**82.8** (+0.1)	**542** (-55%)	**88.0** (+0.2)	**2873** (-29%)	**85.0** (+0.0)	**4400** (-23%)	**50.0** (+16.7)	**9457** (-16%)
R1-14B	Vanilla	93.2	742	92.8	3496	90.0	5484	66.7	9986
	Dynasor	93.4 (+0.2)	596 (-20%)	92.6 (-0.2)	3233 (-8%)	92.5 (+2.5)	4817 (-12%)	63.3 (-3.4)	8941 (-11%)
	SEAL	93.6 (+0.2)	583 (-21%)	92.8 (+)	3139 (-10%)	87.5 (-2.5)	4470 (-18%)	60.0 (-6.7)	8563 (-14%)
	Ours	**93.6** (+0.4)	**438** (-41%)	**92.8** (+0.0)	**2074** (-41%)	**90.0** (+0.0)	**4061** (-26%)	**63.3** (-3.4)	**8132** (-19%)

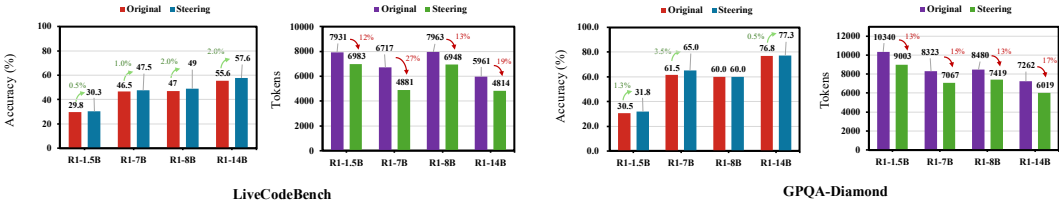


Figure 3: Cross-domain performance of Manifold Steering for overthinking mitigation on LiveCodeBench (code generation) and GPQA-Diamond (disciplinary knowledge).

resulting in zero components under $I - \mathbf{P}_{\mathcal{M}}$, *i.e.*, $\mathcal{M}^{\perp}$, successfully eliminating perturbations.

$$\mathbf{h}'^{(l)}(x) = \mathbf{h}^{(l)}(x) - \alpha \times \mathbf{r}^{(l)}_{overthinking}(\mathbf{r}^{(l)}_{overthinking})^{\top}\mathbf{h}^{(l)}(x_i). \tag{10}$$

The performance of manifold steering is shown in Fig. 2(b) and Sec. 5.2, where we find that, unlike the original paradigm, our manifold steering enables a sustained reduction in token count.

5 Experiments

5.1 Experimental Setups

We begin by briefly outlining the baseline methods, target LRMs, evaluation datasets, and metrics. For more detailed descriptions of the experimental settings, please refer to Appendix A.

Baseline Methods. We compare our manifold steering with two latest baselines, including Dynasor [16] and SEAL [8], both chosen for their ability to maintain the model's original accuracy. For their settings, we both adopt its official setting. To be aware, Dynasor's early stopping often omits the problem-solving process in the final answer, which is impractical for real-world applications. Thus, we require the model to provide a complete solution upon stopping.

Target LRMs. For a comprehensive evaluation, we select the DeepSeek-R1-Distilled series [14], comprising models of varying scales and architectures: DeepSeek-R1-Distill-Qwen (R1-1.5B, R1-7B, R1-14B) and DeepSeek-R1-Distill-Llama-8B (R1-8B). All models use recommended settings: temperature of 0.6, top-p of 0.95, and a maximum token limit of 16,384.

Evaluation Datasets & Metrics. To evaluate the effectiveness of Manifold Steering, we include mathematical datasets of varying difficulty: GSM8K [10], MATH500 [20], AMC2023 [24], and AIME2024 [25]. To further verify the transferability, we use LiveCodeBench [18] for code generation and GPQA-Diamond [32] for expert-level disciplinary knowledge. All datasets are evaluated using **Pass@1** as the task-solving metric and the average token count (**#Tokens**) for overthinking mitigation.

Implementation Details. The data for computing steering directions is filtered using the method outlined in Sec. 3 on the OpenMathInstruct2 dataset [36]. For each model, we specify the layer

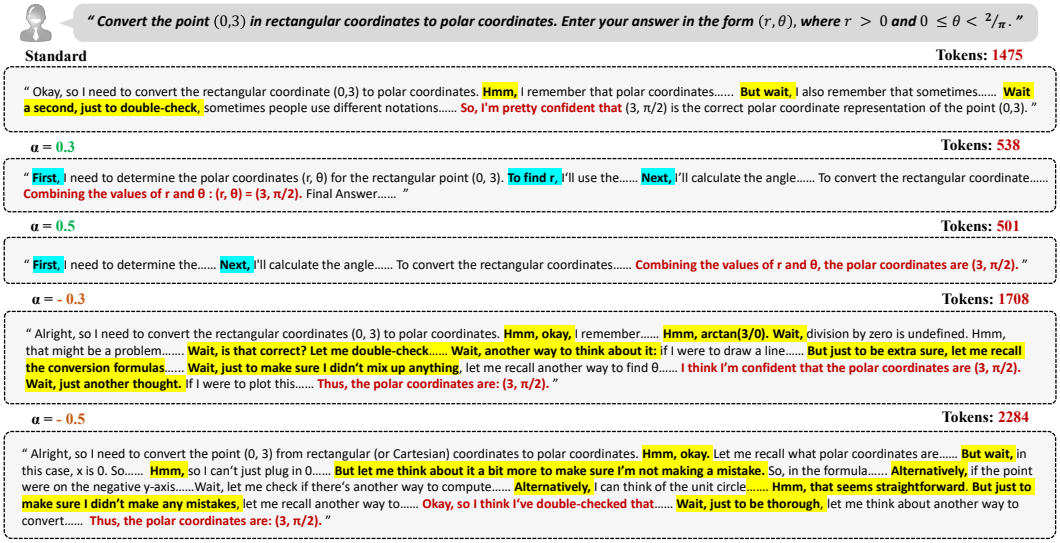


Figure 4: An example of steering overthinking in model outputs. Forward steering yields concise, confident responses, eliminating hesitant phrases, while reverse steering induces verbose outputs.

used to compute the steering direction and the intervention strength α as follows: R1-1.5B (layer 27, $\alpha = 0.7$), R1-7B (layer 27, $\alpha = 0.3$), R1-8B (layer 31, $\alpha = 0.5$), and R1-14B (layer 47, $\alpha = 0.3$). During inference, this direction is applied to all layers as stated in Eq. (3).

