applied sciences

MDPI

Article

Dynamic Closed-Loop Steering for Adaptive and Geometry-Preserving System-2 Reasoning

Deyu Meng [1,*], Tongchuan Xia [2] and Yuanxin Cai [1]

[1] School of Information Communication Engineering, Beijing Information Science and Technology University, No. 55 Taihanglu, Changping, Beijing 102206, China; cai_yuanxin@bistu.edu.cn
[2] School of Computer Science (National Pilot Software Engineering School), BUPT, Beijing University of Posts and Telecommunications, Xitucheng Road 10, Haidian District, Beijing 100876, China; xtc_heartune@bupt.edu.cn
* Correspondence: mengdeyv@gmail.com

Abstract

Large language models (LLMs) are undergoing a paradigm shift from fast generation to deliberate System-2 reasoning, where scaling test-time compute is critical for unlocking complex reasoning capabilities. Yet, more computation does not guarantee better reasoning: under unconstrained test-time scaling, models frequently fall into "overthinking" traps, repeatedly elaborating on incorrect trajectories while wasting token budgets. We propose Dynamic Closed-Loop Steering, a training-free framework that treats reasoning as a monitored control process rather than a static decoding run. The framework follows a sense–decide–act design: lightweight latent signals sense trajectory divergence, a feedback regulator determines adaptive intervention timing and intensity, and a geometry-preserving actuation step redirects hidden states while maintaining their norm structure. This design emphasizes adaptive control and macro-level trajectory observability over static hidden-state addition. Across mathematical and reasoning benchmarks, Dynamic Closed-Loop Steering improves final-answer accuracy while curbing excess token consumption, yielding Pareto-favorable trade-offs across the large majority of evaluated configurations, with graceful degradation near the model's capability floor. Trajectory analysis confirms the method effectively suppresses repetitive self-justification and accelerates recovery from erroneous episodes. These results establish closed-loop, geometry-preserving intervention as a practical foundation for robust System-2 reasoning.

Keywords: trustworthy AI; model robustness; interpretability; inference-time intervention; test-time compute; overthinking; manifold projection

Academic Editor: Douglas O'Shaughnessy

Received: 19 May 2026
Revised: 16 June 2026
Accepted: 22 June 2026
Published: 27 June 2026

1. Introduction

Large language models (LLMs) are undergoing a transition from fast System-1 generation to more deliberate System-2 reasoning, in which scaling test-time compute has become an important route to improved performance on challenging mathematical and logical tasks. As these reasoning-oriented capabilities are increasingly considered for deployment in high-stakes and trust-sensitive settings, their robustness, interpretability, and generation reliability become central concerns. From the perspective of trustworthy AI, it is therefore important not only to improve reasoning accuracy, but also to better understand and regulate how long-horizon reasoning trajectories evolve under extended inference.

Recent empirical studies show that unconstrained test-time scaling can also introduce a recurring failure mode often described as *overthinking*, where a model continues to elaborate

on an incorrect intermediate hypothesis while consuming substantial token budgets [1,2]. In practice, this behavior may appear as prolonged wandering trajectories, repetition, and unstable final answers. Such phenomena do not imply that additional computation is ineffective; rather, they suggest that stronger reasoning performance requires correspondingly stronger mechanisms for monitoring stability and guiding the reasoning process. This issue is closely related to the broader goals of safe and responsible LLM deployment, where robustness under extended inference is as important as peak benchmark accuracy.

Existing attempts to regulate long-horizon reasoning mainly follow two directions: external control over generation length and internal intervention on latent representations. Budget forcing regulates generation length through hard truncation or forced continuation tokens [3], offering a simple external control mechanism but sometimes reducing fluency or coherence. A related line of work, exemplified by ThinkBrake [4], uses a log-probability margin as an external sentence-boundary early-stopping heuristic for halting decoding. Our use of a margin signal is different in scope: it is not copied as a standalone module and is not used as a binary truncation rule, but serves as one adapted internal convergence cue within a broader closed-loop latent steering controller. Inference-time intervention and activation steering instead seek to correct trajectories directly in hidden space [5–8]. These approaches are promising because they operate closer to the model's internal reasoning dynamics. However, many existing implementations still rely on static or uniform injection policies, which may not adequately reflect token-level variation during reasoning. Moreover, when interventions are triggered only after surface-level error cues become visible, earlier hidden states and cached representations may already have drifted away from a desirable reasoning path.

A further issue is geometric consistency. We argue that current activation engineering overlooks a neglected geometric axiom of LLM intervention: hidden-state editing should respect norm stability. Modern LLMs are strongly shaped by LayerNorm or RMSNorm, suggesting that valid hidden states tend to lie near a high-dimensional hyperspherical manifold in which direction mainly carries semantic information while norm reflects a pretrained energy allocation. Under this view, the standard additive update $h_{\text{new}} = h + \alpha v$ can be suboptimal because even a meaningful steering direction may alter $\|h_{\text{new}}\|_2$ and move the state away from its original manifold geometry. We therefore treat semantic drift and state shock as hypothesized contributing factors, rather than as fully established causal mechanisms, for the distribution shifts, repetition, and degraded decoding behavior observed under strong linear interventions. The stability diagnostics in Section 4 provide complementary empirical evidence (as illustrated in Table 1): when norm preservation is removed, model outputs can collapse immediately into nonsensical symbol-like sequences. Consequently, a key question for inference-time intervention is not only whether a method changes the reasoning trajectory, but whether it does so while preserving the geometric structure that supports stable generation. This operational framework is designed to suppress repetitive self-justification, prevent state shock and semantic drift, and direct latent representations back toward valid manifolds where exploration and convergence are balanced.

To address these challenges, we propose Dynamic Closed-Loop Steering (DCLS) (schematically illustrated in Figure 1), a geometry-grounded framework that formulates inference-time intervention as a real-time closed-loop control problem over latent reasoning trajectories. The framework aims to improve correctness and efficiency under fixed test-time budgets while making macro-level trajectory changes observable. Rather than applying static or uniformly scheduled edits, it organizes intervention into a coupled sense–decide–act pipeline: latent sensors estimate whether reasoning is diverging or converging, a feedback regulator determines adaptive intervention timing and intensity,

and the resulting command is executed through a geometry-preserving path in which purification reduces orthogonal interference and *Spherical Linear Actuation* performs norm-preserving angular steering. In this way, sensing determines *when* and *why* to intervene, regulation determines *how much* to intervene, and geometric actuation determines *how* to intervene while preserving hidden-state structure. We use DCLS as the abbreviation for the full framework throughout the experiments.

Table 1. Illustrative token-level degeneration under increasing linear intervention strength on the AIME plane-geometry prompt. Each column corresponds to one intervention strength and first shows the decoded sentence fragment induced by positions 3–9, followed by the selected token, top-k candidates, and probabilities at each position. The transition from coherent text to symbol-like outputs illustrates how norm-disruptive linear intervention can lead to unstable decoding behavior. The bold and italic formatting in the table are used for highlighting parameter values (α), output headers, and model output strings, respectively.

Pos	Top-k		Top-k		Top-k	
	Token	Prob.	Token	Prob.	Token	Prob.
$\alpha = 0.0$	**Output:** *Okay, so I need to solve*					
3	Okay	0.996	Alright	0.004	okay	0.000
4	,	1.000	so	0.000	problem	0.000
5	so	0.852	let	0.148	okay	0.000
6	I	1.000	we	0.000	the	0.000
7	need	0.915	have	0.085	’ve	0.000
8	to	1.000	solve	0.000	solving	0.000
9	solve	0.715	find	0.263	figure	0.022
$\alpha = 0.6$	**Output:** *So, you need to find the*					
3	Okay	0.455	So	0.354	Also	0.167
4	,	0.987	we	0.009	you	0.004
5	you	0.860	we	0.080	problem	0.038
6	need	0.641	have	0.303	’re	0.017
7	to	1.000	a	0.000	the	0.000
8	find	0.580	solve	0.147	also	0.054
9	the	0.949	area	0.017	more	0.012
$\alpha = 1.0$	**Output:** `] } } } nr } }`					
3	]	0.322	}	0.251	\n	0.195
4	}	0.612	\n	0.137	]	0.083
5	}	0.543	nr	0.107	\n	0.065
6	}	0.524	nr	0.248	wikipedia	0.030
7	nr	0.637	nr	0.142	wikipedia	0.059
8	}	0.794	wikipedia	0.058	nr	0.024
9	}	0.848	wikipedia	0.037	nr	0.011

Building on this motivation, our main contributions are summarized as follows.

- We formulate test-time intervention as closed-loop control over latent reasoning trajectories, framing overthinking as a dynamical stability problem in long-horizon LLM reasoning.
- We introduce a geometry-consistent intervention pipeline that combines manifold purification with Spherical Linear Actuation, reducing orthogonal interference and helping maintain hidden-state norm stability during steering.
- We provide empirical evidence that the proposed framework yields Pareto-favorable accuracy–efficiency trade-offs across the large majority of evaluated configurations while maintaining more stable repetition and entropy behavior.

Research hypotheses and questions. We hypothesize that overthinking in long-horizon LLM reasoning can be usefully modeled as a latent trajectory-stability problem, and that

adaptive closed-loop intervention can improve reasoning outcomes when its actuation preserves the model's hidden-state geometry. This hypothesis leads to three research questions: **RQ1:** Can Dynamic Closed-Loop Steering improve final-answer accuracy while avoiding unnecessary token growth? **RQ2:** Does geometry-preserving Spherical Linear Actuation reduce the decoding instability associated with norm-disruptive linear intervention? **RQ3:** Do trajectory-level signals, including repetition, entropy, and error-to-correct transitions, provide observable evidence that the framework changes how models recover from overthinking episodes?

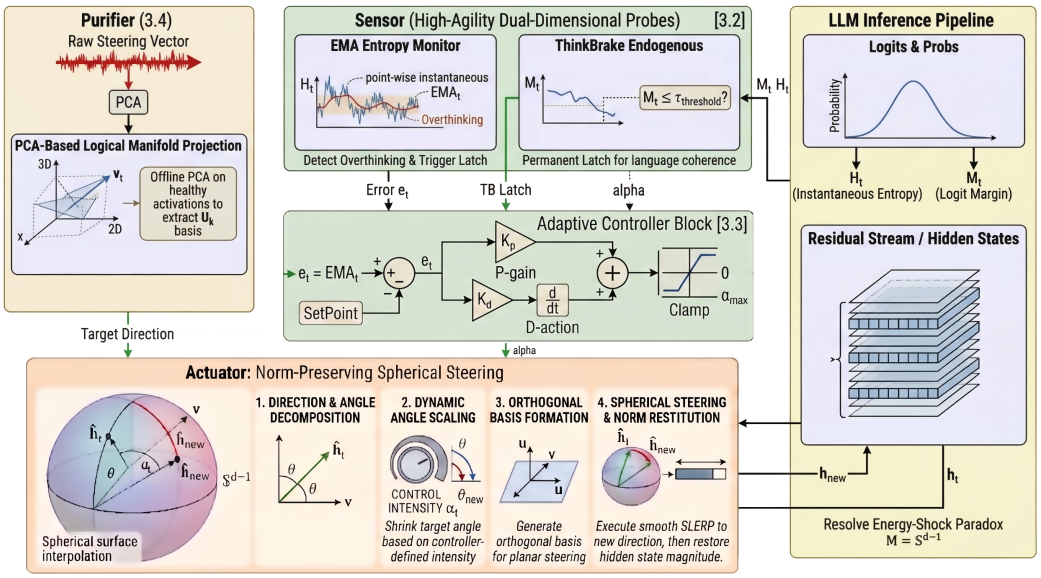


Figure 1. System architecture of the proposed Dynamic Closed-Loop Steering (DCLS) framework. The pipeline follows a sense–regulate–purify–steer cycle: (1) high-agility dual-dimensional sensors (the EMA Entropy Monitor and ThinkBrake Endogenous) detect overthinking or trajectory convergence in the LLM inference pipeline; (2) the adaptive controller block determines the intervention intensity α based on feedback; (3) the purifier filters raw steering vectors using PCA; and (4) the actuator applies norm-preserving spherical steering (SLERP) to guide the residual stream's hidden states, resolving the energy-shock paradox.

2. Related Work

2.1. Scaling Test-Time Compute and Overthinking

Scaling test-time computation has become a central paradigm for improving reasoning in large language models, especially on tasks that require multi-step deduction, planning, and self-correction [3,9–11]. At a high level, this line of work studies how additional inference-time budget can be converted into better problem solving through explicit deliberation rather than direct next-token prediction. Conceptually, this shift is often associated with the distinction between fast heuristic "System 1" behavior and slower, more deliberate "System 2" reasoning [12], where the latter relies on intermediate reasoning traces, search, and verification to overcome the brittleness of purely left-to-right generation.

Along this trajectory, the field has evolved from simple sequential elicitation to increasingly structured reasoning mechanisms. Chain-of-Thought prompting shows that generating intermediate rationales can unlock latent reasoning ability in sufficiently capable models, while zero-shot CoT and synthetic rationale generation further reduce dependence on handcrafted demonstrations [13–15]. Beyond a single reasoning path, self-consistency improves robustness by sampling multiple trajectories and aggregating their conclusions, thereby using agreement across paths as both a decoding strategy and a weak uncertainty signal [16]. More recent search-based frameworks such as Tree of Thoughts extend this idea

by explicitly branching, evaluating, and backtracking over candidate thoughts, moving test-time scaling from longer reasoning traces to deliberate exploration of a structured search space [14,17]. Production reasoning models such as OpenAI o1 and DeepSeek-R1 further anchor this trend, demonstrating that performance on mathematics, coding, and scientific reasoning can continue to improve as more inference-time computation is allocated [9,10,18].

The work most closely related to ours is this broader family of test-time scaling and deliberate reasoning methods, because they share the same objective of improving difficult reasoning by intervening during inference rather than only through larger pretraining budgets. However, existing approaches predominantly treat "more compute" as intrinsically beneficial: they lengthen chains, sample more candidates, or enlarge the search tree, often assuming that better answers will emerge from more extensive deliberation. This assumption is increasingly challenged by recent evidence that excessive reasoning can also induce overthinking, including redundancy, error reinforcement, and inefficient token expenditure [1,2,19].

Our method is motivated precisely by this gap. Rather than merely encouraging the model to think longer, we focus on regulating *when* additional intervention is helpful and *when* it becomes counterproductive. In this sense, our framework complements test-time scaling methods: instead of optimizing only the quantity of reasoning, it seeks to improve the quality and stability of the reasoning trajectory itself, providing adaptive assistance when the model shows signs of confusion while avoiding unnecessary disturbance during semantically healthy generation. This difference is crucial for long-horizon reasoning, where the challenge is not only to allocate more computation, but to prevent that computation from drifting into unproductive or self-reinforcing failure modes.

This distinction also separates our method from logits-level inference-time techniques such as contrastive decoding, contrastive search, and dynamic temperature scaling. Those methods reshape the output distribution or sampling policy at the token level, whereas Dynamic Closed-Loop Steering acts on internal representations through a state-dependent controller. The two families are therefore orthogonal: logits-level decoding can regulate token selection, while latent closed-loop steering regulates the hidden trajectory that gives rise to those token distributions.

2.2. Inference-Time Intervention and Activation Engineering

Inference-time intervention (ITI) and activation engineering study how a model's behavior can be steered by modifying its internal hidden states during the forward pass, without updating model weights through fine-tuning or RL-based retraining [5,7,20]. At a broad level, this research direction treats latent representations as a causal interface for control: instead of supervising only external outputs, it attempts to identify internal directions associated with traits such as truthfulness, safety, honesty, or refusal, and then manipulate these directions online to reshape generation. The standard formulation assumes that a concept can be represented by a steering vector v in latent space, so that the hidden state is updated as $h_{\text{new}} = h + \alpha v$, where α controls intervention strength [21].

Within this paradigm, the field has progressed from concept discovery to increasingly practical control mechanisms. Early work on Concept Activation Vectors (CAVs) established that human-interpretable concepts can often be identified through linear probes over hidden activations, providing a geometric basis for concept-level intervention [22]. Representation Engineering (RepE) extended this logic into a more unified reading-and-control framework, in which latent directions extracted from contrastive stimuli are used to modulate high-level model "mindsets" such as honesty or utility. In parallel, contrastive activation addition and related steering methods showed that averaging activation differences

over positive/negative prompt pairs can yield robust steering vectors that continuously nudge model behavior throughout generation. A more targeted branch of this literature, represented by ITI, further demonstrated that intervention can be localized to a sparse subset of attention heads that track truthfulness, leading to strong gains on benchmarks such as TruthfulQA [5].

The work most closely related to ours is therefore the family of activation steering and inference-time intervention methods that seek to improve model behavior through latent-space editing at generation time. These methods are highly relevant because, like our framework, they intervene online rather than retraining the entire model. However, the dominant operational pattern in this literature is still essentially *open-loop*: once a steering direction is identified, a fixed vector and coefficient are applied uniformly across tokens, layers, or selected modules. Even when the target concept is well chosen, such uniform intervention can ignore the fact that uncertainty and semantic instability vary substantially across reasoning steps, so the same control strength may be insufficient in some moments and excessive in others.

This limitation directly motivates our approach. Rather than treating intervention as a static additive bias, we frame it as a dynamic control problem over the evolving reasoning trajectory. In particular, our method introduces history-aware sensing through EMA to estimate whether the model is entering a genuinely confused state, and it couples this sensing mechanism with adaptive non-linear PD feedback so that the intervention magnitude responds to the current degree of instability rather than remaining fixed. This makes our approach different from standard CAV-, RepE-, and ITI-style steering: we do not only ask *what* direction should be added, but also *when*, *where*, and *how strongly* that intervention should be applied. Such adaptive regulation is especially important for long-horizon reasoning, where dense or overly rigid steering can introduce side effects, over-correction, or control oscillation across extended token trajectories [23].

2.3. Latent Space Geometry and Norm Stability in LLMs

Latent space geometry and norm stability research examines the structural properties of transformer representations, asking how semantic information is organized in high-dimensional hidden space and how normalization constrains the evolution of those states during training and inference. At a broad level, this literature is motivated by the manifold hypothesis, which suggests that although hidden states live in very high ambient dimensions, the effective semantic and syntactic degrees of freedom lie on much lower-dimensional structures [24,25]. This perspective has made geometric regularity, representation anisotropy, and norm control central questions for understanding why modern LLMs remain numerically stable while still supporting rich reasoning behavior. In particular, it motivates a view in which direction and magnitude play different functional roles: angular structure often carries semantic relations, whereas norm can reflect feature salience, activation energy, or confidence [26].

Along this trajectory, the field has evolved from general manifold-based interpretations of hidden states to more mechanistic analyses of normalization and hyperspherical representation learning. Recent studies characterize transformer representations as undergoing layer-wise expansion and compression, suggesting that middle layers enlarge the effective feature space while later layers contract toward task-relevant manifolds [27]. In parallel, work on LayerNorm and RMSNorm has clarified that normalization is not merely an optimization trick but a geometric constraint: LayerNorm projects activations toward a mean-zero hyperplane, whereas RMSNorm preserves directional structure while rescaling magnitude on a hyperspherical manifold [28]. This line of research has become especially influential because RMSNorm-based pre-normalization is now widely adopted in frontier

models, where stable residual norms are essential for deep scaling and long-context training. More recent architectures such as nGPT push this logic further by explicitly enforcing unit-norm representations, thereby decoupling semantic direction from scalar energy and interpreting hidden-state updates as constrained motion on the hypersphere [26].

