

Contrastive Neuron Regulation for Chain-of-Thought Reasoning

Zitao Su^{1,2} Xinyu Tang^{1,2} Wayne Xin Zhao^{1,2*}

¹Gaoling School of Artificial Intelligence, Renmin University of China

²Beijing Key Laboratory of Research on Large Models and Intelligent Governance
suzitao@ruc.edu.cn, batmanfly@gmail.com

Abstract

Chain-of-thought (CoT) reasoning is commonly improved through external mechanisms such as prompting, sampling, and decoding, while the internal units shaping reasoning trajectories remain less understood. We study CoT from the perspective of reasoning trajectory structure and find that incorrect trajectories are more redundant and repetitive, whereas correct ones are more lexically diverse and information-dense. Motivated by this observation, we propose Contrastive Neuron Regulation (CNR), a sparse, backbone-frozen framework for neuron-level regulation of CoT reasoning. CNR constructs correctness-controlled expanded-concise pairs of correct trajectories, identifies MLP neurons associated with the expansion contrast, and sparsely suppresses them during generation. We consider both a fixed-coefficient inference-time intervention and an adaptive variant that learns neuron-specific suppression coefficients on a subject-specific development split while keeping the backbone frozen. Across two reasoning models and seven MATH subjects, CNR improves macro-average accuracy over the no-intervention baseline by 2.6 and 3.2 percentage points, respectively, while intervening on less than 1% of MLP neurons. The code is available at <https://github.com/RUCAIBox/Contrastive-Neuron-Regulation>.

1 Introduction

Chain-of-thought (CoT) reasoning improves LLM performance on complex reasoning tasks by eliciting intermediate steps (Wei et al., 2022; Kojima et al., 2022). Most CoT methods rely on external inference-time mechanisms, including prompting (Zhou et al., 2023; Wang et al., 2023a; Jin and Lu, 2025), sampling and aggregation (Wang et al., 2023b), and search-based decoding (Yao et al., 2023; Besta et al., 2024). These methods control

how reasoning is elicited or selected, but they do not expose the internal units that shape how a reasoning trajectory unfolds. Recent work analyzes CoT-related hidden states (Mehrafarin et al., 2026; Zhang et al., 2025; Yuan et al., 2026), sparse features (Chen et al., 2026), and activations (Li et al., 2025), but sparse neuron-level regulation of CoT reasoning remains relatively underexplored.

In this paper, we study CoT through the lens of *reasoning trajectory structure*: how a reasoning trajectory progresses, repeats, expands, and accumulates information before producing a final answer. Final-answer correctness is an outcome-level property, while reasoning expansion is a process-level property. Our preliminary analysis shows that incorrect-answer trajectories are more redundant and repetitive, whereas correct-answer trajectories are more lexically diverse and information-dense. These observations suggest that answer quality is related to trajectory structure. However, explicit prompts targeting conciseness, redundancy, or verification do not provide a consistently reliable control rule. This suggests that surface-level instructions may not reliably shape trajectory structure.

We therefore turn to internal intervention, where the key challenge is to identify neurons associated with how reasoning trajectories unfold. A direct correct-incorrect contrast is less suitable for neuron discovery because observed trajectory-level structural differences may be entangled with task difficulty, heterogeneous error modes, and final-answer correctness. We instead use correctness-controlled expanded-concise trajectory pairs, where both trajectories solve the same problem correctly but differ in how extensively their reasoning unfolds.

Motivated by this, we propose *Contrastive Neuron Regulation* (CNR), a sparse, backbone-frozen framework for regulating CoT reasoning through neuron-level intervention. CNR scores MLP neurons by contrastive activation differences between expanded and concise correct trajectories, selects

*Corresponding author.

neurons associated with this expansion contrast, and sparsely suppresses them during generation. We consider both a fixed-coefficient inference-time intervention after neuron discovery and an adaptive variant that calibrates neuron-specific suppression coefficients on a subject-specific development split while keeping the backbone frozen. We evaluate CNR on two reasoning models using the MATH dataset. CNR improves macro-average accuracy over BASE by 2.6 percentage points on DeepSeek-R1-Distill-Qwen-1.5B and 3.2 points on Qwen2.5-7B. Compared with the strongest non-CNR single-pass control, RAND, CNR improves macro-average accuracy by 1.4 and 2.7 points, respectively, while intervening on less than 1% of MLP neurons. Our contributions are summarized as follows:

- We introduce a structural view of CoT trajectories that considers outcome-level correctness alongside process-level reasoning expansion.
- We propose *Contrastive Neuron Regulation* (CNR), which identifies MLP neurons associated with correctness-controlled expansion contrasts and regulates them through sparse multiplicative suppression with fixed or learned coefficients.
- We show that sparse neuron-level regulation improves mathematical reasoning on the MATH dataset, outperforming explicit prompt controls and matched sparse neuron-selection controls in macro-average accuracy.

2 Related Work

Chain-of-thought reasoning and external control. Chain-of-thought (CoT) prompting improves LLM reasoning by eliciting intermediate steps (Wei et al., 2022; Kojima et al., 2022). Existing CoT improvements mainly rely on external inference-time mechanisms, including structured prompting (Zhou et al., 2023; Wang et al., 2023a; Jin and Lu, 2025), sampling and aggregation (Wang et al., 2023b), search over reasoning paths (Yao et al., 2023; Besta et al., 2024), and explicit reasoning compression or length control (Sui et al., 2026; Waheed et al., 2025). These methods control prompts, samples, decoding procedures, or output length, but do not directly regulate the model’s internal reasoning computation. CNR instead uses naturally varying CoT trajectories to identify sparse neuron-level targets for inference-time regulation.

Internal analysis, activation steering, and neuron-level intervention. Recent work analyzes internal CoT computation with sparse fea-

tures, hidden-state probes, reasoning-state analyses, and error-awareness signals (Chen et al., 2026; Li et al., 2025; Mehrafarin et al., 2026). Activation-steering methods modify hidden representations to influence model behavior (Zou et al., 2023; Valentino et al., 2026; Zhou et al., 2026), while mechanistic studies show that Transformer MLP units can store or mediate task-relevant information (Geva et al., 2021; Meng et al., 2022). CNR builds on this internal-control perspective but uses a different contrast and intervention method. Instead of relying on correctness differences or dense steering directions, CNR constructs CoT trajectory pairs, identifies MLP neurons associated with reasoning expansion, and regulates them through sparse multiplicative suppression.

3 Preliminary

In this section, we first compare correct and incorrect trajectories, then test explicit prompt control, and finally motivate correctness-controlled expansion contrasts for neuron identification.

3.1 Structural Differences Between Correct and Incorrect Trajectories

To examine whether reasoning success is associated with structural properties of CoT trajectories, we compare trajectories with correct and incorrect final answers. For the same problem, an LLM may produce trajectories that differ in local progression, repeated phrasing, redundancy, expansion, and information density. We refer to these observable properties collectively as *trajectory structure*.

We analyze sampled MATH reasoning trajectories from the same subject categories used in our experiments (Hendrycks et al., 2021). Each trajectory is generated with a step-by-step prompt and is required to end with a boxed answer. We determine correctness by extracting the final answer and comparing it with the reference answer. To focus on the reasoning process rather than answer formatting, we compute all trajectory metrics over the CoT span before the final boxed answer.

To characterize trajectory structure, we use three surface-level metrics. Adjacent sentence similarity measures local redundancy between consecutive reasoning steps. Bigram repeat rate measures repeated local phrasing. Unique token ratio measures lexical diversity and information density. Within each subject, we compare correct-answer and incorrect-answer trajectories using two-sided

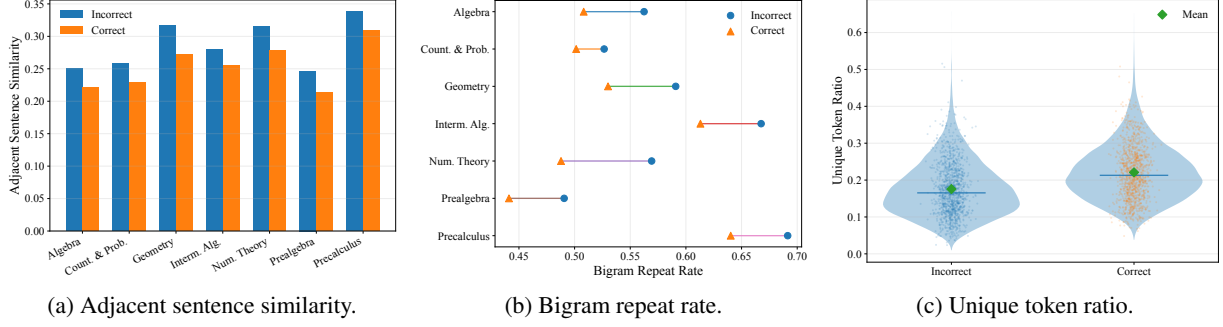


Figure 1: Structural differences between correct and incorrect trajectories.

Mann–Whitney U tests. Details of answer extraction, CoT-span definition, trajectory metrics, and statistical testing are provided in Appendix A.

Figure 1 shows that correct-answer and incorrect-answer trajectories exhibit consistent structural differences. Incorrect-answer trajectories tend to have higher adjacent sentence similarity and bigram repeat rate, indicating stronger local redundancy and repetition. Correct-answer trajectories tend to have higher unique token ratio, indicating greater lexical diversity and denser information content.

3.2 Explicit Control of Observed Trajectory Patterns

The structural differences above suggest a direct intervention question: *Can we improve reasoning by explicitly instructing the model to reduce the observed surface patterns?* To test this, we evaluate prompt-based controls targeting the trajectory properties identified in Section 3.1.

Specifically, CON, which controls over-elaboration, encourages shorter and more direct reasoning; ANTIRED, which controls repetition and local redundancy, discourages repeated reasoning steps; VERIFY, which controls unchecked intermediate reasoning, asks the model to verify key steps before answering; and ANTIREDVER, which jointly controls redundancy and unchecked reasoning, combines anti-redundancy instructions with verification. The full prompt templates are provided in Appendix B.1.

We report the accuracy and CoT-length results of the explicit prompt-control baselines used in this paper on DeepSeek-R1-Distill-Qwen-1.5B (DS-1.5B) and Qwen2.5-7B (QWEN-7B).