5.2 Performance of Manifold Steering in Overthinking Mitigation

We conduct experiments on four mathematical datasets of varying difficulty using four LRMs with different parameter sizes and architectures. Table 1 presents the results, where models are evaluated for accuracy and redundancy reduction. Based on Table 1, we draw the following observations:

Manifold steering achieves the best performance across all models and datasets. Our method consistently outperforms baselines on four out-of-distribution mathematical datasets, with particularly strong results on GSM8K, where it achieves token reduction of 41% $\sim$ 71% while preserving accuracy. This is reasonable, as unlike Dynasor, which relies on external monitoring of the model's certainty, our method modifies model outputs at the more fundamental feature level to mitigate overthinking behavior. Moreover, Dynasor's reliance on external monitoring brings extra computational overhead. We include a comparison of average time cost to demonstrate this drawback in Appendix E, while our method incurs nearly no latency. Compared to SEAL, our manifold steering effectively reduces interference noise, yielding a more precise direction. Additionally, we observe that the overthinking phenomenon diminishes to some extent as model parameter size increases, which is expected, as some overthinking stems from models' inability to solve complex problems.

Overthinking is more pronounced in simpler problems. As presented in Table 1, all methods exhibit more effective overthinking mitigation on simpler datasets, with GSM8K and MATH500 ($\sim$ 40%) showing greater token reduction compared to the more complex AMC2023 and AIME2024 datasets ($\sim$ 20%). This suggests that overthinking is more pronounced in simpler problems, which is reasonable, as complex problems inherently require larger token budgets and may exceed the models' internal capabilities, thereby constraining mitigation effectiveness.

5.3 Cross-Domain Transferability for Overthinking Mitigation

To further investigate the transferability of manifold steering for overthinking mitigation, we assess its performance across two distinct domains: code generation and discipline-specific knowledge, both separate from the mathematical domain used for steering direction extraction. We utilize two representative datasets: **1) LiveCodeBench** [18], a benchmark of coding challenges that probe algorithmic and programming expertise, and **2) GPQA-Diamond** [32], a carefully curated dataset of challenging multiple-choice questions targeting expert-level disciplinary knowledge across various

fields. As shown in Fig. 3, our manifold steering achieves token reduction of 12% $\sim$ 27% across both datasets while maintaining accuracy, demonstrating the generalizability of manifold steering to diverse domains. This cross-domain effectiveness offers multiple benefits: it incurs no additional computational overhead, adapts seamlessly to varied problem structures, and effectively mitigates overthinking without requiring domain-specific fine-tuning.

5.4 Directional Analysis and Hyperparameter Tuning

In this section, we first explore the precise impact of the direction computed by manifold steering on model outputs through case studies, analyzing how steering and its reversal affect response characteristics to better understand directionality's role. Then, we conduct hyperparameter tuning.

Directional Analysis. As shown in Fig. 4, applying the steering direction for overthinking mitigation leads to model outputs that are significantly more concise and confident. Specifically, overthinking behaviors, such as hesitant phrases (e.g., "wait"), frequent shifts in reasoning (e.g., "alternatively"), and repetitive self-checking, are largely eliminated. The model generates streamlined responses with clear, focused reasoning, delivering direct outputs. This effect strengthens as the intervention strength increases to 0.5. In contrast, when reverse steering is applied, the model becomes markedly more hesitant, often repeatedly checking. This leads to verbose outputs filled with excessive caution. Thus, it is crucial to underscore the role of directionality for overthinking.

Hyperparameter Tuning. We use R1-7B model on MATH500 for this analysis, with results for other models in Appendix C. As shown in Fig. 5, our manifold steering direction demonstrates efficacy at a much lower strength of $\alpha = 0.1$. As α increases, token counts continue to decrease, with a remarkable 77.1% reduction observed at $\alpha = 0.5$. This substantial token reduction highlights the purity and effectiveness of our steering direction in mitigating overthinking. However, excessively rapid reasoning, induced by intervention strengths, can hinder the model's ability to thoroughly address complex problems, a phenomenon also observed in human cognition, leading to a decline in accuracy. To balance the trade-off between overthinking mitigation and maintaining accuracy, we select an intervention strength of $\alpha = 0.3$ as the optimal value for robust performance.

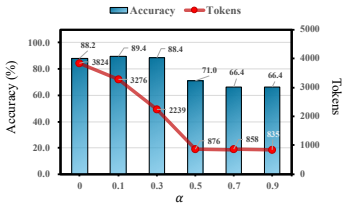


Figure 5: Hyperparameter tuning for strength α in R1-7B

5.5 Cross-Task Transferability of Manifold Steering

In this section, we investigate the applicability of manifold steering to tasks beyond overthinking mitigation, such as refusal feature ablation, to assess its cross-task transferability. Prior studies [2, 45] demonstrate that while steering directions can suppress refusal features in models, some instances persist unless intervention strength is increased, which risks model collapse. Here, we apply our manifold steering method using the Qwen2.5-7B-Instruct as the target LLM, computing the steering direction with the same data as in [2]. As shown in Fig. 6, our method achieves a 100% jailbreak success rate (JSR) on AdvBench [51] while the baseline [2] obtains a JSR of 74% ($\alpha = 2.0$), with all responses verified as valid through manual check, which further validates the robust transferability of manifold steering across diverse tasks and underscores the urgent need for enhanced safety efforts [13, 17, 46, 47, 48] to ensure responsible AI.

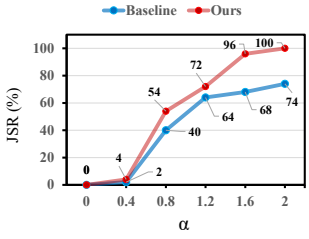


Figure 6: JSR of baseline and ours on Advbench.

6 Discussion and Limitations

Our proposed manifold steering method has demonstrated robust effectiveness in mitigating overthinking, as evidenced by significant token reductions across varying LRMs. However, its applicability to multi-modal large language models remains unexplored. Additionally, while our approach excels in controlling overthinking with minimal accuracy trade-offs, its interaction with highly specialized tasks, such as domain-specific reasoning, e.g., legal or medical analysis, warrants further investiga-

tion. Moreover, the sensitivity of our method to varying intervention strengths suggests potential for optimizing dynamic steering strategies, where the strength adapts to task complexity in real-time.