The work most closely related to ours is this normalization- and geometry-centered literature, because it provides the clearest account of why latent interventions cannot be judged only by their semantic intent. If hidden states occupy structured manifolds and rely on norm stability for numerical health, then any intervention that perturbs them must also respect these geometric constraints. However, much of the existing literature is primarily descriptive or architectural: it explains why representations are anisotropic, why RMSNorm is efficient, or why hyperspherical parameterizations improve optimization, but it does not directly address how an online controller should intervene in a reasoning trajectory while preserving latent-space stability. Conversely, many activation steering methods manipulate hidden states without explicitly accounting for whether the intervention displaces the model toward geometrically unstable or semantically distorted regions.

Our method connects these two perspectives. We treat norm stability and latent geometry not as background properties, but as operational constraints on inference-time control. This is precisely why our framework avoids frequent, fixed-strength perturbations and instead uses adaptive feedback to regulate intervention strength according to the model's evolving state. In effect, our method is designed to preserve semantically meaningful trajectories while reducing unnecessary excursions that could amplify anisotropy, destabilize residual norms, or push the reasoning process into unproductive regions of latent space. This distinction is important for long-horizon reasoning: the key challenge is not only to steer the model toward a better direction, but to do so in a way that remains compatible with the geometry and stability of the underlying representation manifold.

2.4. Degeneration, Hallucination, and Long-Horizon Robustness

Degeneration, hallucination, and long-horizon robustness research studies the failure modes that emerge when autoregressive models generate extended outputs over long contexts. At a broad level, this literature asks why models that are locally fluent can still become globally unstable, drifting away from task intent, factual grounding, or coherent reasoning as generation proceeds [29,30]. The core mechanisms identified in this line of work include exposure bias, semantic drift, entropy collapse, and the snowball effect of hallucinations, all of which arise from the mismatch between teacher-forced training and self-conditioned inference. In this view, long-horizon failure is not simply a decoding artifact but a structural property of autoregressive generation, where small early errors are recursively reintroduced into the context and amplified over time.

Along this trajectory, the field has evolved from early studies of neural text degeneration and repetitive loops to more precise accounts of entropy dynamics, hallucination taxonomy, and long-context semantic instability. Foundational work on degeneration showed that maximum-likelihood training can yield outputs that are fluent yet bland, repetitive, or trapped in self-reinforcing attractors, especially when decoding enters low-entropy loops [31,32]. Subsequent analyses connected this behavior to the softmax bottleneck and anisotropic representation geometry, explaining why models may overestimate unreliable tail probabilities while failing to discriminate effectively among semantically nearby tokens. In parallel, hallucination research clarified how intrinsic and extrinsic errors propagate through sequential generation, giving rise to a snowball effect in which one mistaken step becomes the premise for many later ones. More recent work on long-context robustness further shows that semantic drift, retrieval bias, and "lost-in-the-middle" failures persist

even as context windows scale, indicating that longer context alone does not guarantee stable reasoning over extended trajectories [33].

The work most closely related to ours is therefore the family of studies that analyze degeneration and hallucination as sequential dynamical failures rather than isolated output errors. These studies are especially relevant because, like our framework, they focus on the evolution of the reasoning path itself. However, most existing approaches remain concentrated on either training-time fixes, such as unlikelihood objectives and entropy-aware RL, or decoding-time heuristics, such as contrastive search and tree-based exploration [34]. While these methods are valuable, they often do not provide a direct mechanism for monitoring the model's internal state online and intervening exactly when a trajectory begins to drift into confusion, repetition, or hallucination. As a result, they diagnose the symptoms of long-horizon failure more clearly than they regulate the failure process itself.

Our method is motivated by precisely this gap. Rather than treating degeneration only as a sampling problem or hallucination only as a final-answer error, we view both as manifestations of unstable reasoning dynamics unfolding in latent space. This is why our framework combines history-aware sensing with adaptive feedback control: it aims to detect when the model is entering a confused or self-reinforcing regime and to intervene before that local error compounds into a long-horizon collapse. In this sense, our approach complements prior robustness research by shifting attention from post hoc correction to online trajectory stabilization. The key difference is that we do not merely reduce repetition or hallucination after they appear; we seek to regulate the internal evolution that gives rise to semantic drift, entropy collapse, and error snowballing in the first place.

3. Methodology

3.1. Background Overview: From Overthinking Control to Geometry-Constrained Intervention

The shift from System-1 generation to System-2 reasoning turns autoregressive decoding into a longer-horizon control process: the model must explore intermediate hypotheses, correct wrong branches, and still converge under a finite test-time budget. The core failure mode is *overthinking*, where hidden trajectories remain trapped in high-entropy local attractors and repeatedly elaborate an incorrect latent hypothesis instead of progressing toward a valid solution.

Existing ITI and activation-steering methods mainly rely on open-loop linear editing, typically $h_{t+1} = h_t + \alpha v$. This design is limited in two coupled ways. Dynamically, fixed or uniformly scheduled intervention ignores token-level uncertainty variation and cannot decide *when* stronger correction is needed or *when* withdrawal is safer. Geometrically, because modern LLM hidden states are strongly shaped by LayerNorm/RMSNorm, valid activations lie near a hyperspherical manifold $\mathcal{M} = \mathbb{S}^{d-1}$; linear translation perturbs the norm, injects high-dimensional orthogonal noise, and may push states off-manifold, causing semantic drift, repetition inflation, and severe *state shock*.

Motivated by these coupled limitations, we reformulate inference-time intervention as a geometry-constrained closed-loop control problem on latent trajectories. Rather than treating sensing, intensity selection, and hidden-state editing as separate tricks, we integrate them into a single causal pipeline that monitors reasoning state in real time, computes adaptive steering magnitude, purifies the steering direction, and applies norm-stable manifold motion only when intervention is genuinely beneficial.

3.2. Unified Framework: Sense–Regulate–Purify–Steer–Recover

Our framework treats decoding as a discrete-time non-linear dynamical system on a reasoning manifold (formalized in Algorithm 1). The controlled hidden state is written as

$$h_t^{(\mathrm{ctrl})} = \begin{cases} \mathrm{Perturb}(h_t; \gamma), & \text{if collapse is detected,} \\ \mathrm{Geodesic\_Flow}\Big(h_t,\, \mathrm{Proj}_{U_k}(v_{\mathrm{raw}});\, \alpha_t\Big), & \text{if } M_t > \tau_{\mathrm{brake}}, \\ h_t, & \text{otherwise,} \end{cases}$$

where $\mathrm{Proj}_{U_k}(\cdot)$ denotes manifold-aligned purification, α_t is the adaptive control intensity, M_t is the logit-margin convergence signal, and $\mathrm{Perturb}(\cdot)$ is an anti-collapse recovery operator.

The framework begins with real-time sensing in latent space. Because lexical trigger tokens are delayed relative to residual-stream corruption, we use two complementary probes: an EMA entropy signal for divergence tracking,

$$\mathrm{EMA}_t = \beta H_t + (1-\beta)\mathrm{EMA}_{t-1},$$

and a logit-margin convergence signal,

$$M_t = \log p(y_t^\star) - \log p(y_{\mathrm{term}}),$$

to detect endogenous convergence and latch intervention off near completion. EMA provides a low-pass, history-aware estimate of uncertainty, while the margin signal prevents unnecessary late-stage steering.

The sensed divergence is then translated into steering intensity by an **Adaptive Nonlinear PD Regulator**. Instead of using a fixed proportional gain and hard clipping, we introduce an error-dependent gain schedule and smooth saturation. Let the safe entropy setpoint be SetPoint and define a non-negative error

$$e_t = \max(0, \mathrm{EMA}_t - \mathrm{SetPoint}).$$

The proportional gain increases with error magnitude,

$$K_p(e_t) = K_{p\_\min} + \frac{K_{p\_\max} - K_{p\_\min}}{1 + \exp[-\lambda(e_t - e_{\mathrm{mid}})]},$$

so the controller remains gentle near the safe zone but becomes more decisive once divergence grows. The raw control signal is

$$P_t = K_p(e_t)e_t, \qquad D_t = K_d(e_t - e_{t-1}), \qquad u_t = \max(0, P_t + D_t),$$

followed by tanh-based soft clipping,

$$\alpha_t = \alpha_{\max} \tanh\left(\frac{u_t}{\alpha_{\max}}\right).$$

To prevent fragile high-entropy states from being overly influenced by strong steering, we further apply entropy-scaled suppression (EAST), yielding a final intervention strength α_t' that decays automatically in the very-high-entropy regime; if the logit-margin signal indicates convergence, we force $\alpha_t' = 0$ regardless of the controller output.

Once intensity is determined, the framework applies geometry-consistent actuation in two stages. First, PCA-based projection removes orthogonal noise from the raw steering

vector by restricting it to a task-relevant logical subspace. Given a basis $U_k = [u_1, \ldots, u_k]$ estimated from normal reasoning activations, we compute

$$v_{\text{purified}} = \sum_{i=1}^{k} (v_{\text{raw}} \cdot u_i) u_i.$$

Second, rather than using additive translation, we rotate the hidden state by norm-preserving SLERP. For hidden state h_t, unit target v, and intervention intensity α'_t,

$$\hat{h}_t = \frac{h_t}{\|h_t\|_2}, \quad \theta = \arccos(\hat{h}_t \cdot v), \quad \theta_{\text{new}} = (1 - \alpha'_t)\theta,$$

$$u = \frac{\hat{h}_t - \cos(\theta) v}{\sin(\theta)}, \qquad \hat{h}_{\text{rotated}} = \cos(\theta_{\text{new}}) v + \sin(\theta_{\text{new}}) u,$$

$$h_{\text{new}} = \|h_t\|_2 \, \hat{h}_{\text{rotated}}.$$

This turns intervention into angular motion on the hypersphere, preserving hidden-state energy while still injecting enough geometric momentum to escape incorrect attractors.

Finally, we introduce an **Anti-Collapse Manifold Perturbation** mechanism to handle a failure mode that ordinary steering cannot fix: over-purification. When strong directional control collapses the model into ultra-low-entropy repetitive loops, the PD pathway may become falsely silent because low entropy no longer indicates healthy convergence. We therefore monitor a low-entropy persistence counter together with local N-gram repetition. If both are triggered, the system temporarily suspends normal SLERP and injects a tiny norm-preserving perturbation along a direction orthogonal to the current hidden state. Sampling $v_{\text{noise}} \sim \mathcal{N}(0, I)$, we construct

$$z = v_{\text{noise}} - \frac{v_{\text{noise}} \cdot h_t}{\|h_t\|_2^2} h_t, \qquad z_t = \frac{z}{\|z\|_2},$$

and perturb the state by

$$h'_t = \frac{h_t + \gamma \, \|h_t\|_2 \, z_t}{\|h_t + \gamma \, \|h_t\|_2 \, z_t\|_2} \cdot \|h_t\|_2.$$

After this pulse, the controller enters a short cooldown period with $\alpha = 0$, allowing the model to recover free reasoning from a new manifold location rather than being continuously forced along a collapsed direction.

Novelty Taxonomy and Methodological Boundaries

Dynamic Closed-Loop Steering differs from prior activation-space baselines along three axes. First, compared with CAA-style contrastive activation addition, our method does not apply an unconstrained Euclidean translation; PCA purification and Spherical Linear Actuation keep the update close to the reasoning manifold, reducing the state shock that can arise when additive edits alter hidden-state norm and inject orthogonal noise. Second, compared with SEAL-style entropy-aware latching, our controller is not a static on/off intervention: EMA entropy tracking supplies history-aware endogenous sensing, while the adaptive non-linear PD regulator continuously changes intervention magnitude according to current divergence. Third, compared with static Spherical-Steering or instantaneous vMF gates, our framework is stateful: the logit-margin signal records convergence evidence, the PD derivative term reacts to trajectory changes, and Anti-Collapse Manifold Perturbation explicitly recovers from low-entropy repetitive attractors that a memoryless gate can fail to detect. The core contribution is therefore not a single steering vector, but the integration of **Dual Endogenous Latent Probes**, **Adaptive Non-Linear PD with EAST**, and

Anti-Collapse Manifold Perturbation into one causal closed-loop pipeline. The method is intended to redirect unstable but solvable reasoning trajectories; it cannot supply missing domain knowledge when a model is below the task's capability floor.

Algorithm 1 Dynamic Closed-Loop Steering

Require: Prompt x, model M, raw steering vector v_{raw}, PCA basis U_k, thresholds $(\tau_{\text{entropy}}, \tau_{\text{high}}, \tau_{\text{brake}})$, gains $(K_{p\_\min}, K_{p\_\max}, K_d)$, maximum strength $\alpha_{\max}$, EMA decay β, perturbation scale γ, cooldown length C
Ensure: Generated response R

$R \leftarrow x$; EMA $\leftarrow 0$; $e_{\text{prev}} \leftarrow 0$; $\alpha \leftarrow 0$; $c \leftarrow 0$; first $\leftarrow$ TRUE
$v_{\text{purified}} \leftarrow \text{Normalize}(U_k U_k^\top v_{\text{raw}})$ ▷ PCA projection purification
while not EOS and $|R| < T_{\max}$ **do**
  if LowEntropyPersist(EMA) **and** NGramRepeat(R) **then**
    $h_t \leftarrow \text{HiddenState}(M, R, l^*)$
    $z \leftarrow \text{OrthogonalNoise}(h_t)$; $h_t \leftarrow \text{Renorm}(h_t + \gamma \|h_t\|_2 z)$
    $\alpha \leftarrow 0$; $c \leftarrow C$ ▷ Anti-Collapse pulse and cooldown
  end if
  if $c > 0$ **then**
    $\alpha \leftarrow 0$; $c \leftarrow c - 1$
  end if
  $(\ell_t, h_t) \leftarrow M.\text{Forward}(R, l^*)$
  $h_t^{\text{ctrl}} \leftarrow \text{SLERP}(h_t, v_{\text{purified}}, \alpha)$ ▷ Spherical Linear Actuation
  $\ell_t \leftarrow M.\text{ContinueFrom}(h_t^{\text{ctrl}})$
  $H_t \leftarrow -\sum_y p_t(y) \log p_t(y)$
  if first **then**
    EMA $\leftarrow H_t$; first $\leftarrow$ FALSE
  else
    EMA $\leftarrow \beta H_t + (1 - \beta)$EMA
  end if
  $e_t \leftarrow \max(0, \text{EMA} - \tau_{\text{entropy}})$
  $K_p(e_t) \leftarrow K_{p\_\min} + \dfrac{K_{p\_\max} - K_{p\_\min}}{1 + \exp[-\lambda(e_t - e_{\text{mid}})]}$
  $u_t \leftarrow \max\{0, K_p(e_t) e_t + K_d(e_t - e_{\text{prev}})\}$ ▷ Non-linear adaptive PD
  $\alpha_{\text{raw}} \leftarrow \alpha_{\max} \tanh(u_t / \alpha_{\max})$
  if EMA $> \tau_{\text{high}}$ **then**
    $\alpha \leftarrow \alpha_{\text{raw}} \exp[-\rho(\text{EMA} - \tau_{\text{high}})]$ ▷ EAST boundary protection
  else
    $\alpha \leftarrow \alpha_{\text{raw}}$
  end if
  if LogitMargin$(\ell_t) < \tau_{\text{brake}}$ **then**
    $\alpha \leftarrow 0$ ▷ Latch off near answer convergence
  end if
  $e_{\text{prev}} \leftarrow e_t$
  $y_t \leftarrow \text{Sample}(\ell_t)$; $R \leftarrow R \| y_t$
end while
return R

4. Experiments and Results

4.1. Evaluation Setup and Benchmarks

We evaluate Dynamic Closed-Loop Steering on mathematical reasoning, competition-style reasoning, code logic, and simple symbolic logic benchmarks, using the experimental configuration summarized in Table 2. The main benchmark suite includes MATH500 [35], AIME2024 [36], AIME2025 [37], CRUXEval code-output prediction, and BBH Boolean Expressions. Unless otherwise noted, Qwen3-8B in thinking mode [38] is the primary model for the original MATH500 and trajectory analyses, while cross-model generalization is eval-

uated on Qwen3-8B, DeepSeek-R1-Distill-Qwen-1.5B, and Gemma-4-E2B. The compared methods include the unsteered baseline, Continuous Linear (ITI), CAA, Mitigating Overthinking [39], Spherical-Steering, SEAL, Continuous Spherical-Steering, and our Dynamic Closed-Loop Steering framework. We report Pass@1 as the primary metric and additionally report average generated tokens, repetition rate, perplexity (PPL), and global deep-thinking ratio (DTR) where available.

Table 2. Experimental configuration used for the revised evaluation. All steering variants share the same decoding and hardware settings unless explicitly stated.

Item	Configuration
Models	Qwen3-8B; DeepSeek-R1-Distill-Qwen-1.5B; Gemma-4-E2B
Precision and implementation	BFloat16; PyTorch (version 2.11.0+cu130) and Transformers (version 5.0.0.dev0); single NVIDIA RTX 4090 GPU
Decoding	Temperature 0.6, top-$p = 0.95$, top-$k = 20$, min-$p = 0.05$
Random seed	42 for all controlled runs
Token budgets	32,768 for AIME; 8192 for hard MATH500/CRUXEval; 4096 for full MATH500/BBH
Metrics	Pass@1, average generated tokens, repetition, PPL, DTR, and transition statistics
Controller parameters	EMA smoothing $\beta = 0.1$, entropy trigger 0.15, adapted ThinkBrake-style margin latch $\tau = 0.25$
PD gains and manifold dimension	$K_P = 2.0$, $K_I = 0$, $K_D = 0.5$, PCA dimension $k = 10$

For hyperparameter selection, we follow a fixed protocol rather than tuning separately on each test set. The theoretical upper operating ceiling is $\alpha_{\max} = 0.5$, derived from the soft-clipped controller and used as the strongest safe intervention setting. For simpler or regular-budget tasks, we use the milder $\alpha_{\max} = 0.3$, which gives the best accuracy–efficiency trade-off on full MATH500 under a 4096-token budget and avoids over-intervention. For harder long-budget tasks, including the 116-problem MATH500 difficulty-3/4 subset under an 8192-token budget, we evaluate the stronger $\alpha_{\max} = 0.5$. All remaining controller parameters are fixed across models and benchmarks. The section is organized in five stages: top-line answer-level results, cross-model generalization, generalization to other reasoning domains, trajectory-level analysis, and efficiency/stability diagnostics including expanded ablations and overhead measurements.

All controlled generations use a fixed random seed (42) and identical decoding parameters, so the reported answer-level values correspond to one deterministic evaluation pass per configuration rather than averages over independently resampled runs. We therefore interpret small AIME differences cautiously: because each AIME split contains only 30 problems, a 3.33 percentage-point change corresponds to exactly one additional solved problem. For this reason, answer-level AIME gains are reported together with trajectory-level statistics over many reasoning blocks, human-validated LLM-as-Judge annotations in Appendix A, and activation-vector checksums in Appendix B, which provide complementary evidence about recovery, degradation, and reproducibility.