As shown in Table 1 and Appendix B.2, explicit prompting changes trajectory behavior and sometimes improves accuracy, but its effects are inconsistent across models and subjects. CON, ANTIRED, VERIFY, and ANTIREDVER each help

Model	Setting	Acc.	CoT
DS-1.5B	BASE	65.6	4035.0
	CON	65.8	3160.9
	ANTIRED	64.0	3325.0
	VERIFY	57.7	4313.3
	ANTIREDVER	60.5	3502.7
QWEN-7B	BASE	58.8	763.8
	CON	57.5	760.6
	ANTIRED	58.5	744.4
	VERIFY	60.0	775.3
	ANTIREDVER	58.5	740.4

Table 1: Macro-average results of explicit prompt controls across MATH subjects.

only under limited settings and can also reduce accuracy despite changing CoT length. Thus, the observed correct–incorrect trajectory patterns are informative, but they do not directly yield a reliable external prompt rule.

3.3 Correctness-Controlled Trajectory-Expansion Contrast

The explicit-control results show that external prompts do not reliably improve answer quality through surface-level trajectory instructions. To move beyond surface-level instructions, CNR aims to regulate how the model internally unfolds intermediate reasoning during generation.

To identify internal units relevant to this unfolding process, we focus on reasoning expansion as a controllable trajectory property. A direct correct–incorrect contrast is less suitable for neuron discovery because observed trajectory-level structural differences may be entangled with final-answer correctness, task difficulty, and heterogeneous error modes. Using such a contrast for neuron discovery could select neurons related to failure or correctness rather than to the expansion behavior we intend to regulate.

To reduce these confounds, CNR constructs

correctness-controlled trajectory-expansion contrasts. For the same problem, we compare two trajectories that both reach the correct final answer but differ in how extensively their reasoning unfolds. This keeps the input and final-answer correctness fixed while reducing confounding from task difficulty and heterogeneous error modes.

4 Methodology

As discussed in Section 3, correct–incorrect trajectory analysis reveals an association between answer quality and trajectory structure, while explicit prompt control is unreliable for regulating CoT quality. Building on this motivation, we propose *Contrastive Neuron Regulation* (CNR), a sparse, backbone-frozen framework for regulating CoT reasoning through neuron-level intervention.

As shown in Figure 2, CNR constructs expanded–concise correct pairs, extracts reasoning-span MLP activations, scores neurons by contrastive activation differences, applies sparse suppressive scaling during generation, and, in its adaptive variant, calibrates neuron-specific coefficients while keeping the backbone frozen.

4.1 Correctness-Controlled Pair Construction

CNR constructs trajectory pairs from correct solutions to the same problem to control for final-answer correctness while contrasting different degrees of reasoning expansion. For each problem x_i , we sample multiple CoT outputs and retain the correct trajectories as $\mathcal{Y}_i^{\text{corr}}$. Let $\ell(y)$ denote the number of tokens in the reasoning span before the final boxed answer.

To form a high-contrast expansion pair, we select the most expanded and most concise correct trajectories according to reasoning-span length:

$$y_i^{\text{exp}} = \arg \max_{y \in \mathcal{Y}_i^{\text{corr}}} \ell(y), \quad (1)$$

$$y_i^{\text{con}} = \arg \min_{y \in \mathcal{Y}_i^{\text{corr}}} \ell(y), \quad (2)$$

where y_i^{exp} denotes the expanded correct trajectory and y_i^{con} denotes the concise correct trajectory for the same problem.

To ensure a sufficient expansion gap, we introduce a minimum expansion-ratio threshold τ and retain the pair only when $\ell(y_i^{\text{exp}})/\ell(y_i^{\text{con}}) \geq \tau$.

The retained pairs provide correctness-controlled expansion contrasts for activation extraction and neuron identification. Trajectory

sampling details are provided in Appendix C.1, and pair-retention statistics are reported in Appendix E.

4.2 Reasoning-Span Activation Extraction

To compare internal activation patterns across expanded and concise reasoning trajectories, we extract MLP activations from the reasoning span of each retained pair $(y_i^{\text{exp}}, y_i^{\text{con}})$. Both trajectories are fed into the same frozen model, so the measured activation differences reflect differences between the paired trajectories rather than parameter updates. Let $\mathbf{h}_{l,t} \in \mathbb{R}^{d_{\text{H}}}$ denote the MLP activation vector at layer l and reasoning position t .

Since expanded and concise trajectories have different token lengths, we aggregate token-level activations into neuron-level statistics. For neuron j in layer l , we compute mean activation $\mu_{l,j}$, maximum activation $m_{l,j}$, and standard deviation $\sigma_{l,j}$ over the reasoning span. These statistics summarize average involvement, peak response, and activation variability during reasoning.

For each activation summary $\text{stat} \in \{\mu, m, \sigma\}$, we compute the corresponding paired expansion difference for each neuron:

$$\Delta_{i,l,j}^{\text{stat}} = \text{stat}_{l,j}(y_i^{\text{exp}}) - \text{stat}_{l,j}(y_i^{\text{con}}). \quad (3)$$

A positive value indicates that neuron (l, j) is more active in the expanded correct trajectory than in the concise correct trajectory for the same problem. These paired differences provide the activation-level evidence in our framework for identifying reasoning-expansion-associated neurons. The layer-wise neuron-selection budget is provided in Appendix C.2.

4.3 Reasoning-Expansion-Associated Neuron Identification

To select neurons for intervention, CNR converts the paired activation differences into a layer-wise neuron ranking. A single criterion is insufficient: the mean difference can be driven by noisy shifts, effect size accounts for variability, consistency prevents selection based on a few outlier pairs, the paired-test p -value provides a statistical reliability signal, and correlation with the expansion gap checks whether activation differences vary systematically with the expansion contrast. We therefore combine these complementary signals into a standardized linear score, following activation-based neuron analysis and intervention work that uses both activation magnitude and reliability to iden-

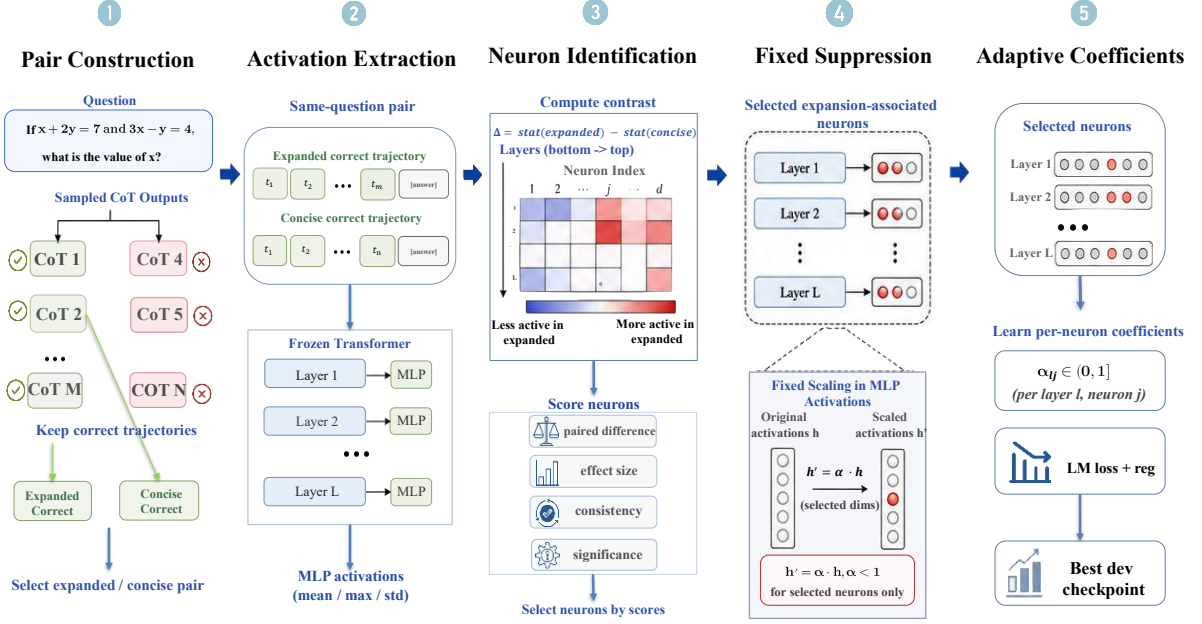


Figure 2: Overview of Contrastive Neuron Regulation. CNR constructs correctness-controlled expanded–concise pairs, selects neurons associated with the expansion contrast, and regulates them through sparse suppression.

tify functionally relevant units (Geva et al., 2021; Meng et al., 2022; Zou et al., 2023).

For each neuron (l, j) , we define the contrastive activation score:

$$\begin{aligned} \text{CAS}_{l,j} &= w_1 z(\bar{\Delta}_{l,j}) - w_2 z(\text{ES}_{l,j}) - w_3 z(\text{cons}_{l,j}) \\ &\quad - w_4 z(-\log_{10} p_{l,j}) - w_5 z(\max(\rho_{l,j}, 0)). \end{aligned} \quad (4)$$

where $\bar{\Delta}_{l,j}$ is the mean paired activation difference, $\text{ES}_{l,j}$ is the paired effect size, $\text{cons}_{l,j}$ is the fraction of pairs with positive activation differences, $p_{l,j}$ is the paired-test p -value, and $\rho_{l,j}$ is the correlation between activation difference and expansion gap. The paired effect size is computed as the mean paired activation difference normalized by the standard deviation of the paired differences. The p -value is obtained from a two-sided one-sample t -test of the paired activation differences against zero, and $\rho_{l,j}$ is computed as the Pearson correlation between the activation difference and the corresponding reasoning-length gap. The operator $z(\cdot)$ denotes layer-wise standardization, which makes heterogeneous score components comparable before linear combination.

We use component weights $w_1 = 0.30$, $w_2 = 0.25$, $w_3 = 0.25$, $w_4 = 0.10$, and $w_5 = 0.10$ for mean activation difference, paired effect size, directional consistency, statistical reliability, and

length-gap correlation, respectively. For each neuron, we compute CAS separately from the mean-, maximum-, and standard-deviation-based activation statistics, and combine the three resulting scores with weights 0.5, 0.3, and 0.2, respectively.