7 Conclusion

In this work, we propose manifold steering, a novel method to address overthinking in LRMs while preserving task performance without additional computational cost. Specifically, by aligning the steering direction with the low-dimensional activation manifold, our approach effectively eliminates the interference noise based on theoretical analysis. Extensive experiments across diverse models and datasets confirm substantial token reductions and robust cross-task transferability. These findings underscore the potential of manifold steering to enhance model efficiency and adaptability, opening new avenues for improving LRMs.

Acknowledgments and Disclosure of Funding

This work was supported by NSFC Projects (62576020, 62276149) and was also supported by the Fundamental Research Funds for the Central Universities.

References

[1] Janice Ahn, Rishu Verma, Renze Lou, Di Liu, Rui Zhang, and Wenpeng Yin. Large language models for mathematical reasoning: Progresses and challenges. *arXiv preprint arXiv:2402.00157*, 2024.

[2] Andy Arditi, Oscar Balcells Obeso, Aaquib Syed, Daniel Paleka, Nina Rimsky, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[3] David D Baek and Max Tegmark. Towards understanding distilled reasoning models: A representational approach. *arXiv preprint arXiv:2503.03730*, 2025.

[4] Nora Belrose. Diff-in-means concept editing is worst-case optimal: Explaining a result by Sam Marks and Max Tegmark, 2023. https://blog.eleuther.ai/diff-in-means/, 2023. Accessed: May 20, 2024.

[5] Leonard Bereska and Efstratios Gavves. Mechanistic interpretability for ai safety–a review. *arXiv preprint arXiv:2404.14082*, 2024.

[6] Nicola Cancedda. Spectral filters, dark signals, and attention sinks. *arXiv preprint arXiv:2402.09221*, 2024.

[7] Yuanpu Cao, Tianrong Zhang, Bochuan Cao, Ziyi Yin, Lu Lin, Fenglong Ma, and Jinghui Chen. Personalized steering of large language models: Versatile steering vectors through bi-directional preference optimization. *Advances in Neural Information Processing Systems*, 37:49519–49551, 2024.

[8] 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.

[9] 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.

[10] 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. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[11] Arthur Conmy, Augustine Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. Towards automated circuit discovery for mechanistic interpretability. *Advances in Neural Information Processing Systems*, 36:16318–16352, 2023.

[12] Alejandro Cuadron, Dacheng Li, Wenjie Ma, Xingyao Wang, Yichuan Wang, Siyuan Zhuang, Shu Liu, Luis Gaspar Schroeder, Tian Xia, Huanzhi Mao, et al. The danger of overthinking: Examining the reasoning-action dilemma in agentic tasks. *arXiv preprint arXiv:2502.08235*, 2025.

[13] Josef Dai, Xuehai Pan, Ruiyang Sun, Jiaming Ji, Xinbo Xu, Mickel Liu, Yizhou Wang, and Yaodong Yang. Safe rlhf: Safe reinforcement learning from human feedback. *arXiv preprint arXiv:2310.12773*, 2023.

[14] DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025.

[15] Joshua Engels, Isaac Liao, Eric J Michaud, Wes Gurnee, and Max Tegmark. Not all language model features are linear. *arXiv e-prints*, pages arXiv–2405, 2024.

[16] Yichao Fu, Junda Chen, Siqi Zhu, Zheyu Fu, Zhongdongming Dai, Aurick Qiao, and Hao Zhang. Efficiently serving llm reasoning programs with certaindex. *arXiv preprint arXiv:2412.20993*, 2024.

[17] Yao Huang, Yitong Sun, Shouwei Ruan, Yichi Zhang, Yinpeng Dong, and Xingxing Wei. Breaking the ceiling: Exploring the potential of jailbreak attacks through expanding strategy space. *arXiv preprint arXiv:2505.21277*, 2025.

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

[19] Xiaoxi Li, Jiajie Jin, Guanting Dong, Hongjin Qian, Yutao Zhu, Yongkang Wu, Ji-Rong Wen, and Zhicheng Dou. Webthinker: Empowering large reasoning models with deep research capability. *arXiv preprint arXiv:2504.21776*, 2025.

[20] Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023.

[21] Peter J Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. Generating wikipedia by summarizing long sequences. *arXiv preprint arXiv:1801.10198*, 2018.

[22] Yule Liu, Jingyi Zheng, Zhen Sun, Zifan Peng, Wenhan Dong, Zeyang Sha, Shiwen Cui, Weiqiang Wang, and Xinlei He. Thought manipulation: External thought can be efficient for large reasoning models. *arXiv preprint arXiv:2504.13626*, 2025.

[23] Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, and Dongmei Zhang. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. *arXiv preprint arXiv:2308.09583*, 2023.

[24] Mathematical Association of America. American mathematics competitions (amc) 10 and 12, 2023. Problems and Answer Keys, 2023.

[25] Mathematical Association of America. American invitational mathematics examination (aime) i and ii, 2024. Problems and Answer Keys, 2024.

[26] Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. *arXiv preprint arXiv:2301.05217*, 2023.

[27] Ansong Ni, Miltiadis Allamanis, Arman Cohan, Yinlin Deng, Kensen Shi, Charles Sutton, and Pengcheng Yin. Next: Teaching large language models to reason about code execution. *arXiv preprint arXiv:2404.14662*, 2024.

[28] OpenAI. Learning to reason with LLMs, 2024.

[29] Nina Panickssery, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt Turner. Steering llama 2 via contrastive activation addition. *arXiv preprint arXiv:2312.06681*, 2023.

[30] Xiao Pu, Michael Saxon, Wenyue Hua, and William Yang Wang. Thoughtterminator: Benchmarking, calibrating, and mitigating overthinking in reasoning models. *arXiv preprint arXiv:2504.13367*, 2025.

[31] Daking Rai, Yilun Zhou, Shi Feng, Abulhair Saparov, and Ziyu Yao. A practical review of mechanistic interpretability for transformer-based language models. *arXiv preprint arXiv:2407.02646*, 2024.

[32] 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.

[33] Bingqing Song, Boran Han, Shuai Zhang, Hao Wang, Haoyang Fang, Bonan Min, Yuyang Wang, and Mingyi Hong. Effectively steer llm to follow preference via building confident directions. *arXiv preprint arXiv:2503.02989*, 2025.

[34] Nishant Subramani, Nivedita Suresh, and Matthew E Peters. Extracting latent steering vectors from pretrained language models. *arXiv preprint arXiv:2205.05124*, 2022.