To establish rigorous statistical significance at the problem level (thereby avoiding the clustering bias of pooling reasoning blocks from the same trajectory), we apply McNemar's test for paired binary outcomes. The paired independent units used in this test are the individual problems (or trajectories), which are solved independently. For each problem, we obtain a paired binary outcome indicating correctness under the baseline and steered configurations. These paired outcomes are compiled into a 2×2 contingency table with the following cell counts: a (both correct), b (baseline correct, steered incorrect; degraded), c (baseline incorrect, steered correct; rescued), and d (both incorrect). Under the null hypothesis that steering has no effect on correctness, the probability of a baseline-only correct outcome equals that of a steered-only correct outcome ($b = c$). The McNemar test statistic (with Yates's continuity correction) is calculated as:

$$\chi^2_{\text{McNemar}} = \frac{(|b - c| - 1)^2}{b + c}, \quad \text{d.f.} = 1 \tag{1}$$

where $b + c$ is the total number of discordant pairs. For Qwen3-8B on the full MATH500 dataset (500 independent problems, $\alpha_{\max} = 0.3$), the contingency table is $a = 331$, $b = 5$, $c = 51$, and $d = 113$. McNemar's test with continuity correction yields $\chi^2 = 36.1607$, $p < 0.0001$ (without correction, $\chi^2 = 37.7857$, $p < 0.0001$), demonstrating that the accuracy improvement from 67.20% to 76.40% is highly statistically significant. For Qwen3-8B on the MATH500 Level 3-4 subset (116 independent problems, $\alpha_{\max} = 0.5$), the contingency table is $a = 100$, $b = 1$, $c = 10$, and $d = 5$. McNemar's test with continuity correction yields $\chi^2 = 5.8182$, $p = 0.0159$ (without correction, $\chi^2 = 7.3636$, $p = 0.0067$), demonstrating that the accuracy improvement from 87.07% to 94.83% is statistically significant ($p < 0.05$). For Gemma-4-E2B on the CRUXEval dataset (800 independent problems, $\alpha_{\max} = 0.5$), the contingency table is $a = 613$, $b = 27$, $c = 39$, and $d = 121$. McNemar's test with continuity correction yields $\chi^2 = 1.8333$, $p = 0.1757$ (without correction, $\chi^2 = 2.1818$, $p = 0.1396$), indicating a positive trend on this larger symbolic reasoning benchmark. For DeepSeek-1.5B on the CRUXEval dataset (800 independent problems, $\alpha_{\max} = 0.3$), the contingency table is $a = 516$, $b = 53$, $c = 55$, and $d = 176$. McNemar's test yields $\chi^2 = 0.0093$, $p = 0.9233$, confirming that both configurations are statistically comparable at the answer level, while steering achieves a Pareto-favorable 8.29% reduction in generated tokens.

4.2. Main Results: Top-Line Performance on MATH500 and AIME

We begin by establishing the answer-level effect of the controller on the two benchmark families. On MATH500, Dynamic Closed-Loop Steering delivers consistent gains across all tested intervention ceilings and reaches the best overall accuracy of 76.40% at $\alpha_{\max} = 0.3$, substantially above the 67.20% baseline. Even at the more aggressive ceiling $\alpha_{\max} = 0.45$, the method still maintains 74.00% accuracy while reducing average token usage from 3045 to 2785. At the benchmark level, the key message is therefore not only that closed-loop steering improves correctness, but that it does so without paying the usual price of uncontrolled compute inflation.

The transfer results on AIME reinforce the same headline while revealing dataset-specific optima. On AIME2025, the baseline reaches 63.33%, while Dynamic Closed-Loop Steering improves to 73.33% at $\alpha_{\max} = 0.35$ and further to 76.67% at $\alpha_{\max} = 0.50$, before easing to 66.66% at $\alpha_{\max} = 0.6$. On AIME2024, the baseline is already stronger at 76.67%, and Dynamic Closed-Loop Steering peaks at 80.00% when $\alpha_{\max} = 0.3$, while the other tested ceilings remain below that mark. Taken together, Tables 3 and 4 show that the method is robustly beneficial on MATH500 and AIME2025, and can also produce gains on AIME2024 when the steering ceiling is tuned appropriately. Having established this top-line advantage, we next ask where it comes from inside the reasoning trajectory itself rather than at the final-answer level alone.

Table 3. Structured summary of results on MATH500 [35]. We report completed runs together with pre-registered comparison baselines; unfilled entries are kept to make the full evaluation design explicit. Best values are shown in bold and second-best values are underlined. ↑ indicates higher is better, and ↓ indicates lower is better.

Method/Setting	Accuracy (↑)	Avg Tokens (↓)	Repetition (↓)
Qwen3 baseline [38]	67.20	3045	**0.11**
CAA [7]	73.74	**1919**	<u>0.12</u>
Spherical-Steering [40]	69.33	2886	<u>0.12</u>
SEAL [41]	<u>74.14</u>	<u>2593</u>	**0.11**
Dynamic Closed-Loop Steering ($\alpha_{\max} = 0.15$)	70.00	2927	**0.11**
Continuous Spherical ($\alpha_{\max} = 0.3$)	74.00	<u>2593</u>	<u>0.12</u>
Dynamic Closed-Loop Steering ($\alpha_{\max} = 0.3$)	**76.40**	2856	<u>0.12</u>
Continuous Spherical ($\alpha_{\max} = 0.45$)	41.20	3416	0.28
Dynamic Closed-Loop Steering ($\alpha_{\max} = 0.45$)	74.00	2785	0.14

Table 4. Structured summary of results on AIME2024 [36] and AIME2025 [37]. We report completed runs together with pre-registered comparison baselines. Best values are shown in bold and second-best values are underlined within each benchmark block. ↑ indicates higher is better, and ↓ indicates lower is better.

Dataset	Method/Setting	Accuracy (↑)	Avg Tokens (↓)	Repetition (↓)
AIME2025	Qwen3 baseline [38]	63.33	17,212	**0.21**
	CAA [7]	60.00	**10,570**	0.23
	Spherical-Steering [40]	66.66	16,727	0.26
	SEAL [41]	70.00	<u>16,208</u>	<u>0.22</u>
	Dynamic Closed-Loop Steering ($\alpha_{max} = 0.3$)	<u>73.33</u>	16,973.6	0.23
	Dynamic Closed-Loop Steering ($\alpha_{max} = 0.35$)	<u>73.33</u>	17,524	**0.21**
	Dynamic Closed-Loop Steering ($\alpha_{max} = 0.50$)	**76.67**	<u>16,166</u>	**0.21**
	Dynamic Closed-Loop Steering ($\alpha_{max} = 0.6$)	66.66	17,493	**0.21**
AIME2024	Qwen3 baseline [38]	<u>76.67</u>	<u>12,904</u>	**0.19**
	CAA [7]	66.66	**10,531**	0.26
	Spherical-Steering [40]	73.33	15,329	0.26
	SEAL [41]	70.00	14,087	<u>0.21</u>
	Dynamic Closed-Loop Steering ($\alpha_{max} = 0.2$)	66.66	15,080	<u>0.21</u>
	Dynamic Closed-Loop Steering ($\alpha_{max} = 0.3$)	**80.00**	13,932	0.22
	Dynamic Closed-Loop Steering ($\alpha_{max} = 0.5$)	73.33	16,022.3	0.25
	Dynamic Closed-Loop Steering ($\alpha_{max} = 0.45$)	70.00	15,965	<u>0.21</u>
	Dynamic Closed-Loop Steering ($\alpha_{max} = 0.6$)	70.00	16,030	0.23

4.3. Stochastic Variability and Seed Robustness

To address concerns regarding the variability and robustness of the results under stochastic decoding, we repeated the main experiments on AIME2024 and AIME2025 using multiple random seeds (Seed $\in \{41, 42, 43\}$). Table 5 summarizes the mean and standard deviation of the Pass@1 accuracy, alongside the average generated tokens and repetition rate across these runs using Qwen3-8B.

Table 5. Variability and robustness analysis on AIME2024 and AIME2025 across multiple random seeds (Seeds 41, 42, and 43) using Qwen3-8B. Best values are shown in bold.

Benchmark/Configuration	Runs	Mean Acc.	Std Acc.	Mean Tokens	Repetition
AIME 2024 Baseline	3	71.11%	6.94%	14,326.1	18.80%
AIME 2024 $\alpha_{max} = 0.3$	3	72.22%	1.92%	15,880.4	22.84%
AIME 2024 $\alpha_{max} = 0.5$	3	**75.56%**	1.93%	15,527.2	22.76%
AIME 2025 Baseline	3	65.55%	3.85%	17,825.0	20.00%
AIME 2025 $\alpha_{max} = 0.3$	3	67.78%	6.94%	18,314.3	22.54%
AIME 2025 $\alpha_{max} = 0.5$	3	**71.11%**	5.09%	17,161.5	21.54%

The results show that both the baseline and steered configurations are highly stable across different random seeds. On AIME2024, our dynamic closed-loop steering with $\alpha_{max} = 0.5$ yields a mean accuracy of 75.56% with a small standard deviation of 1.93% (corresponding to only 1–2 problems of variation, given that each correct answer corresponds to 3.33 percentage points on a 30-problem set), representing a robust +4.45 pp improvement over the baseline (71.11% $\pm$ 6.94%). Similarly, on AIME2025, $\alpha_{max} = 0.5$ achieves the highest mean accuracy of 71.11% $\pm$ 5.09%, representing a +5.56 pp improvement over the baseline (65.55% $\pm$ 3.85%). Notably, the standard deviation of our steered configurations remains comparable to or even lower than the baseline (e.g., standard deviation drops from 6.94% to 1.93% on AIME2024, and stays at a low 5.09% on AIME2025), indicating that closed-loop latent steering does not introduce additional stochastic instability or cause the generation to diverge.

We note that the standard deviations of accuracy naturally overlap between some baseline and steered runs. This is an intrinsic property of small-sample reasoning evaluations like AIME ($N = 30$): because individual problem correctness is binary, the pass rate follows a

binomial distribution. For a model with true accuracy $p \approx 70\%$, the intrinsic binomial sampling noise is $\sigma_{\text{binomial}} = \sqrt{p(1-p)/N} \approx 8.37\%$. Thus, even for a hypothetical model with zero seed sensitivity, random correctness fluctuations across 30 problems would naturally produce standard deviations of ∼8.4%. The fact that our observed standard deviations across seeds are actually lower than this theoretical sampling noise threshold (e.g., 1.92∼5.09%) indicates that the closed-loop steering framework remains stable across the tested seeds.

4.4. Cross-Model Evaluation

To test whether the controller is tied to a single backbone, we extend the AIME evaluation to three model families: Qwen3-8B as the primary dense reasoning model, DeepSeek-R1-Distill-Qwen-1.5B as a much smaller distilled dense model, and Gemma-4-E2B as a compact mixture-of-experts model. Table 6 reports the strongest dynamic setting for each model–benchmark pair, while keeping the decoding setup and controller hyperparameters fixed except for the pre-specified intervention ceiling.

Table 6. Cross-model evaluation on AIME2024 and AIME2025. Token and repetition changes are computed relative to the corresponding unsteered baseline. Best values are shown in bold.

Model	Benchmark	Baseline Acc.	Dynamic Acc.	$\alpha_{\max}$	Δ Acc.	Δ Avg Tokens	Repetition Rate (Dynamic vs. Baseline)
Qwen3-8B	AIME 2025	63.33	**76.67**	0.5	**+13.33 pp**	−1046	0.210 vs. 0.210
DeepSeek-1.5B	AIME 2024	30.00	**33.33**	0.3	**+3.33 pp**	−1440	0.315 vs. 0.319
DeepSeek-1.5B	AIME 2025	**23.33**	20.00	0.3	−3.33 pp	+2595	0.344 vs. 0.393
Gemma-4-E2B	AIME 2024	40.00	**43.33**	0.5	**+3.33 pp**	+83	0.157 vs. 0.154
Gemma-4-E2B	AIME 2025	23.33	**33.33**	0.5	**+10.00 pp**	−63	0.138 vs. 0.146

The results support a qualified but broad generalization claim. Dynamic steering improves accuracy across three model families spanning roughly a five-fold parameter range, from the 1.5B distilled DeepSeek model to Qwen3-8B, and across both dense-transformer and mixture-of-experts architectures. This architecture-agnostic pattern directly supports the claim that the controller is not merely exploiting an idiosyncrasy of Qwen3-8B. The main exception is DeepSeek-R1-Distill-Qwen-1.5B on AIME2025, where accuracy drops by 3.33 percentage points and token usage increases by 15.9%. We interpret this as a capacity-floor and over-reflection effect: when a small model faces problems beyond its knowledge boundary, errors arise from missing capability rather than pathological overthinking. The controller encourages reflective System-2 self-checks, but because the model lacks the knowledge needed to resolve the error, reflection can become a longer unsuccessful loop rather than a successful correction.

4.5. Robustness on Simple Logic Tasks

We evaluate BBH Boolean Expressions as a simple symbolic-logic control task to analyze the behavior of the steering mechanism on simple logic versus highly complex reasoning. Table 7 reports both the full controller and component removals on 250 examples.

Table 7. Task-complexity boundary control on BBH Boolean Expressions with Qwen3-8B. Best values are shown in bold.

Configuration	Accuracy	Avg Tokens
Dynamic Closed-Loop Steering (Full)	**100.0**	1051
w/o Anti-Collapse	**100.0**	**1020**
w/o Manifold	**100.0**	1058
w/o adapted ThinkBrake-style latch	**100.0**	1049
w/o EMA	**100.0**	1122
Baseline	94.40	922

The BBH Boolean results show transparent-observer behavior. All controlled variants correct the 14 baseline errors and reach 100% accuracy, indicating that the intervention does not degrade a simple task where the baseline is already strong. The modest token overhead is concentrated in the cases where the baseline struggles; on already-correct low-entropy trajectories, the controller largely remains passive rather than forcing unnecessary latent-state changes.

4.6. Non-Mathematical Logic: Code Tracing

To test whether the same feedback mechanism transfers beyond mathematical proof-style reasoning, we evaluate CRUXEval code-output prediction. This task requires precise symbolic tracing of Python (version 3.11.14) execution rather than competition mathematics. The results are summarized in Table 8.

Table 8. CRUXEval code-tracing results on DeepSeek-1.5B and Gemma-4-E2B. Best values are shown in bold.

Model	Configuration	Accuracy	Avg Tokens
DeepSeek-1.5B	Baseline	71.13	1290
	Dynamic Closed-Loop Steering ($\alpha_{\max} = 0.3$)	**71.38**	**1183**
	Dynamic Closed-Loop Steering ($\alpha_{\max} = 0.5$)	69.13	5440
Gemma-4-E2B	Baseline	80.00	1120
	Dynamic Closed-Loop Steering ($\alpha_{\max} = 0.3$)	81.00	**1118**
	Dynamic Closed-Loop Steering ($\alpha_{\max} = 0.5$)	**81.50**	1119

These results show domain agnosticism in a non-mathematical logic setting. For DeepSeek-1.5B, the milder $\alpha_{\max} = 0.3$ setting is Pareto-favorable, improving accuracy by 0.25 percentage points while reducing average tokens by 8.29%. For Gemma-4-E2B, the stronger $\alpha_{\max} = 0.5$ setting improves accuracy by 1.50 percentage points with essentially unchanged token footprint. Thus, the controller's suppression of unproductive reasoning loops transfers to code tracing, but the preferred intervention ceiling remains task- and capacity-dependent.

4.7. Computational Overhead Analysis

Finally, we benchmark the runtime cost of the closed-loop controller itself. As shown in Table 9, on identical RTX 4090 hardware and a 32k-token generation benchmark, standard generation reaches 24.07 tokens/s, while Dynamic Closed-Loop Steering reaches 22.90 tokens/s. Peak VRAM remains unchanged at 19.68 GB in both settings. The resulting latency overhead is approximately 4.9%, and the memory overhead is effectively zero because the method stores only a steering vector, scalar controller state, and lightweight entropy statistics.

Table 9. Runtime overhead of Dynamic Closed-Loop Steering on a 32k-token generation benchmark. Best values are shown in bold.

Setting	Throughput (Tokens/s)	Peak VRAM (GB)
Standard generation	**24.07**	19.68
Dynamic Closed-Loop Steering	22.90	19.68

Because the controller often reduces the total number of generated tokens on overthinking-heavy benchmarks, the small per-token overhead does not necessarily imply slower end-to-end inference. In settings such as DeepSeek-1.5B on AIME2024 and CRUXEval, the token reduction can offset or exceed the controller cost.

4.8. Microscopic Level Analysis of Long AIME Reasoning

To explain *why* the answer-level gains in Section 4.2 emerge, we next shift from final answers to full intermediate trajectories on AIME. We therefore introduce an LLM-as-Judge protocol over complete reasoning traces. We use DeepSeek v4 Flash as an external evaluator and provide the gold final answer for each problem as an anchor signal. Because public benchmark reporting for the judge itself is incomplete, we do not treat the judge as a source of ground-truth solving ability; instead, we use it as a structured annotator for step-level state transitions. Full prompt templates, XML formatting details, and output schemas are deferred to Appendix A.

Each generated trace is segmented by double-newline boundaries(\n\n), following recent long-chain-of-thought work that analyzes reasoning at paragraph- or chunk-level granularity [42,43], and each segment is wrapped as an XML block with an explicit block index. This design allows the judge to determine whether a local block is intrinsically correct, intrinsically incorrect, or corrects an earlier mistake. Within each block, the judge also counts correct and incorrect sub-steps. We then classify the final state of the block into one of four mutually exclusive transition types: *correct reasoning* (C), *error reasoning* (E), *correct-to-error* (C→E), and *error-to-correct* (E→C). If a block contains a temporary mistake but ends in a repaired state, we mark it as E→C rather than E.

From these annotations, we compute three core transition metrics. The *True Recovery Rate* measures conditional self-correction ability, i.e., how often the model returns to a correct path after entering an erroneous context. The *True Degradation Rate* measures the opposite failure mode, namely how often a previously correct trajectory gets derailed. Their ratio defines a *Transition Advantage Ratio*, which summarizes whether long-horizon dynamics are biased toward recovery or toward collapse. We additionally analyze four complementary views: temporal stage sensitivity (early/mid/late blocks), error-persistence length, sub-step error severity, and problem-level heterogeneity across easy/medium/hard AIME items. Importantly, because individual reasoning blocks extracted from the same reasoning trajectory are clustered and not independent observations, these block-level transition rates are presented strictly as pooled descriptive statistics to characterize the micro-dynamics of the search process. We do not treat individual blocks as independent samples for inference testing, and all hypothesis tests are performed strictly at the independent problem/trajectory level in Sections 4.1 and 4.9.

Table 10 reveals two different regimes. On AIME2024, Dynamic Closed-Loop Steering is the strongest long-reasoning controller: it improves step accuracy to 87.25% and yields an advantage ratio of 9.215, far above all alternatives. This means the controller is not merely avoiding damage; it actively pushes trajectories back to correct reasoning after local divergence. On AIME2025, the picture is more competitive. Dynamic Closed-Loop Steering still improves over Qwen3 baseline and CAA, and remains better than constant spherical-steering on conditional recovery, but SEAL obtains the strongest transition ratio overall because it combines the highest recovery rate with a slightly lower degradation rate.

Figure 2 gives the same result in a compact geometric view. The upper-right quadrant corresponds to the ideal regime of low degradation and high recovery. Dynamic Closed-Loop Steering remains in this regime across both years, indicating that its gains are not tied to one particular benchmark distribution. By contrast, Bailian/Qwen3 baseline and CAA stay noticeably farther away from the robust-correction frontier, especially on AIME2024, which is exactly the lower-stability setting where long-horizon feedback matters most.

Table 10. Core LLM-as-Judge transition metrics on AIME2024 and AIME2025. Recovery Rate $= \frac{\text{E2C}}{\text{all error contexts}}$, Degradation Rate $= \frac{\text{C2E}}{\text{all correct contexts}}$, and Advantage Ratio $= \frac{\text{Recovery Rate}}{\text{Degradation Rate}}$. Best values are shown in bold and second-best values are underlined within each benchmark block. ↑ indicates higher is better, and ↓ indicates lower is better.

