Probing Reasoning Length Control in Language Models via Neurons

Dongwook Min[1] **Jeonhwi Kang**[1] **Cheoneum Park**[2*] **Seung-Hoon Na**[1*]
[1]Ulsan National Institute of Science and Technology
[2]Hanbat National University
mindw96@unist.ac.kr, kangjeonhwi@unist.ac.kr
parkce@hanbat.ac.kr, nash@unist.ac.kr

Abstract

Reasoning-oriented language models can generate thousands of tokens before answering, yet the internal factors that determine their reasoning length remain poorly understood. This paper studies whether reasoning length can be controlled through sparse neuron interventions, complementing prompt-based, budget-based, and controller-based approaches. The proposed method discovers *reasoning-length-associated neurons* from paired long and concise correct trajectories on a training split, ranks MLP neurons by long-minus-concise activation contrast, and tests the selected neurons on disjoint evaluation splits by scaling their activations during generation. The evaluation covers math, science, and commonsense reasoning tasks, including GSM8K, MATH500, ARC-Challenge, CommonsenseQA, and QASC, under a strict split-separated protocol that never discovers neurons on held-out evaluation examples. In several, but not all, task/model/budget regimes, selected neurons modulate generated length differently from layer-matched random controls, with a modest single-run specificity result on CommonsenseQA accompanied by a small accuracy increase. These results provide task-, model-, and prompt-protocol-specific evidence that sparse MLP neurons can, in some split-controlled regimes, provide neuron-level control signals for generated reasoning length.

1 Introduction

Large language models increasingly solve difficult tasks by producing long chains of intermediate reasoning (Wei et al., 2022; Kojima et al., 2022). Longer reasoning can improve performance by decomposing problems, checking partial solutions, and recovering from local errors. However, longer generations do not uniformly improve reasoning outcomes. On relatively easy math problems or multiple-choice reasoning questions, extended deliberation increases latency, memory pressure, and inference cost; it can also introduce unnecessary assumptions, repetition, distractor analysis, or answer drift. These patterns are commonly associated with overthinking and define a central control problem: adjusting model-generated reasoning length without retraining the model or relying on output truncation.

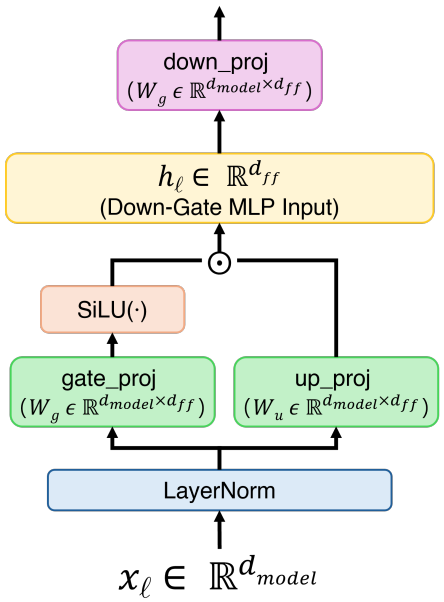


Figure 1: Gated-MLP computation pipeline and neuron definition. At layer ℓ, the hidden state is projected through the gate and up projections, combined as $h_\ell = \mathrm{SiLU}(\mathrm{gate_proj}_\ell(x)) \odot \mathrm{up_proj}_\ell(x)$, and then passed to the down-projection. Each coordinate $h_{\ell,j}$ of this down-projection input vector is treated as one MLP neuron for discovery and intervention.

Throughout this paper, we use *neuron* to denote an activation coordinate inside the Transformer MLP, rather than a vocabulary output unit or a biological neuron. In Qwen-style gated MLP blocks, the down-projection input is

$$h_\ell = \mathrm{SiLU}(\mathrm{gate_proj}_\ell(x)) \odot \mathrm{up_proj}_\ell(x), \quad (1)$$

and one neuron is a layer-coordinate pair (ℓ, j) indexing a coordinate of h_ℓ. Figure 1 illustrates this

*Co-corresponding authors.

Findings of the Association for Computational Linguistics: IJCNLP-AACL 2026, pages 4791–4808
November 6-10, 2026 ©2026 Association for Computational Linguistics

gated-MLP activation space and the neuron definition used throughout the paper.

Most existing approaches control reasoning length from outside the model. Prompts ask for shorter or longer reasoning, decoding budgets impose hard limits, and search-based or adaptive methods regulate inference-time computation (Wang et al., 2023; Yao et al., 2023). These methods are effective external controls, but they do not identify which internal components modulate the realized length of a reasoning trajectory. If reasoning length is partly represented inside the model, direct neuron-level intervention may provide a more fine-grained and interpretable form of test-time control.

Two recent lines of evidence provide empirical context for this hypothesis. First, neuron-level analyses of chain-of-thought reasoning show that high-quality and low-quality trajectories activate different feed-forward neurons, and that intervening on selected neurons can affect reasoning accuracy (Tang et al., 2025). Second, reasoning-strength planning work shows that the number of reasoning tokens a large reasoning model will generate can be predicted from pre-generation activations and modulated by a dense activation direction (Sheng et al., 2025). Taken together, these findings indicate that reasoning behavior is not exclusively an output-side textual phenomenon. They do not, however, establish whether the realized length of reasoning can be modulated through a sparse set of MLP neurons, nor whether length-related signals can be separated from reasoning quality and overthinking.

This paper investigates sparse neuron intervention as a mechanism for reasoning-length control. The analysis focuses on MLP intermediate neurons, specifically coordinates of the gated MLP vector passed into each Transformer block's down-projection. High-ranking neurons are referred to as candidate *reasoning-length-associated neurons*. The cautious terminology is intentional: the discovery step identifies MLP neurons associated with a behavioral contrast, while the intervention claim is established through held-out interventions against matched random controls.

The full method operationalizes this idea by separating discovery from evaluation: candidate neurons are selected on a training split and tested only through held-out intervention. Section 3 provides the full discovery, scoring, and intervention protocol.

Because length and quality are not perfectly separable in real reasoning, we also compare directly against prior reasoning-critical-neuron methods. This comparison evaluates whether neurons selected for reasoning length are competitive with neurons selected for reasoning quality, while preserving the paper's primary focus on controlling generated length.

The empirical evaluation has two components. The main experiments use Qwen3-4B-Thinking-2507 under the model's recommended sampling settings, with practical 4096/8192-token budgets and task-specific budget checks. The evaluation covers math tasks, science-style multiple-choice reasoning, and commonsense reasoning, including GSM8K, MATH500, ARC-Challenge, CommonsenseQA, and QASC. Controlled cross-model evidence with DeepSeek-R1-Distill-Qwen-7B and Qwen3-4B-Instruct-2507 is reported separately. These regimes are not pooled; they separate the main split-controlled results from supporting external-validity checks.

Our contributions are:

- We formulate neuron-level reasoning-length control as a testable mechanistic problem and evaluate whether sparse MLP neurons provide a targeted intervention mechanism for generated reasoning length.
- We adapt differential neuron analysis to correctness-controlled paired trajectories for identifying candidate reasoning-length-associated neurons from long and concise successful solutions.
- We evaluate activation scaling and same-count random controls to test whether selected neurons modulate length beyond generic activation perturbation.
- We compare reasoning-length-associated neuron selection with Tang-style reasoning-critical-neuron ratio scoring under a matched seven-subject MATH protocol.
- We provide a strict split-separated protocol for neuron discovery and held-out evaluation, avoiding leakage from discovering neurons on the same examples used for evaluation.

2 Related Work

Reasoning and inference-time computation. Chain-of-thought prompting elicits intermediate

reasoning steps and improves performance on multi-step tasks (Wei et al., 2022; Kojima et al., 2022). Self-consistency marginalizes over multiple sampled reasoning paths (Wang et al., 2023), while tree-based methods search over intermediate states (Yao et al., 2023). Scratchpad-style work also treats generated tokens as externalized computation (Nye et al., 2021). These methods make reasoning more powerful, but they also make inference-time computation a central design variable. The present work complements these external-control approaches by testing whether sparse internal neurons provide an intervention mechanism for generated reasoning length.

Reasoning-strength planning. Recent work argues that large reasoning models partially plan their reasoning budget before generation. In particular, Sheng et al. (2025) show that reasoning-token counts can be predicted from pre-generation activations and identify a dense activation direction whose magnitude modulates reasoning strength. These results indicate that reasoning length is not merely a surface artifact of decoding or prompt wording. The present work addresses the complementary problem of whether the realized length of reasoning can be modulated through sparse MLP neurons during generation and tested with neuron-level interventions. The distinction is relevant because a dense planning direction may reveal that budget information exists in activation space without identifying a sparse neuron-level intervention mechanism.

