OPENING MOVES: STEERING REASONING THROUGH THE FIRST TOKEN'S MEMORY

Anonymous authors
Paper under double-blind review

ABSTRACT

A reasoning model's first generated token can change thousands of tokens that follow. We study this sensitivity through *opening memory*, the keys and values cached at the opening-token position. Matched-text interventions show that exchanging opening keys or values changes output length without changing the visible opening, and that later cache entries can carry these effects under identical token histories. Based on these observations, we introduce **Opening-Memory Steering (O-Steer)**, which steers long reasoning trajectories through a single edit to the memory written by the opening token. O-Steer uses natural opening tokens to construct a compact edit space, complete-continuation feedback to select a shared memory anchor, and a small problem-dependent controller to adapt the edit. At inference time, the frozen model's opening memory is modified once, after which ordinary decoding proceeds without further intervention. Across 1,919 problems from six math benchmarks, O-Steer reduces mean output length by 40.3% on Qwen3-1.7B and 22.8% on DeepSeek-R1-Distill-Qwen-1.5B, with accuracy decreases of 5.91 and 4.17 percentage points, respectively. These results show that a single opening-memory edit can substantially change the accuracy–length trade-off of long reasoning trajectories.

1 INTRODUCTION

Reasoning models can spend thousands of tokens repeating solutions or continuing to explore after reaching a correct answer (Chen et al., 2025c). Existing approaches reduce this cost by detecting when to stop or by steering generation during reasoning (Yang et al., 2026a; Chen et al., 2025a; Li et al., 2026). In this work, we ask a complementary question: *how much can a long reasoning trajectory be controlled by modifying only its initial memory?*

An *opening token*, such as `Okay`, is the first generated content token after the fixed reasoning scaffold. Changing this token can alter reasoning behavior and output length (Wang & Zhou, 2024; Lin et al., 2025; Sinii et al., 2025). On 1,919 math problems, fixing the opening token of Qwen3-1.7B to `Yep`, selected on the fitting split, reduces the mean output length from 6,499 to 5,330 tokens (18.0%), while accuracy changes from 67.09% to 66.13%. This *Fixed Opening* baseline requires no further intervention during generation.

What carries this effect: the visible opening token, the memory written by the token, or both? We study *opening memory*, defined as the keys and values cached at the opening-token position. Attention sinks show that initial context positions can attract attention despite containing little task-specific information (Xiao et al., 2024); here, we directly intervene on the memory of a generated token without assuming that it is an attention sink. On a 600-problem development panel, exchanging the opening keys or values changes output length while keeping the displayed token, prompt cache, and already computed next-token logits unchanged. Moreover, under identical 128-token histories, exchanging later cache entries changes the remaining output length even after the opening entry is restored. Thus, the effect associated with opening memory can propagate to later memory and remain behaviorally active.

Based on these observations, we introduce **Opening-Memory Steering (O-Steer)**, which steers long reasoning trajectories through a single edit to the memory written by the opening token. Building on one-shot K/V-cache steering (Belitsky et al., 2025), O-Steer uses differences between opening memories to construct a compact edit space. Complete continuations under the resulting edits are then

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

used to select a shared *memory anchor* and train a small problem-dependent controller. At inference time, O-Steer modifies the native opening memory once, without changing the model parameters or visible text, and ordinary decoding proceeds without further intervention.

Across 1,919 problems from six math benchmarks, O-Steer reduces the mean output length by 40.3% on Qwen3-1.7B and 22.8% on DeepSeek-R1-Distill-Qwen-1.5B, with accuracy decreases of 5.91 and 4.17 percentage points, respectively. These results show that editing a single initial memory position can substantially change the accuracy–length trade-off of long reasoning trajectories.

Our contributions are threefold: (i) we use matched-text interventions to separate the effects of opening keys, opening values, and later memory on reasoning; (ii) we propose **O-Steer**, which learns problem-dependent steering from complete-continuation feedback and applies it through a single opening-memory edit; and (iii) we evaluate opening-memory steering across two reasoning models and six math benchmarks, comparing its accuracy–length trade-off with fixed openings, unlearned memory edits, and activation-steering methods.

2 How Opening Tokens Shape Reasoning

Opening tokens affect reasoning length and behavior (Lin et al., 2025; Sinii et al., 2025; Yang et al., 2026b). We examine their accuracy–length trade-offs, isolate the effect of their K/V memory, and test whether that effect extends to later memory under identical text. All analyses use frozen Qwen3-1.7B on 600 development problems: 200 each from GSM8K, MATH, and Omni-MATH (Appendix A.1).

2.1 Opening Choice Changes the Accuracy–Length Trade-off

We force one of sixteen opening tokens, then allow free generation. Each problem receives two generations per opening under the same decoder and a 16K output limit. We measure final-answer accuracy and output tokens, including incorrect and capped outputs.

Figure 1 shows substantial variation from this single-token change. Replacing `Okay` with `Yep` reduces mean length from 5,849 to 4,901 tokens, while observed accuracy is 71.92% and 72.00%, respectively. Shorter outputs do not always retain this accuracy: `Please` produces only 1,003 tokens on average, but accuracy falls to 59.58%.

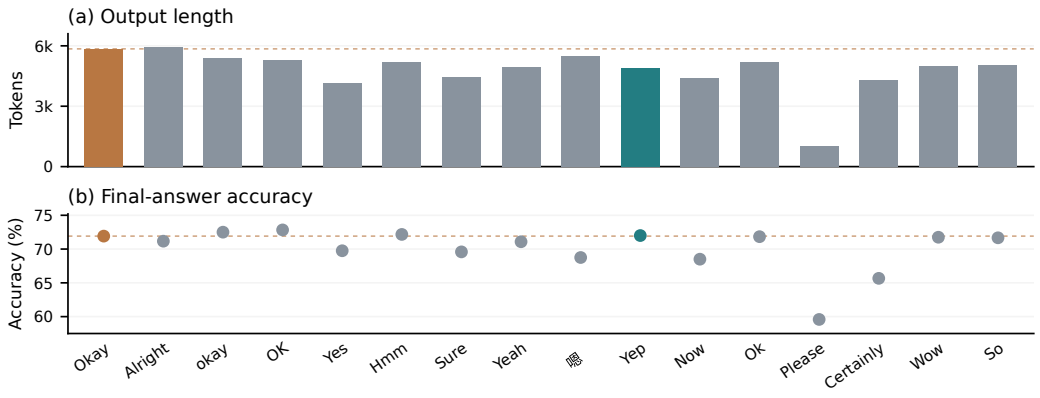

Figure 1: **Different opening tokens expose different accuracy–length trade-offs.** Mean output tokens and final-answer accuracy on 600 problems, with two generations per opening. Brown and teal highlight `Okay` and `Yep`; dashed lines mark the `Okay` reference.

Opening choice exposes scope for shortening, but also the risk of losing correct answers. We next isolate whether the token's stored memory provides a control point for this trade-off.

2.2 Opening Memory Changes the Continuation

Keys determine how strongly later queries attend to the opening position; values supply the information read from it. Building on one-shot cache steering (Belitsky et al., 2025), we test their effects separately while keeping the visible opening token fixed.

We use the `Okay`/`Yep` pair to examine the large length difference above. For each problem x, we obtain both opening memories and cross their keys and values:

$$M_{ab}(x) = \big(K_1(x, a), V_1(x, b)\big), \qquad a, b \in \{\texttt{Okay}, \texttt{Yep}\}. \tag{1}$$

All four conditions display `Okay`. After processing that token, we replace its cache entry with $M_{ab}(x)$ and resume generation, leaving the prompt cache and already computed next-token logits unchanged. Relative to the all-`Okay` memory, the other conditions exchange only keys, only values, or both.

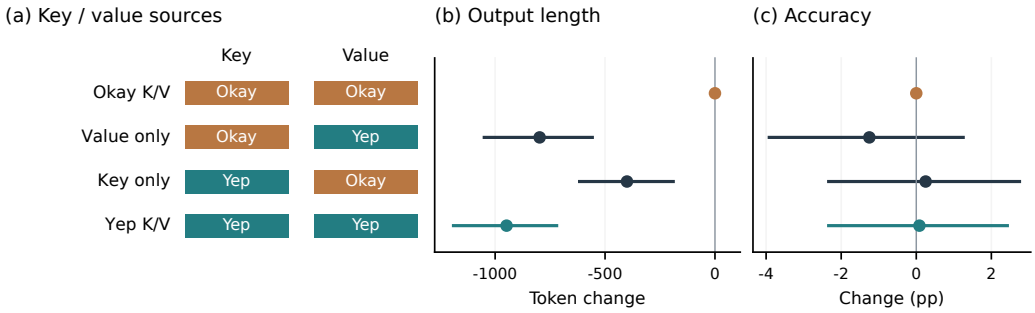

Figure 2: **Changing opening memory changes output length. (a)** Four combinations of `Okay` and `Yep` keys and values, all following the displayed `Okay`. **(b,c)** Accuracy and length changes from the all-`Okay` memory on 600 problems, with two generations per condition.

Exchanging keys alone, values alone, or both reduces mean output length by 400, 798, and 948 tokens, respectively (Figure 2). Values produce the larger individual reduction, but keys also shorten the continuation while the stored values remain unchanged.

Opening memory controls output length even with the visible token fixed. The distinct K/V effects motivate independently adjustable edit strengths. Appendix A.2 relates these interventions to the attention readout and the limits of attention weight as an explanation.

2.3 Opening Effects Extend to Later Memory

The preceding edits allow subsequent tokens to diverge. Does memory still affect output length under identical intervening text? Related work studies downstream caches during prefill and latent reasoning (Li, 2026; Ding et al., 2026); we test the remaining length of explicit reasoning.