Dataset	Method	Step Acc. (↑)	Recovery Rate (↑)	Degradation Rate (↓)	Adv. Ratio (↑)
AIME2024	Qwen3 baseline	64.95	2.43	1.95	1.247
	CAA	65.80	2.55	1.72	1.483
	Spherical-Steering	<u>79.95</u>	<u>5.26</u>	<u>1.28</u>	4.108
	SEAL	79.94	6.02	**1.17**	<u>5.135</u>
	Dynamic Closed-Loop Steering	**87.25**	**13.57**	1.47	**9.215**
AIME2025	Qwen3 baseline	91.23	16.71	1.20	13.883
	CAA	90.52	10.52	1.30	8.071
	Spherical-Steering	**93.89**	14.94	**0.90**	16.689
	SEAL	<u>93.34</u>	**20.98**	<u>0.98</u>	**21.481**
	Dynamic Closed-Loop Steering	92.58	<u>19.34</u>	1.13	<u>17.093</u>

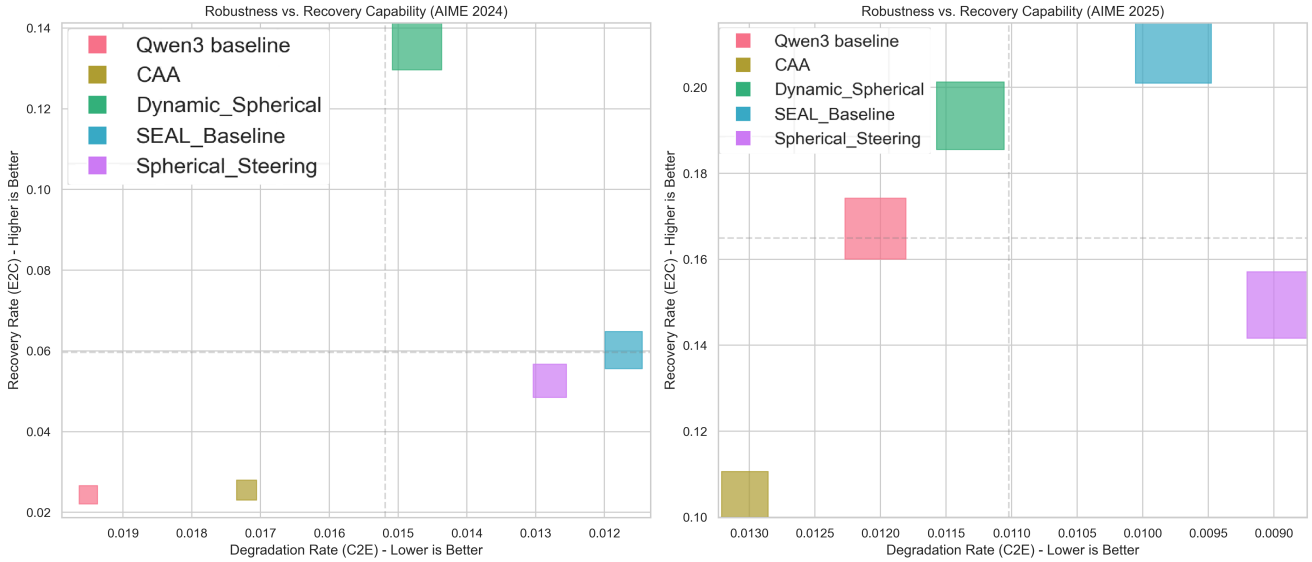


Figure 2. Robustness–correction scatter plot for LLM-as-Judge metrics. The x-axis is Degradation Rate (C-to-E, lower is better; plotted in reverse), the y-axis is Recovery Rate (E-to-C, higher is better), and bubble size reflects Transition Advantage Ratio. The desirable upper-right region corresponds to simultaneously low derailment and high self-correction. Dynamic Closed-Loop Steering occupies this region in both AIME2024 and AIME2025, whereas weaker baselines cluster closer to the low-recovery/high-degradation region.

The temporal view (as detailed in Table 11) sharpens this picture. On AIME2024, Dynamic Closed-Loop Steering performs best among the compared methods in step accuracy at every stage and is especially strong in conditional recovery, reaching 17.39% in the early stage and 14.60% in the mid stage, far above the best non-dynamic alternatives. By contrast, the strongest degradation rates in AIME2024 are achieved by CAA in the early stage and by SEAL in the mid and late stages, which indicates that some baselines are locally conservative but do not recover from errors nearly as well once reasoning drifts. On AIME2025, the field is more competitive: CAA and Spherical-Steering lead early-stage accuracy, Spherical-Steering and SEAL are strongest in mid/late-stage stability, and Spherical-Steering attains the highest late-stage accuracy. Dynamic Closed-Loop Steering nevertheless remains one of the strongest methods in the early and mid portions on recovery, which is consistent with a controller that is most useful during active search rather than only at the final formatting stage. Across both years, late-stage recovery is lower for most methods, consistent with

the intuition that once the trajectory enters final calculation or answer-formatting, earlier logical damage is harder to undo.

Table 11. Temporal/depth analysis on AIME. Early blocks use the first $\min(10, 0.15N)$ segments, late blocks use the last $\min(15, 0.20N)$ segments, and the remainder are mid-stage. Best values are shown in bold and second-best values are underlined within each dataset–stage slice. ↑ indicates higher is better, and ↓ indicates lower is better.

Dataset	Method	Stage	Blocks	Step Acc. (↑)	Recovery Rate (↑)	Degradation Rate (↓)
AIME2024	Qwen3 baseline	Early	300	19.28	0.00	1.10
	CAA	Early	296	69.22	2.20	**0.98**
	Spherical-Steering	Early	300	73.99	2.74	2.64
	SEAL	Early	300	68.91	2.15	1.45
	Dynamic Closed-Loop Steering	Early	300	**83.91**	**17.39**	1.57
	Qwen3 baseline	Mid	7769	67.87	2.77	1.92
	CAA	Mid	8037	66.28	2.77	1.72
	Spherical-Steering	Mid	9000	80.70	5.85	1.17
	SEAL	Mid	9534	80.98	6.73	**1.14**
	Dynamic Closed-Loop Steering	Mid	7771	**87.91**	**14.60**	1.48
	Qwen3 baseline	Late	450	57.71	0.40	3.02
	CAA	Late	444	48.10	0.00	2.53
	Spherical-Steering	Late	450	61.02	0.00	2.97
	SEAL	Late	450	57.18	1.02	1.98
	Dynamic Closed-Loop Steering	Late	450	**76.27**	**1.60**	**1.23**
AIME2025	Qwen3 baseline	Early	295	94.77	54.55	0.70
	CAA	Early	300	**97.90**	30.77	1.05
	Spherical-Steering	Early	300	97.58	**71.43**	**0.00**
	SEAL	Early	300	92.41	17.65	**0.00**
	Dynamic Closed-Loop Steering	Early	299	96.72	30.00	1.04
	Qwen3 baseline	Mid	10,422	91.29	17.02	1.19
	CAA	Mid	9654	90.48	10.68	1.30
	Spherical-Steering	Mid	11,808	**93.79**	15.00	**0.93**
	SEAL	Mid	9817	93.45	**21.85**	1.03
	Dynamic Closed-Loop Steering	Mid	10,328	92.56	19.85	1.12
	Qwen3 baseline	Late	442	88.37	5.88	1.87
	CAA	Late	450	83.10	3.12	1.55
	Spherical-Steering	Late	450	**93.96**	4.55	**0.49**
	SEAL	Late	450	90.77	**6.82**	**0.49**
	Dynamic Closed-Loop Steering	Late	447	89.23	4.88	1.48

Figure 3 visualizes the most striking temporal regularity in the dataset: late-stage correction is a bottleneck for everyone. Even strong models that recover well in the early or middle stages show a marked drop once the trajectory has committed to final computation or answer formatting. This pattern helps explain why an intervention that acts earlier in the reasoning chain is especially valuable.

Table 12 shows the qualitative difference between the two years even more clearly. On AIME2024, Dynamic Closed-Loop Steering dramatically shortens error persistence: the mean erroneous run drops to 1.57 blocks, the maximum run length falls to 14, and 71.44% of error episodes are corrected immediately. This is precisely the desired closed-loop behavior of rapid detection, bounded excursion, and prompt recovery. On AIME2025, absolute differences are smaller because all strong methods already operate in a relatively stable regime; nonetheless, Dynamic Closed-Loop Steering still achieves the best hard-problem advantage ratio, indicating that its main benefit is concentrated on difficult items where long-horizon correction matters most.

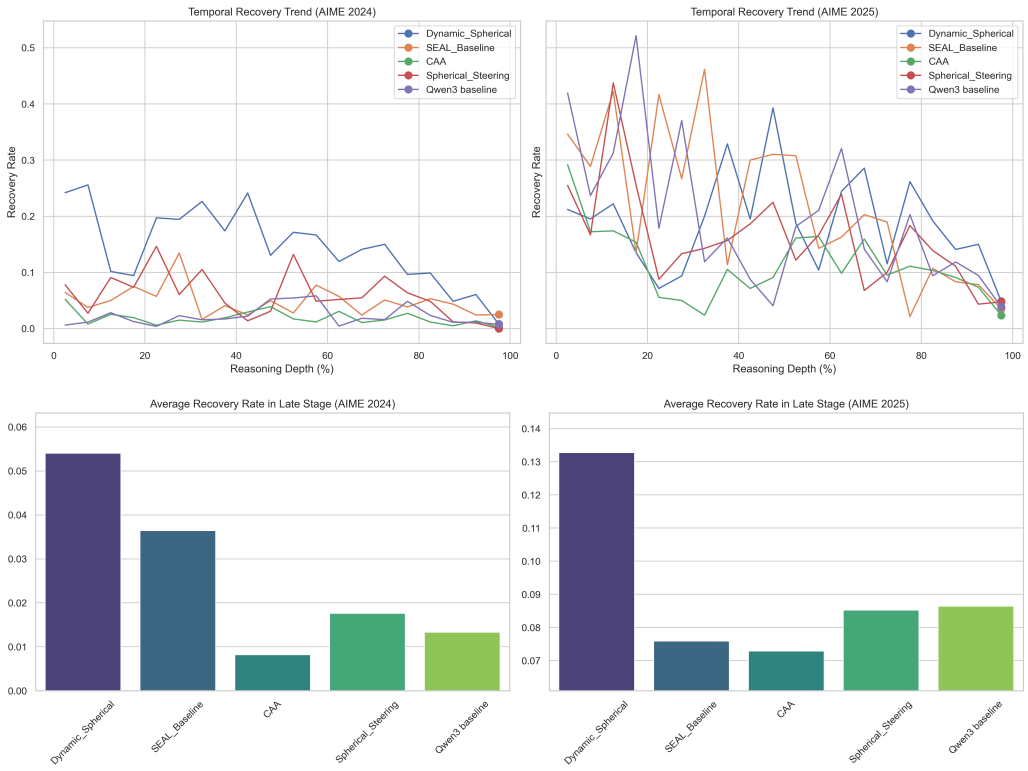


Figure 3. Temporal recovery analysis. Left: recovery-rate trends across Early-to-Mid-to-Late stages for AIME2024 and AIME2025. Right: average late-stage recovery across models. A clear conclusion emerges in both years: once reasoning enters the late stage, self-correction collapses sharply for nearly all methods, implying that missed errors become much harder to repair after the main derivation phase.

Figure 4 makes this micro-mechanism more concrete through Kaplan–Meier survival analysis over error persistence. The x-axis counts how many erroneous blocks have elapsed since the onset of an error episode, and the y-axis reports the probability that the trajectory is still trapped in an erroneous state. A steeper downward curve therefore means faster escape from reasoning dead-ends. Dynamic Closed-Loop Steering shows the most abrupt decline, which indicates the highest hazard rate for exiting error states. Put differently, competing methods exhibit visibly fatter tails, meaning that once they fall into a bad local trajectory, they are more likely to stay there for many additional blocks.

Table 12. Trajectory persistence, sub-step severity, and problem-level heterogeneity on AIME. "Hard-tier Adv." denotes the transition advantage ratio on the hardest 10 problems under the same SEAL-referenced tier partition used in Table 13. Best values are shown in bold and second-best values are underlined within each benchmark block. ↑ indicates higher is better, and ↓ indicates lower is better.

Dataset	Method	Mean Error Run (↓)	Max Error Run (↓)	Immediate Recovery (↑)	Mean Severity (↓)	Hard-Tier Adv. (↑)
AIME2024	Qwen3 baseline	12.05	161	25.77	76.92	0.923
	CAA	9.54	623	<u>44.92</u>	90.41	0.728
	Spherical-Steering	<u>5.63</u>	<u>50</u>	43.98	77.44	0.537
	SEAL	5.96	180	39.55	<u>76.44</u>	<u>0.879</u>
	Dynamic Closed-Loop Steering	**1.57**	**14**	**71.44**	**68.18**	**9.668**
AIME2025	Qwen3 baseline	2.51	<u>33</u>	61.54	71.73	5.656
	CAA	3.06	45	59.82	**61.06**	4.802
	Spherical-Steering	<u>2.43</u>	**30**	**64.88**	<u>62.57</u>	9.513
	SEAL	**2.20**	39	<u>63.19</u>	72.41	<u>10.151</u>
	Dynamic Closed-Loop Steering	2.37	40	58.40	73.43	**13.582**

Table 13. Problem-level heterogeneity on AIME under easy/medium/hard tiering. Each tier contains 10 problems. We report the raw tier-wise Recovery Rate, Degradation Rate, and Transition Advantage Ratio for all methods. Difficulty tiers are defined using SEAL as the reference model because its answer-level accuracy lies near the middle of the compared methods overall, avoiding a partition induced by either an unusually weak or unusually strong anchor. Best values are shown in bold and second-best values are underlined within each dataset–tier slice. ↑ indicates higher is better, and ↓ indicates lower is better.

Dataset	Tier	Method	Recovery Rate (↑)	Degradation Rate (↓)	Adv. Ratio (↑)
AIME2024	Easy	Qwen3 baseline	4.54	1.08	4.218
		CAA	3.09	0.98	3.154
		Spherical-Steering	4.38	**0.51**	8.516
		SEAL	14.13	0.67	**20.956**
		Dynamic Closed-Loop Steering	**17.55**	1.13	15.513
	Medium	Qwen3 baseline	2.49	3.30	0.754
		CAA	2.50	1.78	1.405
		Spherical-Steering	9.45	1.32	7.151
		SEAL	10.68	**1.25**	**8.515**
		Dynamic Closed-Loop Steering	**13.98**	2.22	6.308
	Hard	Qwen3 baseline	1.05	1.14	0.923
		CAA	2.29	3.14	0.728
		Spherical-Steering	1.73	3.23	0.537
		SEAL	2.01	2.29	0.879
		Dynamic Closed-Loop Steering	**10.37**	**1.07**	**9.668**
AIME2025	Easy	Qwen3 baseline	16.05	0.54	29.451
		CAA	9.52	**0.14**	66.159
		Spherical-Steering	18.81	0.53	35.341
		SEAL	**31.25**	0.16	**196.875**
		Dynamic Closed-Loop Steering	25.35	0.38	67.014
	Medium	Qwen3 baseline	**34.19**	0.90	37.849
		CAA	12.73	1.37	9.283
		Spherical-Steering	20.75	**0.60**	34.652
		SEAL	31.37	0.72	**43.608**
		Dynamic Closed-Loop Steering	17.34	1.22	14.178
	Hard	Qwen3 baseline	11.62	2.05	5.656
		CAA	9.27	1.93	4.802
		Spherical-Steering	13.27	**1.40**	9.513
		SEAL	18.36	1.81	10.151
		Dynamic Closed-Loop Steering	**20.43**	1.50	**13.582**

Table 13 makes the tier structure explicit. Importantly, Dynamic Closed-Loop Steering does not only win on the hard split. Its tier-wise results are competitive across easy, medium, and hard problems in both years, which suggests that the controller is broadly useful rather than narrowly specialized. That said, the most striking gains do appear on the hard tier, and this is the regime we emphasize most. On AIME2024-Hard, Dynamic Closed-Loop Steering raises Recovery Rate to 10.37%, while simultaneously reducing Degradation Rate to 1.07%; the resulting Advantage Ratio of 9.668 is an order of magnitude larger than every baseline. On AIME2025-Hard, it again achieves the strongest recovery (20.43%) and the highest advantage ratio (13.582), outperforming the otherwise competitive SEAL and Spherical-Steering baselines. These hard-tier results directly support the claim that closed-loop steering is most valuable when long reasoning requires repeated internal course correction rather than a single clean derivation.

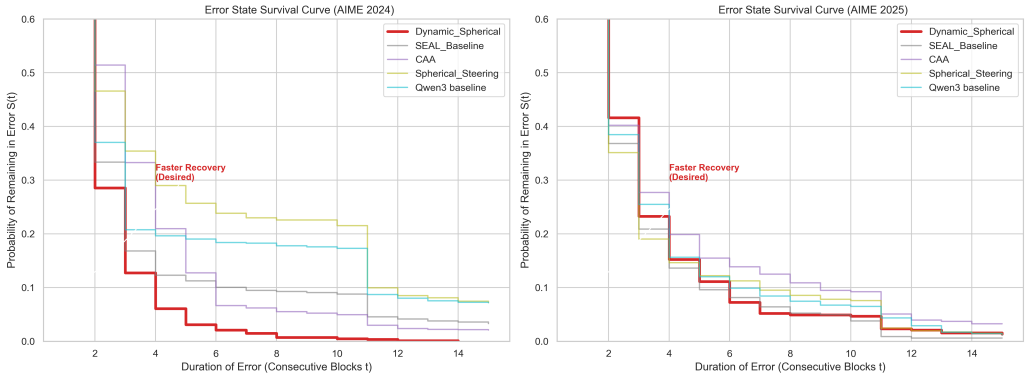


Figure 4. Kaplan–Meier survival curves for error-state persistence on AIME. The x-axis is the number of elapsed erroneous blocks t, and the y-axis is the probability that the trajectory remains in an error state. The steeper decline of Dynamic Closed-Loop Steering indicates much faster escape from reasoning dead-ends, while the fatter tails of baselines reflect prolonged error persistence.

Figure 5 provides the macro-level counterpart to this survival view. As difficulty increases from Easy to Hard, the advantage ratios of the baselines deteriorate markedly, especially on the hardest tier where repeated recovery is indispensable. Dynamic Closed-Loop Steering, by contrast, preserves a clear margin and remains the dominant method on hard problems. This is the visual sense in which the controller "pulls away" exactly when the reasoning chain becomes most fragile and bottlenecked by unrepaired intermediate errors.

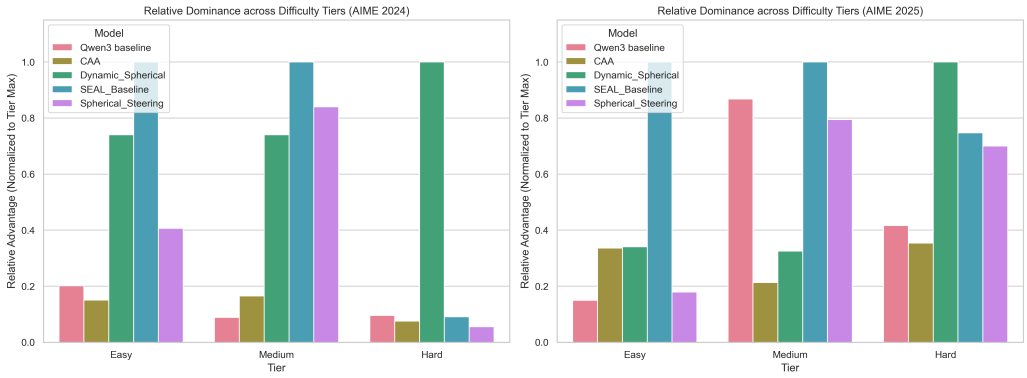


Figure 5. Transition Advantage Ratio across difficulty tiers on AIME. As problems become harder from Easy to Hard, the relative advantage of most baselines shrinks sharply, whereas Dynamic Closed-Loop Steering sustains a much stronger advantage on the hardest tier. This pattern highlights the controller's superior robustness when long reasoning chains require repeated recovery from local errors.