Transformer MLP neurons. Transformer feed-forward layers have been interpreted as high-dimensional feature stores or key-value memories (Geva et al., 2021). Knowledge neuron and model editing work localizes factual associations in parameters or activations (Dai et al., 2022; Meng et al., 2022). Reasoning length is not a single fact, and we do not assign a fixed semantic label to an individual neuron. Instead, we rank MLP neurons by behavioral activation contrasts and validate the resulting set through held-out interventions.

Neuron differential analysis for reasoning quality. The closest prior work is Tang et al. (2025), who identify reasoning-critical FFN neurons by comparing high-quality and low-quality CoT trajectories and then intervene on the selected neurons to improve downstream reasoning accuracy. The proposed method uses a related activation-differential logic, but targets a different behavioral factor. Rather than contrasting high-quality chains against low-quality chains, our main reasoning-length protocol contrasts long and concise successful trajectories for the same question and retains only pairs where both final answers are correct. The design reduces correctness confounds when searching for length-associated neurons.

Quality and length can nevertheless interact. Long reasoning may reflect beneficial verification on difficult examples, but it may also reflect unnecessary deliberation or answer drift. Accordingly, Tang et al. (2025) is used as the most direct methodological baseline, with a same-model, same-split comparison reported in Section 5.1.

Activation steering and causal intervention. Activation steering modifies hidden states to influence model behavior (Turner et al., 2024; Li et al., 2023). A common strategy is to build a contrastive direction from examples that differ in a target behavior and add or subtract that direction during inference. Such directions can be effective, but their sites include both the residual stream and attention-head outputs. Our intervention is more localized: selected MLP down-projection input coordinates are scaled during autoregressive decoding. Matched random controls test whether observed changes are specific to the selected neurons rather than a generic consequence of perturbing MLP activations.

Positioning. Prior reasoning-critical-neuron work shows that reasoning quality can be affected by FFN interventions, and reasoning-strength planning shows that reasoning budget is predictable from activations. The present study connects these ideas while keeping the claims separate: reasoning-critical neurons do not imply the existence of reasoning-length-associated neurons, and a dense budget direction does not imply sparse neuron-level control.

3 Method

Protocol overview. The main protocol has three stages. First, on a discovery split, we generate paired long and concise trajectories from the same model for the same question and retain only pairs where both final answers are correct and the long trajectory is substantially longer. The correctness-controlled construction reduces the direct final-answer correctness confound. Second, we col-

lect MLP down-projection input activations over generated tokens and rank neurons by their paired long-minus-concise activation contrast. Third, on a disjoint evaluation split, we scale the selected neurons by α, measuring accuracy, token count, and matched random controls. Figure 2 summarizes this discovery-and-intervention pipeline.

Terminology. Following common usage in neuron-level interpretability work, *neuron* refers to one MLP activation coordinate. More precisely, each neuron is a layer-coordinate pair in the gated MLP down-projection input. Neurons selected by our score are called *reasoning-length-associated neurons*; they are associated with a long-versus-concise activation contrast and must still be validated by held-out intervention. *Selected neurons* denotes the intervention set, and *matched random neurons* denotes same-count, layer-matched controls. The notation k_{coord} denotes the number of selected MLP neurons, whereas k_{sample} denotes the decoding top-k sampling cutoff; these are distinct quantities throughout the paper.

3.1 Long and Concise Trajectory Discovery

For each discovery item x_i, we generate a long trajectory y_i^L and a concise trajectory y_i^S from the same model. For math tasks, prompts ask the model to reason step by step and place the final answer in `\boxed{}`. For multiple-choice tasks, prompts ask for a standardized answer field containing only the choice letter.

The long/concise contrast is not meant to distinguish correct from incorrect reasoning. Let $|y|$ denote the assistant output length after removing prompt/chat-template tokens and padding. Parser failures and incorrect answers are retained for rollout summaries but not for discovery. Discovery retains only examples where both trajectories are correct and the long trajectory is at least ρ times longer than the concise trajectory:

$$|y_i^L| \geq \rho |y_i^S|, \qquad \rho = 1.2. \tag{2}$$

The paired filter selects length variation among successful solutions; it does not isolate length from verification or explanation style. In addition to the default long/concise prompts, we evaluate stronger detail/derivation prompts to induce clearer long-versus-concise contrasts.

3.2 Neuron Definition and Activation Summary

A neuron is defined as one coordinate of the MLP intermediate vector passed into the down-projection. Concretely, for a gated MLP layer ℓ, the model computes

$$\begin{aligned} h_\ell &= \mathrm{SiLU}(\mathrm{gate_proj}_\ell(x)) \odot \mathrm{up_proj}_\ell(x), \\ h_\ell &\in \mathbb{R}^{d_{\text{ff}}}, \end{aligned} \tag{3}$$

and then applies $\mathrm{down_proj}_\ell$ to map this vector back to the model dimension. Candidate neurons are indexed by a layer-coordinate pair (ℓ, j), where j is a coordinate of h_ℓ (Figure 1). The definition is distinct from a vocabulary or output neuron: it is an activation coordinate inside the gated MLP block.

For each retained trajectory, we average the coordinate's activation over up to the first 256 generated positions, excluding padding. Truncated samples are retained unless otherwise stated, with truncation rates reported for budget-sensitive analyses. This fixed-prefix summary prevents a mechanical bias where long trajectories contribute many more late tokens than concise ones; it does not establish that length is determined within this prefix.

3.3 Scoring Reasoning-Length-Associated Neurons

For pair i, layer ℓ, and coordinate j, let $a_{i,\ell,j}^L$ and $a_{i,\ell,j}^S$ denote the summarized activation for the long and concise trajectories. Paired differences are computed as

$$d_{i,\ell,j} = a_{i,\ell,j}^L - a_{i,\ell,j}^S. \tag{4}$$

Coordinates are ranked by a standardized mean contrast:

$$s_{\ell,j} = \frac{\mathrm{mean}_i(d_{i,\ell,j})}{(\mathrm{std}_i(d_{i,\ell,j}) + \epsilon)/\sqrt{n}}. \tag{5}$$

Here n is the number of retained pairs, $\epsilon = 10^{-6}$, and std is the population standard deviation over pairs. Ranking uses the positive signed score $s_{\ell,j}$, rather than $|s_{\ell,j}|$, so selected neurons are those more active in long successful trajectories than in concise successful trajectories. The main setting selects $k_{\text{coord}} = 512$ neurons.

Tang-aligned activation-ratio variant. For the official MATH comparison in Table 1, we additionally evaluate a Tang-aligned ratio variant. Because raw MLP intermediate activations are signed,

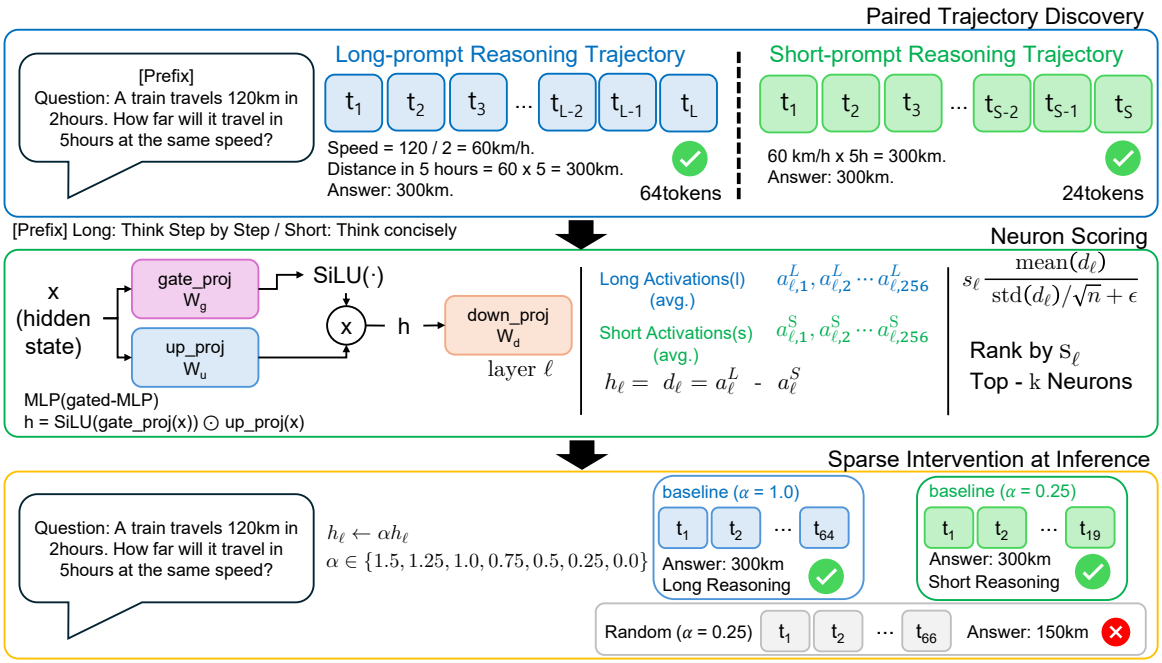


Figure 2: Overview of sparse neuron-level control of reasoning length. The pipeline first constructs paired long and concise reasoning trajectories, then scores MLP neurons by their long–concise activation contrast, and finally scales the selected neurons during held-out inference. The intervention is evaluated by changes in generated length and accuracy relative to matched random-neuron controls.