For each recorded history, we replay identical continuation tokens under `Okay`- or `Yep`-conditioned opening memory. At a 128-token boundary, we separately exchange the opening entry and *later memory*: the K/V entries at positions 2–127. We then process the same 128th token before resuming generation, keeping the visible history identical across all four conditions. Appendix A.3 gives the full protocol.

Exchanging only the opening memory reduces mean remaining output length by 561 tokens. More importantly, exchanging only the later memory reduces it by 368 tokens even after the opening entry is restored to `Okay` (Figure 3). Thus, opening-dependent differences remain behaviorally active in later memory, even under the same visible reasoning history.

Together, these experiments identify opening memory as an initial control point for reasoning: its keys and values independently affect the continuation, and its influence can propagate to later memory. Motivated by these findings, Section 3 introduces O-Steer, which uses opening-induced memory differences to construct candidate edits and learns their strengths from complete-continuation accuracy and length.

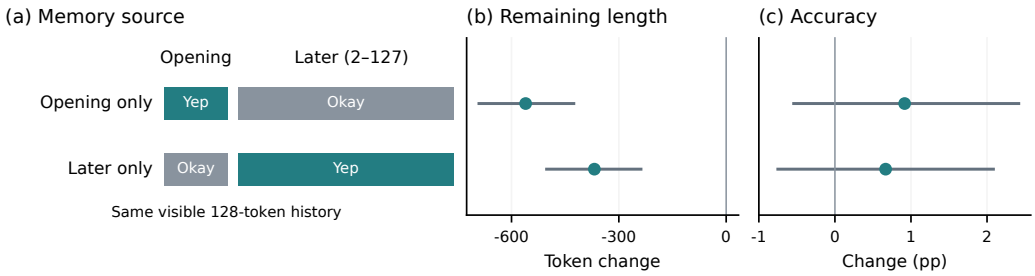

Figure 3: **Later memory retains an opening-dependent effect.** Under identical 128-token histories, **(a)** independently exchanges the opening and later K/V entries; **(b,c)** show changes in remaining output length and accuracy relative to the all-`Okay` cache. Results average over two recorded histories and two repetitions on 600 problems.

3 O-STEER: OPENING-MEMORY STEERING

Section 2 shows that opening memory provides an initial control point for reasoning: its K/V components affect the continuation, and opening-dependent effects can remain active in later memory. Based on these findings, we introduce *O-Steer (Opening-Memory Steering)*, which learns to steer reasoning through a single edit to the memory written by the opening token.

O-Steer has two stages. Offline, natural opening tokens define candidate memory directions, from which we select a shared *memory anchor* and learn a small problem-dependent correction. At inference, O-Steer edits the native opening memory once and then resumes ordinary decoding without further intervention (Figure 4).

3.1 OBJECTIVE AND OPENING-MEMORY INTERVENTION

For a problem x, *Native* denotes generation without a forced opening token or memory edit. Let $m_1(x) = (K_1(x), V_1(x)) \in \mathbb{R}^D$ denote its *opening memory*, obtained by concatenating the keys and values at the opening-token position across layers and key–value heads. Keys are represented in a common positional frame and rotated back when an edit is applied.

After the native opening token is processed, O-Steer applies a single additive edit $\Delta(x) \in \mathbb{R}^D$:

$$m_1(x) \leftarrow m_1(x) + \Delta(x). \tag{2}$$

The model parameters, prompt memory, visible opening token, and already computed next-token logits remain unchanged.

Let $A(\Delta)$ and $L(\Delta)$ denote the expected final-answer accuracy and output length under edit policy Δ, respectively, where the expectation is over problems and decoding randomness. We seek to shorten generation while limiting accuracy degradation:

$$\min_{\Delta} L(\Delta) \quad \text{subject to} \quad A(\Delta) \geq A_{\text{Native}} - \varepsilon, \tag{3}$$

where A_{Native} is the accuracy without intervention and ε is an accuracy tolerance. For example, $\varepsilon = 0.05$ allows an absolute decrease of five percentage points. The same tolerance is used for empirical anchor selection and controller fitting.

3.2 CONSTRUCTING A SHARED MEMORY ANCHOR

We first construct memory directions from natural opening tokens. For each of 16 reference openings c, we compare the memory written by the forced opening with the Native memory on the same training problem:

$$d_c = \frac{1}{n} \sum_{i=1}^{n} \left(m_1(x_i; c) - m_1(x_i; \text{Native}) \right). \tag{4}$$

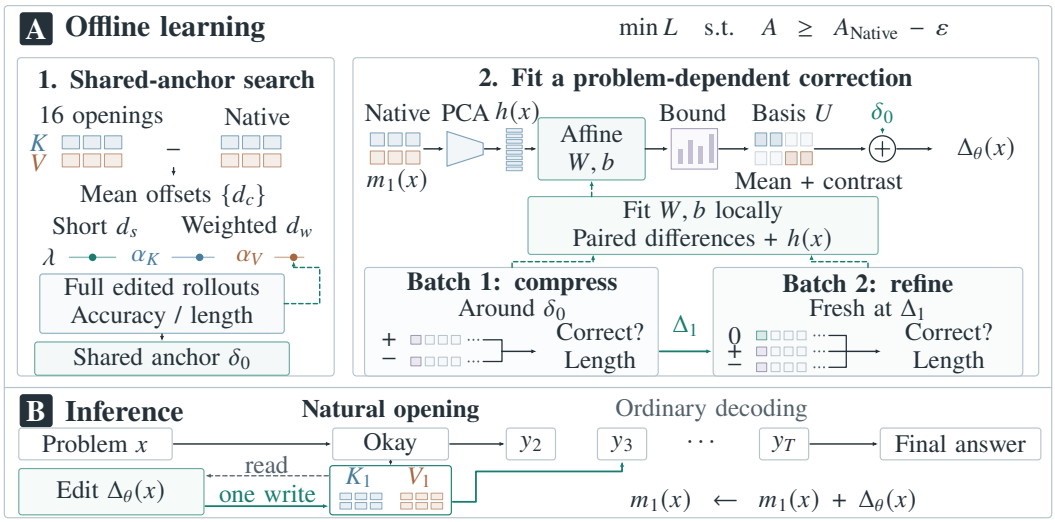

Figure 4: **Overview of O-Steer. (A)** Offline, K/V differences induced by natural opening tokens define candidate edits. Complete-continuation feedback selects a shared memory anchor and learns a problem-dependent correction through compression and refinement. **(B)** At inference, O-Steer reads and edits the native opening memory once, after which ordinary decoding continues without further intervention.

Each $d_c = (d_{c,K}, d_{c,V})$ is therefore an average opening-induced K/V difference that can be applied as an additive edit while retaining the native visible token. We refer to the model-specific set $\{d_c\}_{c=1}^{16}$ as the *opening bank*.

We derive two proposals from this bank. The *short proposal* d_s uses the opening token with the shortest training continuations among those satisfying the accuracy tolerance. The *accuracy-weighted proposal* $d_w = \sum_c w_c d_c$ combines opening directions using nonnegative weights that favor accuracy under a length budget, with regularization toward the short proposal. Appendix B gives the full objective.

We interpolate between the two proposals and independently scale their key and value components:

$$\begin{aligned} d(\lambda) &= (1-\lambda)d_s + \lambda d_w, \\ \delta_\phi &= \big(\alpha_K d_K(\lambda),\, \alpha_V d_V(\lambda)\big), \qquad \phi = (\lambda, \alpha_K, \alpha_V), \end{aligned} \tag{5}$$

where $\lambda \in [0,1]$ and $\alpha_K, \alpha_V \in [0,2]$. The separate K/V strengths are motivated by the distinct effects observed in Section 2.2.

Importantly, a favorable opening token does not imply that its induced memory difference will produce the same outcome when used as an additive edit. We therefore evaluate candidate edits using complete continuations. Constrained Bayesian optimization (Gardner et al., 2014) proposes additional candidates that trade off predicted shortening and accuracy feasibility. After a fixed search budget, we select the shortest measured candidate satisfying the empirical accuracy criterion, with uncertainty penalties on both accuracy and length. Native is included as the zero edit. The resulting $\delta_0 = \delta_{\phi^*}$ is the shared *memory anchor*.

3.3 Problem-Dependent Adaptation

A single shared anchor cannot account for variation across problems. Rather than learning an unrestricted edit in the full K/V space, O-Steer adapts the anchor within a low-dimensional subspace derived from the opening bank.

Let $\mu = \frac{1}{16}\sum_c d_c$ and $r = d(\lambda^*) - \mu$ denote the bank mean and the contrast of the selected proposal from that mean, respectively. We define

$$\mathcal{U} = \operatorname{span}\{(\mu_K, 0),\, (r_K, 0),\, (0, \mu_V),\, (0, r_V)\}\,. \tag{6}$$

After block normalization and removal of dependent directions, its basis forms the columns of $U \in \mathbb{R}^{D\times k}$, where $k \leq 4$. This space allows the controller to adjust common and proposal-specific K/V components separately while remaining close to opening-induced directions.

To condition the edit on the current problem, we summarize the unedited native opening memory using an eight-dimensional PCA feature $h(x) \in \mathbb{R}^8$. PCA is fitted once on the training memories without outcome labels. The resulting feature captures variation in the recipient opening state without using subsequent tokens or answer correctness.

O-Steer predicts a correction to the shared anchor with a single affine layer:

$$\Delta_\theta(x) = \delta_0 + U\,\Pi_\rho(b + Wh(x))\,, \qquad \theta = (b, W), \tag{7}$$

where $b \in \mathbb{R}^k$ is a shared correction, $W \in \mathbb{R}^{k\times 8}$ provides problem-dependent adaptation, and Π_ρ bounds the correction in normalized memory coordinates. Only b and W are learned, giving at most 36 trainable parameters. The anchor, basis, and PCA feature map remain fixed during fitting.

3.4 Learning from Complete-Continuation Feedback

The effect of a memory edit on final accuracy and reasoning length is not directly differentiable through the discrete continuation. We therefore learn the controller from complete-continuation feedback in two stages.