The choice of SEAL for difficulty partitioning is deliberate. In answer-level accuracy, SEAL sits near the middle of the compared groups overall, making it a more neutral anchor than either a weak baseline or the strongest controller. Using such a middle-strength reference avoids defining "easy" and "hard" in a way that is overly optimistic or overly pessimistic. Under this partition, Dynamic Closed-Loop Steering remains strong across all three tiers, which strengthens the conclusion that its gains are not an artifact of a particular difficulty definition.

The radar profile in Figure 6 makes the multi-metric trade-off visually intuitive. The staggered four-row layout prioritizes legibility over strict visual symmetry, with the wider panels used where more horizontal space is needed. Dynamic Closed-Loop Steering does not improve only one dimension; instead, it maintains a broad and relatively balanced profile across correction, stability, and severity. In particular, the recovery axis is where

weaker baselines collapse inward most strongly, reinforcing the view that self-correction is the key differentiator in long reasoning.

Figure 6. Radar-chart capability profile under the LLM-as-Judge evaluation. The panels are arranged in four rows: the first and third rows contain three compact charts, while the second and fourth rows contain two wider charts for readability. The axes summarize answer accuracy, recovery ability, stability (1 − Degradation), convergence advantage, and mildness (1 − Severity). Dynamic Closed-Loop Steering forms the most balanced and expansive profile overall, whereas weaker baselines collapse most visibly along the recovery axis.

The heatmaps in Figure 7 reveal the micro-trajectory signature behind the aggregate statistics. Lower-performing methods produce visually long, contiguous error bands, especially on harder problems, which is exactly the pattern expected from persistent local loops. Dynamic Closed-Loop Steering, by contrast, breaks many of these long failure runs into shorter fragments, consistent with faster escape from error attractors and a higher probability of re-entering correct reasoning.

Figure 8 complements the heatmaps with a direct summary statistic: once an error occurs, how long does it persist? The answer again favors Dynamic Closed-Loop Steering. This matters because a long error run is a concrete operational signature of overthinking: the model is not simply wrong once, but repeatedly elaborating on the wrong latent branch.

Taken together, the LLM-as-Judge results suggest that Dynamic Closed-Loop Steering provides the clearest gains when long reasoning is both fragile and revisable. In the lower-stability AIME2024 regime, it greatly increases the probability of escaping wrong trajectories while keeping degradation low. In the higher-stability AIME2025 regime, the gain shifts from global superiority to a harder-problem advantage: the controller remains especially useful on difficult items that require multiple internal course corrections rather than a single clean derivation.

4.9. Macro-Efficiency and Token-Budget Behavior

With the trajectory-level recovery mechanism in place, we next return to a macro-level question: what does this buy in actual compute usage? A primary symptom of overthinking is budget collapse: trajectories consume the full context window yet fail to converge. In Figure 9, Baseline exhibits a sharp probability mass at the 4000-token boundary, forming a clear truncation wall. This indicates repeated cyclic rationalization with little effective progress.

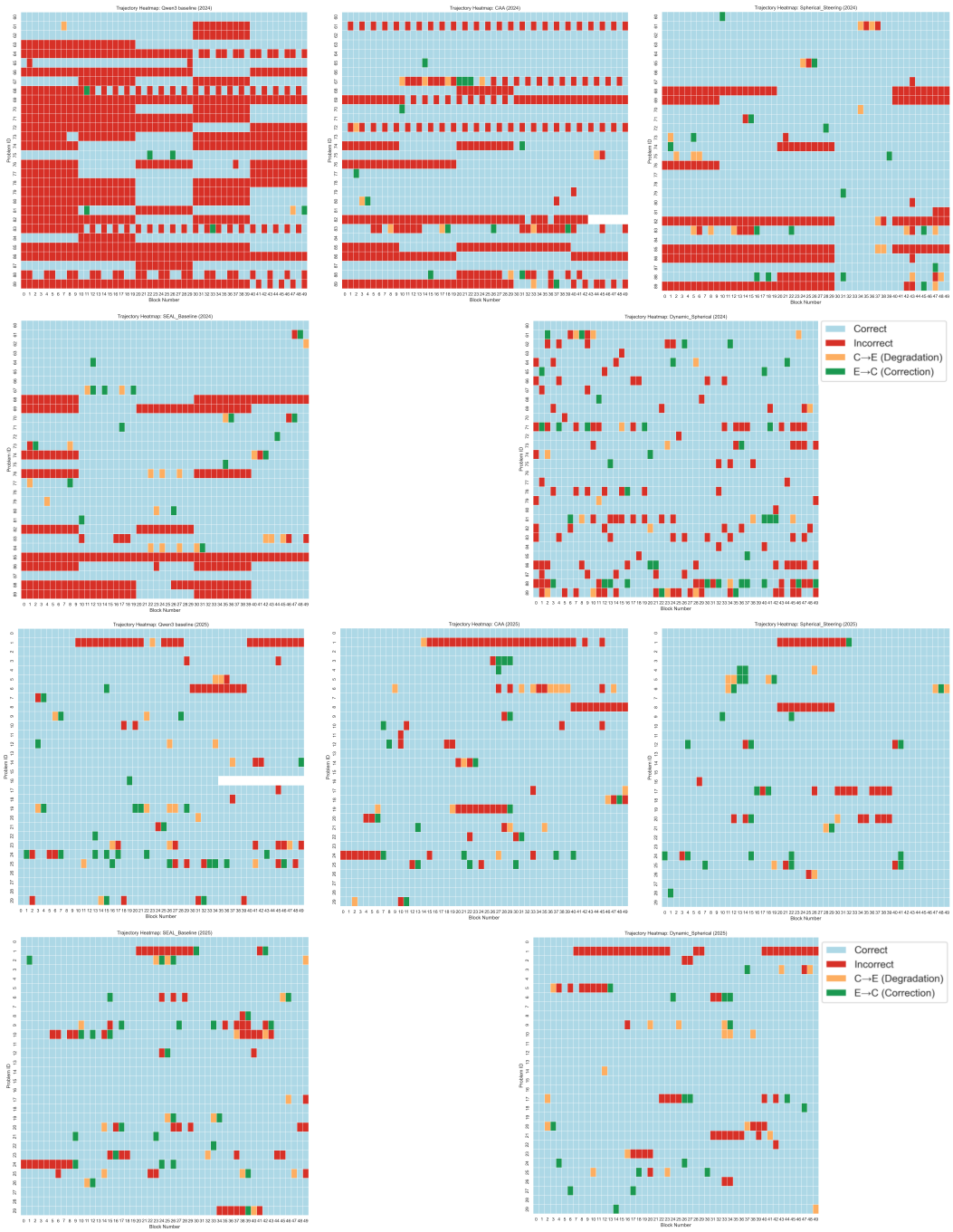


Figure 7. Trajectory heatmaps for all AIME groups, arranged in two rows: AIME2024 on **top** and AIME2025 on **bottom**. In each row, the five panels are ordered from **left** to **right** as Qwen3 baseline, CAA, Spherical-Steering, SEAL, and Dynamic Closed-Loop Steering. The x-axis is the reasoning block index, the y-axis is the problem ID, and color encodes local reasoning state. High-performing models exhibit sparse and quickly corrected error regions, whereas weaker baselines display longer contiguous failure bands that reflect persistent entrapment in erroneous trajectories. Dynamic Closed-Loop Steering most consistently converts long red failure bars into shorter, more recoverable fragments.

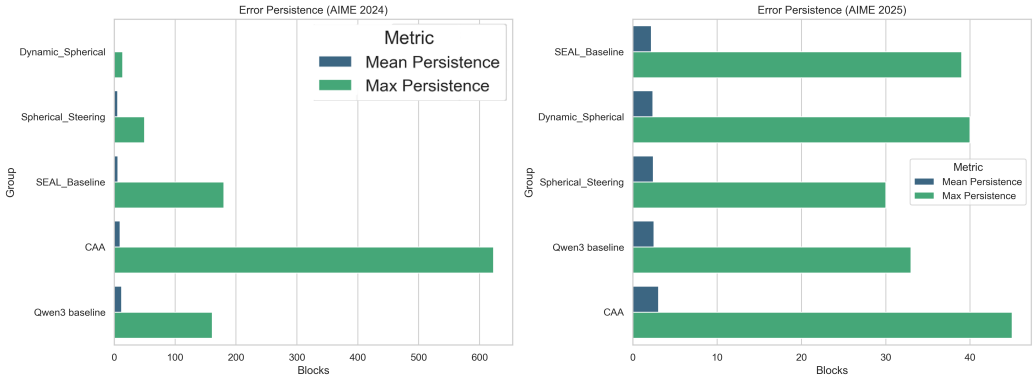
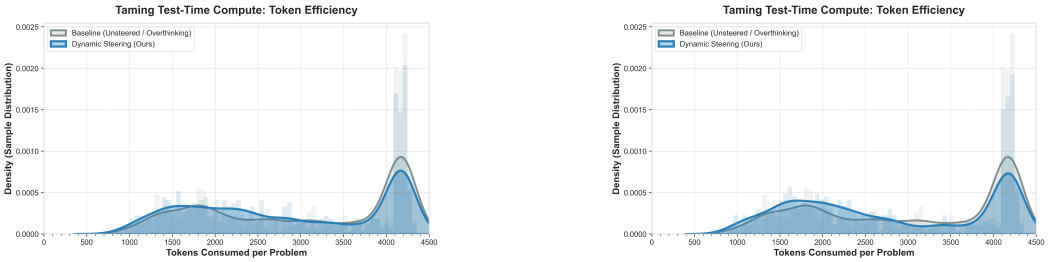


Figure 8. Error-persistence comparison across methods. Horizontal bars summarize the average and maximum length of consecutive error runs, where shorter is better. Dynamic Closed-Loop Steering achieves the shortest sustained error runs overall, especially on AIME2024, confirming that its main benefit is not merely detecting mistakes but preventing them from becoming entrenched.



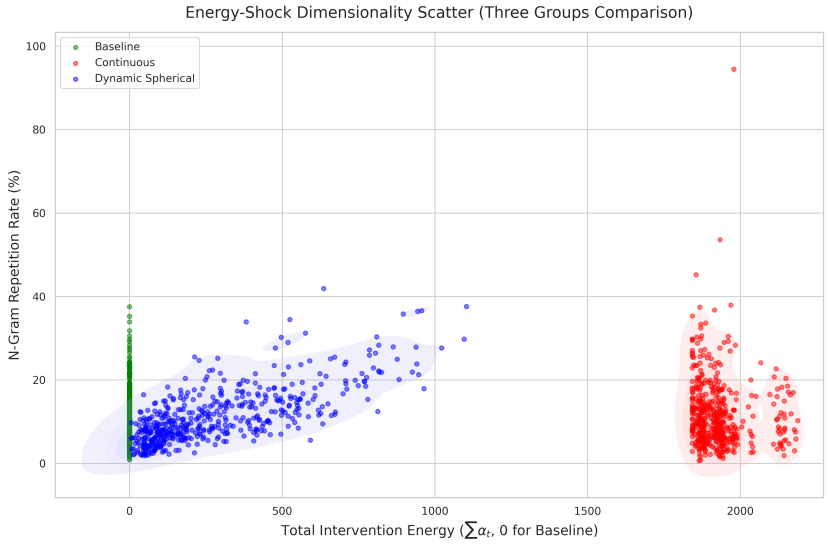
$\alpha = 0.3$ $\alpha = 0.45$

Figure 9. Token efficiency distributions. Dynamic closed-loop steering suppresses the 4k truncation-wall peak and reallocates trajectories to productive completion lengths.

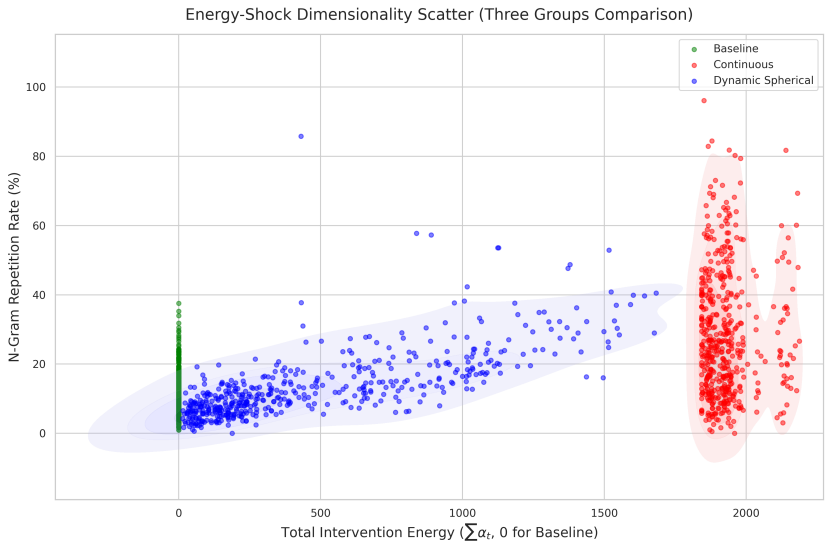
Dynamic steering removes this bottleneck across multiple intervention ceilings. At a mild ceiling ($\alpha_{\max} = 0.15$), Dynamic Closed-Loop Steering already improves accuracy from 67.2% to 70.0% while reducing average token usage from 3045 to 2927. At $\alpha_{\max} = 0.3$, it reaches the best overall accuracy of 76.4% (382/500) with 2856 average tokens, outperforming both Baseline and constant steering. At $\alpha_{\max} = 0.45$, the dynamic controller still preserves a strong 74.0% accuracy at 2785 tokens, whereas continuous forcing becomes unstable. Figure 9 therefore visualizes not only a leftward shift in compute consumption, but a genuine Pareto improvement in the efficiency–accuracy frontier.

4.10. Resolving the Energy–Shock Paradox and Latent Stability

The efficiency gains above would be much less interesting if they were purchased by destabilizing the latent dynamics. We therefore next test whether stronger intervention inevitably damages latent stability. Figure 10 shows that continuous steering accumulates excessive intervention energy and drifts toward the high-repetition regime (surging to 28.17% at $\alpha = 0.45$), consistent with state-shock behavior. The same run also suffers a severe PPL increase to 2.2306 and an accuracy collapse to 41.2%, indicating that stronger open-loop forcing does not buy useful reasoning depth but instead tears the latent state away from its pretrained manifold.



$\alpha = 0.3$



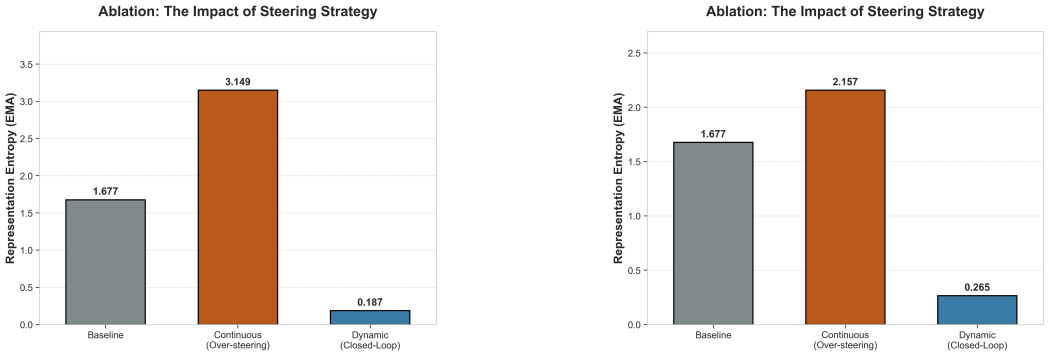
$\alpha = 0.45$

Figure 10. Energy–shock analysis. Continuous steering enters a high-energy/high-repetition region, while Dynamic steering remains in a stable low-shock regime.

Dynamic steering behaves differently. Even at $\alpha = 0.45$, the closed-loop controller keeps repetition at 13.87% and preserves 74.0% accuracy, which means that stronger geometric momentum can remain safe as long as it is conditionally gated and promptly withdrawn. This interpretation is reinforced by entropy ablation in Figure 11: continuous injection inflates internal entropy (peaking around 2.157 in our representation-level statistics), whereas Dynamic steering suppresses it sharply (down to 0.265), indicating a stabilized reasoning manifold rather than forced local perturbation.

To isolate the source of this stability, we expand the ablation from the original 48-problem pilot to the 116-problem MATH500 difficulty-3/4 hard tier. Table 14 reports Qwen3-8B under weak and strong intervention ceilings, while Table 15 reports DeepSeek-1.5B at $\alpha_{\max} = 0.5$. The Qwen results show that the full method reaches 94.83% accuracy at $\alpha_{\max} = 0.5$, improving over the 87.07% baseline by 7.76 percentage points. The component removals reveal distinct mechanisms: removing EMA collapses high-intensity accuracy to 72.41%, confirming that raw entropy triggers are too erratic; removing the adapted ThinkBrake-style latch drops accuracy to 75.86%, showing that the logit-margin latch

prevents unnecessary intervention during final-answer formatting; and removing PCA drops accuracy to 74.14%, consistent with state shock from injecting orthogonal stylistic noise. Anti-Collapse is net-neutral for Qwen accuracy, but on DeepSeek-1.5B it reduces average tokens from 5805.96 to 5614.61 without changing accuracy, showing its role as a safety watchdog that interrupts repetitive low-entropy loops in smaller, less stable models.



$\alpha = 0.3$ $\alpha = 0.45$

Figure 11. Ablation on representation entropy. Dynamic closed-loop control consistently reduces entropy compared with continuous injection.

Table 14. Qwen3-8B ablation on the MATH500 hard tier (116 problems). Best values are shown in bold.

Ablation Configuration	Accuracy ($\alpha_{max} = 0.3$)	Avg Tokens ($\alpha_{max} = 0.3$)	Accuracy ($\alpha_{max} = 0.5$)	Avg Tokens ($\alpha_{max} = 0.5$)
Dynamic Closed-Loop Steering (Full)	88.79	3747	**94.83**	4004
w/o Anti-Collapse	88.79	**3665**	**94.83**	3917
w/o Manifold (no PCA)	**90.52**	3748	74.14	**2850**
w/o adapted ThinkBrake-style latch	87.93	3692	75.86	2897
w/o EMA	86.21	3740	72.41	2977
Baseline	87.07	3723	87.07	3723

Table 15. DeepSeek-1.5B ablation on the MATH500 hard tier (116 problems) at $\alpha_{max} = 0.5$. Best values are shown in bold.