we do not take a signed activation ratio. Instead, for each neuron (ℓ, j), we compute an activation-magnitude ratio:

$$r_{\ell,j} = \frac{\text{mean}_i |a^L_{i,\ell,j}| + \epsilon}{\text{mean}_i |a^S_{i,\ell,j}| + \epsilon}, \qquad \epsilon = 10^{-6}, \quad (6)$$

and rank neurons by $\log r_{\ell,j}$. The MATH row labeled "Ours ratio variant" selects train-derived neurons satisfying $r_{\ell,j} > 1.10$ for each subject, with matched random controls using the same count and layer distribution. This variant is used only for the Tang-aligned MATH comparison; the main reasoning-length score remains the standardized long-minus-concise contrast above. Unlike the default method, the ratio variant uses all long/short pairs without the both-correct or length-ratio filter.

3.4 Sparse Activation Scaling

Given a selected neuron set N, we scale selected neurons during generation:

$$h_{\ell,j} \leftarrow \alpha h_{\ell,j}, \quad (\ell, j) \in N. \quad (7)$$

All unselected neurons are unchanged. Since MLP activations can be positive or negative, $\alpha < 1$ scales selected neurons toward zero rather than guaranteeing a monotonic downstream logit effect. We evaluate $\alpha \in \{1.0, 0.75, 0.5, 0.25, 0.0\}$, where $\alpha = 1.0$ is the no-intervention baseline and $\alpha = 0.0$ zeros selected neurons. Unless stated otherwise, intervention is applied only during generated-token decoding, not prompt prefill.

3.5 Controls and Specificity Tests

Matched random-neuron alpha sweep. For each selected neuron set, we sample k_{coord} random MLP intermediate neurons from the full model, matching the selected-neuron count and layer distribution. Unless otherwise stated, each table uses one fixed random draw with seed 0 rather than an average over many random sets. The control evaluates whether length changes are specific to ranked neurons rather than a generic effect of perturbing the same number of MLP neurons. When we report selected–random token confidence intervals, they are paired example-level bootstrap intervals over the held-out examples; they do not include random-neuron-set variance or decoding-seed variance.

4 Experimental Setup

4.1 Models

The main thinking-model held-out specificity result uses Qwen3-4B-Thinking-2507 (Yang et al., 2025), a compact thinking model whose recommended sampling protocol produces long-form

reasoning. The direct Tang-style MATH comparison uses `Qwen/Qwen2.5-Math-1.5B-Instruct` (Yang et al., 2024) and Llama 3.2 Instruct models (Grattafiori et al., 2024; **?**) under the same official seven-subject MATH protocol. Qwen3-1.7B (Yang et al., 2025) is used for the graded alpha-scaling evaluation. Appendix supporting experiments additionally use DeepSeek-R1-Distill-Qwen-7B (Guo et al., 2025).

4.2 Tasks, Splits, and Leakage Control

Neuron discovery is always performed on a training/development source, while intervention is evaluated on a disjoint held-out split. Evaluation examples are never used for pairing, activation extraction, or neuron scoring. Appendix Table 5 lists exact split sources, including the distinction between the official 5,000-example MATH test and the separate 500-example MATH500 benchmark (Hendrycks et al., 2021; Lightman et al., 2024). Highlighted settings are descriptive choices from parameter sweeps; reported intervals do not adjust for that selection.

4.3 Decoding and Budget Selection

For Qwen3-4B-Thinking-2507, we use the model-card recommended sampling settings available at the time of experimentation: temperature 0.6, top-$p = 0.95$, top-$k_{\text{sample}} = 20$, and min-$p = 0$. The recommended maximum output length for thinking models is large; our evaluated budgets can still truncate outputs, so we report 4096-token runs separately from higher-budget runs. Appendix Section A.2 lists the exact prompt templates, decoding settings, and intervention provenance fields used in the experiments.

4.4 Prompt Contrast Variants

The default long/concise prompt contrast is intentionally simple, but low-contrast prompts can yield weak activation signals. We therefore also use stronger *detail* and *derivation* prompt variants to induce clearer long-versus-concise correct trajectories for discovery.

4.5 Metrics

We report exact-match accuracy, mean and median generated tokens, pairwise token delta relative to $\alpha = 1.0$, and reduced rate. Supporting analyses also report correct-to-correct subsets where both baseline and intervention answers are correct, so compression can be separated from answer degradation. The MATH comparison and symbolic audits use the Qwen equivalence evaluator; GSM8K robustness, transfer, and external-control accuracy retain explicitly labeled local exact-match scores. Evaluation rescoring does not change the local verifier used for GSM8K discovery.

5 Results

The main text focuses on three results that directly support the paper's claims. First, we compare against the closest neuron-differential baseline of Tang et al. (2025) under the same MATH subject protocol. Second, we show that scaling selected neurons produces a graded interventional change in reasoning length. Third, we report a held-out specificity result on the main thinking-model family, comparing selected neurons with matched random neurons. Bold marks the best value within each comparable task/model block.

5.1 Direct Comparison with CoT Neuron Differential Analysis

Tang et al. (2025) identify FFN neurons by contrasting high- and low-quality CoT trajectories and intervening on the selected neurons to improve reasoning accuracy. Table 1 aligns our evaluation with that setup on the official seven-subject MATH split. All non-published baselines are reproduced in the same pipeline, with intervention neurons discovered only from the train-side split. On `Qwen/Qwen2.5-Math-1.5B-Instruct` and `meta-llama/Llama-3.2-3B-Instruct`, the sparse ratio intervention obtains the highest pooled accuracy among the matched-protocol rows; on `meta-llama/Llama-3.2-1B-Instruct`, it does not outperform the no-intervention or reasoning-critical baselines. These small differences are descriptive rather than evidence of a universal mathematical-accuracy improvement. The table uses the Tang-aligned ratio-threshold rule > 1.10 described in Section 3, rather than the default standardized top-k_{coord} score. Appendix Table 7 reports the parser audit motivating the use of the official Qwen mathematical-equivalence parser for this comparison, and Appendix Table 8 details the ratio-threshold analysis that led to this sparse operating point. The comparison is intended as a matched-protocol check rather than as the paper's only evidence for length control. The MATH rows ask whether neurons selected from a length

Method	MATH							
	Algebra	CP	PC	PA	Geometry	IA	NT	Avg.
Qwen2.5-Math-1.5B-IT								
Greedy CoT	**92.59**	70.04	63.55	85.42	64.09	59.69	78.89	75.88
HF no-intervention	92.00	69.20	**65.20**	85.65	64.09	59.80	80.19	76.04
Reasoning-critical neurons (Baseline)	92.42	69.62	64.65	86.11	64.72	59.36	79.63	76.12
Matched random	92.42	69.20	64.84	85.76	63.05	**60.02**	**80.56**	76.10
Ours (ratio)	92.33	**71.94**	64.65	**86.34**	**65.55**	58.80	79.26	**76.30**
Llama 3.2-1B-IT								
Greedy CoT	**36.65**	**16.03**	13.37	34.44	**17.33**	9.41	**17.59**	22.94
HF no-intervention	**36.65**	15.82	13.00	**34.90**	**17.33**	10.08	16.67	**22.98**
Reasoning-critical neurons (Baseline)	**36.65**	15.82	13.00	**34.90**	**17.33**	10.08	16.67	**22.98**
Matched random	30.83	13.71	11.54	31.23	**17.33**	9.41	16.67	20.48
Ours (ratio)	35.21	15.82	**13.74**	33.07	17.12	**10.52**	15.00	22.28
Llama 3.2-3B-IT								
Greedy CoT	62.17	31.22	27.66	55.45	**33.40**	21.37	36.30	41.38
HF no-intervention	62.26	30.38	27.66	54.54	31.52	22.15	**37.04**	41.20
Reasoning-critical neurons (Baseline)	**62.76**	29.75	27.66	55.11	31.32	22.15	**37.04**	41.34
Matched random	57.29	26.37	28.02	53.50	29.44	21.15	33.33	38.72
Ours (ratio)	61.25	**32.70**	**28.75**	**57.06**	31.32	**23.03**	34.81	**41.64**