Starting from $b = W = 0$, we generate continuations under paired positive and negative Gaussian perturbations of the correction coefficients, using matched decoding randomness. Differences in final-answer correctness and output length provide local smoothed response estimates. Together with $h(x)$, these estimates define a bounded update that first minimizes predicted length subject to the accuracy constraint in Equation (3), and then improves predicted accuracy without increasing the attained length. We call the resulting controller the *compressed policy*.

Because the local response estimate may become inaccurate after this update, we collect a second batch of fresh continuations around the compressed policy. We re-estimate the local responses and again minimize length under the same accuracy floor. If $\widehat{L}_\star$ denotes the resulting local length estimate, we then maximize estimated accuracy subject to $L \leq (1+\eta)\widehat{L}_\star$, while retaining the accuracy and trust-region constraints. We use $\eta = 0.05$, which permits a small relative increase in length for accuracy recovery and is distinct from the absolute accuracy tolerance ε.

The two feedback stages update the same b and W, so the deployed policy retains the single affine controller in Equation (7). The fitting constraints are based on local empirical estimates and therefore do not guarantee that the accuracy floor will hold on unseen problems. Appendix B provides the response estimator, optimization constraints, and treatment of locally infeasible cases.

Inference. Given a problem x, the model first generates and processes its native opening token. O-Steer extracts $h(x)$ from the resulting opening memory, computes $\Delta_\theta(x)$, and applies the edit once at that cache position. Ordinary decoding then continues without further intervention. No candidate generation, answer verification, or online controller update is required.

4 Experiments

We evaluate whether a single opening-memory edit can shorten reasoning while preserving final-answer accuracy, and compare O-Steer with opening-token baselines and steering methods that intervene during decoding.

4.1 Experimental Setup

We evaluate frozen Qwen3-1.7B and DeepSeek-R1-Distill-Qwen-1.5B (Qwen Team, 2025; DeepSeek-AI, 2025) on 1,919 problems from six math benchmarks: 400 GSM8K, 400 MATH, 90 AIME, 83 AMC, 272 Minerva Math, and 674 OlympiadBench (Cobbe et al., 2021; Hendrycks et al., 2021; Lewkowycz et al., 2022; He et al., 2024). Each method generates two solutions per problem using BF16 inference, temperature 0.6, top-p 0.95, top-k 20, and a 16,384-token output limit. We report

final-answer accuracy and mean output length over all generations, including incorrect and capped outputs, with each problem weighted equally.

A separate 600-problem development panel is split into 480 fitting and 120 validation problems. All model-specific choices, including anchor selection and controller fitting, use the fitting split; Mean-V uses the full 600-problem panel without answer labels. Appendix C provides calibration details and matched controls for the SEAL and ReBalance execution paths. The evaluation suite was fixed but consulted during method development.

4.2 BASELINES

Opening-based baselines. **Native** performs ordinary generation without intervention. **Fixed Opening** forces only the first content token—`Yep` for Qwen and `I` for DeepSeek—selected on the fitting split as the shortest opening within two accuracy points of Native. This baseline tests whether choosing a favorable opening token alone is sufficient. **Mean-V** applies a shared, label-free value offset to the native opening memory while retaining its keys and visible token, testing whether a fixed, unlearned memory edit is sufficient.

Steering baselines. **SEAL** (Chen et al., 2025a) applies a fixed hidden-state direction at thought boundaries to suppress reflection and transition relative to execution. **ReBalance** (Li et al., 2026) adapts steering during decoding using online confidence and its variation. These methods provide comparisons with approaches that intervene repeatedly during reasoning. We calibrate their released implementations on the same 480 fitting problems and evaluate them under the common protocol; Appendix C provides implementation details and matched controls.

O-Steer. **O-Steer (ours)** learns a shared opening-memory anchor and a problem-dependent correction from complete-continuation feedback, and applies the resulting edit once after the native opening token (Section 3).

4.3 RESULTS ACROSS MODELS

Table 1 reports accuracy and mean output length across the six benchmarks for both models. The reported O-Steer results use the compressed policy from Section 3.4, with $\varepsilon = 0.04$.

Qwen: O-Steer provides stronger compression. Fixed Opening reduces mean output length by 18.0% with a 0.96-point decrease in accuracy, while Mean-V achieves a 10.6% reduction with a 0.18-point decrease. O-Steer reduces mean length to 3,879 tokens, 40.3% below Native and 27.2% below Fixed Opening, with 61.18% accuracy. SEAL preserves higher accuracy (67.59%) while reducing the mean length to 5,626 tokens. Thus, O-Steer reaches a substantially shorter operating point, with a larger accuracy decrease. Across benchmarks, its accuracy decrease ranges from 4.00 points on MATH to 9.74 points on Minerva.

DeepSeek: O-Steer exhibits a different trade-off. O-Steer reduces mean output length by 22.8%, from 5,309 to 4,096 tokens, with 51.41% accuracy. Fixed Opening produces shorter outputs (2,878 tokens) but lower accuracy (49.87%), whereas Mean-V provides little reduction in length (0.4%). SEAL reaches 54.64% accuracy at 4,005 tokens, giving a better aggregate point estimate than O-Steer on both metrics, while ReBalance remains closer to Native accuracy. These results indicate that the effect of one-shot opening-memory steering varies across models and is not uniformly preferable to steering during decoding.

Accuracy preservation under the fitting constraint. O-Steer's observed accuracy decreases are 5.91 points on Qwen and 4.17 points on DeepSeek, compared with the four-point fitting tolerance. Thus, the empirical constraint used during fitting does not necessarily transfer to held-out continuations. This discrepancy reflects the use of local response estimates for controller fitting: an update predicted to satisfy the accuracy constraint can produce a different trade-off under actual generation. Aggregate comparisons with external methods are descriptive; matched controls and paired intervals are reported in Appendix C.5.

4.4 What Does Problem-Dependent Learning Add?

Before controller fitting, the shared memory anchor achieves 64.07% accuracy with 4,499 mean output tokens on Qwen, and 53.80% accuracy with 4,798 tokens on DeepSeek. O-Steer further reduces these lengths by 13.8% and 14.6%, respectively, with accuracy decreases of 2.89 and 2.40 points relative to the anchor. Thus, learning the correction provides additional compression beyond shared-anchor selection. Appendix B.6 analyzes the resulting changes in K/V edit magnitudes and benchmark-level outcomes.

At inference, the learned policy retains a single affine controller and a single opening-memory edit. The reported token reductions measure generation length; they do not include offline fitting cost or imply equivalent reductions in wall-clock latency.

Table 1: **Accuracy–length trade-offs on 1,919 problems.** Accuracy and mean output length are reported on separate rows; bold indicates the highest accuracy or shortest output in each column. Overall annotations show changes from Native in accuracy (percentage points) and output length (↓ reduction; ↑ increase). Fixed Opening uses `Yep` for Qwen and `I` for DeepSeek; O-Steer uses $\varepsilon = 0.04$.

Qwen3-1.7B								
Method	**Metric**	**GSM8K**	**MATH**	**AIME**	**AMC**	**Minerva**	**Olympiad**	**Overall**
Native	Acc. ↑	**88.50**	**89.12**	28.89	65.66	**47.98**	54.30	67.09
	Tokens ↓	2,101	4,626	13,278	9,102	6,451	9,016	6,499
Fixed Opening	Acc. ↑	86.38	87.75	28.89	70.48	43.57	54.82	66.13 (-0.96)
	Tokens ↓	1,377	3,704	12,152	7,889	4,763	7,642	5,330 ↓18.0%
Mean-V	Acc. ↑	88.38	88.12	**34.44**	69.88	45.22	54.30	66.91 (-0.18)
	Tokens ↓	1,631	4,042	12,535	8,470	5,184	8,360	5,808 ↓10.6%
SEAL	Acc. ↑	88.00	**89.12**	30.56	**72.29**	47.06	**55.34**	**67.59** (+0.50)
	Tokens ↓	1,654	3,727	12,723	8,072	5,129	8,061	5,626 ↓13.4%
ReBalance	Acc. ↑	88.38	87.62	29.44	68.67	45.59	54.53	66.65 (-0.44)
	Tokens ↓	1,962	4,532	13,238	8,925	6,132	8,749	6,302 ↓3.0%
O-Steer ($\varepsilon = .04$)	Acc. ↑	83.25	85.12	19.44	57.23	38.24	49.18	61.18 (-5.91)
	Tokens ↓	**861**	**2,659**	**10,295**	**6,169**	**2,921**	**5,641**	**3,879** ↓40.3%
DeepSeek-R1-Distill-Qwen-1.5B								
Method	**Metric**	**GSM8K**	**MATH**	**AIME**	**AMC**	**Minerva**	**Olympiad**	**Overall**
Native	Acc. ↑	73.12	78.88	**21.67**	**60.24**	29.60	**45.77**	55.58
	Tokens ↓	448	3,344	12,623	8,203	4,932	8,180	5,309
Fixed Opening	Acc. ↑	**75.75**	73.12	16.67	49.40	24.26	35.53	49.87 (-5.71)
	Tokens ↓	450	**1,595**	**9,632**	**4,291**	**1,067**	**4,735**	**2,878** ↓45.8%
Mean-V	Acc. ↑	70.12	79.00	18.33	58.43	28.68	43.47	53.80 (-1.78)
	Tokens ↓	453	3,267	12,775	8,261	4,692	8,228	5,287 ↓0.4%
SEAL	Acc. ↑	74.00	80.25	19.44	57.23	**31.99**	41.47	54.64 (-0.94)
	Tokens ↓	459	2,412	10,496	6,445	2,753	6,393	4,005 ↓24.6%
ReBalance	Acc. ↑	74.00	**81.62**	21.11	58.43	29.78	44.14	**55.68** (+0.10)
	Tokens ↓	449	2,887	11,732	7,502	4,129	7,548	4,806 ↓9.5%
O-Steer ($\varepsilon = .04$)	Acc. ↑	70.62	76.62	18.89	55.42	27.21	38.65	51.41 (-4.17)
	Tokens ↓	**413**	2,351	11,182	7,132	2,558	6,619	4,096 ↓22.8%

5 Related Work

Reasoning budgets and opening choices. Chain-of-thought prompting and self-consistency improve reasoning through intermediate steps and multiple sampled solutions (Wei et al., 2022; Kojima et al., 2022; Wang et al., 2023b). Overthinking in reasoning models (Chen et al., 2025c) has motivated adaptive compute allocation, budget forcing, and early exit (Snell et al., 2025; Muennighoff et al., 2025; Yang et al., 2026a), as well as training methods that optimize reasoning length or the accuracy–compute trade-off (Ma et al., 2025; Arora & Zanette, 2025). The beginning of generation

provides another control point: first-step branching and sampling can alter reasoning paths (Wang & Zhou, 2024; Chen et al., 2025b), while leading words, learned opening-token biases, and token-level triggers can affect reasoning behavior (Lin et al., 2025; Sinii et al., 2025; Yang et al., 2026b). Our work extends this line from the choice of an opening token to the internal memory written by that token, and studies how this memory affects subsequent reasoning.