To ensure that selected neurons are directionally aligned with the expanded side of the contrast, we apply threshold filters before taking the top-ranked neurons in each layer:

$$\mathcal{N}_l = \{j : \bar{\Delta}_{l,j} \geq \Delta_{\min}, \text{ES}_{l,j} \geq \text{ES}_{\min}, \text{cons}_{l,j} \geq \text{cons}_{\min}, p_{l,j} \leq p_{\max}\}. \quad (5)$$

We use $\Delta_{\min} = 0.01$, $\text{ES}_{\min} = 0.20$, $\text{cons}_{\min} = 0.65$, and $p_{\max} = 0.05$. The resulting candidate set $\mathcal{N}_l$ contains neurons that are consistently and reliably more active on the expanded side of the correctness-controlled contrast. Among the neurons that satisfy these filters, we retain the highest-scoring neurons in each layer according to the fixed layer-wise budget described in Appendix C.2.

4.4 Fixed Suppressive Inference-Time Regulation

To test whether regulating the selected neurons affects generation behavior, fixed-coefficient CNR intervenes on their activations during inference (Meng et al., 2022; Zou et al., 2023). For each

selected neuron $j \in \mathcal{N}_l$, we apply a multiplicative suppression coefficient $\alpha \in [0, 1]$:

$$h'_{l,t,j} = \begin{cases} \alpha h_{l,t,j}, & j \in \mathcal{N}_l, \\ h_{l,t,j}, & j \notin \mathcal{N}_l, \end{cases}$$

where $\alpha = 1$ recovers the no-intervention baseline, while smaller values impose stronger suppression. The intervention is applied at every generation step but only to the selected neurons, leaving all backbone parameters unchanged. This fixed-coefficient setting serves as a diagnostic version of CNR for testing how suppression strength affects accuracy and CoT length. The intervention itself is inference-time only after the data-driven neuron-selection stage. Fixed- α sweep settings are described in Appendix C.3, and full diagnostic sweep results are reported in Appendix D.

4.5 Adaptive Coefficient Learning

Fixed suppression uses the same coefficient for all selected neurons, although different selected neurons may require different suppression strengths. To allow finer-grained regulation, adaptive CNR calibrates neuron-specific suppression coefficients on a development split while keeping the backbone model frozen. Unlike fixed-coefficient CNR, the adaptive variant includes a lightweight coefficient-calibration stage before test-time evaluation while keeping all backbone parameters frozen.

For each selected neuron (l, j) , we introduce a trainable scalar $r_{l,j}$ during adaptation and map it to a bounded coefficient:

$$\alpha_{l,j} = \alpha_{\min} + (\alpha_{\max} - \alpha_{\min})\sigma(r_{l,j}),$$

where $\alpha_{\min}$ and $\alpha_{\max}$ define the allowed suppression range, and the sigmoid mapping keeps each coefficient within this range. The intervention then becomes $h'_{l,t,j} = \alpha_{l,j}h_{l,t,j}$ for $j \in \mathcal{N}_l$. The initial coefficient α_{init} controls the starting suppression strength before learning.

Only the coefficient parameters $\{r_{l,j}\}$ are optimized. During training, the intervention is active in the forward pass, and the language-modeling loss is computed under teacher forcing on reference solutions. Prompt tokens are masked out so that the objective focuses on solution generation rather than prompt reconstruction. For each subject, we hold out a development split for checkpoint selection.

The training objective is

$$\mathcal{L} = \mathcal{L}_{\text{LM}} + \lambda \mathcal{L}_{\text{reg}},$$

where $\mathcal{L}_{\text{LM}}$ measures solution-token prediction quality and $\mathcal{L}_{\text{reg}}$ discourages excessive deviation from the initial coefficient:

$$\mathcal{L}_{\text{reg}} = \frac{1}{|\mathcal{L}_{\text{sel}}|} \sum_{l \in \mathcal{L}_{\text{sel}}} \frac{1}{|\mathcal{N}_l|} \sum_{j \in \mathcal{N}_l} (\alpha_{l,j} - \alpha_{\text{init}})^2,$$

where $\mathcal{L}_{\text{sel}}$ is the set of layers containing selected neurons, and λ controls the strength of the coefficient regularization. After training, we select the checkpoint with the lowest development-set negative log-likelihood and apply the learned coefficients for inference at test time. Adaptive-learning details and hyperparameters are provided in Appendix C.4 and Table 10.

5 Experiments

In this section, we first describe the experimental setup, then report the main results and conduct detailed analyses.

5.1 Experimental Setup

Datasets. We evaluate on the MATH dataset (Hendrycks et al., 2021), covering seven subjects: *algebra*, *counting_and_probability*, *geometry*, *intermediate_algebra*, *number_theory*, *prealgebra*, and *precalculus*. Neuron identification is performed on the training split, and final results are reported on the test split. For adaptive coefficient learning, we specifically hold out 10% of the training data as a subject-specific development split.

Models. We use two open-weight reasoning models: DeepSeek-R1-Distill-Qwen-1.5B (DeepSeek-AI, 2025) and Qwen2.5-7B (Yang et al., 2024). We abbreviate them as DS-1.5B and QWEN-7B, respectively. All backbone parameters remain frozen; the adaptive variant optimizes only neuron-level suppression coefficients.

Trajectory construction and analysis. For the preliminary trajectory-level analysis, we sample 8 CoT outputs per problem with temperature 0.7, top- p 0.95, maximum generation length 2048, and a fixed random seed. Correctness is determined by extracting and normalizing the final boxed answer, and trajectory metrics are computed over the reasoning span before the final answer.

For CNR neuron discovery, we independently sample training trajectories using a maximum generation length of 8192. We retain only trajectories

Model	Setting	Alg.	C&P	Geo.	IAlg.	NTh.	PAlg.	PCalc.	Avg.
DS-1.5B	BASE	85.2	55.9	58.7	55.1	67.4	78.1	59.0	65.6
	CON	85.2	59.1	55.7	59.4	71.1	74.4	55.5	65.8
	ANTIRED	82.5	60.6	56.6	56.6	70.0	71.0	50.7	64.0
	VERIFY	76.7	50.0	50.9	51.7	60.4	65.6	48.9	57.7
	ANTIREDVER	76.1	59.3	51.8	54.7	65.4	64.6	51.6	60.5
	RAND	85.7	58.6	60.0	57.6	69.2	77.6	59.0	66.8
	ACT	71.0	32.1	60.5	60.9	47.2	40.5	46.2	51.2
	CNR	89.5	59.3	61.4	59.7	66.8	81.1	59.7	68.2
QWEN-7B	BASE	79.0	54.0	50.5	48.6	60.2	68.7	50.4	58.8
	CON	81.1	52.7	46.8	45.6	61.3	68.2	46.5	57.5
	ANTIRED	81.4	51.7	46.8	47.7	63.9	72.0	46.3	58.5
	VERIFY	82.7	57.0	45.7	47.6	65.4	71.8	49.6	60.0
	ANTIREDVER	81.3	51.3	45.7	49.3	63.3	69.9	48.9	58.5
	RAND	81.9	51.7	50.5	44.8	62.5	74.8	48.7	59.3
	ACT	83.8	53.6	50.5	21.5	59.1	69.6	45.1	54.7
	CNR	81.9	58.0	48.9	49.5	65.7	78.4	51.8	62.0

Table 2: Subject-wise single-pass accuracy on MATH. CON, ANTIRED, VERIFY, and ANTIREDVER are explicit prompt controls; RAND and ACT are sparse neuron-selection controls. RAND reports the mean over five random seeds.

whose final answers are correct and construct same-question expanded–concise pairs from them. For each problem, the longest and shortest correct trajectories are selected according to reasoning-span length, and the pair is retained only when its expansion ratio is at least $\tau = 1.5$.

Details of answer extraction and trajectory metrics are provided in Appendix A; trajectory-construction details and pair-retention statistics are provided in Appendices C and E.

Compared methods. BASE denotes the no-intervention greedy decoding setting. Our main reported CNR setting is the adaptive variant, which applies learned neuron-specific suppression coefficients to neurons selected by the correctness-controlled expansion contrast. We compare it with single-pass external and internal controls. External controls include CON, ANTIRED, VERIFY, and ANTIREDVER, which explicitly target conciseness, redundancy reduction, verification, or their combination. Internal sparse controls include RAND and ACT, which use matched random neurons and matched top-activation neurons. We further evaluate LENCAP and ACTSTEER: LENCAP tests hard output-length control, while ACTSTEER adapts activation-steering ideas to dense trajectory-expansion directions (Zou et al., 2023).

Implementation details. CNR selects 50 MLP neurons per layer for DS-1.5B and 80 MLP neurons per layer for QWEN-7B. Both models have 28 layers in our experiments. Adaptive coefficients

are learned in per-neuron mode with $\alpha_{\min} = 0$, $\alpha_{\max} = 1$, and $\alpha_{\text{init}} = 0.5$. We train the coefficient parameters for 3 epochs with training batch size 2, evaluation batch size 4, learning rate 10^{-2} , and regularization weight $\lambda = 10^{-3}$. The maximum sequence length is 8192, training uses bfloat16 precision, and the checkpoint with the lowest development-set negative log-likelihood is selected for test evaluation. Additional implementation details are provided in Appendix C.

Computational budget. Experiments used eight NVIDIA A800 80GB PCIe GPUs and approximately 800 GPU-hours in total, covering trajectory sampling, activation extraction, neuron identification, coefficient learning, diagnostic controls, and test evaluation.

5.2 Main Results

Table 2 reports subject-wise single-pass accuracy on MATH. The primary comparison is between BASE and CNR. CNR improves macro-averaged accuracy from 65.6 to 68.2 on DS-1.5B and from 58.8 to 62.0 on QWEN-7B, corresponding to gains of 2.6 and 3.2 percentage points. The gains are not uniform across subjects: relative to BASE, accuracy decreases from 67.4 to 66.8 on Number Theory for DS-1.5B and from 50.5 to 48.9 on Geometry for QWEN-7B. We therefore characterize the improvement as a macro-level gain rather than a uniform subject-wise improvement.

Compared with explicit prompt controls, including CON, ANTIRED, VERIFY, and ANTIREDVER,

Dataset	Base	CNR	Δ	α
MathVista	41.0	45.0	+4.0	0.3
MathVerse	51.0	52.0	+1.0	0.5
CLUE-SNLI	41.0	44.0	+3.0	0.7

Table 3: Out-of-domain fixed- α diagnostic on 200 sampled examples per dataset.