Table 1: Seven-subject official MATH comparison using the Qwen mathematical equivalence parser. "Reasoning-critical neurons" denotes our same-pipeline reproduction of Tang et al. (2025), while "Ours" uses train-derived activation-ratio scores with threshold > 1.10. Avg. denotes pooled accuracy over the 5,000-example MATH test split. Differences are descriptive unless paired tests are reported.

contrast remain competitive with a closely related quality-neuron intervention when both are evaluated under the same split, parser, and subject protocol. The mixed Llama 3.2 results are informative in this respect: they show that the sparse ratio variant is not a model-agnostic accuracy-improvement method, reinforcing the need to report length changes, matched random controls, and task-specific held-out analyses separately.

5.2 Alpha Scaling Gives a Graded Length-Control Signal

Table 2 tests whether reducing α shortens reasoning more for selected neurons than for matched random neurons. On Qwen3-1.7B GSM8K, selected-neuron scaling yields a monotonic mean-token reduction from $\alpha = 0.75$ to 0.0, whereas the random control has near-zero median changes and much smaller mean reductions (Appendix Figure 3). Accuracy changes are treated as task- and model-specific.

At $\alpha = 0.25$, symbolic rescoring gives 88.25% accuracy versus 83.09% (+5.16pp, 95% CI $[+3.49, +6.82]$); the token change is -343.5, CI $[-380.1, -307.0]$. Five layer-matched random sets give -2.4 ± 34.0 tokens, while selected interventions across three decoding seeds give -319.1 ± 26.0 (mean±SD). These setting-specific results supplement the exact-match table (Appendix C).

Setting	α	Acc.	Δ Acc.	Mean Tok.	Δ Tok.	Med. Δ
HF no-int.	1.00	83.09	–	2297.3	–	–
Selected	0.75	84.31	+1.21	2191.7	-105.7	-33
Selected	0.50	84.53	+1.44	2072.8	-224.5	-159
Selected	0.25	**87.11**	**+4.02**	1953.9	-343.5	-250
Selected	0.00	86.05	+2.96	**1896.0**	**-401.4**	**-303**
Matched rand.	0.75	83.32	+0.23	2288.0	-9.3	0
Matched rand.	0.50	84.61	+1.52	2256.9	-40.4	0
Matched rand.	0.25	83.32	+0.23	2259.4	-37.9	0
Matched rand.	0.00	84.84	+1.74	2243.9	-53.4	0

Table 2: GSM8K alpha sweep with Qwen3-1.7B, derivation-prompt discovery, $k_{\text{coord}} = 512$, and max-new-tokens 4096. Deltas are relative to HF no-intervention. Accuracy uses the original normalized exact-match evaluator.

5.3 Main Thinking-Model Held-Out Specificity

Table 3 reports the most selective held-out result for the main Qwen3-4B-Thinking-2507 family. On CSQA with an 8192-token budget, selected-neuron scaling gives a small single-run accuracy

Setting	α	Acc.	Δ Acc.	Mean Tok.	Δ Tok.
HF no-int.	1.00	59.38	–	2464.7	–
Selected	0.25	**60.20**	**+0.82**	**2433.7**	**-31.0**
Matched rand.	0.25	59.21	-0.16	2492.7	+28.0

Table 3: Main Qwen3-4B-Thinking-2507 held-out specificity result on CSQA validation with train-derived CSQA neurons and max-new-tokens 8192. The selected-neuron row has a paired selected–random token-delta gap of -59.0 tokens with an approximate 95% example-bootstrap confidence interval of $[-101.5, -16.5]$. The accuracy difference is a single-run descriptive result.

increase $(+0.82)$ and reduces mean length by 31 tokens, whereas the matched random intervention slightly reduces accuracy and increases length. The selected–random paired token gap is -59.0 tokens with an approximate 95% example-bootstrap interval of $[-101.5, -16.5]$. Although the effect size is modest, this setting uses train-derived neurons, held-out validation examples, matched random controls, and an 8192-token decoding budget.

5.4 Additional Analyses from Review

Review-stage analyses on GSM8K/Qwen3-1.7B provide additional controls. Across long/base evaluation prompts, selected–random token gaps are -234.2 $[-271.6, -198.6]$ and -249.8 $[-286.1, -214.6]$. Both length intervals exclude zero; the long-prompt exact-accuracy gap is $+1.21$pp $[-0.30, +2.73]$. Concise prompting is a strong control: adding neuron scaling removes 211.6 tokens $[-246.6, -176.5]$, with an exact-accuracy difference of $+0.23$pp $[-1.29, +1.67]$. A difference was not detected; this does not establish equivalence. Lexical verification markers decrease 19.2% alongside a 15.0% token reduction, characterizing output behavior without assigning individual-neuron semantics. Under symbolic scoring, 97 examples improve and 29 deteriorate; 70 improvements move from capped baselines below the cap, while 17 remain capped. These patterns associate gains with budget pressure without establishing causal mediation. These observations are limited to the tested model and settings. Complete tables, evaluator-specific statistics, and failure cases appear in Appendix C.

6 Discussion

The results support two conclusions. First, reasoning length is not purely an output-level artifact of prompting or decoding: sparse MLP neurons can provide a targeted intervention mechanism for generation length in some settings. Second, specificity is essential. Specificity evidence is strongest in settings where selected neurons shorten generation more than layer-matched random neurons while preserving accuracy.

This distinction matters because several simpler mechanisms can also shorten outputs. A smaller token budget truncates the answer, and a concise prompt can change the entire input distribution. In contrast, the intervention studied here leaves the prompt and decoding budget fixed and changes only a sparse set of MLP activation coordinates during generated-token decoding. The alpha sweep therefore tests a more local question: whether reducing the contribution of neurons associated with longer successful trajectories changes the realized length of otherwise comparable generations. The random-neuron controls are important for this interpretation, since scaling arbitrary MLP coordinates can also perturb generation length or accuracy.

The evidence also suggests that reasoning length and reasoning quality are partially entangled rather than independent factors. The default both-correct discovery filter is designed to reduce the most direct correctness confound, but it does not guarantee that the selected neurons encode only length. Long successful trajectories may contain verification, alternative derivations, or task-specific styles that are useful on some examples and unnecessary on others. This helps explain why the MATH comparison shows small gains under a sparse ratio threshold, whereas the same intervention family does not uniformly improve every model or subject. The method should therefore be viewed as a way to expose and test length-associated internal features, not as an unconditional recipe for improving accuracy.

From a practical perspective, the desired control is not simply "make the model shorter." Shorter reasoning is beneficial when the omitted tokens are redundant, repetitive, or distractor-driven, but harmful when they contain necessary computation. A useful intervention should therefore provide a continuous length-control parameter whose effect can be checked against task accuracy and matched random controls. This framing is consistent with the observed task-dependent behavior: the most informative results are those in which selected neurons produce a larger length change than random neurons without a corresponding accuracy collapse.

The findings should not be interpreted as evidence for universal, model-independent "reasoning neurons." The selected neurons are associated with reasoning length under a given model, task distribution, prompt protocol, and scoring procedure. Accordingly, reasoning-length-associated neurons are treated as candidate behavioral features validated by intervention rather than fixed semantic objects.

7 Conclusion

This paper presented a method for identifying and manipulating candidate reasoning-length-associated MLP neurons. By contrasting long and concise correct trajectories for the same input, the method selects neurons preferentially active during longer successful reasoning. Scaling these neurons modulates output length in some settings more strongly than layer-matched random controls, although the evidence varies across tasks, models, and budgets. The split-separated framework offers a lightweight way to test neuron-level control signals for generated reasoning length under specific task and prompt regimes.