Representation and cache steering. Activation addition, inference-time intervention, and task or function vectors steer model behavior through internal representations (Rimsky et al., 2024; Li et al., 2023; Hendel et al., 2023; Todd et al., 2024), while prefix/prompt tuning and methods such as ReFT and LoFiT learn continuous inputs or localized representation changes (Li & Liang, 2021; Lester et al., 2021; Wu et al., 2024; Yin et al., 2024). SoftCoT and CODI represent intermediate reasoning in continuous space (Xu et al., 2025; Shen et al., 2025), and Soft Reasoning performs verifier-guided search over the first generated token's embedding (Zhu et al., 2025). More directly related to memory steering, KV Cache Steering applies a one-shot post-prefill K/V edit to guide reasoning (Belitsky et al., 2025), whereas DIRECTER adjusts cache steering during decoding for instruction following (Kang & Kim, 2026). O-Steer instead derives a compact edit space from memories induced by natural opening tokens, learns a problem-dependent edit from complete-continuation feedback, and applies it once to the native opening memory. Its learning procedure uses constrained Bayesian optimization for search under measured feasibility constraints (Gardner et al., 2014) and perturbation-based updates related to evolution strategies and zeroth-order adaptation (Wierstra et al., 2014; Salimans et al., 2017; Maheswaranathan et al., 2019; Malladi et al., 2023).

Analyzing the role of memory. Activation patching, circuit analysis, and interchange interventions test the behavioral effects of internal states (Zhang & Nanda, 2024; Wang et al., 2023a; Geiger et al., 2022; 2024), while work on subspace patching, self-repair, and circuit faithfulness clarifies how such interventions should be interpreted (Makelov et al., 2024; Rushing & Nanda, 2024; Hanna et al., 2024). Attention-flow, value-aware, and contextual-mixing analyses characterize how information is read and propagated through attention (Abnar & Zuidema, 2020; Kobayashi et al., 2020; Ferrando et al., 2022), and attention-sink studies examine initial context positions (Xiao et al., 2024). Recent work further investigates downstream K/V caches during prefill and latent reasoning (Li, 2026; Ding et al., 2026). Our analysis focuses on opening memory: crossed K/V interventions separate opening-key and opening-value effects, while cache exchanges under identical visible histories test whether opening-dependent effects remain active in later memory.

6 Discussion and Conclusion

The first generated token writes an internal memory that can influence the subsequent reasoning trajectory. Through matched-text interventions, we separate the effects of its keys and values and show that opening-dependent effects can remain active in later memory. Based on these findings, we introduce O-Steer, which learns a problem-dependent edit to the opening memory and applies it once before ordinary decoding continues. Across two reasoning models, O-Steer substantially reduces output length, although the observed accuracy decreases can exceed the tolerance used during fitting.

Our analysis is limited to one model and one opening-token pair, while the evaluation covers two small reasoning models and mathematical reasoning tasks. The evaluation suite also informed method development, and reductions in output tokens do not necessarily translate into proportional reductions in wall-clock latency. Extending opening-memory steering to larger models and broader reasoning tasks, and improving generalization of the accuracy constraint, are important directions for future work. More broadly, our results suggest that the memory written at the beginning of generation provides a compact control point for shaping long reasoning trajectories.

REPRODUCIBILITY STATEMENT

Appendix A specifies the memory interventions and supporting measurements. Appendix B gives the proposal objective, controller, and fitting procedure; Appendix C provides baseline calibration, outcome definitions, and matched comparisons.

AI USE STATEMENT

Generative AI assisted research design and analysis, code development and review, and drafting the manuscript, appendix, and figures. The authors take responsibility for the final content.

REFERENCES

Samira Abnar and Willem Zuidema. Quantifying attention flow in transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4190–4197. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.385. URL https://aclanthology.org/2020.acl-main.385/.

Daman Arora and Andrea Zanette. Training language models to reason efficiently. In *Advances in Neural Information Processing Systems*, volume 38, pp. 60770–60808. Curran Associates, Inc., 2025. doi: 10.52202/085713-2032. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/579b5b84311e122584dedab1d3b7613f-Paper-Conference.pdf.

Max Belitsky, Dawid J. Kopiczko, Michael Dorkenwald, M. Jehanzeb Mirza, James R. Glass, Cees G. M. Snoek, and Yuki M. Asano. KV cache steering for controlling frozen LLMs. *arXiv preprint arXiv:2507.08799*, 2025. URL https://arxiv.org/abs/2507.08799v2.

Runjin Chen, Zhenyu Zhang, Junyuan Hong, Souvik Kundu, and Zhangyang Wang. SEAL: Steerable reasoning calibration of large language models for free. In *Second Conference on Language Modeling*, 2025a. URL https://openreview.net/forum?id=klPszYDIRT.

Weizhe Chen, Zhicheng Zhang, Guanlin Liu, Renjie Zheng, Wenlei Shi, Chen Dun, Zheng Wu, Xing Jin, and Lin Yan. Flaming-hot initiation with regular execution sampling for large language models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 7133–7142. Association for Computational Linguistics, 2025b. doi: 10.18653/v1/2025.findings-naacl.396. URL https://aclanthology.org/2025.findings-naacl.396/.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. Do NOT think that much for 2+3=? On the overthinking of long reasoning models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 9487–9499. PMLR, 2025c. URL https://proceedings.mlr.press/v267/chen25bx.html.

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

DeepSeek-AI. DeepSeek-R1-Distill-Qwen-1.5B: Model card. Hugging Face, 2025. URL https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Qwen-1.5B.

Yi Ding, Lijun Huang, and Menglin Yang. SCIT: Testing causal cache carriers in latent chain-of-thought models. *arXiv preprint arXiv:2608.27265*, 2026. URL https://arxiv.org/abs/2608.27265.

Javier Ferrando, Gerard I. Gállego, and Marta R. Costa-jussà. Measuring the mixing of contextual information in the transformer. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 8698–8714. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.emnlp-main.595. URL https://aclanthology.org/2022.emnlp-main.595/.

Jacob R. Gardner, Matt J. Kusner, Zhixiang Xu, Kilian Q. Weinberger, and John P. Cunningham. Bayesian optimization with inequality constraints. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pp. 937–945, 2014. URL https://proceedings.mlr.press/v32/gardner14.html.

Atticus Geiger, Zhengxuan Wu, Hanson Lu, Josh Rozner, Elisa Kreiss, Thomas Icard, Noah Goodman, and Christopher Potts. Inducing causal structure for interpretable neural networks. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 7324–7338. PMLR, 2022. URL https://proceedings.mlr.press/v162/geiger22a.html.

Atticus Geiger, Zhengxuan Wu, Christopher Potts, Thomas Icard, and Noah Goodman. Finding alignments between interpretable causal variables and distributed neural representations. In *Proceedings of the Third Conference on Causal Learning and Reasoning*, volume 236 of *Proceedings of Machine Learning Research*, pp. 160–187. PMLR, 2024. URL https://proceedings.mlr.press/v236/geiger24a.html.

Michael Hanna, Sandro Pezzelle, and Yonatan Belinkov. Have faith in faithfulness: Going beyond circuit overlap when finding model mechanisms. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=TZ0CCGDcuT.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3828–3850. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.211. URL https://aclanthology.org/2024.acl-long.211/.

Roee Hendel, Mor Geva, and Amir Globerson. In-context learning creates task vectors. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 9318–9333. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.findings-emnlp.624. URL https://aclanthology.org/2023.findings-emnlp.624/.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/file/be83ab3ecd0db773eb2dc1b0a17836a1-Paper-round2.pdf.

Minjae Kang and Jaehyung Kim. Enhancing instruction following of LLMs via activation steering with dynamic rejection. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/c05a399e5e2312f65c94cc5388b80828-Abstract-Conference.html.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. Attention is not only a weight: Analyzing transformers with vector norms. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 7057–7075. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-main.574. URL https://aclanthology.org/2020.emnlp-main.574/.

Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*, volume 35, pp. 22199–22213. Curran Associates, Inc., 2022. doi: 10.52202/068431-1613. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/8bb0d291acd4acf06ef112099c16f326-Paper-Conference.pdf.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3045–3059. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.emnlp-main.243. URL https://aclanthology.org/2021.emnlp-main.243/.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 3843–3857, 2022. doi: 10.52202/068431-0278. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/18abbeef8cfe9203fdf9053c9c4fe191-Paper-Conference.pdf.