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

[36] Shubham Toshniwal, Wei Du, Ivan Moshkov, Branislav Kisacanin, Alexan Ayrapetyan, and Igor Gitman. Openmathinstruct-2: Accelerating ai for math with massive open-source instruction data. *arXiv preprint arXiv:2410.01560*, 2024.

[37] Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*, 2023.

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[39] Yiming Wang, Pei Zhang, Jialong Tang, Haoran Wei, Baosong Yang, Rui Wang, Chenshu Sun, Feitong Sun, Jiran Zhang, Junxuan Wu, et al. Polymath: Evaluating mathematical reasoning in multilingual contexts. *arXiv preprint arXiv:2504.18428*, 2025.

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

[41] 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.

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

[43] Dayu Yang, Tianyang Liu, Daoan Zhang, Antoine Simoulin, Xiaoyi Liu, Yuwei Cao, Zhaopu Teng, Xin Qian, Grey Yang, Jiebo Luo, et al. Code to think, think to code: A survey on code-enhanced reasoning and reasoning-driven code intelligence in llms. *arXiv preprint arXiv:2502.19411*, 2025.

[44] Dian Yu, Baolin Peng, Ye Tian, Linfeng Song, Haitao Mi, and Dong Yu. Siam: Self-improving code-assisted mathematical reasoning of large language models. *arXiv preprint arXiv:2408.15565*, 2024.

[45] Lei Yu, Virginie Do, Karen Hambardzumyan, and Nicola Cancedda. Robust LLM safeguarding via refusal feature adversarial training. In *The Thirteenth International Conference on Learning Representations*, 2025.

[46] Yichi Zhang, Yao Huang, Yitong Sun, Chang Liu, Zhe Zhao, Zhengwei Fang, Yifan Wang, Huanran Chen, Xiao Yang, Xingxing Wei, et al. Multitrust: A comprehensive benchmark towards trustworthy multimodal large language models. *Advances in Neural Information Processing Systems*, 37:49279–49383, 2024.

[47] Yichi Zhang, Zihao Zeng, Dongbai Li, Yao Huang, Zhijie Deng, and Yinpeng Dong. Realsafe-r1: Safety-aligned deepseek-r1 without compromising reasoning capability. *arXiv preprint arXiv:2504.10081*, 2025.

[48] Yichi Zhang, Siyuan Zhang, Yao Huang, Zeyu Xia, Zhengwei Fang, Xiao Yang, Ranjie Duan, Dong Yan, Yinpeng Dong, and Jun Zhu. Stair: Improving safety alignment with introspective reasoning. *arXiv preprint arXiv:2502.02384*, 2025.

[49] Yuxiang Zheng, Dayuan Fu, Xiangkun Hu, Xiaojie Cai, Lyumanshan Ye, Pengrui Lu, and Pengfei Liu. Deepresearcher: Scaling deep research via reinforcement learning in real-world environments. *arXiv preprint arXiv:2504.03160*, 2025.

[50] Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*, 2023.

[51] Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*, 2023.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We clearly state our main contributions in the abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have discussed it in Sec. 6.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We have provided it in Appendix B.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We have provided it in Appendix A.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: Data and code will be available when the paper is public.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We have included in Sec. 5.1 and Appendix A.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Our experiments' results are stable, and the LLM's temperature for evaluation is set to 0.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We have included it in Sec. 5.1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: We have followed the code of ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We have included it in Sec. 6.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: [NA]

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: What we use are all public resources, and we obtain the owners' permission.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: [NA]

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: [NA]

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: [NA]

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [No]

Justification: We only use LLM for evaluation.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.

A Implement Details

In this section, we will introduce the implementation details of our approach, focusing on the data selection process for computing the steering direction to mitigate overthinking in each model and the specific configurations of the baseline methods used for comparison.

A.1 Data Selection for Direction Computation

To construct representative datasets for computing the steering direction to mitigate overthinking, we begin by randomly sampling 20k questions from the OpenMathInstruct-2 training set [36]. For each model, we generate five independent responses per question using the official sampling configuration: a temperature of 0.6, top-p of 0.95, and a maximum length of 16k tokens. These responses form the basis for constructing two model-specific datasets, the **Redundant set** ($D_{\text{redundant}}$) and the **Concise set** (D_{concise}), as described below:

- **Redundant set ($D_{\text{redundant}}$):** This dataset includes questions where all five responses exceed 16k tokens without terminating and contain more than 20 times of hesitation keywords (e.g., "wait", "alternatively", etc.). To capture meaningful overthinking behavior, we process the responses using the following template, truncating the response at the occurrence of the hesitation keyword:

  ```
  <|begin_of_sentence|><|User|>{instruction}<|Assistant|><think>\n
  {partial_response}{hesitation keyword}
  ```

  The truncation at a hesitation keyword is reasonable because overthinking typically emerges after a certain point in the response, rather than immediately upon encountering the question. Moreover, through activation visualization, we observe no significant differences in the activation patterns of different hesitation keywords. Thus we choose "wait" as a consistent marker here.
- **Concise set (D_{concise}):** This dataset includes questions where all five responses are under 1k tokens and contain none of the hesitation keywords. The template for these responses includes only the instruction without the response, as they inherently represent concise and focused outputs:

  ```
  <|begin_of_sentence|><|User|>{instruction}<|Assistant|><think>\n
  ```

These selection criteria ensure that $D_{\text{redundant}}$ captures responses exhibiting excessive verbosity and hesitation, while D_{concise} represents efficient and direct responses, providing a clear contrast for computing the steering direction representing overthinking. To ensure high-quality data, we retain only 500 samples for each dataset after applying the selection criteria and double checking. For computing the steering direction, we follow [50] to sample 100 samples from each dataset and employ the IsolationForest algorithm to filter out outliers. For manifold subspace estimation, we utilize the entire set of 500 samples from each dataset to capture the full representational structure.

A.2 Baseline Methods

As stated in Sec. 5.1, we select two latest baselines, Dynasor [16] and SEAL [8], for their ability to preserve the original accuracy in reasoning tasks. Below, we detail the specific settings for them:

General Setting. All large reasoning models adopt the official recommended settings with a temperature of 0.6, top-p of 0.95, and a maximum length of 16k tokens.