Ablation Configuration	Accuracy	Avg Tokens
Dynamic Closed-Loop Steering (Full)	**59.48**	5614
w/o Anti-Collapse	**59.48**	5806
w/o Manifold (no PCA)	56.03	5720
w/o adapted ThinkBrake-style latch	56.03	4621
w/o EMA	56.90	5386
Baseline	57.76	**2470**

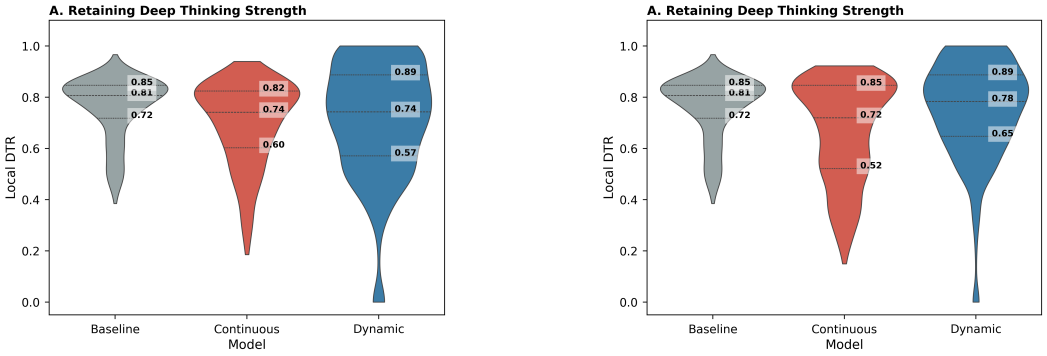
4.11. Preserving System-2 Deliberation: DTR Analysis

A natural counterargument at this stage is that lower entropy and fewer tokens might simply reflect shallower reasoning. Figure 12 argues against that interpretation. Continuous steering shifts the DTR distribution downward (median dropping to 0.67), suggesting persistent cognitive friction under token-wise forcing. Dynamic steering, by contrast, preserves a substantially higher deep-thinking profile: at $\alpha = 0.45$, global DTR remains 0.7528, the median stays around 0.75, and the upper quartile expands to about 0.89. In other words, the controller suppresses pathological loops without erasing the model's capacity for deep multi-layer deliberation.

4.12. Micro-Dynamics of the Closed-Loop Sensor

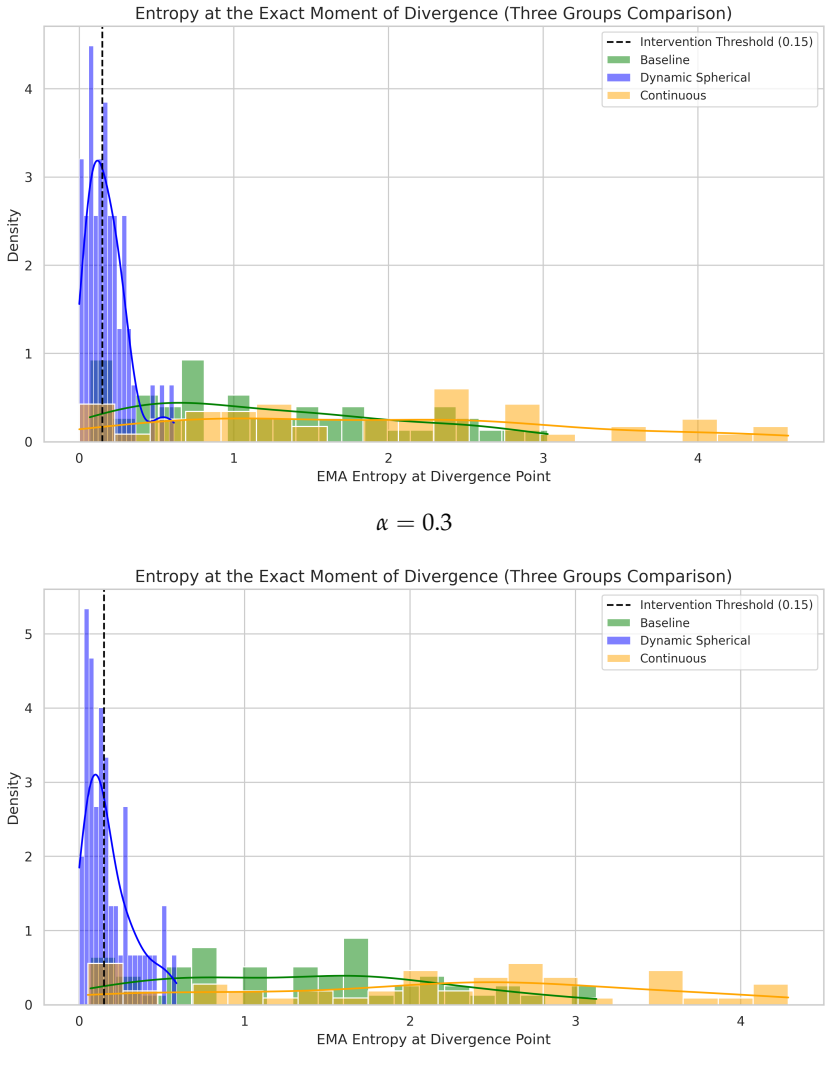
Having argued that the controller preserves deep deliberation, we now examine whether it also intervenes in the right places and at the right time. To verify trigger

precision, we analyze event-level dynamics around divergence onset. Figure 13 shows that entropy peaks are tightly aligned with the predefined trigger band (roughly 0.10–0.25), and controller responses occur immediately after threshold crossing. This supports a causal sense–decide–act loop rather than delayed heuristic correction.



$\alpha = 0.3$ $\alpha = 0.45$

Figure 12. Distribution of local deep-thinking ratio (DTR). Dynamic steering preserves or improves high-depth reasoning compared with continuous injection.



$\alpha = 0.3$

$\alpha = 0.45$

Figure 13. Divergence anatomy. EMA probe detects semantic divergence near threshold intervals and triggers immediate feedback intervention.

Single-sample intervention traces in Figure 14 further show rapid post-trigger entropy reduction together with adaptive steering window lengths. This behavior is consistent with the statistics in Table 3: the dynamic controller activates on every problem, but only sustains intervention over a limited fraction of tokens (42.05% at $\alpha = 0.3$ and 47.95% at $\alpha = 0.45$), and then exits through endogenous stopping rather than fixed-length forcing.

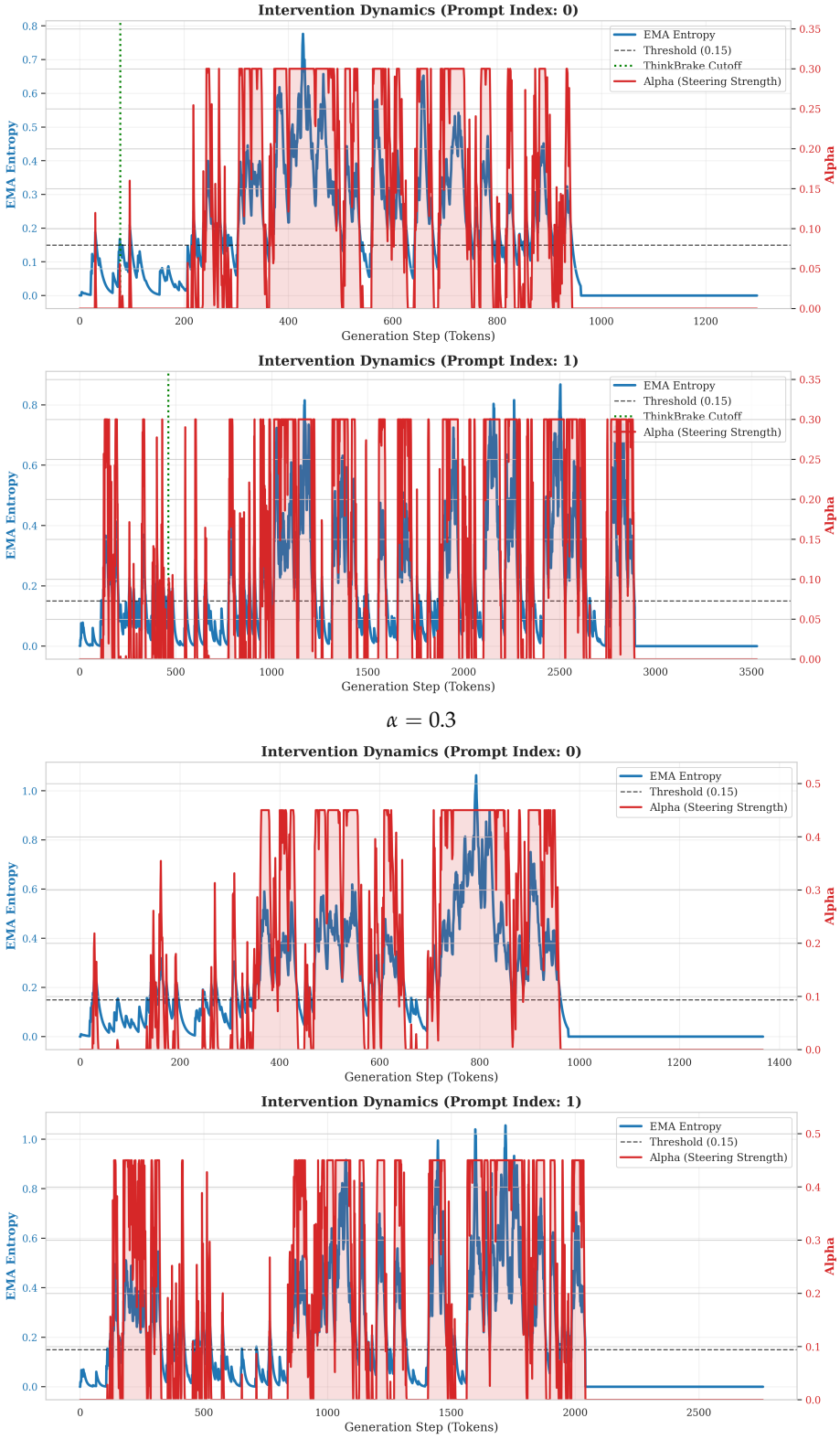


Figure 14. Single-sample intervention trajectories. Steering spikes follow threshold crossing and are terminated adaptively by endogenous convergence signals.

A further diagnostic then clarifies why this closed-loop sensing matters. When we temporarily disable the dynamic non-linear shaping and inspect the intervention-strength distribution induced by a more direct PD-style controller, the resulting window-level α values are not unimodal or approximately Gaussian. Instead, Figure 15 shows a clear bimodal pattern, indicating that the model's entropy excursions naturally separate into two qualitatively different regimes.

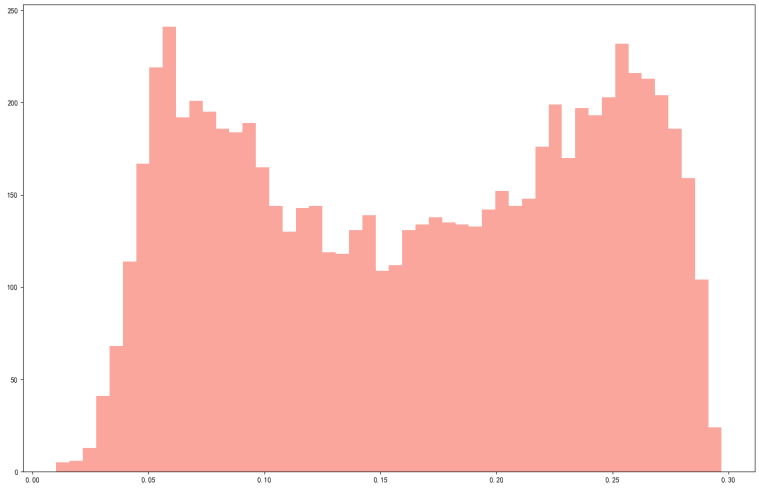


Figure 15. Bimodal distribution of window-level intervention strength at $\alpha_{\max} = 0.3$. The low-α mode corresponds to shallow lexical/syntactic uncertainty, while the high-α mode corresponds to persistent semantic/logical divergence. This directly argues against uniform token-level threshold intervention.

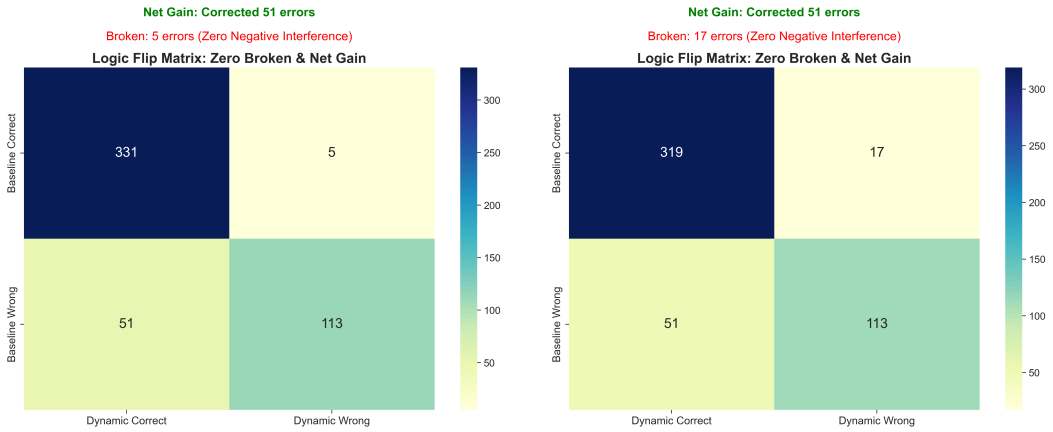
The low-α peak corresponds to shallow and short-lived uncertainty: lexical continuation, subword completion, connective selection, and local syntactic transitions. These events can raise entropy above threshold for a moment, but they are usually resolved within only a few tokens. Because EMA already smooths such fluctuations, our controller keeps them in a weak-intervention regime. By contrast, the high-α peak corresponds to persistent semantic and logical crossroads: branch exploration, constraint checking, self-verification, and multi-step hypothesis competition. These are precisely the locations where stronger steering is useful because uncertainty is not merely stylistic but structurally tied to reasoning failure or divergence.

This observation is important methodologically. In prior token-level intervention schemes, any token whose entropy crosses a threshold is treated as equally intervention-worthy. Our bimodal result shows that this assumption is false: many above-threshold tokens belong to a harmless "how to say it" regime rather than a dangerous "what to think" regime. Uniformly forcing both categories with the same strength would inject noise precisely where the model is only organizing surface semantics, damaging fluency without offering logical benefit. The role of EMA smoothing plus adaptive intensity control is therefore not cosmetic; it is what allows the system to distinguish lexical stutters from genuine reasoning divergence and to avoid over-intervening on semantically healthy tokens.

4.13. Micro-Logic Reliability and Escape Velocity

The final question is whether intervention behaves as a trustworthy correction operator rather than a source of random disturbance. In Figure 16, we present the problem-level logic-flip reliability matrices evaluated on the full MATH500 dataset ($N = 500$), where each problem serves as an independent unit of analysis. Under $\alpha_{\max} = 0.3$, our framework yields a substantial 10:1 rescue-to-break ratio, correcting 51 previously failed problems while only degrading 5 previously successful ones. This corresponds to a 2×2 contingency table with the following cell counts: $a = 331$ (both correct), $b = 5$ (baseline correct, steered incorrect;

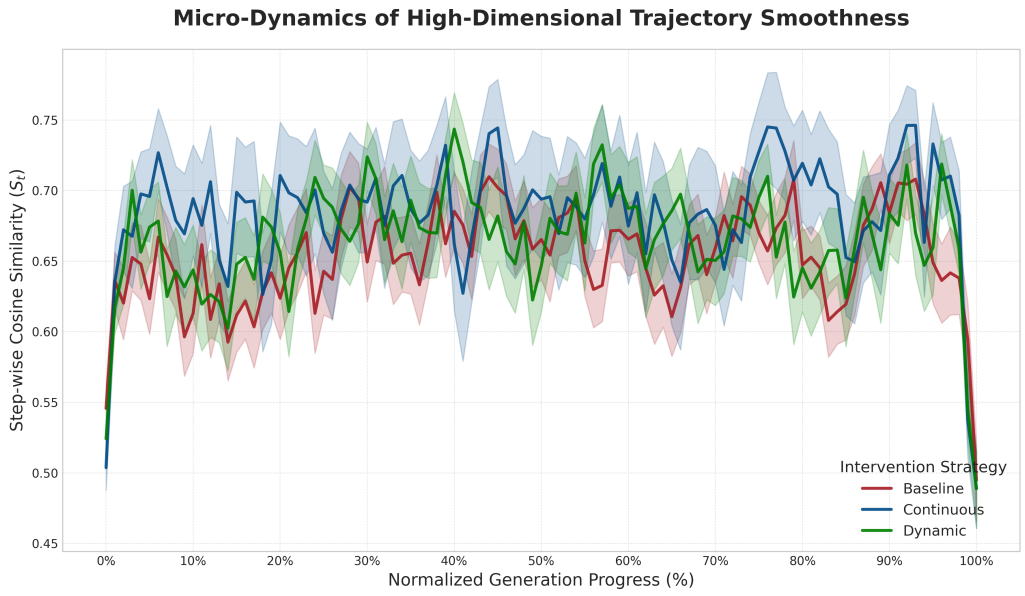
degraded), $c = 51$ (baseline incorrect, steered correct; rescued), and $d = 113$ (both incorrect). Applying McNemar's test with Yates's continuity correction on these discordant pairs yields $\chi^2_{\text{McNemar}} = 36.16$ ($p < 0.0001$; uncorrected $\chi^2_{\text{McNemar\_raw}} = 37.79$, $p < 0.0001$), demonstrating that the problem-level rescue effect is highly statistically significant. At $\alpha = 0.45$, rescue remains high while breakage increases, exposing a measurable correction–friction trade-off that mirrors the macro-level shift from maximum safety to more aggressive correction.



$\alpha = 0.3$ $\alpha = 0.45$

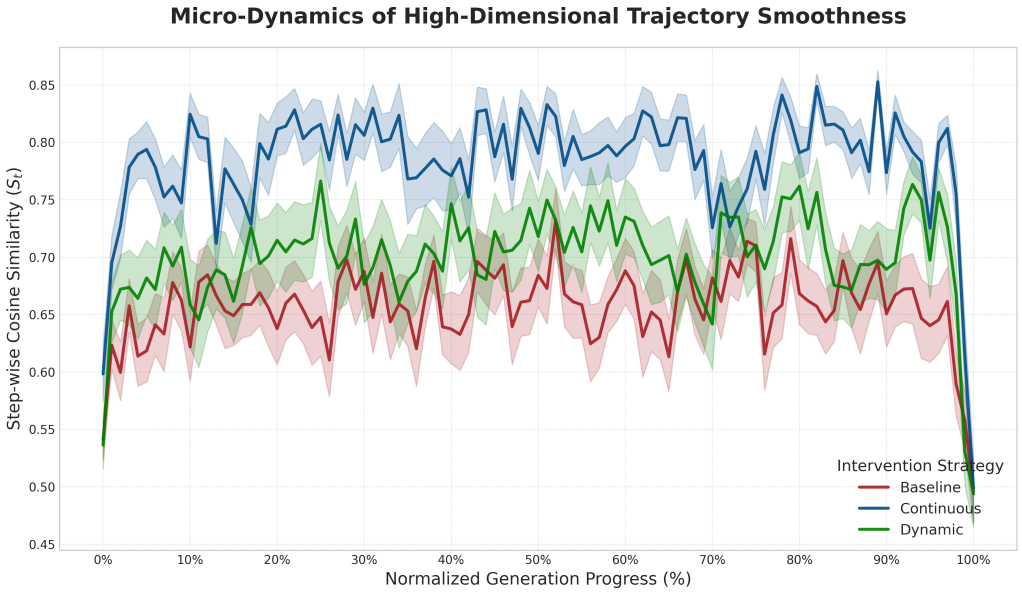
Figure 16. Logic-flip reliability matrices. Dynamic steering provides strong net-positive correction with controllable collateral risk.

Trajectory smoothness analysis in Figure 17 completes this picture by revealing an escape-velocity effect: weak intervention may remain partially entangled with baseline dynamics on hard samples, while stronger intervention provides sufficient geometric momentum to decouple from rigid local minima and improve global success rate. This is precisely why $\alpha = 0.3$ delivers the cleanest micro-level reliability, whereas $\alpha = 0.45$ secures the best macro-level trade-off between answer accuracy and compute efficiency.