CNR achieves the best macro-average accuracy on both models. This suggests that directly instructing the model to be concise, reduce redundancy, or verify intermediate steps is less reliable than regulating expansion-associated internal units. Compared with sparse neuron-selection controls, including RAND and ACT, CNR also outperforms matched random-neuron suppression and top-activation-neuron suppression. This indicates that the correctness-controlled trajectory-expansion signal is important for selecting effective neurons.

Detailed BASE–CNR comparisons, including CoT length and subject-wise deltas, are reported in Appendix F. Prompt templates and subject-wise explicit-control results are provided in Appendices B.1 and B.2. Sparse neuron-selection control details are provided in Appendix C.5.

To provide a lightweight check beyond MATH, we sample 200 examples from each of MathVista (Lu et al., 2024), MathVerse (Zhang et al., 2024), and CLUE-SNLI (Bowman et al., 2015; Xu et al., 2020). For each dataset, we re-run pair construction, activation extraction, contrastive neuron scoring, and sparse neuron selection, and then evaluate DS-1.5B using fixed- α CNR without adaptive coefficient learning. As shown in Table 3, CNR improves accuracy on all three sampled subsets. These results provide a lightweight indication that the CNR procedure can be applied beyond the original MATH subjects, although the evaluation remains small-scale and fixed- α performance may depend on the target dataset.

5.3 Ablation and Diagnostic Analysis

We conduct ablation and diagnostic analyses to test whether the gains of CNR can be explained by simpler control mechanisms. Specifically, we examine hard length capping, dense activation steering, sparse neuron-selection controls, and cross-subject neuron overlap. A further fixed- α CAS component sensitivity analysis is reported in Appendix D.2.

Model	Setting	Acc.	CoT
DS-1.5B	BASE	65.6	4035.0
	CNR	68.2	3783.1
	LENCAP	66.7	3324.0
	ACTSTEER	68.0	3915.6
QWEN-7B	BASE	58.8	763.8
	CNR	62.0	712.0
	LENCAP	57.3	519.5
	ACTSTEER	58.7	730.2

Table 4: Macro-average results for length capping and dense activation steering.

5.3.1 Length Capping and Dense Activation Steering

Table 4 reports macro-average diagnostic results across subjects. LENCAP tests whether CNR works mainly by shortening outputs. ACTSTEER tests whether the same trajectory-expansion signal is sufficient when applied as a dense steering direction rather than sparse neuron-level regulation. Full subject-wise LENCAP results are provided in Appendix G, and full ACTSTEER results are provided in Appendix H.

LENCAP substantially reduces CoT length, especially on QWEN-7B, but it does not match CNR’s accuracy. This suggests that the effect of CNR is not reducible to hard output-length truncation. ACTSTEER performs competitively on DS-1.5B, but it does not improve over BASE on QWEN-7B. This indicates that the trajectory-expansion signal is informative, but dense steering does not consistently replace sparse neuron-level regulation.

5.3.2 Sparse Neuron-Selection Controls

The sparse controls in Table 2 test whether the effect comes from suppressing any matched neuron set. RAND uses the same layer-wise neuron budget as CNR but selects neurons randomly. ACT selects generally high-activation neurons instead of neurons identified by correctness-controlled trajectory-expansion contrasts. Their construction details are provided in Appendix C.5.

On both models, CNR achieves higher macro-average accuracy than RAND and ACT. These results suggest that the gains are not solely explained by arbitrary sparse suppression or by suppressing high-activation neurons. They further support the usefulness of the correctness-controlled expansion contrast for neuron selection.

To assess robustness to random neuron selection, we additionally evaluate RAND over five random seeds. Paired-bootstrap analysis shows that CNR

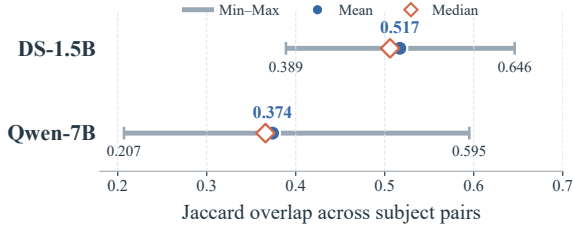


Figure 3: Subject-wise overlap of selected reasoning-expansion-associated neurons.

exceeds the five-seed RAND baseline by 1.40 percentage points on DS-1.5B (95% CI [0.35, 2.46], $p = 0.0076$) and by 2.73 points on QWEN-7B (95% CI [1.53, 3.92], $p < 10^{-4}$). Across the 14 model–subject comparisons, CNR outperforms the mean of the five RAND seeds in 11 cases and individual random-seed runs in 55 of 70 comparisons. A summary of the multi-seed comparison is provided in Appendix C.5.

To separate trajectory-guided neuron selection from coefficient learning, we also evaluate an Adaptive RAND control. Adaptive RAND uses the same development split, coefficient-learning objective, coefficient bounds, training epochs, and neuron budget as adaptive CNR, but replaces the CNR-selected neurons with random neurons. CNR improves over Adaptive RAND by 2.13 macro points on DS-1.5B and 3.99 points on QWEN-7B, and performs better in 13 of the 14 subject–model settings. The full subject-wise comparison is reported in Table 12 in Appendix C.5.

5.3.3 Subject-Wise Neuron Overlap

We further analyze whether selected reasoning-expansion-associated neurons are shared across MATH subjects. For each subject, we represent selected neurons as layer–neuron pairs and compute pairwise Jaccard overlap between subject-specific selected sets. Figure 3 summarizes the minimum, maximum, mean, and median overlap across subject pairs.

DS-1.5B shows higher mean overlap than QWEN-7B, suggesting that its selected expansion-associated neurons are more shared across subjects. QWEN-7B shows lower overlap, indicating stronger subject specificity. Thus, selected neurons are neither fully universal nor fully subject-specific. This pattern suggests that the selected neurons contain both shared and subject-specific structure. Additional overlap definitions and intersection statistics are provided in Appendix I.

6 Conclusion

In this paper, we presented Contrastive Neuron Regulation (CNR), a sparse neuron-level framework for controlling chain-of-thought reasoning by regulating MLP neurons associated with reasoning expansion. CNR constructs correctness-controlled expanded–concise trajectory pairs, identifies expansion-associated neurons, and learns suppressive coefficients while keeping the backbone frozen. Experiments on two reasoning models and seven MATH subjects showed that CNR improved macro-averaged accuracy over the no-intervention baseline and outperformed matched sparse neuron-selection controls while intervening on less than 1% of MLP neurons. Diagnostic analyses showed that these gains were not explained by explicit prompt control, hard length capping, dense activation steering, or arbitrary sparse suppression.

7 Limitations

CNR has several limitations. First, reasoning-expansion-associated neurons are identified through activation contrasts and validated by intervention, but this does not fully disentangle reasoning expansion from length-correlated generation dynamics or other discourse-level factors. Second, CNR relies on observable CoT trajectories, which may not perfectly reflect the model’s latent reasoning process. Third, correctness-controlled pair construction requires multiple correct trajectories with sufficient expansion variation, so neuron discovery may favor problems that the model can already solve in diverse ways. Finally, our main experiments focus on MATH and two Qwen-family reasoning models. Although we include a small out-of-domain diagnostic, broader validation across model families, larger models, and more diverse reasoning domains remains open. We also do not compare against the full range of data-driven lightweight adaptation baselines, such as adaptive ACT, development-tuned prompt controls, learned dense steering, PEFT/LoRA, or distillation. These comparisons would help more clearly characterize the advantages and limitations of CNR relative to alternative lightweight adaptation strategies.

Acknowledgments

This work was partially supported by National Natural Science Foundation of China under Grant No. 92470205, Beijing Natural Science Foundation under Grant No. L2611017, and Beijing Major Sci-

ence and Technology Project under Contract No. Z251100008425002.

References

- Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, and Torsten Hoeffler. 2024. [Graph of thoughts: Solving elaborate problems with large language models](#). In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2024, February 20-27, 2024, Vancouver, Canada*, pages 17682–17690. AAAI Press.
- Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. [A large annotated corpus for learning natural language inference](#). In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 632–642. The Association for Computational Linguistics.
- Xi Chen, Aske Plaat, and Niki van Stein. 2026. [How does chain of thought think? mechanistic interpretability of chain-of-thought reasoning with sparse autoencoding](#). In *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2026, Singapore, January 20-27, 2026*, pages 30297–30305. AAAI Press.
- DeepSeek-AI. 2025. [Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning](#). *CoRR*, abs/2501.12948.
- Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. [Transformer feed-forward layers are key-value memories](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 5484–5495. Association for Computational Linguistics.
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. [Measuring mathematical problem solving with the MATH dataset](#). In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.
- Ziqi Jin and Wei Lu. 2025. [Self-harmonized chain of thought](#). In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2025 - Volume 1: Long Papers, Albuquerque, New Mexico, USA, April 29 - May 4, 2025*, pages 1153–1174. Association for Computational Linguistics.
- Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. [Large language models are zero-shot reasoners](#). In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.
- Zihao Li, Xu Wang, Yuzhe Yang, Ziyu Yao, Haoyi Xiong, and Mengnan Du. 2025. [Feature extraction and steering for enhanced chain-of-thought reasoning in language models](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025, Suzhou, China, November 4-9, 2025*, pages 10893–10913. Association for Computational Linguistics.
- Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. 2024. [Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts](#). In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.
- Houman Mehrafarin, Amit Parekh, and Ioannis Konstas. 2026. [When chain-of-thought fails, the solution hides in the hidden states](#). *CoRR*, abs/2604.23351.
- Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. [Locating and editing factual associations in GPT](#). In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.
- Yi Sui, Chaozhuo Li, and Dawei Song. 2026. [Think less, know more: State-aware reasoning compression with knowledge guidance for efficient reasoning](#). *CoRR*, abs/2604.09150.
- Marco Valentino, Geonhee Kim, Dhairya Dalal, Zhixue Zhao, and André Freitas. 2026. [Mitigating content effects on reasoning in language models through fine-grained activation steering](#). In *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2026, Singapore, January 20-27, 2026*, pages 33314–33322. AAAI Press.
- Abdul Waheed, Chancharik Mitra, Laurie Z. Wang, Deva Ramanan, and Bhiksha Raj. 2025. [Less is more tokens: Efficient math reasoning via difficulty-aware chain-of-thought distillation](#). *CoRR*, abs/2509.05226.
- Boshi Wang, Sewon Min, Xiang Deng, Jiaming Shen, You Wu, Luke Zettlemoyer, and Huan Sun. 2023a. [Towards understanding chain-of-thought prompting](#).