Dynasor. We adopt the official settings for Dynasor. The configuration probes the model every 32 tokens with a "Probe-In-The-Middle" technique and injects a "Final Answer" prompt at each iteration to ensure complete solutions upon early termination. Generation stops when the Certaindex metric ($\tilde{H}$) exceeds a predefined confidence threshold. To be aware, Dynasor's early stopping often omits the problem-solving process in the final answer, which is impractical for real-world applications. Thus, we require the model to provide a complete solution in the final answer upon stopping.

SEAL. We adopt the official settings for SEAL [8], using 1k training samples from the Math dataset [20] to extract the reasoning steering vector. Reasoning processes are segmented into thoughts using "\n\n" delimiters, classified as execution, reflection, or transition via keyword-based rules

(e.g., "Alternatively" for transition, "Wait" for reflection). The steering vector is computed at layer 20 as $S = \bar{H}_E - \bar{H}_{RT}$, where $\bar{H}_E$ and $\bar{H}_{RT}$ are average representations of execution and reflection/transition thoughts, respectively. During greedy decoding, hidden states of "\n\n" tokens at layer 20 are adjusted as $\tilde{H} = H + 1.0 \cdot S$.

B Proofs

B.1 Proof of Theorem 4.1

Proof. We derive the expected noise norm of the interference component $\mathbf{r}_{other}$, the part of the overthinking direction $\mathbf{r}^{(l^*)}$ in the orthogonal complement $\mathcal{M}^\perp$ of the low-dimensional manifold $\mathcal{M}$. The theorem states:

$$\mathbb{E}[\|\mathbf{r}_{other}\|_2^2] = \text{tr}\left((\mathbf{I} - \mathbf{P}_\mathcal{M})\mathbf{\Sigma}_{\text{noise}}^{(l)}\right), \quad \mathbf{\Sigma}_{\text{noise}}^{(l)} = \frac{\mathbf{C}^{(l)}}{|D_{\text{redundant}}|} + \frac{\mathbf{C}^{(l)}}{|D_{\text{concise}}|},$$

where $\mathbf{P}_\mathcal{M} = \mathbf{U}^{(l)}[:, 1:k](\mathbf{U}^{(l)}[:, 1:k])^\top$, and $\mathbf{U}^{(l)}[:, 1:k]$ are the top-k principal components of the activation covariance $\mathbf{C}^{(l)}$. We build on prior findings that $\mathcal{M}$ is low-dimensional, identified via PCA on $D_{\text{reasoning}} = D_{\text{redundant}} \cup D_{\text{concise}}$, with $k = 10$ capturing over 70% of the variance, validating the linear manifold assumption.

Step 1: Define the overthinking direction $\mathbf{r}^{(l^*)}$. Per Eq. (2), $\mathbf{r}^{(l^*)} = \mathbf{r}_{overthinking} + \mathbf{r}_{other}$, where $\mathbf{r}_{overthinking} \in \mathcal{M}$ captures the shift between redundant and concise reasoning, and $\mathbf{r}_{other} \in \mathcal{M}^\perp$ is interference. We model:

$$\mathbf{r}^{(l^*)} = \frac{1}{|D_{\text{redundant}}|} \sum_{x_i \in D_{\text{redundant}}} \mathbf{h}^{(l)}(x_i) - \frac{1}{|D_{\text{concise}}|} \sum_{x_i \in D_{\text{concise}}} \mathbf{h}^{(l)}(x_i).$$

Assume activations $\mathbf{h}^{(l)}(x_i) \sim \mathcal{N}(\boldsymbol{\mu}_{\text{set}}, \mathbf{C}^{(l)})$, with $\boldsymbol{\mu}_{\text{redundant}}$ or $\boldsymbol{\mu}_{\text{concise}}$ for each dataset, and $\mathbf{C}^{(l)}$ estimated over $D_{\text{reasoning}}$. The covariance is:

$$\mathbb{E}[\mathbf{r}^{(l^*)}\mathbf{r}^{(l^*)\top}] = \frac{\mathbf{C}^{(l)}}{|D_{\text{redundant}}|} + \frac{\mathbf{C}^{(l)}}{|D_{\text{concise}}|}.$$

Step 2: Define $\mathcal{M}$ and derive $\mathbf{I} - \mathbf{P}_\mathcal{M}$. The manifold $\mathcal{M}$ is spanned by the top-k eigenvectors of $\mathbf{C}^{(l)} = \frac{1}{N-1}\mathbf{A}^{(l)}(\mathbf{A}^{(l)} - \bar{\mathbf{A}}^{(l)})^\top$, where $\mathbf{A}^{(l)} = [\mathbf{h}^{(l)}(x_1), \ldots, \mathbf{h}^{(l)}(x_N)]$, and $\bar{\mathbf{A}}^{(l)} = \frac{1}{N}\sum_{i=1}^N \mathbf{h}^{(l)}(x_i)$. The eigendecomposition $\mathbf{C}^{(l)} = \mathbf{U}^{(l)}\mathbf{\Lambda}^{(l)}(\mathbf{U}^{(l)})^\top$ yields $\mathbf{U}^{(l)}[:, 1:k]$, and:

$$\mathbf{P}_\mathcal{M} = \mathbf{U}^{(l)}[:, 1:k](\mathbf{U}^{(l)}[:, 1:k])^\top.$$

The projection onto $\mathcal{M}^\perp$ is $\mathbf{I} - \mathbf{P}_\mathcal{M}$, as it removes the $\mathcal{M}$-component. Since $\mathbf{U}^{(l)}[:, 1:k]$ is orthonormal, $\mathbf{P}_\mathcal{M}$ is idempotent and symmetric, so:

$$(\mathbf{I} - \mathbf{P}_\mathcal{M})^2 = \mathbf{I} - \mathbf{P}_\mathcal{M}, \quad (\mathbf{I} - \mathbf{P}_\mathcal{M})^\top = \mathbf{I} - \mathbf{P}_\mathcal{M}.$$

PCA's linear basis ensures $\mathcal{M}^\perp$ captures the $d - k$ dimensions of noise, critical when $d \gg k$.

Step 3: Define $\mathbf{r}_{other}$. Since $\mathbf{r}_{overthinking} \in \mathcal{M}$, the interference is:

$$\mathbf{r}_{other} = (\mathbf{I} - \mathbf{P}_\mathcal{M})\mathbf{r}^{(l^*)}.$$

This isolates noise in $\mathcal{M}^\perp$, which disrupts normal abilities due to high-dimensional computation.