$\alpha = 0.3$

Figure 17. *Cont.*



$\alpha = 0.45$

Figure 17. Trajectory smoothness and volatility. Crossing a critical intervention threshold yields effective decoupling from overthinking attractors.

5. Discussion

5.1. Redefining Overthinking: From Epistemic Deficit to Geometric Energy Barrier

Much prior work on test-time scaling assumes that longer deliberation monotonically improves reasoning. Our empirical results challenge this assumption. The observed truncation-wall behavior at the 4000-token boundary indicates that unconstrained System-2 generation can enter a high-entropy deadlock, where additional tokens reflect oscillatory self-justification rather than productive search.

Under this lens, overthinking is not simply a knowledge blind spot. It is a dynamical conflict between strong heuristic priors and downstream logical constraints. The proposed controller acts as a closed-loop latent steering controller: it injects geometric momentum precisely when divergence emerges, breaks latent oscillation, and restores convergence under a fixed compute budget.

5.2. Geometric Shift: From State Shock to Norm-Preserving Rotation

Our results also expose a geometric limitation of linear intervention ($h \leftarrow h + \alpha v$). In long-horizon reasoning, additive forcing perturbs hidden-state magnitude and introduces orthogonal noise components, which can manifest as severe state shock and repetition inflation (>28%).

The PCA+SLERP pipeline addresses this at the mechanism level. PCA projection filters steering signals into a task-relevant logical subspace, while SLERP performs angular correction with exact norm preservation. This replaces amplitude pushing with manifold-consistent rotation, substantially reducing state shock and preserving residual-stream stability during intervention.

5.3. Escape Velocity and Why Closed-Loop Control Matters

Static or uniformly scheduled intervention lacks feedback on instantaneous reasoning state, making it vulnerable to either under-correction or over-steering. By combining an EMA divergence probe with an Adaptive Non-linear PD Regulator, our framework performs error-driven adaptive control in latent space.

Micro-dynamics reveal an escape-velocity regime: below a critical intervention strength, trajectories may remain trapped near overthinking attractors; above it, transiently stronger steering can decouple trajectories and recover reasoning progress. Importantly, this stronger push remains safe because the adapted ThinkBrake-style margin latch provides endogenous withdrawal once convergence evidence appears.

5.4. Evaluation Implications: DTR as Post-Hoc Metric, EMA as Real-Time Proxy

DTR is highly informative for assessing deliberative depth, but evaluating Jensen–Shannon Divergence across all L Transformer layers imposes an $O(L)$ computational overhead, making it prohibitive for real-time control. EMA entropy is therefore utilized as a highly sensitive, $O(1)$ physical proxy, while DTR is retained as a post-hoc reference evaluation metric.

This division of labor is empirically supported: the dynamic method reduces global entropy yet preserves strong local deep-thinking behavior, indicating that entropy suppression need not imply collapse into shallow System-1 generation.

5.5. Broad Impacts for Trustworthy AI

Closed-loop trajectory control is particularly relevant for high-stakes domains in which unbounded hallucination loops are not merely inefficient but operationally risky. In clinical decision support, financial forecasting, and other trust-sensitive settings, a model that repeatedly elaborates on an incorrect latent branch may produce persuasive but unstable rationales. Recent discussions of trustworthy and explainable AI in clinical and financial contexts emphasize that deployed systems require transparency, controllability, and reliable failure boundaries [44,45]. Our framework contributes to this agenda by making macro-level reasoning dynamics more observable: entropy excursions, convergence signals, recovery events, and degradation events can be monitored rather than inferred only from the final answer.

This does not by itself certify a model for high-risk deployment. Rather, the contribution is infrastructural: a sense–decide–act controller provides a mechanism for detecting and interrupting runaway reasoning loops under a fixed compute budget. Such a mechanism may complement external auditing, human oversight, and domain-specific verification systems by exposing when the model is entering unstable deliberation. In larger distributed or multi-agent reasoning systems, related ideas from verifiable coded computation can further support auditable execution and integrity checks for outsourced or parallelized inference pipelines [46].

5.6. Limitations and Future Directions

Several limitations remain. A key limitation of the hyperparameter selection protocol is that we currently lack a quantitative, rigorous method to partition task complexity thresholds a priori, and dynamically scheduling the operating ceiling $\alpha_{\max}$ represents an important direction for future work. First, our results reveal a capacity-floor effect. When a small model operates near the lower bound of its reasoning capability, as in DeepSeek-R1-Distill-Qwen-1.5B on AIME2025 or difficult MATH500 subsets, steering cannot create missing knowledge or reasoning capacity. In such cases, the controller may keep the model in a reflective self-checking mode without enabling it to solve the problem, thereby increasing token usage. Future systems should therefore include capability-aware bypass or early-stopping criteria that detect when additional controlled deliberation is unlikely to help.

Second, the current evaluation focuses on text-only internal representations. Multimodal reasoning introduces heterogeneous latent spaces across visual, textual, and tool-use tokens, and it is not yet clear how a single entropy signal or steering direction should

align these spaces. Extending Dynamic Closed-Loop Steering to multi-modal or tool-augmented reasoning will require new cross-space alignment methods and safety checks for modality-specific state shock.

Third, our interpretability claim is intentionally limited. The framework improves macro-level trajectory observability through measurable signals such as entropy, repetition, recovery, and degradation, but this is not equivalent to proving that the resulting reasoning traces are genuinely more understandable to human end users. Establishing human-facing semantic interpretability requires a statistically powered human-evaluation study, ideally with domain experts and task-specific criteria.

Future work should also investigate training-time integration. A promising route is to encode norm-preserving steering constraints directly into reinforcement-learning objectives, and to extend closed-loop geometric control to multi-agent, MoE, multi-modal, or tool-use settings for simultaneous regulation of logic, factuality, and safety.

6. Conclusions

We introduced Dynamic Closed-Loop Steering, a training-free framework that reformulates inference-time intervention as closed-loop geometric control over latent reasoning trajectories. Rather than treating long reasoning as a fixed decoding process, the framework organizes intervention into a sense–decide–act loop: trajectory sensors monitor divergence and convergence, an adaptive non-linear regulator determines intervention strength, and geometry-preserving actuation redirects hidden states while respecting their norm structure.

The revised evaluation supports three main conclusions. First, the framework improves reasoning behavior across multiple task families, including mathematical reasoning, competition-style AIME problems, code-output reasoning, and simple Boolean logic. Second, the cross-model results indicate that the mechanism is not limited to a single backbone: it transfers across model scales from 1.5B to 8B parameters and across dense, distilled, and MoE architectures. Third, the empirical pattern is best described as Pareto-favorable rather than universally Pareto-optimal: in the large majority of highlighted configurations, accuracy improves while token usage decreases, while the remaining cases trade modest token increases for accuracy gains.

More broadly, the results suggest that overthinking can be productively studied as a trajectory-stability problem. The framework does not claim to make model reasoning fully semantically interpretable to human users; instead, it improves the observability and controllability of macro-level reasoning trajectories. This distinction is important: closed-loop steering can help expose when a model is diverging, recovering, or approaching convergence, while stronger claims about human-facing interpretability require dedicated user studies. We therefore view Dynamic Closed-Loop Steering as a practical step toward adaptive, geometry-preserving, and empirically observable System-2 reasoning in large language models.

Author Contributions: Conceptualization, D.M.; methodology, D.M.; software, D.M.; validation, D.M. and T.X.; formal analysis, D.M.; investigation, D.M. and T.X.; resources, D.M. and T.X. and Y.C.; data curation, D.M.; writing—original draft preparation, D.M.; writing—review and editing, D.M., T.X. and Y.C.; visualization, D.M.; supervision, Y.C.; project administration, Y.C.; funding acquisition, Y.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Beijing Municipal High-Level Overseas Talent Funding,Organization Department of the CPC Beijing Municipal Committee, grant number 004.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The experimental code and data supporting the findings of this study are available at https://github.com/DeYuDaDa/Closed-Loop-Steering (accessed on 14 June 2026). The code used to reproduce the baseline methods is available at https://github.com/DeYuDaDa/CLSS-Baseline (accessed on 14 June 2026).

Acknowledgments: During the preparation of this manuscript, the authors used Gemini 3.1 Pro and DeepSeek v4 Flash only for language polishing, grammar checking, and typesetting assistance. These tools were not used to generate experimental data, conduct statistical analyses, design the study, or make scientific conclusions. All experiments, data processing, figures, and quantitative analyses reported in this manuscript were performed and verified by the authors. The authors have reviewed and edited all AI-assisted language output and take full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial intelligence
LLM	Large language model
CoT	Chain-of-Thought
EMA	Exponential moving average
PD	Proportional–derivative
PCA	Principal component analysis
SLERP	Spherical linear interpolation
ITI	Inference-time intervention
CAA	Contrastive activation addition
SEAL	Steering vectors as energy-based activation likelihoods
DTR	Deep-thinking ratio
PPL	Perplexity
AIME	American Invitational Mathematics Examination
MATH500	A 500-problem subset of the MATH benchmark
EAST	Entropy-scaled adaptive suppression threshold
C2E	Correct-to-error transition
E2C	Error-to-correct transition

Appendix A. LLM-as-Judge Protocol for AIME Long-Reasoning Analysis

This appendix summarizes the evaluation protocol used for the AIME long-reasoning analysis in Section 4.3. Instead of judging only the final answer, we evaluate how a solution evolves across the reasoning process. This allows us to measure not only whether a model eventually succeeds, but also whether it stays on a correct path, drifts into error, or recovers after making a mistake.

Appendix A.1. Evaluation Setup

We first align the AIME problems, reference solutions, gold answers, and model generations from different experimental groups into a unified evaluation set. This ensures that all methods are judged on the same problems and under the same scoring protocol. The purpose of this alignment step is not methodological novelty, but consistency: every compared system is evaluated against the same reference information.

Appendix A.2. Trajectory Segmentation and Tagged Context

For each problem, the model's solution is divided into consecutive reasoning blocks using paragraph-level breaks. These blocks preserve the original order of the derivation and provide a natural unit for local analysis. We then assign each block an explicit index so that the judge can evaluate not only the content of a block itself, but also its position within the broader reasoning trajectory.

This block-based design is important because errors in long reasoning rarely appear as a single isolated sentence. More often, a block either continues a correct line of thought, introduces a wrong step, or repairs an earlier mistake. By evaluating the trace block by block, we can describe how reasoning changes over time rather than collapsing everything into a single final outcome.

Appendix A.3. Judging Inputs and Decision Rule

Each judgment is based on four pieces of information: the original problem, a reference solution, the gold final answer, and the model's reasoning trace. For each block, the judge records how many sub-steps are correct, how many are incorrect, and which type of reasoning transition the block represents.

We use four transition labels throughout the paper:

- **Correct**: the block ends in a correct local reasoning state;
- **Error**: the block ends in an incorrect local reasoning state;
- **C2E**: the block begins on a correct path but exits in error;
- **E2C**: the block begins from an erroneous state but repairs the trajectory and ends correctly.

The key rule is that the label is determined by the *final semantic state* of the block. In other words, if a block briefly goes astray but corrects itself before the end, it is treated as a recovery rather than as a pure error. This rule is central to our analysis because the goal is to capture self-correction dynamics in long reasoning, not simply to count local mistakes.

Illustrative Input Format

For clarity, below we show a simplified example of the type of structured input provided to the judge. In the actual pipeline, the judge receives the problem, a reference solution, the gold final answer, and the indexed reasoning blocks together, so that each block can be evaluated in context rather than in isolation.

```
Problem: Find the value of x if 2x + 3 = 11.
Reference answer: x = 4
Gold final answer: 4

Model trace:
<block_1>
2x + 3 = 11, so 2x = 8.
</block_1>

<block_2>
Thus x = 5.
</block_2>
```

In this toy example, the first block would be judged as locally correct, while the second block would be judged as an error because it ends in an incorrect final state. This example is intentionally simple, but it illustrates the exact logic of the block-level evaluation protocol used in the full AIME setting.

Table A1. Summary of the appendix judging protocol for AIME long-reasoning analysis.

Component	Role in the Pipeline
Problem alignment	Map evaluation indices to official AIME items and attach the problem, solution, and gold answer.
Block segmentation	Split each reasoning trace by double newlines and preserve the original chronological order.
Tagged context	Wrap blocks with explicit indices so the judge can reason about local content and trajectory position jointly.
Per-block outputs	Record total sub-steps, correct sub-steps, incorrect sub-steps, and one transition label for each block.
Transition rule	Use the block's final semantic state to distinguish stable correctness, stable error, degradation, and recovery.

Appendix A.4. From Block Judgments to Trajectory Analysis

After block-level judgments are produced, we aggregate them into the trajectory statistics reported in the main text. These include step accuracy, recovery rate, degradation rate, and other measures of reasoning stability. The advantage of this two-stage design is that all methods are first evaluated under the same local rubric, and only then compared through higher-level summary metrics.

The same protocol is applied to all reasoning systems compared in the paper, including our dynamic controller and the external baselines. This shared evaluation setting allows us to attribute differences in the reported trajectory metrics to differences in reasoning behavior rather than to differences in annotation rules.

Appendix A.5. Scope and Interpretation

This protocol is intended to complement, rather than replace, final-answer evaluation. A model may produce the correct answer after a highly unstable chain of reasoning, while another may follow a mostly correct path but fail near the end. The LLM-as-judge analysis therefore provides an additional view of model behavior: it reveals when reasoning remains stable, when it deteriorates, and when it successfully repairs itself. In this sense, it is especially useful for understanding the behavioral effect of closed-loop steering on long mathematical derivations.

Appendix A.6. Human-Expert Validation of the Trajectory Judge

To validate the reliability of the LLM-as-Judge protocol used for microscopic trajectory analysis, we conducted a rigorous human-expert agreement study. We randomly sampled 200 state-transition instances from the revised AIME evaluation traces, ensuring an equal representation (50 samples each) of the four LLM transition categories: Correct (C), Error (E), Derailment (C→E), and Self-Correction (E→C).

Two human experts independently annotated each sampled transition using three behavioral labels: *productive*, *repetitive*, and *hallucinatory*. To compare these three human labels against the four automated LLM labels, we apply a deterministic many-to-one mapping based on the final semantic state:

- **C and E→C** map to **Productive**, because their logic is sound or an original error is constructively resolved.
- **C→E** maps to **Hallucinatory**, because originally correct reasoning is suddenly derailed by a factually incorrect assertion.
- **E** maps to **Repetitive**, because when a trajectory remains stuck in an error state without recovering, the model is trapped in a stagnant loop and fails to generate new productive logic.

Agreement and Confusion Matrix

The human–human agreement between the two independent experts was highly consistent, achieving a Cohen's $\kappa = 0.84$. We resolved the remaining disagreements to form a final human consensus. The expert consensus labels were then compared against the deterministically mapped LLM judge labels. As shown in Table A2, the human-judge agreement reached Cohen's $\kappa = 0.82$, demonstrating strong reliability. This high agreement confirms that the automated judge's local transitions are securely aligned with expert behavior labeling, thereby supporting the aggregate recovery and degradation analyses reported in the main text.

Table A2. Confusion matrix comparing human-expert consensus against the mapped LLM trajectory judge on 200 sampled transitions (Cohen's $\kappa = 0.82$).

	LLM Productive	LLM Repetitive	LLM Hallucinatory	Total (Human)
Human Productive	92	2	3	97
Human Repetitive	4	43	4	51
Human Hallucinatory	4	5	42	51
Total (LLM)	100	50	50	200

Appendix A.7. Trajectory Segmentation Heuristic and Its Bias

Our trajectory analysis uses double-newline paragraph breaks (`\n\n`) as a practical heuristic for segmenting long reasoning traces into blocks. This choice is consistent with recent long-chain-of-thought inference-time studies that treat paragraph- or chunk-level boundaries as natural units for guided reasoning and compression analysis [42,43]. This choice is not a claim that paragraph boundaries are a perfect cognitive decomposition of reasoning. It may merge two short logical moves into one block, or split a single extended derivation when the model inserts an extra paragraph break for style.

To estimate the magnitude of this bias, we manually inspected 50 complete reasoning trajectories and compared the paragraph-level segmentation against coherent thought blocks identified by human readers. In 92% of the inspected trajectories, the double-newline heuristic correctly framed locally coherent reasoning units for the purposes of transition-level analysis. The remaining cases primarily involved formatting irregularities rather than systematic label inversion. Therefore, while the segmentation is heuristic, it is sufficiently stable for the macro-level statistics reported in the paper, especially because all compared methods are segmented by the same rule.

Appendix B. Activation Vector Construction and Plausibility Analysis

This appendix describes how the activation vectors used in our steering method were constructed and why we consider them to be behaviorally meaningful. At a high level, the vectors are derived from contrastive reasoning pairs: one side reflects careful, self-monitoring, verification-oriented reasoning, while the other side reflects shallow, over-confident, or error-prone reasoning. By contrasting these two modes at the hidden-state level, we obtain directions in activation space that are intended to capture a "deliberate reasoning" tendency rather than surface lexical style alone.

Appendix B.1. Data Sources for Contrastive Pairs

The contrastive data used for vector extraction comes from two complementary sources. The first source is a prompt-based synthetic dataset generated with Qwen3-Max. In this part, we explicitly asked the model to create paired internal monologues in which the positive example expresses strong System-2 behavior—for example, checking assumptions, verifying units, revisiting premises, and considering edge cases—whereas the

negative example reflects weak metacognition, overconfidence, or intuitive but unreliable reasoning. We used 317 contrastive pairs in total, with prompts designed to elicit reasoning failures such as factual errors, causal confusion, hasty generalization, gambler's fallacy, and invalid mathematical inference.

The second source is a filtered Claude Opus reasoning dataset collected from Hugging Face. This dataset provides long-form worked solutions with relatively rich reasoning traces, especially in mathematical and educational problem-solving settings. Compared with the prompt-generated pairs, these samples are less explicitly contrastive at the wording level, but they contribute higher-quality and more natural reasoning trajectories. In practice, the two sources play complementary roles: the synthetic pairs sharpen the behavioral distinction we want to extract, while the externally collected reasoning traces improve the naturalness and coverage of the target reasoning style.

Appendix B.2. Prompt Design for Synthetic Contrastive Data

The synthetic portion of the dataset was designed to target a very specific cognitive contrast: reflective verification versus lazy acceptance. Instead of asking for generic "good" and "bad" answers, we prompted the model to produce paired inner-monologue completions. The positive completion was required to sound like a careful reasoner who questions intermediate assumptions and corrects potential mistakes. The negative completion was required to sound like a reasoner who relies on superficial plausibility, dismisses uncertainty, or prematurely accepts an answer.

This design is important because the goal of the activation vector is not merely to separate correct text from incorrect text. Rather, we want to isolate a direction associated with *how* the model reasons: whether it slows down, checks itself, and maintains epistemic caution, as opposed to drifting toward confident but weakly justified conclusions. The prompts were therefore written to emphasize reasoning stance and metacognitive behavior, not only final correctness.

Illustrative Contrastive Examples

Representative prompt-completion pairs include cases such as checking whether 15 is prime, evaluating whether a small personal sample supports a universal claim, distinguishing correlation from causation, identifying gambler's fallacy, correcting a historical date, and recognizing that the triangle inequality does not imply equilateral structure. Across these examples, the positive completions tend to revise assumptions and explicitly verify the logic, while the negative completions tend to endorse the user's flawed intuition on the basis of superficial plausibility.