- An empirical study of what matters. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL 2023, Toronto, Canada, July 9-14, 2023, pages 2717–2739. Association for Computational Linguistics.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023b. [Self-consistency improves chain of thought reasoning in language models](#). In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. [Chain-of-thought prompting elicits reasoning in large language models](#). In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.
- Liang Xu, Hai Hu, Xuanwei Zhang, Lu Li, Chenjie Cao, Yudong Li, Yechen Xu, Kai Sun, Dian Yu, Cong Yu, Yin Tian, Qianqian Dong, Weitang Liu, Bo Shi, Yiming Cui, Junyi Li, Jun Zeng, Rongzhao Wang, Weijian Xie, and 13 others. 2020. [CLUE: A chinese language understanding evaluation benchmark](#). In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 4762–4772. International Committee on Computational Linguistics.
- An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 22 others. 2024. [Qwen2.5 technical report](#). *CoRR*, abs/2412.15115.
- Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. [Tree of thoughts: Deliberate problem solving with large language models](#). In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.
- Aojie Yuan, Zhiyuan Julian Su, Haiyue Zhang, Yi Nian, and Yue Zhao. 2026. Hidden error awareness in chain-of-thought reasoning: The signal is diagnostic, not causal. *arXiv preprint arXiv:2605.09502*.
- Anqi Zhang, Yulin Chen, Jane Pan, Chen Zhao, Aurojit Panda, Jinyang Li, and He He. 2025. [Reasoning models know when they’re right: Probing hidden states for self-verification](#). *CoRR*, abs/2504.05419.
- Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Yu Qiao, Peng Gao, and Hongsheng Li. 2024. [MATHVERSE: does your multi-modal LLM truly see the diagrams in visual math problems?](#) In *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part VIII*, Lecture Notes in Computer Science, pages 169–186. Springer.
- Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V. Le, and Ed H. Chi. 2023. [Least-to-most prompting enables complex reasoning in large language models](#). In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.
- Jingjing Zhou, Gaoxiang Cong, Li Su, and Liang Li. 2026. [Star: Sensitive trajectory regulation for unlearning in large reasoning models](#). In *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2026, Singapore, January 20-27, 2026*, pages 35121–35129. AAAI Press.
- Andy Zou, Long Phan, Sarah Li Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, and 2 others. 2023. [Representation engineering: A top-down approach to AI transparency](#). *CoRR*, abs/2310.01405.

A Trajectory Metrics and Answer Extraction

This appendix provides the details of answer extraction, correctness evaluation, and trajectory-level metrics used in the preliminary analysis.

A.1 Answer Extraction and Correctness Evaluation

For each generated solution, we extract the model’s predicted final answer from the output text. We first search for the last `\boxed{}` expression and use its content as the predicted answer. If no boxed expression is found, we fall back to explicit answer patterns such as “final answer” or “answer”. If these patterns are also absent, we use the last numeric-like span in the generated output.

Both predicted and reference answers are normalized before comparison. The normalization removes common LaTeX wrappers, whitespace, trailing punctuation, and other superficial formatting differences. We then apply exact matching, numeric equivalence, and symbolic equivalence when possible. A generated trajectory is labeled as final-answer-correct if the normalized predicted answer matches the normalized reference answer under one of these matching rules.

For trajectory-level analysis, we define the CoT span as the generated text before the final boxed answer. If no boxed answer is found, we use the text before the extracted fallback answer when possible. All trajectory metrics below are computed only over this CoT span.

A.2 Trajectory Metrics

We use three surface-level metrics to characterize local structural properties of CoT trajectories. Although these metrics do not directly measure semantic or logical validity, they provide lightweight indicators of redundancy, repetition, and lexical diversity.

Unique token ratio. After lowercasing and tokenizing a trajectory into $w_1, \dots, w_T$, the unique token ratio is defined as

$$\text{UTR}(y) = \frac{|\{w_t : 1 \leq t \leq T\}|}{T}.$$

A higher value indicates greater lexical diversity.

Bigram repeat rate. Let

$$\mathcal{B}_2(y) = \{(w_t, w_{t+1})\}_{t=1}^{T-1}$$

denote the multiset of adjacent token bigrams in trajectory y . The bigram repeat rate is

$$\text{BRR}(y) = 1 - \frac{|\text{unique}(\mathcal{B}_2(y))|}{|\mathcal{B}_2(y)|}.$$

A higher value indicates stronger repeated local phrasing.

Adjacent sentence similarity. We split each CoT span into sentences and represent each sentence as a token-count vector. Let $s_1, \dots, s_K$ denote the extracted sentences, and let $c(s_k)$ be the token-count vector of sentence s_k . Adjacent sentence similarity is

$$\text{AdjSim}(y) = \frac{1}{K-1} \sum_{k=1}^{K-1} \cos(c(s_k), c(s_{k+1})).$$

A higher value indicates stronger local redundancy or weaker step-to-step progression.

A.3 Statistical Testing

For each MATH subject, we compare final-answer-correct and incorrect trajectories using the three metrics above. Because metric distributions are not assumed to be Gaussian, we use two-sided Mann–Whitney U tests to compare the two groups. The preliminary analysis reports subject-wise group trends and visual distributional summaries rather than using these tests as selection criteria for CNR neurons. Neuron identification in CNR instead uses correctness-controlled trajectory-expansion pairs, as described in Section 4.

B External Explicit-Control Baselines

This appendix reports additional details for the explicit prompt-control baselines discussed in Section 3.2. These baselines modify only the system instruction and do not intervene on internal activations. Macro-average results are reported in Table 1; here we provide the full prompt templates and subject-wise results.

B.1 Prompt Templates

• BASE

You are a helpful math assistant.
Solve the problem step by step and put your final answer in `\boxed{}`.

• CON

You are a concise mathematical reasoning assistant.
Solve the problem step by step, but keep the reasoning as short and direct as possible.
Put your final answer in `\boxed{}`.

- **ANTIRED**

You are a helpful math assistant.
Solve the problem step by step.
Avoid repeating the same idea, equation, or sentence.
Each step should add new information toward the solution.
Keep the reasoning concise but complete, and put your final answer in `\boxed{}`.

- **VERIFY**

You are a careful mathematical reasoning assistant.
Solve the problem step by step.
Before giving the final answer, verify the key steps and check whether the final result satisfies the problem conditions.
If you find an error, correct it.
Put your final answer in `\boxed{}`.

- **ANTIREDVER**

You are a careful mathematical reasoning assistant.
Solve the problem step by step.
Avoid repeated or redundant steps, and make sure each step adds useful information.
Before giving the final answer, verify the key steps and check whether the result satisfies the problem conditions.
Keep the reasoning concise but complete, and put your final answer in `\boxed{}`.

For models evaluated with chat formatting, the prompt is instantiated as follows.

```
<|im_start|>system
{system prompt}
<|im_end|>
<|im_start|>user
{problem}
<|im_end|>
<|im_start|>assistant
```

For non-chat formatting, the same system prompt and problem are placed under `### System`, `### User`, and `### Assistant` fields.

B.2 Subject-Wise Results

Tables 5–8 provide subject-wise accuracy and CoT-length results for the explicit prompt controls. Accuracy tables show whether explicit control improves final-answer quality. CoT-length tables show whether the prompts change trajectory behavior. We abbreviate ANTIREDVER as AR+VER in compact tables.

Setting	Alg.	C&P	Geo.	IAlg.	NTh.	PAlg.	PCalc.	Avg.
CON	85.2	59.1	55.7	59.4	71.1	74.4	55.5	65.8
ANTIRED	82.5	60.6	56.6	56.6	70.0	71.0	50.7	64.0
VERIFY	76.7	50.0	50.9	51.7	60.4	65.6	48.9	57.7
AR+VER	76.1	59.3	51.8	54.7	65.4	64.6	51.6	60.5

Continued on next column

Setting	Alg.	C&P	Geo.	IAlg.	NTh.	PAlg.	PCalc.	Avg.
---------	------	-----	------	-------	------	-------	--------	------

Table 5: DS-1.5B explicit-control accuracy.

Setting	Alg.	C&P	Geo.	IAlg.	NTh.	PAlg.	PCalc.	Avg.
CON	1723	3511	3786	4123	3140	1916	3927	3161
ANTIRED	1934	3568	3795	4347	3261	2114	4256	3325
VERIFY	2810	4916	4788	5082	4473	3274	4851	4313
AR+VER	2174	3658	4053	4659	3505	2254	4216	3503

Table 6: DS-1.5B explicit-control length.

Setting	Alg.	C&P	Geo.	IAlg.	NTh.	PAlg.	PCalc.	Avg.
CON	81.1	52.7	46.8	45.6	61.3	68.2	46.5	57.5
ANTIRED	81.4	51.7	46.8	47.7	63.9	72.0	46.3	58.5
VERIFY	82.7	57.0	45.7	47.6	65.4	71.8	49.6	60.0
AR+VER	81.3	51.3	45.7	49.3	63.3	69.9	48.9	58.5

Table 7: Qwen2.5-7B explicit-control accuracy.

Setting	Alg.	C&P	Geo.	IAlg.	NTh.	PAlg.	PCalc.	Avg.
CON	580	597	655	1072	806	651	964	761
ANTIRED	521	583	708	1177	746	405	1072	744
VERIFY	583	612	748	1168	912	459	945	775
AR+VER	569	569	655	1139	856	401	995	740

Table 8: Qwen2.5-7B explicit-control CoT length.

C Implementation Details and Hyperparameters

This appendix summarizes the implementation details and hyperparameters used for trajectory construction, neuron identification, inference-time intervention, adaptive coefficient learning, and neuron-selection controls.

C.1 Trajectory Sampling

For the preliminary correct–incorrect trajectory analysis, we sample 8 CoT outputs per problem with temperature 0.7, top- p 0.95, maximum generation length 2048, and a fixed random seed. These samples are used only for the trajectory-level analysis described in Section 3.

For CNR neuron discovery, we independently sample multiple CoT trajectories from each MATH training problem using a maximum generation length of 8192. We retain only trajectories whose final answers are correct and construct same-question trajectory-expansion pairs from these correct trajectories. For each retained problem, the expanded trajectory is the longest correct trajectory and the concise trajectory is the shortest correct trajectory according to reasoning-span length. A

pair is retained only when

$$\frac{\ell(y_i^{\text{exp}})}{\ell(y_i^{\text{con}})} \geq \tau,$$

where $\tau = 1.5$.