Step 4: Compute the squared norm. Calculate:

$$\|\mathbf{r}_{other}\|_2^2 = [(\mathbf{I} - \mathbf{P}_\mathcal{M})\mathbf{r}^{(l^*)}]^\top(\mathbf{I} - \mathbf{P}_\mathcal{M})\mathbf{r}^{(l^*)} = \mathbf{r}^{(l^*)\top}(\mathbf{I} - \mathbf{P}_\mathcal{M})\mathbf{r}^{(l^*)},$$

using the idempotence of $\mathbf{I} - \mathbf{P}_\mathcal{M}$.

Step 5: Take the expectation. Compute:

$$\mathbb{E}[\|\mathbf{r}_{other}\|_2^2] = \mathbb{E}[\mathbf{r}^{(l^*)\top}(\mathbf{I} - \mathbf{P}_\mathcal{M})\mathbf{r}^{(l^*)}] = \text{tr}((\mathbf{I} - \mathbf{P}_\mathcal{M})\mathbb{E}[\mathbf{r}^{(l^*)}\mathbf{r}^{(l^*)\top}]).$$

Substitute:

$$\mathbb{E}[\mathbf{r}^{(l^*)}\mathbf{r}^{(l^*)\top}] = \mathbf{\Sigma}_{\text{noise}}^{(l)} = \frac{\mathbf{C}^{(l)}}{|D_{\text{redundant}}|} + \frac{\mathbf{C}^{(l)}}{|D_{\text{concise}}|}.$$

Thus:

$$\mathbb{E}[\|\mathbf{r}_{other}\|_2^2] = \text{tr}\left((\mathbf{I} - \mathbf{P}_\mathcal{M})\mathbf{\Sigma}_{\text{noise}}^{(l)}\right).$$

□

B.2 Proof of Theorem 4.2

Proof. We derive the mean activation shift $\Delta\boldsymbol{\mu}^{(l)}$ at layer l due to the intervention (applied as in Eq. (4).) along the overthinking direction $\mathbf{r}^{(l^*)}$, showing its norm is proportional to $\alpha\|\mathbf{r}_{other}\|_2$, and establish the layer-wise amplification of the shift at layer $l+1$. The theorem builds on Theorem 4.1.

Step 1: Derive the mean activation shift. The intervention at layer l is:

$$\mathbf{h}^{(l)'}(x_i) = \mathbf{h}^{(l)}(x_i) - \alpha[(\mathbf{r}^{(l^*)})^\top \mathbf{h}^{(l)}(x_i)]\mathbf{r}^{(l^*)},$$

with $\alpha > 0$. The mean activation before intervention is:

$$\boldsymbol{\mu}^{(l)} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{h}^{(l)}(x_i),$$

and post-intervention:

$$\boldsymbol{\mu}^{(l)'} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{h}^{(l)'}(x_i) = \frac{1}{N}\sum_{i=1}^{N}\left(\mathbf{h}^{(l)}(x_i) - \alpha[(\mathbf{r}^{(l^*)})^\top \mathbf{h}^{(l)}(x_i)]\mathbf{r}^{(l^*)}\right).$$

Compute:

$$\boldsymbol{\mu}^{(l)'} = \boldsymbol{\mu}^{(l)} - \alpha\frac{1}{N}\sum_{i=1}^{N}[(\mathbf{r}^{(l^*)})^\top \mathbf{h}^{(l)}(x_i)]\mathbf{r}^{(l^*)}.$$

The mean shift is:

$$\Delta\boldsymbol{\mu}^{(l)} = \boldsymbol{\mu}^{(l)'} - \boldsymbol{\mu}^{(l)} = -\alpha\frac{1}{N}\sum_{i=1}^{N}[(\mathbf{r}^{(l^*)})^\top \mathbf{h}^{(l)}(x_i)]\mathbf{r}^{(l^*)},$$

matching the first part of Eq. (7).

Step 2: Decompose the shift and isolate $\mathbf{r}_{other}$ contribution. Since $\mathbf{r}^{(l^*)} = \mathbf{r}_{\mathcal{M}} + \mathbf{r}_{other}$ with $\mathbf{r}_{\mathcal{M}} \perp \mathbf{r}_{other}$, we can decompose:

$$(\mathbf{r}^{(l^*)})^\top \mathbf{h}^{(l)}(x_i) = (\mathbf{r}_{\mathcal{M}})^\top \mathbf{h}^{(l)}(x_i) + (\mathbf{r}_{other})^\top \mathbf{h}^{(l)}(x_i).$$

Thus:

$$\begin{aligned}
\Delta\boldsymbol{\mu}^{(l)} &= -\alpha\frac{1}{N}\sum_{i=1}^{N}[(\mathbf{r}_{\mathcal{M}})^\top \mathbf{h}^{(l)}(x_i) + (\mathbf{r}_{other})^\top \mathbf{h}^{(l)}(x_i)](\mathbf{r}_{\mathcal{M}} + \mathbf{r}_{other}) \\
&= -\alpha\frac{1}{N}\sum_{i=1}^{N}[(\mathbf{r}_{\mathcal{M}})^\top \mathbf{h}^{(l)}(x_i)]\mathbf{r}_{\mathcal{M}} - \alpha\frac{1}{N}\sum_{i=1}^{N}[(\mathbf{r}_{\mathcal{M}})^\top \mathbf{h}^{(l)}(x_i)]\mathbf{r}_{other} \\
&\quad - \alpha\frac{1}{N}\sum_{i=1}^{N}[(\mathbf{r}_{other})^\top \mathbf{h}^{(l)}(x_i)]\mathbf{r}_{\mathcal{M}} - \alpha\frac{1}{N}\sum_{i=1}^{N}[(\mathbf{r}_{other})^\top \mathbf{h}^{(l)}(x_i)]\mathbf{r}_{other}.
\end{aligned}$$

Let:

$$s_{\mathcal{M}} = \frac{1}{N}\sum_{i=1}^{N}[(\mathbf{r}_{\mathcal{M}})^\top \mathbf{h}^{(l)}(x_i)], \quad s_{other} = \frac{1}{N}\sum_{i=1}^{N}[(\mathbf{r}_{other})^\top \mathbf{h}^{(l)}(x_i)].$$

Then:

$$\Delta\boldsymbol{\mu}^{(l)} = -\alpha s_{\mathcal{M}}\mathbf{r}_{\mathcal{M}} - \alpha s_{\mathcal{M}}\mathbf{r}_{other} - \alpha s_{other}\mathbf{r}_{\mathcal{M}} - \alpha s_{other}\mathbf{r}_{other}.$$