Limitations

First, exact-match evaluation is brittle for mathematical expressions, especially on MATH500; symbolic verification is necessary for robust evaluation. We additionally report symbolic rescoring for the principal GSM8K and MATH500 sweeps. Second, the selected neurons are MLP intermediate dimensions, not human-interpretable semantic units. Third, results depend on prompt protocol, sampling, and maximum output length, so practical 4096-token and high-budget settings must be reported separately; prompt-induced contrasts may not recover prompt-invariant reasoning-length mechanisms. Fourth, most accuracy changes are single-run descriptive results, and our matched random controls use fixed random draws unless otherwise stated, so they do not capture all random-set or decoding-seed uncertainty. The principal GSM8K analysis includes five random sets and three decoding seeds; other settings have more limited coverage. Fifth, generated-token-only activation scaling and all-token or weight-scaling approximations are different interventions; backend comparisons should be treated as approximation checks rather than identical replacements. Finally, necessity and sufficiency of the selected neurons are not established.

Acknowledgments

This work was supported by the Institute of Information & communications Technology Planning & Evaluation (IITP) grants funded by the Korea government (MSIT): No. RS-2020-II201336, Artificial Intelligence graduate school support (UNIST); and No. RS-2026-25523906, Hyper-scale Industrial AI Research Support (R&D) Program, Development of an industry-specified multimodal hyper-scale foundation model.

References

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

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

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pages 8493–8502.

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*, pages 5484–5495.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Daya Guo, Dejian Yang, Haowei Zhang, and 1 others. 2025. Deepseek-r1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645(8081):633–638.

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*, volume 1.

Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. 2020. QASC: A dataset for question answering via sentence composition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8082–8090.

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*.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023. Inference-time intervention: Eliciting truthful answers from a language model. In *Advances in Neural Information Processing Systems*.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, Charles Sutton, and Augustus Odena. 2021. Show your work: Scratchpads for intermediate computation with language models. *arXiv preprint arXiv:2112.00114*.

Leheng Sheng, An Zhang, Zijian Wu, Weixiang Zhao, Changshuo Shen, Yi Zhang, Xiang Wang, and Tat-Seng Chua. 2025. On reasoning strength planning in large reasoning models. In *Advances in Neural Information Processing Systems*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4149–4158.

Yiru Tang, Kun Zhou, Yingqian Min, Wayne Xin Zhao, Jing Sha, Zhichao Sheng, and Shijin Wang. 2025. Enhancing chain-of-thought reasoning via neuron activation differential analysis. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 16151–16159.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. 2024. Steering language models with activation engineering. *Preprint*, arXiv:2308.10248.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. 2024. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*.

An Yang and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

A Extended Results and Reproducibility Details

The main text reports the tables needed to support the central claims. This appendix provides protocol details, prompts, parser checks, and supporting results needed to reproduce the study and interpret the main tables. For numeric appendix tables, bold marks the best value within the comparable task/model block. Higher is better for accuracy, accuracy delta, pair count, and reduction rate; lower is better for token count and token delta when the goal is compression. Qualitative comparison tables do not have an ordered best value. For delta columns, unfavorable values are not bolded solely because they are numerically maximal within a block.

A.1 Model and Split Details

A.2 Prompt Templates, Decoding, and Provenance

All rollouts use the model tokenizer's chat template when available. Each example is formatted as a single user message. For `base`, the user message contains only the question. For all other prompt styles, the user message is the question followed by a blank line and the instruction below. The assistant generation is then started with the model's normal generation prompt. The intervention is therefore evaluated on generated tokens only; prior conversation history is not modified.

Model	Where Used	Role
Qwen3-4B-Thinking-2507	CSQA main result; appendix details	Main thinking-model held-out specificity
`Qwen/Qwen2.5-Math-1.5B-Instruct`; Llama 3.2 1B/3B-IT	Official MATH comparison	Tang-style reproduction and direct comparison
Qwen3-1.7B	GSM8K alpha-scaling table	Graded length-control evaluation
Qwen3-4B	Appendix MATH details	Non-Thinking Qwen3 check
DeepSeek-R1-Distill-Qwen-7B Qwen3-4B-Instruct-2507	Appendix supporting experiments	Cross-model supporting evidence

Table 4: Model roles. Results are not pooled across these model families; each row is interpreted within its own protocol. Model citations are provided in Section 4.

Task	Discovery source	Eval source	Role
GSM8K (Cobbe et al., 2021)	train	test	math
Official MATH	competition-MATH first 7,500	competition-MATH last 5,000	math comparison
MATH500 (Hendrycks et al., 2021; Lightman et al., 2024)	non-overlapping MATH train examples	500-example MATH500 test	math
ARC-C (Clark et al., 2018)	train	test	science MCQ
CSQA (Talmor et al., 2019)	train	validation	commonsense MCQ
QASC (Khot et al., 2020)	train	validation	science/commonsense MCQ

Table 5: Split-separated task protocol. Held-out evaluation splits are never used for neuron discovery. The protocol distinguishes the official seven-subject MATH comparison, which uses the 12,500-example competition-MATH order with 7,500 train-side examples and 5,000 held-out test examples, from MATH500 on the separate 500-example benchmark. The exact split membership is fixed before neuron discovery and stored in the experiment metadata; examples are never reassigned after seeing evaluation outputs. For MATH500, overlap is removed on the train side before activation extraction. QASC uses validation as the labeled held-out split because the test split does not provide usable answer labels. HellaSwag-scale (Zellers et al., 2019) full discovery is deferred because exhaustive paired discovery at that scale is substantially more expensive than the benchmarks studied here.

Math prompt templates.

- `base`: no additional instruction.
- `long`: "Please reason step by step, and put your final answer within `\boxed{}`."
- `short`: "Please reason concisely step by step, and put your final answer within `\boxed{}`."
- `neutral`: "Put your final answer within `\boxed{}`."
- `detail_long`: "Please solve the problem using detailed step-by-step reasoning. Include all necessary intermediate steps before giving the final answer within `\boxed{}`."
- `detail_short`: "Please solve the problem using concise step-by-step reasoning. Include only the essential steps before giving the final answer within `\boxed{}`."
- `derivation_long`: "Please solve the problem with a detailed derivation. Do not skip intermediate algebraic or logical steps. Put your final answer within `\boxed{}`."
- `derivation_short`: "Please solve the problem with a minimal derivation. Skip routine algebraic or obvious steps, but keep enough reasoning to justify the answer. Put your final answer within `\boxed{}`."
- `strong_long`: "Please solve carefully and thoroughly. Think through the problem step by step, include the necessary intermediate derivations, and verify your solution before giving the final answer within `\boxed{}`."

- `strong_short`: "Please solve using the fewest necessary steps. Avoid unnecessary explanation and do not include verification unless essential. Put your final answer within `\boxed{}`."

Multiple-choice prompt templates.

- `base`: no additional instruction.
- `long`: "Please reason step by step. Please show your choice in the answer field with only the choice letter, e.g., `"answer": "C"`."
- `short`: "Please reason concisely step by step. Please show your choice in the answer field with only the choice letter, e.g., `"answer": "C"`."
- `neutral`: "Please show your choice in the answer field with only the choice letter, e.g., `"answer": "C"`."
- `detail_long` / `derivation_long`: "Please reason through the question in detailed steps before choosing. Please show your choice in the answer field with only the choice letter, e.g., `"answer": "C"`."
- `detail_short` / `derivation_short`: "Please reason concisely and include only the essential steps. Please show your choice in the answer field with only the choice letter, e.g., `"answer": "C"`."
- `strong_long`: "Please reason carefully and thoroughly before choosing. Please show your choice in the answer field with only the choice letter, e.g., `"answer": "C"`."
- `strong_short`: "Please answer using the fewest necessary reasoning steps. Please show your choice in the answer field with only the choice letter, e.g., `"answer": "C"`."

The `strong_*` prompts are documented for reproducibility but are not used for main discovery because the strong-long wording explicitly encourages verification, which introduces a verification confound for length-only discovery.

Which prompts are used where. The default split protocol uses `long/short` for discovery and `long` for held-out evaluation. The official MATH seven-subject comparison uses train-side discovery and base-prompt held-out evaluation; its "Ours" row uses the sparse ratio-threshold rule in Table 8. The Qwen3-1.7B GSM8K graded-intervention table uses `derivation_long/derivation_short` for discovery and `derivation_long` for evaluation. The CSQA main specificity table uses default `long/short` discovery and `long` evaluation. The detail-prompt setting uses `detail_long/detail_short` for discovery and `detail_long` for evaluation. The base-prompt setting keeps the train-derived default neuron set fixed but evaluates with `base`.