Bojie Li. Models take notes at prefill: Kv cache can be editable and composable. *arXiv preprint arXiv:2606.17107*, 2026. URL https://arxiv.org/abs/2606.17107.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. In *Advances in Neural Information Processing Systems*, volume 36, pp. 41451–41530. Curran Associates, Inc., 2023. doi: 10.52202/075280-1797. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/81b8390039b7302c909cb769f8b6cd93-Abstract-Conference.html.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4582–4597. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.acl-long.353. URL https://aclanthology.org/2021.acl-long.353/.

Yulin Li, Tengyao Tu, Li Ding, Junjie Wang, Huiling Zhen, Yixin Chen, Yong Li, and Zhuotao Tian. Efficient reasoning with balanced thinking. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=cJseWJJ5IM.

Zhengkai Lin, Zhihang Fu, Ze Chen, Chao Chen, Liang Xie, Wenxiao Wang, Deng Cai, Zheng Wang, and Jieping Ye. Controlling thinking speed in reasoning models. In *Advances in Neural Information Processing Systems*, volume 38, pp. 78300–78347. Curran Associates, Inc., 2025. doi: 10.52202/085713-2623. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/70d46c12643e8f056a693ed43472dc35-Paper-Conference.pdf.

Xinyin Ma, Guangnian Wan, Runpeng Yu, Gongfan Fang, and Xinchao Wang. CoT-valve: Length-compressible chain-of-thought tuning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6025–6035. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.300. URL https://aclanthology.org/2025.acl-long.300/.

Niru Maheswaranathan, Luke Metz, George Tucker, Dami Choi, and Jascha Sohl-Dickstein. Guided evolutionary strategies: augmenting random search with surrogate gradients. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 4264–4273, 2019. URL https://proceedings.mlr.press/v97/maheswaranathan19a.html.

Aleksandar Makelov, Georg Lange, Atticus Geiger, and Neel Nanda. Is this the subspace you are looking for? an interpretability illusion for subspace activation patching. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/70b8505ac79e3e131756f793cd80eb8d-Abstract-Conference.html.

Sadhika Malladi, Tianyu Gao, Eshaan Nichani, Alex Damian, Jason D. Lee, Danqi Chen, and Sanjeev Arora. Fine-tuning language models with just forward passes. In *Advances in Neural Information Processing Systems*, volume 36, pp. 53038–53075, 2023. doi: 10.52202/075280-2308. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/a627810151be4d13f907ac898ff7e948-Abstract-Conference.html.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 20275–20321. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.1025. URL https://aclanthology.org/2025.emnlp-main.1025/.

Qwen Team. Qwen3-1.7B: Model card. Hugging Face, 2025. URL https://huggingface.co/Qwen/Qwen3-1.7B.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering Llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15504–15522. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.828. URL https://aclanthology.org/2024.acl-long.828/.

Cody Rushing and Neel Nanda. Explorations of self-repair in language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 42836–42855. PMLR, 2024. URL https://proceedings.mlr.press/v235/rushing24a.html.

Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*, 2017. URL https://arxiv.org/abs/1703.03864.

Zhenyi Shen, Hanqi Yan, Linhai Zhang, Zhanghao Hu, Yali Du, and Yulan He. CODI: Compressing chain-of-thought into continuous space via self-distillation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 677–693. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.36. URL https://aclanthology.org/2025.emnlp-main.36/.

Viacheslav Sinii, Nikita Balagansky, Gleb Gerasimov, Daniil Laptev, Yaroslav Aksenov, Vadim Kurochkin, Alexey Gorbatovski, Boris Shaposhnikov, and Daniil Gavrilov. Small vectors, big effects: A mechanistic study of RL-induced reasoning via steering vectors. *arXiv preprint arXiv:2509.06608*, 2025. URL https://arxiv.org/abs/2509.06608v4. Version 4, February 2026.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *International Conference on Learning Representations*, volume 2025, pp. 10131–10165, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/1b623663fd9b874366f3ce019fdfdd44-Paper-Conference.pdf.

Eric Todd, Millicent L. Li, Arnab Sen Sharma, Aaron Mueller, Byron C. Wallace, and David Bau. Function vectors in large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/4ae163cb8788970e53b4fd9578141139-Abstract-Conference.html.

Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: A circuit for indirect object identification in GPT-2 small. In *The Eleventh International Conference on Learning Representations*, 2023a. URL https://openreview.net/forum?id=NpsVSN6o4ul.

Xuezhi Wang and Denny Zhou. Chain-of-thought reasoning without prompting. In *Advances in Neural Information Processing Systems*, volume 37, pp. 66383–66409. Curran Associates, Inc., 2024. doi: 10.52202/079017-2123. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/7a8e7fd295aa04eac4b470ae27f8785c-Paper-Conference.pdf.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*, 2023b. URL https://openreview.net/forum?id=1PL1NIMMrw.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837. Curran Associates, Inc., 2022. doi: 10.52202/

068431-1800. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf.

Daan Wierstra, Tom Schaul, Tobias Glasmachers, Yi Sun, Jan Peters, and Jürgen Schmidhuber. Natural evolution strategies. *Journal of Machine Learning Research*, 15(27):949–980, 2014. URL https://www.jmlr.org/papers/v15/wierstra14a.html.

Zhengxuan Wu, Aryaman Arora, Zheng Wang, Atticus Geiger, Dan Jurafsky, Christopher D. Manning, and Christopher Potts. ReFT: Representation finetuning for language models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 63908–63962. Curran Associates, Inc., 2024. doi: 10.52202/079017-2041. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/75008a0fba53bf13b0bb3b7bff986e0e-Abstract-Conference.html.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/5e5fd18f863cbe6d8ae392a93fd271c9-Abstract-Conference.html.

Yige Xu, Xu Guo, Zhiwei Zeng, and Chunyan Miao. SoftCoT: Soft chain-of-thought for efficient reasoning with LLMs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 23336–23351. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.1137. URL https://aclanthology.org/2025.acl-long.1137/.

Chenxu Yang, Qingyi Si, Yongjie Duan, Zheliang Zhu, Chenyu Zhu, Qiaowei Li, Minghui Chen, Zheng Lin, and Weiping Wang. Dynamic early exit in reasoning models. In *The Fourteenth International Conference on Learning Representations*, 2026a. URL https://openreview.net/forum?id=NpU7ZXafRi.

Van Yang, Shouren Wang, Debargha Ganguly, Xinpeng Li, Chaoda Song, Vikash Singh, Vipin Chaudhary, and Xiaotian Han. Mid-Think: Training-free intermediate-budget reasoning via token-level triggers. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 6024–6038. Association for Computational Linguistics, 2026b. doi: 10.18653/v1/2026.findings-acl.299. URL https://aclanthology.org/2026.findings-acl.299/.

Fangcong Yin, Xi Ye, and Greg Durrett. LoFiT: Localized fine-tuning on LLM representations. In *Advances in Neural Information Processing Systems*, volume 37, pp. 9474–9506. Curran Associates, Inc., 2024. doi: 10.52202/079017-0301. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/122ea6470232ee5e79a2649243348005-Abstract-Conference.html.

Fred Zhang and Neel Nanda. Towards best practices of activation patching in language models: Metrics and methods. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/06a52a54c8ee03cd86771136bc91eb1f-Abstract-Conference.html.

Qinglin Zhu, Runcong Zhao, Hanqi Yan, Yulan He, Yudong Chen, and Lin Gui. Soft reasoning: Navigating solution spaces in large language models through controlled embedding exploration. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 80427–80447. PMLR, 2025. URL https://proceedings.mlr.press/v267/zhu25ae.html.

A Supporting Analyses of Opening Memory

A.1 Opening-Token Outcomes and Intervention Boundary

This appendix supports the three analyses in Section 2: opening-dependent outcomes, separate K/V effects, and opening-dependent effects retained in later memory. All experiments use Qwen3-1.7B on 600 development problems, with 200 each from GSM8K, MATH, and Omni-MATH, two continuations per condition, and a 16,384-token output limit. Sampling and outcome definitions follow Section 4.1 and Appendix C.4. The *opening token* is the first generated content token after the reasoning scaffold, rather than a beginning-of-sequence or delimiter token. Table 2 reports the complete opening-token results underlying Section 2.1.

Table 2: Opening-token outcomes on the common 600-problem panel, with two generations per token. Means include incorrect and capped outputs; Native reproduces the `Okay` paths in this experiment.

Opening token	Accuracy (%)	Output tokens	Capped (%)
Okay	71.92	5,848.71	13.67
Alright	71.17	5,925.86	12.67
okay	72.50	5,372.40	10.67
OK	72.83	5,269.47	10.33
Yes	69.75	4,161.85	5.42
Hmm	72.17	5,175.27	9.67
Sure	69.58	4,455.11	7.67
Yeah	71.08	4,960.03	9.92
`U+55EF`	68.75	5,469.19	9.83
Yep	72.00	4,900.74	7.92
Now	68.50	4,379.10	6.00
Ok	71.83	5,177.24	9.67
Please	59.58	1,002.73	0.50
Certainly	65.67	4,290.25	5.75
Wow	71.75	4,985.48	8.33
So	71.67	5,057.88	8.00

For each problem, we obtain the `Okay` and `Yep` opening memories from the same prompt. The full token/K/V factorial crosses the displayed token with the sources of its keys and values, yielding eight conditions. Each intervention is applied after the displayed token has been processed; the prompt cache and already computed second-token logits are retained. In these runs, the second-token distributions after top-k/top-p truncation have singleton support and coincide. Consequently, all four matched-K/V token comparisons have identical recorded suffixes across their 1,200 pairs. This verifies the intervention boundary for the tested decoder; it does not imply that visible opening tokens are irrelevant under arbitrary decoding rules.