C.2 Neuron Selection Budget

CNR uses a sparse layer-wise neuron selection budget. For DeepSeek-R1-Distill-Qwen-1.5B, we select 50 MLP neurons per layer. For Qwen2.5-7B, we select 80 MLP neurons per layer. Both models have 28 layers in our experiments. This corresponds to intervening on less than 1% of all MLP intermediate neurons.

Model	Layers	MLP dim.	Neurons/layer	Ratio
DS-1.5B	28	8960	50	0.56%
Qwen-7B	28	18944	80	0.42%

Table 9: Neuron intervention budget. Percentages are computed over all MLP intermediate neurons across layers.

C.3 Fixed-Coefficient Intervention

For fixed suppression, all selected neurons share the same coefficient α . We evaluate

$$\alpha \in \{1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1, 0.0\}.$$

The case $\alpha = 1.0$ corresponds to the no-intervention baseline. Smaller values impose stronger suppression. Fixed- α sweeps are used as diagnostic references for understanding the effect of suppression strength.

C.4 Adaptive Coefficient Learning

Adaptive coefficient learning assigns a neuron-specific coefficient $\alpha_{l,j}$ to each selected neuron. Only the coefficient parameters are trained; all backbone model parameters remain frozen. For each MATH subject, the corresponding training split is divided into a coefficient-learning split and a held-out development split with development ratio 0.1.

The training examples are constructed from the dataset question and reference solution. The prompt is used as the input prefix, and the dataset solution is used as the teacher-forcing target. Prompt tokens are masked out in the labels, so the language-modeling loss is computed only on solution tokens. During learning, the intervention is active in the forward pass.

Each selected neuron has a trainable raw parameter $r_{l,j}$ mapped to a bounded coefficient:

$$\alpha_{l,j} = \alpha_{\min} + (\alpha_{\max} - \alpha_{\min})\sigma(r_{l,j}).$$

We use the per-neuron setting with $\alpha_{\min} = 0$, $\alpha_{\max} = 1$, and $\alpha_{\text{init}} = 0.5$. The objective is

$$\mathcal{L} = \mathcal{L}_{\text{LM}} + \lambda \mathcal{L}_{\text{reg}},$$

where $\mathcal{L}_{\text{LM}}$ is the token-level language-modeling loss and

$$\mathcal{L}_{\text{reg}} = \frac{1}{|\mathcal{L}_{\text{sel}}|} \sum_{l \in \mathcal{L}_{\text{sel}}} \frac{1}{|\mathcal{N}_l|} \sum_{j \in \mathcal{N}_l} (\alpha_{l,j} - \alpha_{\text{init}})^2.$$

The regularization coefficient is $\lambda = 10^{-3}$.

After each training epoch, we evaluate development-set negative log-likelihood. The checkpoint with the lowest development NLL is selected, and its coefficients are fixed and applied to the test set. The main hyperparameters are summarized in Table 10. We use random seed 42 for adaptive coefficient learning. In the reported per-neuron setting, adaptive CNR optimizes only 1,400 scalar suppression coefficients for DS-1.5B and 2,240 for Qwen-7B, while all backbone parameters remain frozen.

Hyperparameter	Value
Backbone parameters	Frozen
Trainable parameters	Neuron-level coefficients only
Coefficient mode	Per-neuron
$\alpha_{\min}, \alpha_{\max}$	0, 1
α_{init}	0.5
Development ratio	0.1
Epochs	3
Random seed	42
Training batch size	2
Evaluation batch size	4
Learning rate	10^{-2}
Regularization weight	10^{-3}
Checkpoint selection	Lowest development NLL
Maximum sequence length	8192
Precision	bfloat16

Table 10: Hyperparameters for adaptive coefficient learning.

C.5 Neuron-Selection Controls

We use two neuron-selection controls with the same layer-wise intervention budget as CNR. Both controls apply the same multiplicative suppression rule as fixed-coefficient CNR, but replace the CNR-selected neuron set with alternative matched neuron sets. For each model–subject setting, the suppression coefficient is fixed and shared across all selected neurons. For RAND, the same coefficient is used across all five random seeds.

Random-neuron control. The random-neuron control samples neurons with the same cardinality and layer distribution as the CNR-selected neurons. Results are averaged over five random seeds. This control tests whether suppressing an arbitrary matched set of neurons is sufficient to reproduce the observed gains.

Top-activation control. The top-activation control selects, in each layer, the same number of neurons as CNR, but ranks neurons by average activation magnitude over training CoT activations rather than by correctness-controlled trajectory-expansion differences. In our implementation, the top-activation control excludes the original CNR neurons, yielding a non-overlapping high-activation neuron set. This control tests whether suppressing generally high-activation neurons is sufficient, without using the trajectory-expansion signal.

Multi-Seed Random-Neuron Control. To quantify the robustness of this control, Table 11 summarizes the paired-bootstrap comparison between CNR and the five-seed RAND baseline.

Model	CNR– RAND5	95% CI	p	CNR > RAND
DS-1.5B	+1.40	[0.35, 2.46]	0.0076	6/7
Qwen-7B	+2.73	[1.53, 3.92]	$< 10^{-4}$	5/7

Table 11: Five-seed random-neuron control and paired-bootstrap comparison.

Across both models, CNR outperforms the mean of the five RAND seeds in 11 of 14 subject–model settings and individual random-seed runs in 55 of 70 comparisons. The margin is smaller on DS-1.5B, so we interpret this result as statistical support for a modest macro-level advantage over matched random sparse suppression rather than uniform dominance across subjects.

Matched adaptive random-neuron control. To separate trajectory-guided neuron selection from coefficient learning, we additionally evaluate Adaptive RAND. Adaptive RAND uses the same development split, coefficient-learning objective, coefficient bounds, training epochs, and neuron budget as adaptive CNR, but replaces the CNR-selected neurons with random neurons.

Subject	Adaptive RAND	CNR	CNR–Adaptive RAND
DS-1.5B			
Algebra	85.59	89.47	+3.88
Count./Prob.	58.02	59.28	+1.27
Geometry	58.46	61.38	+2.92
Interm. Alg.	57.59	59.69	+2.10
Number Theory	70.37	66.85	-3.52
Prealgebra	77.04	81.06	+4.02
Precalculus	55.49	59.71	+4.21
Macro Avg.	66.08	68.20	+2.13
QWEN-7B			
Algebra	79.53	81.89	+2.36
Count./Prob.	54.85	58.02	+3.16
Geometry	47.39	48.85	+1.46
Interm. Alg.	45.85	49.50	+3.65
Number Theory	62.96	65.74	+2.78
Prealgebra	68.31	78.42	+10.10
Precalculus	47.44	51.83	+4.40
Macro Avg.	58.05	62.04	+3.99

Table 12: Matched Adaptive RAND control. Adaptive RAND uses the same coefficient-learning setup as adaptive CNR while replacing trajectory-guided neuron selection with random neuron selection.

CNR outperforms Adaptive RAND in 13 of 14 subject–model settings, with the only exception being Number Theory on DS-1.5B. The macro-average gains over Adaptive RAND are 2.13 points for DS-1.5B and 3.99 points for QWEN-7B. This result supports the contribution of trajectory-guided neuron selection beyond learned sparse suppression coefficients alone.

C.6 Evaluation Metrics

The primary metric is final-answer accuracy. We also report CoT length, measured as the mean number of reasoning tokens before the final boxed answer. CoT length is used only for analysis and is not an optimization target.

D Subject-wise Fixed-Coefficient Sweep Results

This appendix reports diagnostic fixed-coefficient CNR variants and full subject-wise fixed- α sweeps for DeepSeek-R1-Distill-Qwen-1.5B and Qwen2.5-7B. The $\alpha = 1.0$ row corresponds to the no-intervention baseline. Fixed- α results are used to analyze suppression-strength sensitivity and should be interpreted as oracle-style diagnostic references rather than deployment-realistic baselines. The main paper reports CNR, which learns neuron-specific suppression coefficients on a development split while keeping the backbone model frozen.

D.1 Summary of Fixed-Coefficient Variants

We summarize two fixed-coefficient variants before reporting the full sweeps. G-CNR uses a single fixed suppression coefficient shared across all subjects within each model. SW-CNR selects the best fixed coefficient separately for each subject from the test-set sweep. Both variants are diagnostic references: G-CNR summarizes the effect of a model-level shared coefficient, while SW-CNR provides an oracle-style upper reference for fixed-coefficient selection. Because SW-CNR uses subject-wise test-set selection, it should not be interpreted as a deployment-realistic baseline.

Model	Setting	Coefficient selection	Acc.	Δ
DS-1.5B	BASE	No intervention	65.6	–
	G-CNR	Shared fixed $\alpha = 0.20$	68.1	+2.5
	SW-CNR	Subject-wise best fixed α	69.3	+3.7
	CNR	Learned neuron-specific coefficients	68.2	+2.6
Qwen-7B	BASE	No intervention	58.8	–
	G-CNR	Shared fixed $\alpha = 0.40$	60.5	+1.7
	SW-CNR	Subject-wise best fixed α	62.9	+4.1
	CNR	Learned neuron-specific coefficients	62.0	+3.2

Table 13: Macro-average summary of diagnostic fixed-coefficient variants. Accuracy is reported in percentage points. Δ is computed relative to BASE.

Setting	Alg.	C&P	Geo.	IAlg.	NTh.	PAIg.	PCalc	Avg.
BASE	85.2	55.9	58.7	55.1	67.4	78.1	59.0	65.6
G-CNR	89.6	60.8	54.7	52.8	75.2	83.4	60.1	68.1
SW-CNR	89.6	60.8	59.9	56.1	75.2	83.4	60.1	69.3
CNR	89.5	59.3	61.4	59.7	66.8	81.1	59.7	68.2

Table 14: DS-1.5B diagnostic fixed-coefficient variant accuracy. All numbers are accuracies in percentage points.