Decoding and intervention settings. For Qwen3-4B-Thinking-2507 and related Qwen3 thinking-model runs, sampling uses temperature 0.6, top-$p = 0.95$, top-$k_{\text{sample}} = 20$, and min-$p = 0$, following the model recommendation. vLLM is used for non-intervened rollouts when possible and HF generation for neuron intervention. Intervention is applied only to generated tokens, not to prompt tokens. Unless otherwise stated, the selected neuron set has $k_{\text{coord}} = 512$, activations are summarized over the first 256 generated positions, and candidate neurons are layer-coordinate pairs in the gated MLP down-projection input.

Provenance fields. Intervention outputs store the train-side neuron source in metadata, including `source_task`, `source_dataset`, `source_split`, `source_model`, `source_max_tokens`, `source_k` (the selected-neuron count), and `source_score_path`. Rows missing this provenance are not used for main claims. The guard prevents discovery on the same held-out split used for evaluation, which would leak test labels and invalidate the intervention interpretation.

A.3 Protocol and Parser Details

Table 6 clarifies the relation to Tang et al. (2025). The methods share the broad neuron-differential template, but differ in the target behavior: quality/correctness in the prior work versus reasoning length among correct trajectories in our setting.

Table 7 records the official-parser rediscovery audit. The large gap between strict and official labels indicates that test-output regrading alone is insufficient: train-side discovery labels also require the same mathematical-equivalence parser.

Table 8 records the exploratory CP pilot that motivated the sparse ratio threshold. It uses all paired trajectories and the earlier local exact-match evaluator; it is not the seven-subject official-parser run. The ratio score uses $\log((\text{mean}_i|a^L_{i,\ell,j}| +$

Dimension	Tang et al. (2025)	Ours
Target behavior	CoT quality / correctness; high-quality chains are contrasted with low-quality chains.	Reasoning length / overthinking; long correct chains are contrasted with concise correct chains for the same input.
Discovery filter	Correctness and manual quality filtering over sampled MATH trajectories.	Both-correct filtering plus length-ratio threshold $\rho = 1.2$, with discovery restricted to train split.
Neuron score	Ratio of average FFN activation between high- and low-quality chains, followed by thresholding.	Paired long-minus-concise standardized activation contrast over MLP down-projection input neurons, followed by ranking and selecting k_{coord} neurons.
Intervention	Stimulate or suppress selected reasoning-critical neurons to improve or damage reasoning accuracy.	Scale selected reasoning-length-associated neurons by α during generated-token decoding to test length and accuracy changes.
Primary metric	Accuracy gain over greedy CoT.	Accuracy–length trade-off, token reduction, and matched-random specificity.
Direct comparison status	Published baseline to reproduce.	Seven-subject Qwen2.5-Math reproduction and sparse ratio-threshold analysis under the same split-separated evaluation principle.

Table 6: Methodological comparison against the closest prior neuron differential method.

Subject	Train Official Acc.	Strict Acc.	Recovered / Lost	Quality Pairs	Length Pairs	Ratio > 1.10 Neurons
Algebra	**95.54**	**89.99**	403 / 16	94	**94**	81
CP	75.45	65.66	302 / 0	135	42	71
PC	76.58	66.99	291 / 5	110	41	53
PA	92.41	81.85	**509 / 0**	101	62	66
Geometry	61.55	55.34	216 / 0	181	34	60
IA	70.91	65.29	303 / 12	**294**	64	55
NT	79.78	77.42	82 / 0	145	60	**42**

Table 7: Official MATH train rediscovery audit with `Qwen/Qwen2.5-Math-1.5B-Instruct`. "Strict" is the earlier local string/equivalence evaluator; "Official" uses the Qwen mathematical parser and grader.

$\epsilon)/(\text{mean}_i|a^S_{i,\ell,j}| + \epsilon))$, not a raw signed activation ratio. Broad ratio selection reduces output length at the cost of substantial accuracy degradation, whereas the sparse $r_{\ell,j} > 1.10$ set is more selective.

A.4 Main-Model Alpha and k_{coord} Diagnostics

Table 9 gives the broader Qwen3-4B-Thinking-2507 held-out sweep used to contextualize the main CSQA result.

Table 10 reports the selected-set-size comparison for the CSQA specificity result.

Table 11 reports supporting GSM8K prompt and selected-set-size comparisons.

A.5 Prompt and Budget Details

Table 12 summarizes the prompt contrasts and budgets used to construct long/concise discovery pairs.

A.6 Supporting Prior Experiments

Tables 13 and 14 preserve supporting cross-model evidence. These rows are not pooled with the new main experiments; they document earlier model/task combinations in which selected-neuron intervention produced correctness-preserving compression relative to matched random controls.

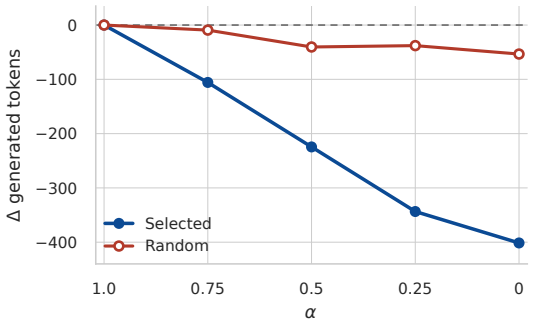


Figure 3: Token-side companion to Table 2 for the Qwen3-1.7B GSM8K alpha sweep. Selected neurons show a graded reduction as α decreases, while matched random neurons remain comparatively flat.

Condition	Selection	Neurons	Acc.	Δ Acc.	Mean Tok.	Δ Tok.	Median Tok.
HF no-intervention	–	–	62.24	–	519.84	–	438.5
Diff all-pairs	top-ranked	512	61.81	-0.43	525.69	+5.85	437.0
Ratio	top-ranked	512	60.76	-1.48	528.95	+9.11	450.0
Ratio random	top-ranked	512	59.70	-2.54	531.07	+11.23	449.0
Ratio	top 10%	25,088	56.54	-5.70	**484.08**	**-35.76**	**423.0**
Ratio random	top 10%	25,088	56.33	-5.91	511.96	-7.88	442.5
Ratio	> 1.10	71	**64.14**	**+1.90**	519.18	-0.66	433.5
Ratio random	> 1.10	71	60.76	-1.48	528.10	+8.26	442.5
Ratio	> 1.05	1,901	61.81	-0.43	506.43	-13.41	435.5
Ratio random	> 1.05	1,901	62.03	-0.21	513.48	-6.36	449.5

Table 8: Exploratory MATH CP comparison of difference-based and ratio-based neuron scoring with `Qwen/Qwen2.5-Math-1.5B-Instruct`. Deltas are relative to HF no-intervention on the same 474 CP test examples, using the earlier local exact-match evaluator.

Task	Budget	Setting	α	Acc.	Δ Acc.	Mean Tok.	Δ Tok.
GSM8K	4096	Baseline	1.00	**91.74**	–	1405.4	–
GSM8K	4096	Selected	0.25	90.83	-0.91	**1372.4**	**-33.1**
GSM8K	4096	Matched random	0.25	91.43	-0.31	1396.8	-8.6
MATH500	4096	Baseline	1.00	46.00	–	**3031.4**	–
MATH500	4096	Selected	0.25	**46.20**	**+0.20**	3050.2	+18.8
ARC-C	4096	Baseline	1.00	**74.23**	–	1649.7	–
ARC-C	4096	Selected	0.25	72.78	-1.45	**1587.2**	**-62.5**
ARC-C	4096	Matched random	0.25	73.46	-0.77	1642.7	-7.0
CSQA	4096	Baseline	1.00	58.23	–	2380.2	–
CSQA	4096	Selected	0.25	**60.69**	**+2.46**	**2342.6**	**-37.6**
CSQA	4096	Matched random	0.25	56.84	-1.39	2404.1	+23.9
CSQA	8192	Baseline	1.00	59.38	–	2464.7	–
CSQA	8192	Selected	0.25	**60.20**	**+0.82**	**2433.7**	**-31.0**
CSQA	8192	Matched random	0.25	59.21	-0.16	2492.7	+28.0
QASC	4096	Baseline	1.00	**57.45**	–	2298.7	–
QASC	4096	Selected	0.25	55.40	-2.05	2304.6	+5.9
QASC	4096	Matched random	0.25	55.40	-2.05	**2264.4**	**-34.3**

Table 9: Qwen3-4B-Thinking-2507 held-out results with $k_{\text{coord}} = 512$, shown at the key operating point $\alpha = 0.25$. Math accuracy retains the original normalized exact-match scoring; symbolic MATH500 results are reported in Appendix C.