The K–V interaction in mean output length, defined as the joint effect minus the sum of the two individual effects, is $+249.84$ tokens. Its original interval includes zero, so these measurements do not resolve whether the two effects combine additively. The length differences for key-only and value-only edits are -400.29 ($[-615.77, -189.83]$) and -797.52 ($[-1049.53, -557.96]$) tokens, respectively. Intervals retain the original 18-endpoint comparison family (99.7222%), with repeated outputs averaged within problem before benchmark-stratified resampling.

A.2 Attention Readout and Same-History Replay

For one head, let p denote attention on the opening token (*opening attention*), v its value, and b the normalized attention-weighted value of the other positions. Holding the query and the other keys and values fixed gives

$$\Delta o = W_O \left[p\Delta v + \Delta p(v - b) + \Delta p \Delta v \right]. \tag{8}$$

Here W_O is the output-projection block for that head. The three terms correspond to a value change at fixed routing, a routing change at fixed values, and their interaction. Thus, attention mass alone cannot measure the effect of a memory edit (Kobayashi et al., 2020).

The identity is local: later queries and cache writes can also change during an edited rollout. Editing a stored entry does not recompute existing entries at other layers. However, its readout changes a later token's residual state, which propagates through subsequent layers and can alter newly written K/V. A single cache edit can therefore affect downstream computation beyond the edited position.

We replay the same recorded `Okay` continuation under all eight token/K/V conditions. Table 3 uses all 600 problems at tokens 32 and 128. The longer replay curves use the 592 problems reaching token 512; 23 have passed the reasoning-close marker by that point. Key-only and value-only edits move opening attention in opposite directions, although both shorten free generation. Reducing opening attention is therefore not a common explanation of the two length effects.

Table 3: Mean opening attention (%), across layers and query heads, under the same replayed text. All conditions display `Okay`; $n = 600$ at each checkpoint.

Opening memory	Token 32	Token 128
Okay K/V	0.729	0.413
Yep K only	0.432	0.204
Yep V only	0.810	0.444
Yep K/V	0.482	0.222

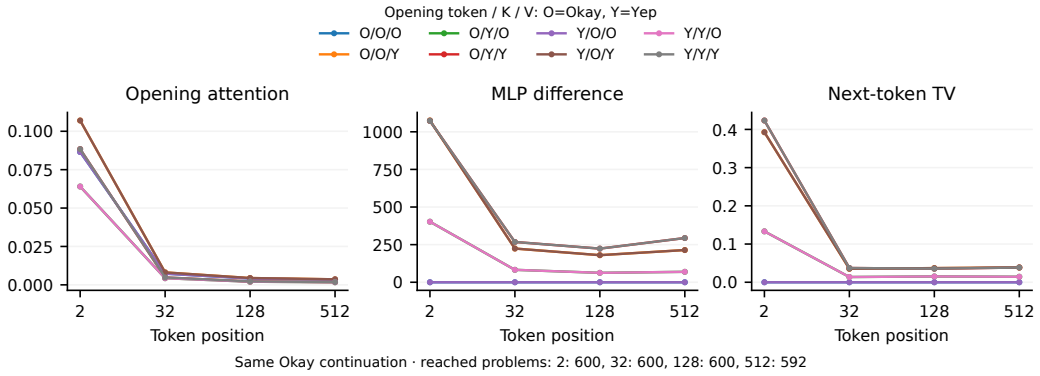

Figure 5: **Same-history replay.** Direct opening attention decreases with generation depth while differences in downstream computation remain. O and Y denote `Okay` and `Yep` sources; curves with matching K/V sources overlap. Attention is averaged over layers and query heads. MLP differences are norms across all 28 layers; total variation (TV) compares the next-token sampling distribution with the all-`Okay` condition. Each checkpoint uses all histories reaching it.

On the common 592 problems, joint-exchange opening attention is 0.484%, 0.221%, and 0.182% at tokens 32, 128, and 512, while next-token sampling TV is 0.0361, 0.0352, and 0.0386. Thus, low direct attention can coexist with persistent differences in downstream computation. These within-backend replay measurements do not identify the opening token as an attention sink: that would require comparison with other positions (Xiao et al., 2024).

A.3 Exchanging Opening and Later Memory

For each problem, we use two recorded receiver histories, one from `Okay` and one from `Yep`. All histories reach the fixed 128-token boundary. For each history, we replay its continuation tokens under either opening-memory source. At the fixed boundary, we independently exchange the opening entry and *later memory*, defined as the K/V entries at positions 2–127, and then process the same 128th token before resuming free generation. The prompt cache and visible history are matched across conditions. Four cache combinations, two receiver histories, and two repetitions give 9,600 continuations in total.

Opening-only and later-only exchanges reduce remaining output length by 560.54 and 368.38 tokens, with the original 99.1667% intervals (six endpoints) $[-690.93, -425.97]$ and $[-501.80, -238.23]$.

Table 4: Opening- and later-memory exchanges under matched text. Each row averages two histories and two repetitions on 600 problems; length counts tokens generated after the exchange boundary.

Opening	Later	Accuracy (%)	Remaining tokens	Capped (%)
Okay	Okay	71.29	5,696.87	12.96
Yep	Okay	72.21	5,136.33	10.33
Okay	Yep	71.96	5,328.49	10.83
Yep	Yep	71.88	4,766.42	8.63

The length interaction is -1.52 tokens ($[-182.54, +178.61]$). Thus, opening-dependent differences can remain behaviorally active in later memory even after the opening entry is restored. The experiment does not identify a unique carrier or determine what fraction of the original free-generation effect is mediated by later memory.

B O-STEER: LEARNING DETAILS

This appendix specifies O-Steer's proposal construction, anchor selection, and two controller updates in Section 3. Construction and fitting use $n = 480$ training problems; 120 separate development problems provide post-fitting diagnostics. The opening bank, feature map, anchor, and controller are model-specific. The chosen accuracy tolerance ε is used from proposal construction through controller fitting. Changing it can alter the short opening token, proposal mixture, search trajectory, and learned policy. The underlying 16-vector opening bank and unsupervised feature map are independent of this tolerance.

B.1 MEMORY COORDINATES AND PROPOSAL CONSTRUCTION

We flatten K/V across layers and key–value heads and undo the rotary positional transform on keys before comparing memories. At injection, the key edit is rotated to the target position. The bank uses one opening memory per training problem and opening token; accuracy and length use two complete continuations. The 16 opening tokens are fixed from first-token probabilities before their outcomes are measured.

Let a_c, ℓ_c denote opening token c's training accuracy and mean length, and let a_N denote the corresponding Native accuracy. The short opening token is $s = \arg\min_{c: a_c \geq a_N - \varepsilon} \ell_c$, with ties broken by higher accuracy and then token index. This construction requires at least one eligible reference token. Define the short-centered prior $p_c = 0.8\,\mathbf{1}[c = s] + 0.2/16$ and length budget $B = 1.2\ell_s$. The accuracy-weighted proposal solves

$$w^* = \underset{w \geq 0,\ \sum_c w_c = 1}{\arg\max} \left\{ \sum_c w_c a_c - \tau D_{\mathrm{KL}}(w \| p) \right\} \quad \text{subject to} \quad \sum_c w_c \ell_c \leq B, \qquad \tau = 0.02. \tag{9}$$

Here $D_{\mathrm{KL}}(w\|p) = \sum_c w_c \log(w_c/p_c)$. The regularizer keeps the proposal near the short opening token while allowing accuracy-favoring alternatives. With $\nu \geq 0$ enforcing the normalized constraint $\sum_c w_c \ell_c/\ell_s \leq B/\ell_s$, $w_c \propto p_c \exp\{(a_c - \nu\ell_c/\ell_s)/\tau\}$. We use $\nu = 0$ if feasible and otherwise solve the scalar budget equation. These quantities summarize opening-token outcomes; they do not guarantee the behavior of the corresponding additive memory edit. We therefore evaluate actual interventions next.

B.2 SELECTING THE ANCHOR FROM ACTUAL EDITS

The family in Equation (5) has $\lambda \in [0, 1]$ and $\alpha_K, \alpha_V \in [0, 2]$. We first evaluate value-only and joint K/V edits for each of the two proposals at unit strength: four candidates with $480 \times 2 = 960$ continuations each. Native provides the zero edit and matched references. For each candidate, we average accuracy and length differences from Native within each problem and then compute their means and standard errors across problems.

Two independent Gaussian-process models predict the accuracy and normalized-length differences. They use Matérn-$5/2$ kernels on the unit-scaled parameter box, length scale 0.5, signal standard

Table 5: Cumulative variance of normalized, centered native opening memories explained by the leading principal components.

Model	1 component	2	4	8 (used)
Qwen	68.69%	76.60%	83.92%	88.30%
DeepSeek	59.34%	78.25%	92.19%	95.11%

deviations 0.05 and 0.15, and observation variances given by squared standard errors with a 10^{-10} floor. Constrained expected improvement (Gardner et al., 2014) scores a candidate by

$$\mathrm{EI}_L(\phi)\ \Pr(D_A(\phi) \geq -\varepsilon \mid \mathcal{D})\,, \tag{10}$$

where D_A is its accuracy difference from Native and $\mathcal{D}$ contains the measured training outcomes. Improvement is defined relative to the shortest eligible measured edit. We select from 1,024 fixed Sobol points and their boundary-face projections, using two rounds of three candidates. Posterior-mean fantasies separate candidates within each batch; only measured outcomes enter subsequent observations and final selection.

For measured candidate j, let $\bar{D}_{Aj}, \bar{D}_{Lj}$ denote its mean paired differences and s_{Aj}, s_{Lj} their standard errors. We select

$$j^* = \operatorname*{arg\,min}_{j:\, \bar{D}_{Aj} - s_{Aj} \geq -\varepsilon} (\bar{D}_{Lj} + s_{Lj}), \tag{11}$$

with higher accuracy, smaller normalized norm, and candidate index as tie breakers. Native is feasible with exactly zero paired differences. The one-standard-error penalties discourage uncertain choices; they do not guarantee held-out accuracy.