Setting	Alg.	C&P	Geo.	IAlg.	NTh.	PAIg.	PCalc	Avg.
BASE	79.0	54.0	50.5	48.6	60.2	68.7	50.4	58.8
G-CNR	80.0	60.1	49.7	49.2	63.0	69.7	51.8	60.5
SW-CNR	82.4	60.1	50.5	49.4	67.2	78.0	52.4	62.9
CNR	81.9	58.0	48.9	49.5	65.7	78.4	51.8	62.0

Table 15: Qwen-7B diagnostic fixed-coefficient variant accuracy. All numbers are accuracies in percentage points.

D.2 CAS Component Sensitivity

We conduct an additional fixed- α diagnostic on DS-1.5B to examine the sensitivity of neuron selection to individual CAS components. This diagnostic uses component weights 0.30/0.25/0.25/0.10/0.10 for mean activation difference, effect size, directional consistency, statistical reliability, and length correlation, respectively. We select 50 MLP neurons per layer over 28 layers, resulting in 1400 selected neurons. All variants use the same fixed suppression coefficient $\alpha = 0.10$ and the same 8192-token evaluation budget. This diagnostic uses the same CAS component weights as the main CNR setting and removes one component at a time to assess its contribution to neuron selection.

Variant	Macro Acc.	Δ	Mean Jaccard
Full CAS rerun	62.63	0.00	100.0
w/o mean difference	59.81	-2.82	56.4
w/o effect size	63.28	+0.65	85.2
w/o consistency	63.49	+0.86	88.3
w/o p -value	63.13	+0.50	88.9
w/o length correlation	64.22	+1.59	81.9

Table 16: Fixed- α CAS component sensitivity diagnostic on DS-1.5B. Δ is measured relative to the full CAS rerun, and Mean Jaccard denotes overlap with the neuron set selected by the full rerun.

Removing the mean-difference component causes the largest reduction in macro accuracy and the largest shift in the selected neuron set. At the same time, several ablated variants outperform the full fixed- α rerun. The current CAS weighting should therefore be viewed as a practical design choice rather than an optimized weighting scheme.

D.3 Full Fixed-Coefficient Sweeps

The following tables provide the full subject-wise fixed- α sweeps. They show that the effect of suppression strength is non-monotonic and subject-dependent, motivating neuron-specific coefficient learning.

α	Algebra	C&P	Geometry
Accuracy (%)			
BASE (1.00)	85.2	55.9	58.7
0.90	85.1	57.6	58.7
0.80	87.6	58.2	58.2
0.70	88.8	57.2	59.9
0.60	87.4	58.6	59.9
0.50	89.3	57.6	57.8
0.40	88.0	58.9	58.2
0.30	89.6	58.2	56.4

Continued on next column

α	Algebra	C&P	Geometry
0.20	89.6	60.8	54.7
0.10	89.3	60.1	57.4
0.00	89.4	58.2	53.9
CNR	89.5	59.3	61.4
CoT Tok Mean			
BASE (1.00)	2532.0	4569.3	4503.1
0.90	2512.6	4574.7	4470.7
0.80	2339.0	4494.0	4510.9
0.70	2302.6	4405.8	4415.2
0.60	2266.6	4410.1	4448.5
0.50	2122.2	4328.9	4463.6
0.40	2117.1	4421.5	4520.4
0.30	2087.6	4430.0	4565.0
0.20	2100.1	4313.5	4628.3
0.10	2045.8	4348.5	4626.2
0.00	2053.1	4459.0	4758.1
CNR	2175.2	4315.2	4267.3

Table 17: DS-1.5B fixed- α sweep: first subject group.

α	Interm. Alg.	Num. Th.	Prealg.	Precalc.
Accuracy (%)				
BASE (1.00)	55.1	67.4	78.1	59.0
0.90	53.1	68.0	79.0	59.0
0.80	56.1	70.9	81.1	58.1
0.70	54.7	73.1	81.5	59.3
0.60	54.3	74.1	80.5	57.5
0.50	55.0	73.0	82.3	57.7
0.40	53.7	74.1	82.3	57.3
0.30	54.3	73.1	82.7	59.2
0.20	52.8	75.2	83.4	60.1
0.10	51.9	70.7	82.9	56.4
0.00	52.4	73.5	83.1	58.8
CNR	59.7	66.8	81.1	59.7
CoT Tok Mean				
BASE (1.00)	5014.1	4211.0	2816.8	4598.8
0.90	5020.2	4112.0	2762.0	4653.6
0.80	4996.3	3793.3	2507.4	4619.4
0.70	5101.9	3725.6	2494.7	4620.0
0.60	5048.2	3587.1	2482.5	4676.8
0.50	5068.6	3497.0	2449.8	4557.5
0.40	5148.7	3388.3	2414.3	4615.2
0.30	5068.3	3548.6	2370.5	4495.5
0.20	5031.3	3331.7	2379.2	4395.2
0.10	5114.0	3476.7	2226.3	4621.7
0.00	5184.5	3451.2	2384.5	4419.4
CNR	4793.9	4137.9	2456.9	4335.0

Table 18: DS-1.5B fixed- α sweep: remaining subjects.

α	Algebra	C&P	Geometry
Accuracy (%)			
BASE (1.00)	79.0	54.0	50.5
0.90	80.0	58.2	49.3
0.80	81.9	54.6	49.5
0.70	82.4	54.9	47.2
0.60	79.8	55.9	47.2
0.50	76.6	59.1	47.6
0.40	80.0	60.1	49.7

Continued on next column

α	Algebra	C&P	Geometry
0.30	81.5	58.0	49.1
0.20	80.6	57.2	49.9
0.10	77.5	54.6	50.3
0.00	71.7	53.4	49.5
CNR	81.9	58.0	48.9
CoT Tok Mean			
BASE (1.00)	546.0	606.6	763.0
0.90	515.4	552.9	747.0
0.80	536.5	568.3	762.2
0.70	518.0	519.2	709.4
0.60	612.2	547.0	770.7
0.50	692.1	540.1	687.1
0.40	636.5	473.8	709.0
0.30	717.3	512.8	691.4
0.20	644.7	601.9	708.5
0.10	881.3	731.3	759.1
0.00	1214.8	724.0	720.6
CNR	495.4	539.3	701.3

Table 19: Qwen-7B fixed- α sweep: first subject group.

α	Interm. Alg.	Num. Th.	Prealg.	Precalc.
Accuracy (%)				
BASE (1.00)	48.6	60.2	68.7	50.4
0.90	48.1	61.1	68.4	50.7
0.80	48.1	62.8	66.6	48.9
0.70	48.7	62.0	67.7	51.1
0.60	48.8	66.3	67.1	51.6
0.50	47.7	67.2	69.4	52.4
0.40	49.2	63.0	69.7	51.8
0.30	47.3	60.2	75.1	49.5
0.20	48.0	61.9	76.1	49.5
0.10	49.4	58.3	78.0	52.0
0.00	47.8	55.7	75.0	50.2
CNR	49.5	65.7	78.4	51.8
CoT Tok Mean				
BASE (1.00)	1103.8	738.6	436.1	1152.2
0.90	1171.4	729.5	407.6	1185.2
0.80	1108.1	701.3	414.2	1038.8
0.70	1075.9	667.0	388.1	1201.4
0.60	1211.4	717.2	386.0	1116.2
0.50	1149.8	700.2	436.3	1068.4
0.40	1060.0	782.5	409.6	1170.6
0.30	1096.1	732.8	367.4	1182.0
0.20	1131.9	900.5	436.6	1159.2
0.10	1147.8	1030.0	418.5	1157.9
0.00	1129.6	1139.5	611.9	1269.2
CNR	1066.7	702.9	372.2	1106.0

Table 20: Qwen-7B fixed- α sweep: remaining subjects.

E Pair-Retention Statistics

This appendix provides detailed pair-retention statistics for correctness-controlled trajectory-expansion pair construction. For each training problem, CNR samples multiple CoT trajectories, retains correct trajectories, constructs a trajectory-expansion pair when possible, and keeps the pair only if it satisfies the expansion-ratio threshold. In

implementation, the expanded correct trajectory and the concise correct trajectory are instantiated as the longest and shortest correct trajectories for the same problem. The retained pairs are used for contrastive neuron scoring. The skipped count aggregates all pair-construction failure modes, including insufficient correct samples, answer-extraction failures, and failure to satisfy the expansion-ratio threshold.

Overall, DeepSeek-R1-Distill-Qwen-1.5B retains 4409 out of 7500 training questions, corresponding to an overall pair-retention rate of 58.8%. Qwen2.5-7B retains 4608 out of 7500 training questions, corresponding to an overall pair-retention rate of 61.4%. The retained pairs generally have median expansion ratios close to 2.0, while some subjects contain much larger maximum ratios, especially for Qwen2.5-7B. This indicates that the pair-construction process yields a broad range of trajectory-expansion contrasts.

Pair-retention difficulty bias. Because pair construction requires multiple correct trajectories with sufficient expansion variation, the retained problems may differ systematically from the full training distribution. An additional difficulty analysis indicates a modest easy-case skew. The largest observed shift occurs in Geometry: the full MATH training set has mean difficulty level 3.96, whereas the retained Geometry problems have mean difficulty levels 3.52 for DS-1.5B and 3.53 for QWEN-7B. Similar but smaller shifts are observed in other subjects. This selection effect may influence which neurons are selected, so we do not interpret the resulting neuron sets as universal causal mechanisms of reasoning expansion.

Subject	Total	Retained	Skipped	Rate
Algebra	1744	1118	626	64.1
Count. & Prob.	771	458	313	59.4
Geometry	870	507	363	58.3
Interm. Alg.	1295	789	506	60.9
Num. Th.	869	500	369	57.5
Prealgebra	1205	710	495	58.9
Precalculus	746	327	419	43.8
Total	7500	4409	3091	58.8

Table 21: DS-1.5B pair-retention statistics.

Subject	Total	Retained	Skipped	Rate
Algebra	1744	1117	627	64.0
Count. & Prob.	771	459	312	59.5

Continued on next column

Subject	Total	Retained	Skipped	Rate
Geometry	870	533	337	61.3
Interm. Alg.	1295	804	491	62.1
Num. Th.	869	528	341	60.8
Prealgebra	1205	748	457	62.1
Precalculus	746	419	327	56.2
Total	7500	4608	2892	61.4

Table 22: Qwen-7B pair-retention statistics.