Since $\mathbf{r}_{\mathcal{M}} \perp \mathbf{r}_{other}$:

$$\begin{aligned}
\|\Delta\boldsymbol{\mu}^{(l)}\|_2^2 &= \alpha^2\|s_{\mathcal{M}}\mathbf{r}_{\mathcal{M}} + s_{other}\mathbf{r}_{\mathcal{M}}\|_2^2 + \alpha^2\|s_{\mathcal{M}}\mathbf{r}_{other} + s_{other}\mathbf{r}_{other}\|_2^2 \\
&= \alpha^2(s_{\mathcal{M}} + s_{other})^2\|\mathbf{r}_{\mathcal{M}}\|_2^2 + \alpha^2(s_{\mathcal{M}} + s_{other})^2\|\mathbf{r}_{other}\|_2^2.
\end{aligned}$$

Let $s = s_{\mathcal{M}} + s_{other}$. Then:

$$\|\Delta\boldsymbol{\mu}^{(l)}\|_2 = \alpha|s|\sqrt{\|\mathbf{r}_{\mathcal{M}}\|_2^2 + \|\mathbf{r}_{other}\|_2^2}.$$

By Theorem 4.1, when the error component $\mathbf{r}_{other}$ is present (i.e., $\|\mathbf{r}_{other}\|_2 > 0$), it contributes to the total norm. The dominant term depends on the relative magnitudes of $\|\mathbf{r}_{\mathcal{M}}\|_2$ and $\|\mathbf{r}_{other}\|_2$. Assuming $\mathbf{h}^{(l)}(x_i) \sim \mathcal{N}(\boldsymbol{\mu}_{\text{set}}, \mathbf{C}^{(l)})$ and $|s|$ is a positive constant, we obtain:

$$\|\Delta\boldsymbol{\mu}^{(l)}\|_2 \propto \alpha\|\mathbf{r}_{other}\|_2,$$

when $\|\mathbf{r}_{other}\|_2$ dominates, completing Eq. (7).

Step 3: Derive the layer-wise amplification from $\mathbf{r}_{other}$. For layer $l+1$, the activation is:

$$\mathbf{h}^{(l+1)}(x_i) = \sigma\left(\mathbf{W}^{(l+1)}\text{Attn}(\mathbf{h}^{(l)}(x_i))\right),$$

and post-intervention:

$$\mathbf{h}^{(l+1)'}(x_i) = \sigma\left(\mathbf{W}^{(l+1)}\text{Attn}(\mathbf{h}^{(l)'}(x_i))\right),$$

where $\mathbf{W}^{(l+1)}$ combines MLP and attention weights, Attn is the attention mechanism, and σ is GeLU. The mean shift is:

$$\Delta\boldsymbol{\mu}^{(l+1)} = \boldsymbol{\mu}^{(l+1)'} - \boldsymbol{\mu}^{(l+1)}, \quad \boldsymbol{\mu}^{(l+1)'} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{h}^{(l+1)'}(x_i), \quad \boldsymbol{\mu}^{(l+1)} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{h}^{(l+1)}(x_i).$$

To isolate the $\mathbf{r}_{other}$ contribution, decompose the single-input shift at layer l:

$$\begin{aligned}\Delta\mathbf{h}^{(l)}(x_i) &= \mathbf{h}^{(l)'}(x_i) - \mathbf{h}^{(l)}(x_i)\\ &= -\alpha[(\mathbf{r}^{(l^*)})^\top\mathbf{h}^{(l)}(x_i)]\mathbf{r}^{(l^*)}\\ &= -\alpha[(\mathbf{r}_{\mathcal{M}})^\top\mathbf{h}^{(l)}(x_i) + (\mathbf{r}_{other})^\top\mathbf{h}^{(l)}(x_i)](\mathbf{r}_{\mathcal{M}} + \mathbf{r}_{other}).\end{aligned}$$

The norm is:

$$\begin{aligned}\|\Delta\mathbf{h}^{(l)}(x_i)\|_2 &= \alpha|(\mathbf{r}^{(l^*)})^\top\mathbf{h}^{(l)}(x_i)|\|\mathbf{r}^{(l^*)}\|_2\\ &= \alpha|(\mathbf{r}_{\mathcal{M}})^\top\mathbf{h}^{(l)}(x_i) + (\mathbf{r}_{other})^\top\mathbf{h}^{(l)}(x_i)|\sqrt{\|\mathbf{r}_{\mathcal{M}}\|_2^2 + \|\mathbf{r}_{other}\|_2^2}.\end{aligned}$$

The component from $\mathbf{r}_{other}$ can be isolated by considering its contribution:

$$\|\Delta\mathbf{h}^{(l)}(x_i)\|_2 \geq \alpha|(\mathbf{r}_{other})^\top\mathbf{h}^{(l)}(x_i)|\|\mathbf{r}_{other}\|_2,$$

when the $\mathbf{r}_{other}$ term is significant. Propagate to layer $l+1$:

$$\Delta\mathbf{h}^{(l+1)}(x_i) = \mathbf{h}^{(l+1)'}(x_i) - \mathbf{h}^{(l+1)}(x_i) \approx \sigma'\left(\mathbf{W}^{(l+1)}\text{Attn}'(\mathbf{h}^{(l)}(x_i))\Delta\mathbf{h}^{(l)}(x_i)\right),$$

where Attn$'$ and σ' are the Jacobians of attention and GeLU. The attention softmax and GeLU have minimum amplification factors $\gamma_{\text{attn}}, \gamma_\sigma > 0$, and the linear transformation by $\mathbf{W}^{(l+1)}$ satisfies:

$$\|\mathbf{W}^{(l+1)}\mathbf{x}\|_2 \geq \sigma_{\min}(\mathbf{W}^{(l+1)})\|\mathbf{x}\|_2.$$

Thus:

$$\|\Delta\mathbf{h}^{(l+1)}(x_i)\|_2 \geq \gamma_{\text{attn}}\gamma_\sigma\sigma_{\min}(\mathbf{W}^{(l+1)})\|\Delta\mathbf{h}^{(l)}(x_i)\|_2.$$

Focusing on the $\mathbf{r}_{other}$ contribution:

$$\|\Delta\mathbf{h}^{(l+1)}(x_i)\|_2 \geq \gamma_{\text{attn}}\gamma_\sigma\sigma_{\min}(\mathbf{W}^{(l+1)})\alpha|(\mathbf{r}_{other})^\top\mathbf{h}^{(l)}(x_i)|\|\mathbf{r}_{other}\|_2.$$