Task	k_{coord}	Setting	Acc.	Δ Acc.	Mean Tok.	Δ Tok.	Selected–Random Δ Tok.
CSQA	64	Selected	58.56	-0.82	2451.6	-13.1	-15.7 $[-58.0, +26.6]$
CSQA	64	Matched random	60.03	+0.66	2467.3	+2.5	–
CSQA	256	Selected	58.89	-0.49	2453.8	-10.9	-26.3 $[-67.7, +15.1]$
CSQA	256	Matched random	58.64	-0.74	2480.1	+15.4	–
CSQA	512	Selected	**60.20**	**+0.82**	**2433.7**	**-31.0**	**-59.0** $[-101.5, -16.5]$
CSQA	512	Matched random	59.21	-0.16	2492.7	+28.0	–

Table 10: CSQA max-new-tokens 8192 selected-set-size sensitivity for Qwen3-4B-Thinking-2507 at $\alpha = 0.25$. All rows use train-derived CSQA neurons and the same held-out validation baseline.

B Reproducibility Details

- Main selected neuron count: $k_{\text{coord}} = 512$.
- Selected-set-size comparisons: $k_{\text{coord}} \in \{64, 256, 512\}$ where reported.

Eval prompt	Neuron set	Setting	Acc.	Δ Acc.	Mean Tok.	Δ Tok.
Long	$k_{\text{coord}} = 64$	Baseline	**91.74**	–	1405.4	–
Long	$k_{\text{coord}} = 64$	Selected, $\alpha = 0.25$	90.52	-1.21	1408.8	+3.3
Long	$k_{\text{coord}} = 64$	Matched random, $\alpha = 0.25$	91.21	-0.53	**1390.1**	**-15.3**
Base	$k_{\text{coord}} = 512$	Baseline	82.26	–	1625.0	–
Base	$k_{\text{coord}} = 512$	Selected, $\alpha = 0.25$	87.26	+5.00	**1349.5**	**-275.5**
Base	$k_{\text{coord}} = 512$	Matched random, $\alpha = 0.25$	**87.64**	**+5.38**	1363.8	-261.2

Table 11: GSM8K prompt and neuron-set comparisons with Qwen3-4B-Thinking-2507 and max-new-tokens 4096. Deltas are relative to the prompt-matched baseline.

Model	Task	Prompt contrast	Budget	Pair count	Mean long-concise Δ
Qwen3-4B-Thinking-2507	MATH500	default long/concise	8192	122	+249.7
Qwen3-4B-Thinking-2507	MATH500	detail long/concise	8192	295	**+1076.0**
Qwen3-4B-Thinking-2507	MATH500	derivation long/concise	8192	225	+769.9
Qwen3-4B-Thinking-2507	ARC-C	detail long/concise	4096	477	+793.0
Qwen3-4B-Thinking-2507	CSQA	detail long/concise	4096	312	+961.0
Qwen3-4B-Thinking-2507	QASC	detail long/concise	4096	194	+811.5
Qwen3-4B-Thinking-2507	QASC	detail long/concise	8192	183	+980.2
Qwen3 1.7B	GSM8K	derivation long/concise	4096	4947	+595.7
Qwen3-4B	MATH500	derivation long/concise	4096	4525	+673.0

Table 12: Prompt contrast and budget-sensitivity experiments. Pair count is the number of retained both-correct pairs satisfying the length-ratio filter.

Task	Model	Setting	α	Acc.	Δ Acc.	Mean Tok.	Δ Tok. / %
GSM8K	DeepSeek-R1-Distill-Qwen-7B	Baseline	1.00	78.54	–	612.6	–
GSM8K	DeepSeek-R1-Distill-Qwen-7B	Selected	0.25	77.79	-0.75	**590.3**	**-22.3 / -3.6%**
GSM8K	DeepSeek-R1-Distill-Qwen-7B	Matched random	0.25	**78.62**	**+0.08**	628.2	+15.6 / +2.5%
MATH500	DeepSeek-R1-Distill-Qwen-7B	Baseline	1.00	50.80	–	2483.9	–
MATH500	DeepSeek-R1-Distill-Qwen-7B	Selected	0.25	50.60	-0.20	**2107.8**	**-376.1 / -15.1%**
MATH500	DeepSeek-R1-Distill-Qwen-7B	Matched random	0.25	**51.20**	**+0.40**	2488.5	+4.6 / +0.2%
ARC-C	DeepSeek-R1-Distill-Qwen-7B	Baseline	1.00	71.42	–	635.5	–
ARC-C	DeepSeek-R1-Distill-Qwen-7B	Selected	0.25	**73.72**	**+2.30**	**593.9**	**-41.6 / -6.5%**
ARC-C	DeepSeek-R1-Distill-Qwen-7B	Matched random	0.25	71.84	+0.43	639.2	+3.7 / +0.6%
CSQA	DeepSeek-R1-Distill-Qwen-7B	Baseline	1.00	55.86	–	640.7	–
CSQA	DeepSeek-R1-Distill-Qwen-7B	Selected	0.25	**57.90**	**+2.05**	**606.5**	**-34.2 / -5.3%**
CSQA	DeepSeek-R1-Distill-Qwen-7B	Matched random	0.25	55.69	-0.17	645.0	+4.3 / +0.7%
ARC-C	Qwen3-4B-Instruct-2507	Baseline	1.00	91.89	–	459.4	–
ARC-C	Qwen3-4B-Instruct-2507	Selected	0.25	91.81	-0.08	**441.3**	**-18.1 / -3.9%**
ARC-C	Qwen3-4B-Instruct-2507	Matched random	0.25	**92.24**	**+0.34**	455.8	-3.6 / -0.8%
CSQA	Qwen3-4B-Instruct-2507	Baseline	1.00	80.59	–	468.6	–
CSQA	Qwen3-4B-Instruct-2507	Selected	0.25	80.34	-0.25	**457.4**	**-11.2 / -2.4%**
CSQA	Qwen3-4B-Instruct-2507	Matched random	0.25	**81.24**	**+0.66**	470.0	+1.4 / +0.3%

Table 13: Controlled cross-model supporting evidence.

- Activation site: MLP down-projection input.
- Pair filtering threshold: $\rho = 1.2$.
- Activation summary: first 256 generated positions.
- Scoring: positive signed standardized long-minus-concise contrast, population standard deviation, and $\epsilon = 10^{-6}$.
- Model identifiers: Qwen3-4B-Thinking-2507 denotes `Qwen/Qwen3-4B-Thinking-2507`; Qwen3-4B denotes `Qwen/Qwen3-4B`;

Task	Model	Setting	n	Base Tok.	Interv. Tok.	Mean Δ Tok. / %	Med. Δ	Red. Rate
MATH500	DeepSeek-R1-Distill-Qwen-7B	Selected	231	1875.5	**1490.8**	**-384.6 / -20.5%**	**-207.0**	**64.9**
MATH500	DeepSeek-R1-Distill-Qwen-7B	Matched random	**233**	**1844.1**	1870.3	+26.2 / +1.4%	+2.0	43.8
ARC-C	DeepSeek-R1-Distill-Qwen-7B	Selected	**747**	539.7	**498.7**	**-41.0 / -7.6%**	**-33.0**	**62.2**
ARC-C	DeepSeek-R1-Distill-Qwen-7B	Matched random	731	**534.3**	537.1	+2.8 / +0.5%	+0.0	49.9
CSQA	DeepSeek-R1-Distill-Qwen-7B	Selected	557	535.7	**495.6**	**-40.1 / -7.5%**	**-34.0**	**60.7**
CSQA	DeepSeek-R1-Distill-Qwen-7B	Matched random	**567**	**533.1**	533.7	+0.5 / +0.1%	+2.0	48.7
ARC-C	Qwen3-4B-Instruct-2507	Selected	1058	437.6	**420.5**	**-17.1 / -3.9%**	**-18.0**	**61.3**
ARC-C	Qwen3-4B-Instruct-2507	Matched random	**1059**	**437.0**	435.3	-1.8 / -0.4%	-2.0	53.0
CSQA	Qwen3-4B-Instruct-2507	Selected	933	**421.4**	**412.1**	**-9.3 / -2.2%**	**-7.0**	**53.8**
CSQA	Qwen3-4B-Instruct-2507	Matched random	**950**	425.3	423.7	-1.7 / -0.4%	+0.0	49.3

Table 14: Correct-to-correct pairwise token changes for $\alpha = 0.25$ in controlled supporting experiments. Each row uses its own both-correct subset, so selected and random rows need not contain the same examples.