Subject	Mean	Median	Min	Max
Algebra	2.23	1.94	1.50	12.37
Count. & Prob.	2.31	2.02	1.50	11.08
Geometry	2.26	1.95	1.50	10.84
Interm. Alg.	2.28	2.00	1.50	12.92
Num. Th.	2.32	2.03	1.50	13.65
Prealgebra	2.21	1.91	1.50	10.73
Precalculus	2.36	2.03	1.50	11.94

Table 23: DS-1.5B expansion-ratio statistics for retained pairs.

Subject	Mean	Median	Min	Max
Algebra	2.19	1.91	1.50	14.75
Count. & Prob.	2.26	1.97	1.50	13.28
Geometry	2.32	2.00	1.50	18.37
Interm. Alg.	2.35	2.03	1.50	19.42
Num. Th.	2.31	2.00	1.50	15.66
Prealgebra	2.16	1.88	1.50	12.41
Precalculus	2.42	2.05	1.50	20.13

Table 24: Qwen-7B expansion-ratio statistics for retained pairs.

F CNR Subject-Wise Details

This appendix provides detailed subject-wise comparisons between the no-intervention baseline and CNR. CNR learns per-neuron suppression coefficients on a subject-specific development split while keeping the backbone model frozen. Accuracy is reported in percentage points. CoT length is reported as an analysis metric and is measured by the mean number of reasoning tokens before the final boxed answer. Δ Acc. and Δ CoT are computed as CNR minus BASE.

Subject	Base	CNR	Δ Acc.
Algebra	85.2	89.5	+4.3
Count. & Prob.	55.9	59.3	+3.4
Geometry	58.7	61.4	+2.7
Interm. Alg.	55.1	59.7	+4.6
Num. Th.	67.4	66.8	-0.6
Prealgebra	78.1	81.1	+3.0
Precalculus	59.0	59.7	+0.7
Macro Avg.	65.6	68.2	+2.6

Table 25: DS-1.5B: subject-wise accuracy comparison between BASE and CNR.

Subject	Base	CNR	Δ CoT
Algebra	2532.0	2175.2	-356.8
Count. & Prob.	4569.3	4315.2	-254.1
Geometry	4503.1	4267.3	-235.8
Interm. Alg.	5014.1	4793.9	-220.2
Num. Th.	4211.0	4137.9	-73.1
Prealgebra	2816.8	2456.9	-359.9
Precalculus	4598.8	4335.0	-263.8
Macro Avg.	4035.0	3783.1	-252.0

Table 26: DS-1.5B: subject-wise CoT length comparison between BASE and CNR.

Subject	Base	CNR	Δ Acc.
Algebra	79.0	81.9	+2.9
Count. & Prob.	54.0	58.0	+4.0
Geometry	50.5	48.9	-1.6
Interm. Alg.	48.6	49.5	+0.9
Num. Th.	60.2	65.7	+5.5
Prealgebra	68.7	78.4	+9.7
Precalculus	50.4	51.8	+1.4
Macro Avg.	58.8	62.0	+3.2

Table 27: Qwen-7B: subject-wise accuracy comparison between BASE and CNR.

Subject	Base	CNR	Δ CoT
Algebra	546.0	495.4	-50.6
Count. & Prob.	606.6	539.3	-67.3
Geometry	763.0	701.3	-61.7
Interm. Alg.	1103.8	1066.7	-37.1
Num. Th.	738.6	702.9	-35.7
Prealgebra	436.1	372.2	-63.9
Precalculus	1152.2	1106.0	-46.2
Macro Avg.	763.8	712.0	-51.8

Table 28: Qwen-7B: subject-wise CoT length comparison between BASE and CNR.

G Length-Control Baseline

This appendix reports the full results of the LENCAP baseline. LENCAP uses the no-intervention model and controls only the maximum generation length. For each model and subject, we run a small grid of maximum generation lengths around the corresponding CNR CoT length and select the cap whose average CoT length is closest to CNR.

The purpose of this baseline is to test whether the gains of CNR can be reproduced by hard length capping alone. The results show that length capping does not reproduce the CNR gains on average. On DeepSeek-R1-Distill-Qwen-1.5B, LENCAP improves over BASE but remains below CNR. On Qwen2.5-7B, LENCAP reduces CoT length substantially but decreases macro-average accuracy below the no-intervention baseline.

Subject	BASE	CNR	LENCAP	Cap
Algebra	85.2	89.5	83.0	3264
Count. & Prob.	55.9	59.3	58.0	6464
Geometry	58.7	61.4	59.1	6400
Interm. Alg.	55.1	59.7	57.8	7168
Num. Th.	67.4	66.8	69.6	6208
Prealgebra	78.1	81.1	78.1	3712
Precalculus	59.0	59.7	61.0	6528
Macro Avg.	65.6	68.2	66.7	—

Table 29: DS-1.5B length-control baseline: accuracy.

Subject	BASE	CNR	LENCAP	Diff.
Algebra	2532.0	2175.2	1830.0	345.2
Count. & Prob.	4569.3	4315.2	3762.0	553.2
Geometry	4503.1	4267.3	3798.3	469.1
Interm. Alg.	5014.1	4793.9	4545.0	248.9
Num. Th.	4211.0	4137.9	3412.8	725.1
Prealgebra	2816.8	2456.9	1869.9	587.0
Precalculus	4598.8	4335.0	4049.9	285.1
Macro Avg.	4035.0	3783.1	3324.0	459.1

Table 30: DS-1.5B length-control baseline: CoT length.

Subject	BASE	CNR	LENCAP	Cap
Algebra	79.0	81.9	75.6	768
Count. & Prob.	54.0	58.0	54.2	832
Geometry	50.5	48.9	50.1	1024
Interm. Alg.	48.6	49.5	47.5	1600
Num. Th.	60.2	65.7	58.1	1024
Prealgebra	68.7	78.4	66.6	576
Precalculus	50.4	51.8	49.1	1664
Macro Avg.	58.8	62.0	57.3	—

Table 31: Qwen-7B length-control baseline: accuracy.

Subject	BASE	CNR	LENCAP	Diff.
Algebra	546.0	495.4	398.3	97.1
Count. & Prob.	606.6	539.3	415.3	124.0
Geometry	763.0	701.3	500.0	201.3
Interm. Alg.	1103.8	1066.7	759.5	307.2
Num. Th.	738.6	702.9	493.2	209.7
Prealgebra	436.1	372.2	299.3	72.9
Precalculus	1152.2	1106.0	770.7	335.3
Macro Avg.	763.8	712.0	519.5	192.5

Table 32: Qwen-7B length-control baseline: CoT length.

H Activation-Steering Baseline

This appendix reports dense activation-steering results. Unlike CNR, which selects sparse MLP neurons and applies multiplicative suppression, ACTSTEER uses the same correctness-controlled expanded-concise activation pairs to construct dense layer-wise steering directions. For each layer, we compute

$$v_l = \mathbb{E}[h_l(y^{\text{con}}) - h_l(y^{\text{exp}})],$$

and apply

$$h'_{l,t} = h_{l,t} + \beta v_l$$

during generation. A positive β steers activations toward the concise side of the expanded–concise contrast. We use a fixed steering coefficient $\beta = 0.1$ for all subjects.

Subject	BASE	CNR	ACTSTEER
Algebra	85.2	89.5	86.6
Count. & Prob.	55.9	59.3	60.3
Geometry	58.7	61.4	60.3
Interm. Alg.	55.1	59.7	58.6
Num. Th.	67.4	66.8	68.5
Prealgebra	78.1	81.1	79.4
Precalculus	59.0	59.7	61.9
Macro Avg.	65.6	68.2	68.0

Table 33: DS-1.5B dense activation-steering baseline: accuracy.

Subject	CNR	ACTSTEER
Algebra	2175.2	2451.3
Count. & Prob.	4315.2	4398.6
Geometry	4267.3	4364.2
Interm. Alg.	4793.9	4936.6
Num. Th.	4137.9	4000.1
Prealgebra	2456.9	2694.8
Precalculus	4335.0	4563.3
Macro Avg.	3783.1	3915.6

Table 34: DS-1.5B dense activation-steering baseline: CoT length.

Subject	BASE	CNR	ACTSTEER
Algebra	79.0	81.9	76.9
Count. & Prob.	54.0	58.0	54.4
Geometry	50.5	48.9	52.0
Interm. Alg.	48.6	49.5	48.0
Num. Th.	60.2	65.7	59.8
Prealgebra	68.7	78.4	67.9
Precalculus	50.4	51.8	52.0
Macro Avg.	58.8	62.0	58.7

Table 35: Qwen-7B dense activation-steering baseline: accuracy.

Subject	CNR	ACTSTEER
Algebra	495.4	506.8
Count. & Prob.	539.3	657.7
Geometry	701.3	617.6
Interm. Alg.	1066.7	1068.0
Num. Th.	702.9	720.1
Prealgebra	372.2	404.9
Precalculus	1106.0	1136.4
Macro Avg.	712.0	730.2

Table 36: Qwen-7B dense activation-steering baseline: CoT length.

I Additional Subject-Wise Neuron Overlap Details

The main text reports the summary of subject-wise neuron-overlap statistics in Figure 3. This appendix provides the definition and additional intersection statistics.

For each model and subject, we represent the selected neurons as a set of layer–neuron pairs:

$$\mathcal{S}_s = \{(l, j) : j \in \mathcal{N}_l^{(s)}\},$$

where s denotes the MATH subject and $\mathcal{N}_l^{(s)}$ is the selected neuron set at layer l . For two subjects a and b , we compute the Jaccard overlap:

$$J(\mathcal{S}_a, \mathcal{S}_b) = \frac{|\mathcal{S}_a \cap \mathcal{S}_b|}{|\mathcal{S}_a \cup \mathcal{S}_b|}.$$

A larger Jaccard value indicates that two subjects share more selected neurons.

The selected neuron sets show nontrivial overlap across subjects. DeepSeek-R1-Distill-Qwen-1.5B has a higher mean overlap than Qwen2.5-7B, suggesting that its selected expansion-associated neurons are more shared across MATH subjects. Qwen2.5-7B still exhibits substantial overlap, but its lower minimum and mean values suggest stronger subject specificity.

Model	Mean Inter.	Min Inter.	Max Inter.	Neurons /Subj.
DS-1.5B	951.3	784	1099	1400
Qwen-7B	1205.4	768	1671	2240

Table 37: Intersection statistics for subject-wise neuron overlap. Intersection denotes the number of shared selected layer–neuron pairs between two subjects.