B.3 Correction Space and Problem Features

For $t \in \{K, V\}$, let $s_t^2 = 16^{-1} \sum_c \|d_{c,t}\|_2^2$ and let S repeat these block scales on its diagonal. We scale the common and contrast vectors in Equation (6), perform separate K/V singular-value decompositions, and retain independent directions. Rescaling gives U, with $k \leq 4$ columns and metric $M = U^\top S^{-2} U$. The deployed coefficient projection is

$$\Pi_\rho(z) = z \min\left\{1, \frac{\rho}{\sqrt{z^\top M z}}\right\}, \qquad \rho = 0.5, \qquad \Pi_\rho(0) = 0. \tag{12}$$

This projection bounds the total correction to the anchor, including both learned updates; the anchor itself is not subject to the bound. Even a value-only anchor can receive a key correction because its proposal defines both components of U.

For the controller input, let $\widetilde{m}(x)$ denote the flattened, position-aligned native opening memory and $\bar{m}$ its training mean. Let S_f contain the separate root-mean-square magnitudes of native K and V coordinates. PCA is fitted to $z(x) = S_f^{-1}(\widetilde{m}(x) - \bar{m})$ without outcome labels. If P_8 contains its eight leading orthonormal eigenvectors and Λ_8 their eigenvalues, define

$$u(x) = \Lambda_8^{-1/2} P_8^\top z(x), \qquad h(x) = \mathrm{clip}_{\sqrt{8}}(u(x)) - \overline{\mathrm{clip}_{\sqrt{8}}(u)}, \tag{13}$$

where $\mathrm{clip}_r(u) = u \min\{1, r/\|u\|_2\}$, with zero mapped to zero, and the bar denotes the training mean. Whitening places the retained components on comparable scales, while clipping limits extreme inputs. This fixed transform is applied to the realized, unedited native opening at inference. It depends on the problem and sampled opening, but not on subsequent tokens or answer labels.

We fix eight input components to limit the number of fitted coefficients. Table 5 reports the variance retained by the saved feature map on its 480 training memories, before whitening and clipping. Eight is a fixed capacity choice rather than a dimension selected by evaluation performance. The input features summarize the recipient state, while the at-most-four output directions in U determine the available corrections.

B.4 Two Batches of Local Feedback

Write $\theta_t = \mathrm{vec}_{\mathrm{row}}([b_t\ W_t]) \in \mathbb{R}^{9k}$ for the current controller parameters, stacking the rows of the $k \times 9$ parameter matrix. The first batch starts at $\theta_0 = 0$; the second starts from the compressed policy

with parameters θ_1. Let $h_{ij} \in \mathbb{R}^8$ denote the features of the realized native opening memory for problem i and repetition $j \in \{1, 2\}$. Draw $\xi_{ij} \sim \mathcal{N}(0, I_k)$ and generate under the edits

$$\delta_0 + U\Pi_\rho\big(b_t + W_t h_{ij} \pm \sigma\xi_{ij}\big), \qquad \sigma = 0.25. \tag{14}$$

The two signs share the same pre-edit state and decoding randomness. The first batch reuses the measured anchor outcomes as its center. The second additionally generates an unperturbed continuation, measuring the current controller rather than extrapolating from the anchor. The two batches reuse perturbation directions and decoding randomness; the second measures fresh continuations at the updated controller rather than providing an independent sample.

Let $q_A(Y) = C(Y)$ denote binary final-answer correctness and $q_L(Y) = T(Y)/T_{\max}$ normalized length, with $T_{\max} = 16384$. For either outcome q, define the problem-level response estimate

$$g_{q,i} = \frac{1}{2}\sum_{j=1}^{2} \mathrm{vec}_{\mathrm{row}}\left[\frac{q(Y_{ij}^{+}) - q(Y_{ij}^{-})}{2\sigma}\xi_{ij}(1, h_{ij}^{\top})\right] \in \mathbb{R}^{9k}. \tag{15}$$

A generation terminating before an editable state contributes zero response. This is an antithetic Gaussian response estimate in coefficient space, with the projection included in each measured intervention. At a fixed center, its expectation over independent perturbations gives the derivative of a Gaussian-smoothed response (Salimans et al., 2017). We use it as a local slope around measured, unperturbed outcomes; it is not the exact derivative of the deployed zero-perturbation policy. The backbone is neither differentiated nor updated.

For a parameter increment $u \in \mathbb{R}^{9k}$, let a_i denote the measured center's accuracy difference from Native and l_i its normalized length, both averaged within problem. With $v_i = (a_i, g_{A,i}^{\top})^{\top}$, $t_u = (1, u^{\top})^{\top}$, and $\Sigma_A = \mathrm{Cov}_i(v_i)/n + 10^{-6}I$, we use

$$\widehat{D}_A^{-}(u) = \bar{v}^{\top} t_u - \sqrt{t_u^{\top}\Sigma_A t_u}. \tag{16}$$

For the first batch, the upper length estimate has the same form with $v_i = (l_i, g_{L,i}^{\top})^{\top}$, a plus sign on its standard-error term, and a factor of $T_{\max}$ to convert back to tokens. For the second batch, we estimate length relative to its newly measured center:

$$\widehat{L}^{+}(u) = T_{\max}\left[\bar{l} + \bar{g}_L^{\top} u + \sqrt{u^{\top}\Sigma_g u}\right], \qquad \Sigma_g = \mathrm{Cov}_i(g_{L,i})/n + 10^{-6}I. \tag{17}$$

The predicted extra cost is exactly zero at $u = 0$: this estimate penalizes uncertainty in the change rather than uncertainty in the measured center's mean length. Replicates are grouped by problem in all covariance estimates. These one-standard-error penalties are fitting heuristics, not confidence bounds valid after adaptive anchor and controller selection.

We bound each update by $u^{\top}Hu \le \rho^2$, where $H = M \otimes \overline{\psi\psi^{\top}} + R \in \mathbb{R}^{9k\times 9k}$, $\psi_{ij} = (1, h_{ij}^{\top})^{\top}$, and $\otimes$ denotes the Kronecker product. The average is over exposed training states. The diagonal matrix R penalizes each intercept by 0.05 and each feature coefficient by 1. This parameter-space bound regularizes the fitted response; the pointwise projection in Equation (12) bounds the resulting edits.

B.5 Compression and Accuracy Recovery

In each batch, we first solve

$$u_c = \arg\min_u \widehat{L}^{+}(u) \quad \text{subject to} \quad \widehat{D}_A^{-}(u) \ge -\varepsilon, \qquad u^{\top}Hu \le \rho^2. \tag{18}$$

The first batch then maximizes the same accuracy estimate without increasing the attained length. In the second batch, set $\widehat{L}_\star = \widehat{L}^{+}(u_c)$ and solve

$$\begin{aligned} u^* &= \arg\max_u \widehat{D}_A^{-}(u) \\ \text{subject to} \quad \widehat{L}^{+}(u) &\le (1+\eta)\widehat{L}_\star, \quad \widehat{D}_A^{-}(u) \ge -\varepsilon, \quad u^{\top}Hu \le \rho^2, \end{aligned} \qquad \eta = 0.05. \tag{19}$$

For a nonpositive predicted length, no additional length allowance is added. Here ε controls the allowed accuracy decrease, whereas η permits a small relative increase in length for accuracy recovery.

The final parameters are $\theta_2 = \theta_1 + u^*$; both increments enter $b + Wh$ before the single projection in Equation (7).

If the center is locally accuracy-infeasible, we first attempt to restore feasibility within the trust region. The second batch additionally bounds this attempt by its measured center length plus the η allowance. If the floor remains unreachable, the first batch retains the anchor; the second retains any accepted recovery step and records the unmet target. A failed solve retains the last valid step. The resulting small constrained problems are solved with SLSQP using analytic derivatives of the response estimates. These estimates are fitting surrogates and do not guarantee deployed accuracy.

The four initial candidates, six search candidates, first paired batch, and second center/paired batch require 3,840, 5,760, 1,920, and 2,880 training outputs, respectively, beyond the Native references. The 120-problem validation set compares the compressed and refined policies without selecting between them. Evaluation measures their actual accuracy–length trade-off on the common suite.

B.6 How Learning Changes the Anchor

Changes to the edit. We compare the shared anchor δ_0 with the compressed policy $\Delta_{\theta_1}(x)$ reported in Table 1, using the 960 native opening states encountered during fitting. Table 6 uses the block normalization S defined above. The mean correction is about 16% of the anchor norm. Learning slightly reduces the key-edit magnitude and increases the value-edit magnitude in both models. Mean cosine similarity between the anchor and final edit exceeds 0.99 in each K/V block, indicating that the controller primarily adjusts edit strengths within the anchor's neighborhood.

Table 6: Learned edits relative to the shared anchor, averaged over 960 training opening states. Correction size uses normalized joint K/V norms; K and V magnitudes are percentages of the corresponding anchor block.

Model	Correction / anchor	Final K / anchor K	Final V / anchor V
Qwen	15.70%	94.21%	113.59%
DeepSeek	16.24%	99.26%	118.63%

Changes to generation. On the 1,919 evaluation problems, learning reduces mean output length from 4,499 to 3,879 tokens on Qwen and from 4,798 to 4,096 on DeepSeek. The original paired intervals for these differences are -620 $[-794, -445]$ and -702 $[-917, -487]$ tokens, respectively, at 99.6875% confidence. Accuracy changes from 64.07% to 61.18% and from 53.80% to 51.41%. Thus, controller learning provides additional compression beyond the shared anchor, but does not recover accuracy overall.

Figure 6 further resolves these changes by benchmark. All six Qwen benchmarks become shorter. DeepSeek shows a more heterogeneous pattern: GSM8K improves in both accuracy and length, whereas AIME becomes slightly longer and less accurate. Thus, the learned correction changes generation beyond shared-anchor selection, with task-dependent trade-offs. These results do not by themselves identify which K/V changes cause shortening or separate the contribution of problem-dependent adaptation from that of a learned shared correction.