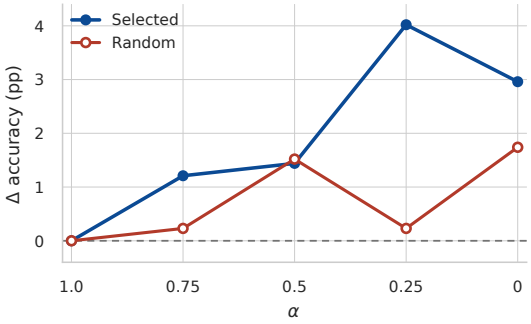


Figure 4: Accuracy-side companion to Figure 3 for the Qwen3-1.7B GSM8K alpha sweep under the original exact-match evaluator. Accuracy deltas are descriptive single-run changes relative to the no-intervention row in Table 2.

DeepSeek-R1-Distill-Qwen-7B denotes `deepseek-ai/DeepSeek-R1-Distill-Qwen-7B`. Experiment metadata record local checkpoint revisions and backend versions when available.

- Primary intervention: generated-token-only activation scaling.
- Matched random control: same count and layer distribution as the selected neurons, using a fixed random seed unless otherwise stated.
- Qwen3-4B-Thinking-2507 sampling: temperature 0.6, top-$p = 0.95$, top-$k_{\text{sample}} = 20$, min-$p = 0$.
- Budget selection: short rollout first; full run uses the smallest max length with acceptable truncation and answer-format failure rates.

C Additional Analyses

The following analyses supplement the original experiments. Unless otherwise specified, GSM8K results use Qwen3-1.7B, all 1,319 test questions, 512 derivation-derived coordinates, temperature 0.6, a 4096-token output budget, and selected-neuron scaling at $\alpha = 0.25$.

C.1 Random Sets and Decoding Seeds

Five layer-matched random sets (subset seeds 0–4) are evaluated at $\alpha \in \{0.25, 0\}$ with decoding seed 0. Separately, decoding seeds 0–2 are evaluated with paired no-intervention and selected-neuron runs at $\alpha = 0.25$, holding the discovered set fixed. These are complementary checks, not a five-by-three factorial experiment. At $\alpha = 0.25$, the random-set token change is -2.4 ± 34.0 (mean±sample SD), ranging from -37.9 to $+48.1$; the selected change is -343.5. Across decoding seeds, selected changes are $-343.5, -322.1, -291.7$ tokens, averaging -319.1 ± 26.0. Original normalized exact-match accuracy gains are $+4.02, +1.14, +3.03$ percentage points (pp), averaging $+2.73 \pm 1.46$pp. This evidence is specific to this setting; five random draws do not support a normal-reference significance test.

C.2 Prompt Transfer and External Controls

Prompt transfer holds the derivation-derived neuron set fixed while changing the evaluation prompt to `long` or question-only `base`. Each prompt has its own baseline and layer-matched random control, with decoding seed 0. Selected–baseline token changes are -232.8 and -293.9, respectively. Table 15 gives the selected–random comparisons.

The long-prompt interval containing zero concerns accuracy, not length. Transfer to two prompts does not establish prompt invariance.

Evaluation prompt	Token difference [95% CI]	Exact accuracy difference (pp) [95% CI]
	-234.2	$+1.21$
`long`	$[-271.6, -198.6]$	$[-0.30, +2.73]$
	-249.8	$+2.35$
`base`	$[-286.1, -214.6]$	$[+0.83, +3.87]$

Table 15: Selected minus matched random on GSM8K/Qwen3-1.7B. Accuracy uses the original normalized exact-match evaluator. Intervals use 10,000 paired bootstrap samples over questions and condition on the observed runs.

Table 16 compares external controls under the same model, questions, sampling, decoding seed, and HF backend. Concise prompting uses `derivation_short`; the hard budget keeps `derivation_long` and reduces the cap to 2048. The combination also scales the same coordinates. Relative to concise prompting, combining both interventions changes length by -211.6 tokens $[-246.6, -176.5]$ and exact accuracy by $+0.23$pp $[-1.29, +1.67]$, using paired bootstrap intervals; exact McNemar $p = 0.84$. This is not an equivalence or non-inferiority test.

Condition	Exact acc.	Tokens	s/ex.
Long prompt, no intervention	83.09	2297	12.6
Hard cap 2048	69.29	1741	5.4
Concise prompt	86.88	1698	10.1
Selected, $\alpha = 0.25$	87.11	1954	10.8
Concise + selected	87.11	1487	9.6

Table 16: External controls on GSM8K/Qwen3-1.7B. Tokens are mean generated tokens; time is amortized generation seconds per example at batch size 16, not single-query latency. All accuracies are exact-match percentages.

Concise prompting alone is shorter than scaling alone and has similar measured accuracy. Hard truncation lowers accuracy by 13.80pp: 209 of its 216 baseline-correct answers becoming incorrect reach the cap. Its timing therefore does not represent a quality-preserving speedup. Timing excludes discovery cost; dense steering and adaptive decoding are not compared.

C.3 Symbolic Scoring and Failure Analysis

Table 17 audits unchanged outputs using the Qwen2.5-Math answer extraction and mathematical-equivalence code (Yang et al., 2024). MATH500 references are treated as already extracted final answers. Rescoring evaluation outputs does not change the discovery labels or neuron sets. Robustness, prompt-transfer, and external control accuracy above retains the original exact-match scorer.

Condition	Exact acc.	Symbolic acc.	Recovered	Lost
Task / model: GSM8K / Qwen3-1.7B				
Baseline	83.09	83.09	19	19
Selected, $\alpha = 0.25$	87.11	88.25	21	6
Random, $\alpha = 0.25$	83.32	83.85	20	13
Task / model: MATH500 / Qwen3-4B-Thinking				
Baseline	46.00	48.80	27	13
Selected, $\alpha = 0.25$	46.20	49.60	30	13
Random, $\alpha = 0.25$	47.00	48.60	20	12

Table 17: Scoring audit on 1,319 GSM8K and 500 MATH500 examples, including the complete MATH500 random run. Recovered/lost count labels changing from incorrect/correct to correct/incorrect when switching scorers.

Identical aggregate GSM8K baseline accuracy masks 19 recovered and 19 lost labels. Paired correctness analyses must therefore use one scorer consistently. Under symbolic evaluation, the selected GSM8K intervention fixes 97 examples and breaks 29, giving $+5.16$pp $[+3.49, +6.82]$. Of the 97 fixes, 87 have capped baselines; 70 move below the cap and 17 remain capped. Of the 29 failures, 18 intervened outputs reach the cap; 4 are shorter, 9 equally long, and 16 longer than baseline. A cap hit means stored generated length at least 4096, not an independently recorded termination reason. These associations do not establish causal mediation by truncation. MATH500 selected/random accuracy changes order under rescoring, but the differences remain small.

Two symbolic correct-to-incorrect examples illustrate the limits of compression-based explanations. For `gsm8k_test_1003`, the intervention answers monthly cost (360) instead of the requested earnings (480), while length increases from 3078 to 3926 tokens. For `gsm8k_test_106`, the correct baseline answer 72 changes to 64; length increases from 3733 to 4096, with unresolved checking of incompatible volume calculations. These cases are illustrations, not an exhaustive error taxonomy.

C.4 Layers and Lexical Markers

In the 36-layer Qwen3-4B-Thinking-2507, layers 16–27 (zero-based) contain 62.5%, 42.4%, 55.1%, and 39.6% of the 512 selected coordinates for GSM8K, CSQA, ARC-C, and MATH500, respectively. These values describe that 4B model, rather than the 28-layer Qwen3-1.7B used in the principal sweep, and do not assign semantics to individual neurons.

For Qwen3-1.7B GSM8K outputs, a lexical proxy counts case-insensitive regular-expression matches in 16 groups: wait; hmm; actually; let me check/verify/reconsider/rethink/see; mistake/incorrect/error; reconsider/rethink; another way/alternative; hold on; not necessarily; on second thought; check; verify/verification; double-check; make sure; sanity check; and wrong. Overlapping groups are counted separately. Density is the per-example mean of $1000c_i/\max(t_i, 1)$, where c_i counts matches and t_i counts generated tokens. Selected scaling changes density from 10.06 to 9.51 per 1000 tokens (paired change -0.55, 95% CI $[-0.71, -0.40]$); random scaling changes it by $+0.07$ $[-0.08, +0.22]$. These counts are not verified reasoning operations. Discovery long/short densities also differ (10.23 versus 8.75), so the contrast does not isolate length from verification-related language.