The mean shift norm is:

$$\|\Delta\boldsymbol{\mu}^{(l+1)}\|_2 = \left\|\frac{1}{N}\sum_{i=1}^{N}\Delta\mathbf{h}^{(l+1)}(x_i)\right\|_2.$$

Assume the layer-wise propagation amplifies the previous shift by $\gamma > 1$, reflecting attention and non-linear effects across layers. Combining the amplification of the existing shift and the new $\mathbf{r}_{other}$ contribution:

$$\|\Delta\boldsymbol{\mu}^{(l+1)}\|_2 \geq \gamma\|\Delta\boldsymbol{\mu}^{(l)}\|_2 + \alpha\gamma_{\text{attn}}\gamma_\sigma\sigma_{\min}(\mathbf{W}^{(l+1)})|(\mathbf{r}_{other})^\top\mathbf{h}^{(l)}(x_i)|\|\mathbf{r}_{other}\|_2,$$

matching Eq. (8). This shows that the $\mathbf{r}_{other}$ component causes layer-wise amplification through both the accumulated shift (first term) and the direct contribution at each layer (second term).

Step 4: Analyze the amplification mechanism. The amplification factors $\gamma > 1$, $\gamma_{\text{attn}}, \gamma_\sigma > 0$, and non-zero $\sigma_{\min}(\mathbf{W}^{(l+1)})$ ensure that perturbations from $\mathbf{r}_{other}$ grow across layers. The first term $\gamma\|\Delta\boldsymbol{\mu}^{(l)}\|_2$ represents the propagation of accumulated shift, while the second term represents the fresh perturbation introduced at layer $l+1$ due to $\mathbf{r}_{other}$. This dual mechanism ensures the shift grows across layers, disrupting the model's normal abilities. □

C Hyperparameter Tuning

In this section, we present the results of tuning the intervention strength α across four models: DeepSeek-R1-Distill-Qwen-1.5B, DeepSeek-R1-Distill-Qwen-7B, DeepSeek-R1-Distill-Llama-8B, and DeepSeek-R1-Distill-Qwen-14B on MATH500 [20]. As shown in Fig. 7, to achieve an optimal balance between efficiency and accuracy, we ultimately select $\alpha = 0.7$ for R1-1.5B, $\alpha = 0.3$ for R1-7B, $\alpha = 0.5$ for R1-8B, and $\alpha = 0.3$ for R1-14B.

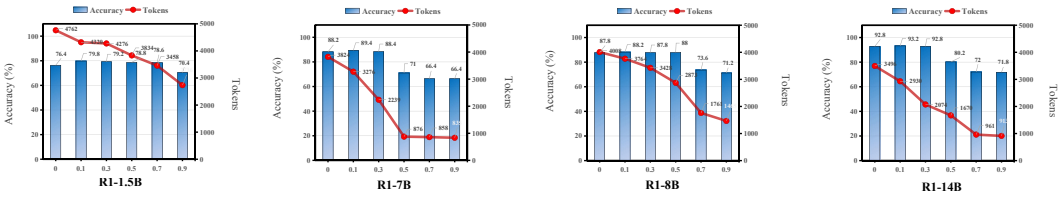


Figure 7: Impact of intervention strength α on the token reduction and accuracy of R1-1.5B, R1-7B, R1-8B, and R1-14B on the MATH500 dataset

D Layer Selection for Manifold Steering

The selection of intervention layers is critical for the effectiveness of Manifold Steering. We conduct a layer-wise analysis across multiple model sizes to determine the optimal intervention points. As shown in the tables below, we evaluate the performance across different layers by measuring accuracy and tokens on the MATH500. The results demonstrate that later layers consistently achieve better performance: Layer 27 for R1-1.5B and R1-7B, Layer 31 for R1-8B, and Layer 47 for R1-14B.

Table 2: Layer-wise performance analysis for R1-1.5B on MATH500.

	Vanilla	**Layer 1**	**Layer 5**	**Layer 10**	**Layer 15**	**Layer 20**	**Layer 25**	**Layer 27**
Accuracy (%)	76.4	76.6	77.0	76.4	57.6	74.8	67.4	78.6
# Tokens	4762	4472	4434	4223	1469	3930	1179	3458

Table 3: Layer-wise performance analysis for R1-7B on MATH500.

	Vanilla	**Layer 1**	**Layer 5**	**Layer 10**	**Layer 15**	**Layer 20**	**Layer 25**	**Layer 27**
Accuracy (%)	88.2	88.4	88.0	88.2	84.4	80.6	72.2	88.4
# Tokens	3824	3685	3665	3701	2713	1906	1070	2239

Table 4: Layer-wise performance analysis for R1-8B on MATH500.

	Vanilla	**Layer 1**	**Layer 5**	**Layer 10**	**Layer 15**	**Layer 20**	**Layer 25**	**Layer 30**	**Layer 31**
Accuracy (%)	87.8	87.2	87.8	88.2	75.6	86.4	71.8	87.6	88.0
# Tokens	4009	3896	3820	3654	2950	3280	1856	2975	2873

Table 5: Layer-wise performance analysis for R1-14B on MATH500.

	Vanilla	**Layer 1**	**Layer 5**	**Layer 10**	**Layer 15**	**Layer 20**	**Layer 25**	**Layer 30**	**Layer 35**	**Layer 40**	**Layer 45**	**Layer 47**
Accuracy (%)	92.8	92.4	92.4	92.0	92.2	92.6	89.8	80.4	87.4	84.6	82.4	92.8
# Tokens	3496	3384	3420	3095	2958	2857	2398	1814	2207	1836	1625	2074

E Time Latency Analysis

In this section, we analyze the time latency for the DeepSeek-R1-Distill-Qwen-7B model on the Math500 dataset [20], comparing our approach with Dynasor [16] and SEAL [8]. All experiments are conducted on an Ubuntu 22.04 system with A800 GPUs. We find that Dynasor exhibits the

significantly longest time latency, which is reasonable due to its frequent probing of intermediate states and its unsuitability for parallel processing of large reasoning models. For SEAL, although both SEAL and our method introduce negligible additional computational cost, SEAL's token reduction is less effective than ours, resulting in higher time latency.

Table 6: Average Time Latency on Math500 for different overthinking-mitigation methods in R1-7B.

Methods	Original	Dynasor	SEAL	Ours
Time Latency (s)	1.74	39.89	1.37	1.05