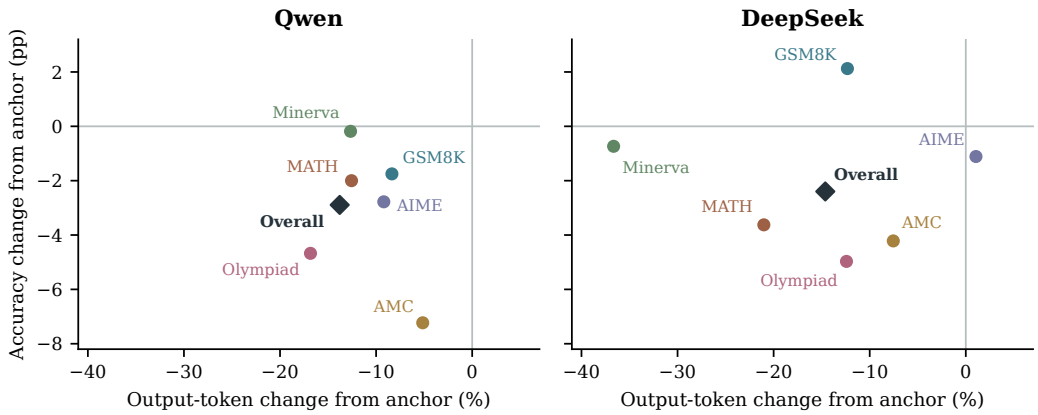

Figure 6: **From the shared anchor to O-Steer.** Changes in mean output length and final-answer accuracy on each evaluation benchmark. Leftward points indicate shorter outputs; upward points indicate higher accuracy. Diamonds show the problem-weighted mean over all 1,919 problems.

C Baselines and Evaluation Details

C.1 Fixed Opening Selection

For the Fixed Opening baseline, sixteen candidate opening tokens are fixed from fitting-set first-token probabilities before comparing their outcomes. Let $\widehat{A}_c$ and $\widehat{L}_c$ denote the fitting accuracy and mean output length for opening token c. We select

$$c^* = \arg \min_{c:\, \widehat{A}_c \geq \widehat{A}_{\text{Native}} - 0.02} \widehat{L}_c. \tag{20}$$

The resulting fitting accuracy floors are 70.1875% for Qwen and 59.9792% for DeepSeek, selecting `Yep` and `I`, respectively. The selected token is fixed across all evaluation problems; selection is not repeated for individual benchmarks or questions. Table 7 reports the corresponding fitting results. This procedure is independent of the accuracy tolerance used to fit O-Steer. The two models use different candidate sets: Qwen's set does not contain `I`, whereas DeepSeek's set does not contain `Yep`.

Table 7: Fixed Opening selection on 480 fitting problems. The same Native-minus-two-percentage-point rule selects both tokens.

Model	Selected token	Native accuracy (%)	Token accuracy (%)	Output tokens
Qwen	`Yep`	72.19	72.60	4,911
DeepSeek	`I`	61.98	60.73	2,724

C.2 Mean-V

Mean-V constructs a shared value offset without using answer correctness. For a 600-problem bank, let $V(x, c)$ concatenate the values at the first content-token position across layers and key–value heads, with `Okay` as the reference opening token. Let Π_{16} project onto a rank-16 subspace fitted to these value differences without labels. The shared offset is

$$\bar{\Delta}_V = \Pi_{16} \left[\frac{1}{600 \cdot 16} \sum_{x,c} \big(V(x, c) - V(x, \texttt{Okay}) \big) \right]. \tag{21}$$

The same sixteen-token reference library is used to construct a separate bank for each model. This library differs from the model-specific candidate set used for DeepSeek's Fixed Opening baseline,

and no memory vector is transferred between models. Mean-V is therefore a label-free reference rather than an ablation of a particular component of O-Steer.

After the native opening token is processed, we set $V_1' = V_1 + \bar{\Delta}_V$ while retaining K_1, the prompt memory, the visible opening token, and the already computed second-token logits. Generation then continues without further intervention. This baseline tests whether a shared, unlearned value edit is sufficient to affect the accuracy–length trade-off.

C.3 Published-Method Ports

SEAL and ReBalance are calibrated separately for each model on its 480 fitting problems using Native traces. The base-model weights remain frozen, and calibration does not use answer-correctness labels. All reported results use our common sampling protocol rather than the benchmark-specific settings of the source papers.

SEAL. Following Chen et al. (2025a), keyword labels separate execution from reflection/transition paragraphs. Their mean hidden-state difference defines the steering direction applied at reasoning paragraph separators. Our port uses the fixed layer-20 rule and a unit-magnitude coefficient, with the sign chosen to suppress the reflection/transition component. No per-benchmark tuning is performed.

ReBalance. We follow the released implementation of Li et al. (2026). Lexicon and confidence labels define the hidden-state prototypes, and layer selection uses grouped fitting-set confidence prediction. During decoding, paragraph confidence and its variation determine the steering strength and sign. We retain the released code's offline geometric selected-token confidence and online arithmetic maximum-probability confidence. The reported scores therefore reflect these model-specific ports.

Both ports use the same model weights and evaluation settings as the opening-memory experiments, with eager execution and matched Native/Mean-V controls. The matched controls are reported below.

C.4 Data and Outcome Definitions

Section 2 uses 600 development problems, balanced across GSM8K, MATH, and Omni-MATH. We use a fixed split of 480 problems for fitting and 120 for validation, keeping all openings and repeated generations of each problem in the same partition. Opening banks and controllers are fitted separately for each model.

The evaluation set contains 1,919 problems: 400 GSM8K, 400 MATH, 90 AIME, 83 AMC, 272 Minerva Math, and 674 OlympiadBench. We generate two solutions per problem to estimate sampling variability, without best-of-two selection. Aggregate results weight problems equally; the 600-problem analysis likewise weights its three benchmarks equally.

We count all generated tokens, including incorrect and capped outputs, up to the 16,384-token limit. Accuracy is determined from the final mathematical answer after a valid reasoning-close marker and completed output. Output length excludes prompt and fixed scaffold tokens; it does not measure useful reasoning or wall-clock latency.

Reported intervals are based on problem-level paired differences. Repeated generations are first averaged within each problem, followed by benchmark-stratified resampling. The comparison families are retained in the tables below. The evaluation suite was fixed but not fully sequestered, as its results were consulted during method development.

C.5 Matched Controls and Paired Uncertainty

The main table uses the original opening-memory Native and Mean-V results. Table 8 reports their independently measured counterparts from the SEAL/ReBalance execution, together with the corresponding published-method results. All means are computed over the same 1,919 problems, allowing each method to be compared with its matched execution-path control.

Table 8: Matched controls for SEAL and ReBalance on 1,919 problems, with two generations per method. Accuracy and length use all outputs.

Model	Method	Accuracy (%)	Output tokens
Qwen	Native	67.46	6,485
Qwen	Mean-V	66.86	5,838
Qwen	SEAL	67.59	5,626
Qwen	ReBalance	66.65	6,302
DeepSeek	Native	55.73	5,297
DeepSeek	Mean-V	54.35	5,234
DeepSeek	SEAL	54.64	4,005
DeepSeek	ReBalance	55.68	4,806

The paired intervals in Table 9 compare each method with its matched Native control. We retain the original 32-endpoint comparison family and its 99.84375% intervals, although only contrasts with Native are reported here. The 800-problem group consists of GSM8K and MATH, while the 1,119-problem group consists of AIME, AMC, Minerva Math, and OlympiadBench. Repeated outputs are first averaged within each problem, followed by benchmark-stratified resampling. The combined main table reports descriptive means and does not redefine these comparison groups or introduce an additional significance test.

Table 9: Accuracy and length differences from each method's matched Native control. Intervals retain the original 99.84375% level. Accuracy differences are percentage points; negative token differences mean shorter outputs.

Group	Method	Δ accuracy [interval]	Δ tokens [interval]
Qwen			
1119	SEAL	+0.18 [-2.11, 2.46]	-949 [-1142, -742]
1119	ReBalance	-1.03 [-3.27, 1.21]	-186 [-378, 13]
800	SEAL	+0.06 [-1.76, 1.89]	-735 [-893, -584]
800	ReBalance	-0.50 [-2.25, 1.25]	-179 [-321, -31]
DeepSeek			
1119	SEAL	-1.61 [-4.39, 1.16]	-1930 [-2224, -1634]
1119	ReBalance	-0.31 [-2.87, 2.21]	-723 [-980, -464]
800	SEAL	-0.38 [-2.94, 2.01]	-400 [-600, -202]
800	ReBalance	+0.31 [-1.81, 2.31]	-167 [-314, -24]

C.6 O-STEER: PAIRED COMPARISONS

The reported O-Steer controller is the compressed policy from Section 3.4. Its proposal bank uses a fixed two-point training accuracy floor, while measured-anchor selection and controller fitting use $\varepsilon = 0.04$. Each policy uses two continuations per problem. The original problem-paired bootstrap intervals below have confidence level 99.6875%, accounting for the predeclared comparison family. Accuracy differences are reported in percentage points and length differences in output tokens. No policy is selected using these comparisons.

Table 10: O-Steer accuracy and length differences from Native on 1,919 problems, with the original 99.6875% paired intervals. Accuracy is in percentage points; length is in output tokens.

Model	Reference	Accuracy difference	Token difference
Qwen	Native	-5.91 $[-7.98, -3.88]$	-2621 $[-2820, -2418]$
DeepSeek	Native	-4.17 $[-6.29, -2.08]$	-1213 $[-1431, -1013]$

These results establish substantial shortening for both models, while neither model establishes the four-point accuracy-preservation target on evaluation.