Appendix B.3. Activation Extraction Procedure

For each contrastive pair, we run the model and extract hidden activations from a model-specific target internal layer located at approximately two-thirds of the network depth: Layer 24 for Qwen3-8B (36 layers), Layer 23 for Gemma-4-E2B (35 layers), and Layer 19 for DeepSeek-R1-Distill-Qwen-1.5B (28 layers). For the primary Qwen3-8B configuration, the hidden dimensionality is 4096. The contrast between the positive and negative sides of each pair is then used to define candidate steering directions. Intuitively, if a consistent behavioral distinction exists across many pairs, these pairwise differences should accumulate into a coherent low-dimensional subspace corresponding to careful reasoning behavior.

The motivation for using internal activations in this way is that hidden states can preserve abstract behavioral regularities even when the prompts vary substantially on the surface. A prime-number example, a historical-fact example, and a causal-reasoning

example may look unrelated lexically, but they can still share a common internal signature if they all instantiate the same high-level pattern of reflective verification versus impulsive acceptance.

Appendix B.4. Why the Data Sources Are Complementary

The two data sources serve different but compatible purposes. The Qwen3-Max pairs are useful because they are intentionally contrastive: they give the extraction procedure a clean signal about the difference between strong and weak reasoning behavior. However, synthetic data alone can sometimes overemphasize stylistic regularities introduced by the prompt template. The external Opus reasoning traces help counterbalance this by contributing more natural reasoning trajectories with broader linguistic variation.

From the perspective of representation learning, this combination is desirable. One source provides clarity of supervision; the other provides diversity and realism. As a result, the extracted vector is less likely to encode only a narrow writing style and more likely to reflect a reusable reasoning tendency that generalizes across domains and prompt forms.

Appendix B.5. PCA-Based Plausibility Analysis

To evaluate whether the extracted contrasts form a meaningful structure rather than random noise, we performed principal component analysis (PCA) on the resulting activation differences. The PCA summary for the primary Qwen3-8B configuration is shown in Table A3. The analysis uses 317 contrastive pairs, the two-thirds-depth target layer (Layer 24 for Qwen3-8B), a hidden dimension of 4096, and a retained PCA dimension of 10.

Table A3. PCA summary for the extracted activation-difference vectors.

Principal Component	Explained Variance (%)	Cumulative Variance (%)
PC1	16.7148	16.7148
PC2	5.3094	22.0242
PC3	3.8135	25.8377
PC4	3.0559	28.8936
PC5	2.9727	31.8663
PC6	2.2368	34.1031
PC7	1.9649	36.0680
PC8	1.6808	37.7487
PC9	1.4963	39.2450
PC10	1.4764	40.7215

Several aspects of this spectrum are encouraging. First, the first principal component accounts for 16.71% of the variance, followed by a sharp drop to 5.31% for the second component. This elbow-shaped decay suggests that the contrastive activations are not distributed isotropically across the full 4096-dimensional space. Instead, there appears to be a dominant direction that captures a substantial portion of the shared behavioral difference across pairs. In the context of our construction, this is exactly what we would hope to observe if the dataset consistently reflects a common reasoning contrast.

Second, the top 10 principal components explain 40.72% of the variance. In an absolute sense, this number may appear moderate, but in a 4096-dimensional hidden space it is in fact a fairly concentrated result. Capturing more than 40% of the total variance with only 10 retained components indicates that the extracted differences occupy a structured and anisotropic subspace rather than being spread uniformly across thousands of dimensions. This is consistent with the view that a relatively small number of internal directions dominate the distinction between reflective reasoning and error-prone shortcutting.

Third, the residual variance is not necessarily undesirable. In high-dimensional language-model activations, a large portion of variance is expected to reflect lexical content, prompt phrasing, positional effects, and other context-specific details that are not central to the reasoning trait we aim to control. From a steering perspective, PCA acts as a purification step: it preserves the most coherent shared directions while filtering away diffuse components that may encode irrelevant or potentially harmful nuisance variation.

Appendix B.6. Interpretation and Limitations

Taken together, these observations support the plausibility of the extracted activation vectors. The contrastive data is behaviorally targeted, the extracted hidden-state differences show substantial low-dimensional structure, and the dominant PCA directions are consistent with the hypothesis that the model internally represents a distinction between careful verification-oriented reasoning and weaker heuristic reasoning. This does not prove that any single component corresponds to a perfectly isolated "truth" or "anti-hallucination" axis, but it does provide evidence that the vectors capture a nontrivial and reusable behavioral signal.

At the same time, this interpretation should be understood cautiously. Because part of the contrastive dataset is synthetic, some of the recovered structure may still reflect prompt-induced regularities. Likewise, PCA identifies directions of shared variance, not necessarily causally pure mechanisms. For this reason, we treat the PCA analysis as a plausibility check rather than as definitive proof. Its role in the paper is to show that the extracted vectors are structured, concentrated, and behaviorally interpretable enough to justify their use in the downstream closed-loop steering experiments.

Appendix B.7. PCA Dimensionality Sensitivity

We further evaluated the sensitivity of the PCA purification dimension k, since the purified steering vector must retain enough reasoning-relevant signal while filtering orthogonal nuisance variation. Table A4 and Figure A1 report a controlled comparison for $k \in \{1, 10, 50\}$.

Table A4. Sensitivity of PCA purification dimension k. Best values are shown in bold.

PCA Dimension (k)	Accuracy	Correct Problems	Repetition Rate	Avg Tokens
$k = 1$ (Under-expression)	25.00%	8/32	0.5229	3419.0
$k = 10$ (Optimal)	**28.13%**	**9/32**	**0.5444**	**3153.0**
$k = 50$ (Orthogonal noise)	15.63%	5/32	0.6453	3580.0

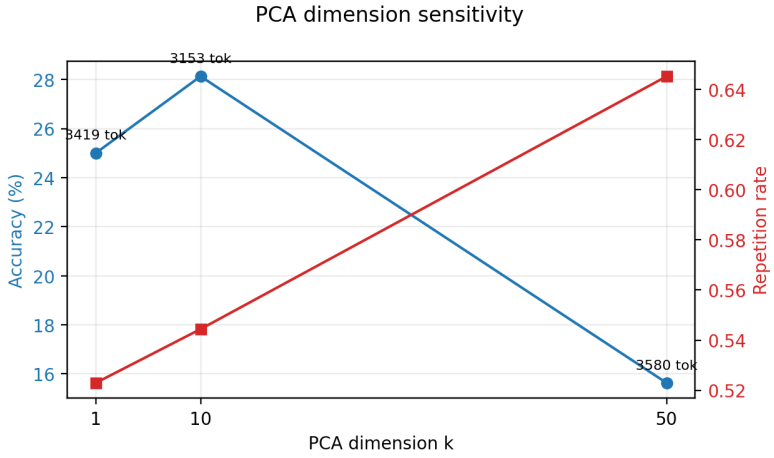


Figure A1. PCA dimension sensitivity. The annotation above each accuracy point reports the corresponding average generated tokens.

The results show a clear trade-off. With $k = 1$, the subspace is too narrow: it captures only the strongest contrastive direction and under-represents the multi-dimensional structure of verification-oriented System-2 reasoning, leading to under-expression and only 25.00% accuracy. With $k = 10$, the retained subspace captures 40.72% of the cumulative activation-difference variance in the 4096-dimensional hidden space. This is the best empirical operating point, reaching 28.13% accuracy while also producing the lowest token footprint. With $k = 50$, the projection begins to reintroduce low-variance orthogonal noise, including stylistic features, formatting tendencies, and positional artifacts. Injecting these noisy directions back into hidden states through SLERP destabilizes the autoregressive process and causes state shock, as reflected by the repetition spike to 0.6453 and the accuracy collapse to 15.63%. This sensitivity analysis supports $k = 10$ as a practical midpoint between under-expression and over-permissive noisy steering.

Appendix B.8. MD5 Checksums for Activation-Vector Reproducibility

To improve reproducibility, we report MD5 checksums for the raw and PCA-purified activation vectors used in the revised experiments. These checksums uniquely identify the vector artifacts used for each model family and allow independent reruns to verify that the same steering directions are loaded.

Table A5. MD5 checksums of activation vectors used in the revised evaluation.

Model	Raw Vector MD5	Purified Vector MD5
Qwen3-8B	`79c829b4d4d4e2b60206a67e3e8e1a93`	`3071192381d751312197cb00212dd4f7`
DeepSeek-R1-Distill-Qwen-1.5B	`c8cd1c895b4a61ff69f2ab2ac321f386`	`ba24c89109ce1a34539961db1f380e9f`
Gemma-4-E2B	`a0683c7d05d872616214184e845ef728`	`c4a3a3cfe7f7f48d0a200f0b7a1756bf`

References

1. Chen, X.; Xu, J.; Liang, T.; He, Z.; Pang, J.; Yu, D.; Song, L.; Liu, Q.; Zhou, M.; Zhang, Z.; et al. Do NOT think that much for 2+3=? On the overthinking of o1-like LLMs. *arXiv* **2025**, arXiv:2412.21187.
2. Ghosal, S.S.; Chakraborty, S.; Reddy, A.; Lu, Y.; Wang, M.; Manocha, D.; Huang, F.; Ghavamzadeh, M.; Bedi, A.S. Does thinking more always help? Mirage of test-time scaling in reasoning models. *arXiv* **2025**, arXiv:2506.04210.
3. Muennighoff, N.; Yang, Z.; Shi, W.; Li, X.L.; Fei-Fei, L.; Hajishirzi, H.; Zettlemoyer, L.; Liang, P.; Candès, E.; Hashimoto, T. s1: Simple test-time scaling. *arXiv* **2025**, arXiv:2501.19393.
4. Oh, M.; Song, S.; Lee, S.; Jo, S.; Jo, Y. ThinkBrake: Efficient Reasoning via Log-Probability Margin Guided Decoding. *arXiv* **2025**, arXiv:2510.00546.
5. Li, K.; Patel, O.; Viégas, F.; Pfister, H.; Wattenberg, M. Inference-time intervention: Eliciting truthful answers from a language model. *Adv. Neural Inf. Process. Syst.* **2024**, *36*, 41451–41530.
6. Turner, A.M.; Thiergart, L.; Leech, G.; Udell, D.; Vazquez, J.J.; Mini, U.; MacDiarmid, M. Steering language models with activation engineering. *arXiv* **2024**, arXiv:2308.10248.
7. Rimsky, N.; Gabrieli, N.; Schulz, J.; Tong, M.; Hubinger, E.; Turner, A.M. Steering Llama 2 via contrastive activation addition. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics, Bangkok, Thailand, 11–16 August 2024.
8. Zou, A.; Phan, L.; Chen, S.; Campbell, J.; Guo, P.; Ren, R.; Pan, A.; Yin, X.; Mazeika, M.; Dombrowski, A.-K.; et al. Representation engineering: A top-down approach to AI transparency. *arXiv* **2023**, arXiv:2310.01405.
9. OpenAI. *Learning to Reason with LLMs*; OpenAI: San Francisco, CA, USA, 2024.
10. DeepSeek-AI; Guo, D.; Yang, D.; Zhang, H.; Song, J.; Wang, P.; Zhu, Q.; Xu, R.; Zhang, R.; Ma, S.; et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv* **2025**, arXiv:2501.12948.
11. Ji, Y.; Li, J.; Xiang, Y.; Ye, H.; Wu, K.; Yao, K.; Xu, J.; Mo, L.; Zhang, M. A survey of test-time compute: From intuitive inference to deliberate reasoning. *arXiv* **2025**, arXiv:2501.02497.
12. Zhang, D.; Li, Z.Z.; Zhang, M.L.; Zhang, J.; Liu, Z.; Yao, Y.; Xu, H.; Zheng, J.; Chen, X.; Zhang, Y.; et al. From system 1 to system 2: A survey of reasoning large language models. *IEEE Trans. Pattern Anal. Mach. Intell.* **2025**, *48*, 3335–3354. [CrossRef]
13. Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q.V.; Zhou, D. Chain-of-thought prompting elicits reasoning in large language models. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 24824–24837. [CrossRef]
14. Zhang, X.; Du, C.; Pang, T.; Liu, Q.; Gao, W.; Lin, M. Chain of preference optimization: Improving chain-of-thought reasoning in llms. *Adv. Neural Inf. Process. Syst.* **2024**, *37*, 333–356. [CrossRef]

15. Lightman, H.; Kosaraju, V.; Burda, Y.; Edwards, H.; Baker, B.; Lee, T.; Leike, J.; Schulman, J.; Sutskever, I.; Cobbe, K. Let's Verify Step by Step. *arXiv* **2023**, arXiv:2305.20050.
16. Wang, X.; Wei, J.; Schuurmans, D.; Le, Q.; Chi, E.; Narang, S.; Chowdhery, A.; Zhou, D. Self-consistency improves chain of thought reasoning in language models. *arXiv* **2022**, arXiv:2203.11171.
17. Yao, S.; Yu, D.; Zhao, J.; Shafran, I.; Griffiths, T.; Cao, Y.; Narasimhan, K. Tree of Thoughts: Deliberate problem solving with large language models. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 11809–11822. [CrossRef]
18. Yang, W.; Ma, S.; Lin, Y.; Wei, F. Towards thinking-optimal scaling of test-time compute for llm reasoning. *Adv. Neural Inf. Process. Syst.* **2026**, *38*, 43605–43631.
19. Brown, B.; Juravsky, J.; Ehrlich, R.; Clark, R.; Le, Q.V.; Ré, C.; Mirhoseini, A. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv* **2024**, arXiv:2407.21787.
20. Sofroniew, N.; Kauvar, I.; Saunders, W.; Chen, R.; Henighan, T.; Hydrie, S.; Citro, C.; Pearce, A.; Tarng, J.; Gurnee, W.; et al. Emotion concepts and their function in a large language model. *arXiv* **2026**, arXiv:2604.07729.
21. Kim, B.; Wattenberg, M.; Gilmer, J.; Cai, C.; Wexler, J.; Viegas, F.; Sayres, R. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In Proceedings of the International Conference on Machine Learning, Stockholm Sweden, 10–15 July 2018; pp. 2668–2677.
22. Schnoor, E.; Tiomoko, M.; Said, J.; Jung, A.; Samek, W. Concept activation vectors: A unifying view and adversarial attacks. In Proceedings of the ICASSP 2026–2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2026; pp. 3481–3485.
23. Wang, T.; Ma, Y.; Liao, K.; Yang, C.; Zhang, Z.; Wang, J.; Liu, X. Token-Aware Editing of internal activations for large language model alignment. In Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, Suzhou, China, 4–9 November 2025.
24. Whiteley, N.; Gray, A.; Rubin-Delanchy, P. Statistical exploration of the manifold hypothesis. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **2026**, *88*, 353–385. [CrossRef]
25. Farghly, T.; Potaptchik, P.; Howard, S.; Deligiannidis, G.; Pidstrigach, J. Diffusion models and the manifold hypothesis: Log-domain smoothing is geometry adaptive. *Adv. Neural Inf. Process. Syst.* **2026**, *38*, 122795–122840.
26. Loshchilov, I.; Hsieh, C.P.; Sun, S.; Ginsburg, B. ngpt: Normalized transformer with representation learning on the hypersphere. In Proceedings of the International Conference on Learning Representations, Singapore, 24–28 April 2025; Volume 2025, pp. 74014–74038.
27. Joshi, A.; Bhatt, D.; Modi, A. Geometry of decision making in language models. *Adv. Neural Inf. Process. Syst.* **2026**, *38*, 144139–144196.
28. Gupta, A.; Ozdemir, A.; Gong, C.; Anumanchipalli, G. Geometric interpretation of layer normalization and a comparative analysis with rmsnorm. In Proceedings of the Findings of the Association for Computational Linguistics: EACL 2026, Rabat, Morocco, 24–29 March 2026; pp. 375–407.
29. Ji, Z.; Lee, N.; Frieske, R.; Yu, T.; Su, D.; Xu, Y.; Ishii, E.; Bang, Y.J.; Madotto, A.; Fung, P. Survey of hallucination in natural language generation. *ACM Comput. Surv.* **2023**, *55*, 1–38. [CrossRef]
30. Holtzman, A.; Buys, J.; Du, L.; Forbes, M.; Choi, Y. The curious case of neural text degeneration. *arXiv* **2019**, arXiv:1904.09751.
31. Welleck, S.; Kulikov, I.; Roller, S.; Dinan, E.; Cho, K.; Weston, J. Neural text generation with unlikelihood training. *arXiv* **2019**, arXiv:1908.04319.
32. Su, Y.; Lan, T.; Wang, Y.; Yogatama, D.; Kong, L.; Collier, N. A contrastive framework for neural text generation. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 21548–21561. [CrossRef]
33. Gallifant, J.; Fiske, A.; Levites Strekalova, Y.A.; Osorio-Valencia, J.S.; Parke, R.; Mwavu, R.; Martinez, N.; Gichoya, J.W.; Ghassemi, M.; Demner-Fushman, D.; et al. Peer review of GPT-4 technical report and systems card. *PLoS Digit. Health* **2024**, *3*, e0000417. [CrossRef] [PubMed]
34. Hao, Z.; Wang, H.; Liu, H.; Luo, J.; Yu, J.; Dong, H.; Lin, Q.; Wang, C.; Chen, J. Rethinking entropy interventions in rlvr: An entropy change perspective. *arXiv* **2025**, arXiv:2510.10150.
35. Hendrycks, D.; Burns, C.; Kadavath, S.; Arora, A.; Basart, S.; Tang, E.; Song, D.; Steinhardt, J. Measuring mathematical problem solving with the MATH dataset. *arXiv* **2021**, arXiv:2103.03874.
36. Mathematical Association of America. American Invitational Mathematics Examination 2024. 2024. Available online: https://maa.org/maa-invitational-competitions/ (accessed on 16 June 2026).
37. Mathematical Association of America. American Invitational Mathematics Examination 2025. 2025. Available online: https://maa.org/maa-invitational-competitions/ (accessed on 16 June 2026).
38. Yang, A.; Li, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Gao, C.; Huang, C.; Lv, C.; et al. Qwen3 technical report. *arXiv* **2025**, arXiv:2505.09388.
39. Huang, Y.; Chen, H.; Ruan, S.; Zhang, Y.; Wei, X.; Dong, Y. Mitigating Overthinking in Large Reasoning Models via Manifold Steering. *arXiv* **2025**, arXiv:2505.22411.

40. You, Z.; Deng, C.; Chen, H. Spherical Steering: Geometry-Aware Activation Rotation for Language Models. *arXiv* **2026**, arXiv:2602.08169.
41. Chen, R.; Zhang, Z.; Hong, J.; Kundu, S.; Wang, Z. SEAL: Steerable Reasoning Calibration of Large Language Models for Free. *arXiv* **2025**, arXiv:2504.07986.
42. Yang, W.; Yue, X.; Chaudhary, V.; Han, X. Speculative thinking: Enhancing small-model reasoning with large model guidance at inference time. *arXiv* **2025**, arXiv:2504.12329.
43. Wang, Y.; Luo, H.; Yao, H.; Huang, T.; He, H.; Liu, R.; Tan, N.; Huang, J.; Cao, X.; Tao, D.; et al. R1-compress: Long chain-of-thought compression via chunk compression and search. *arXiv* **2025**, arXiv:2505.16838.
44. Kumar, S.; Imambi, S. MIMIC-EYE: A secure and explainable multi-modal deep learning framework for clinical decision support. *Int. J. Comput. Methods Exp. Meas.* **2025**, *13*, 484–506. [CrossRef]
45. Turgay, S.; Aydin, A.; Erdogan, S.; Yıldırım, M.; Kavacık, M. Enhancing stock market forecasting through deep learning and decentralized data integrity: A blockchain-integrated framework. *J. Intell. Manag. Decis.* **2025**, *4*, 118–136. [CrossRef]
46. Kim, W.; Kruglik, S.; Kiah, H.M. Verifiable coded computation of multiple functions. *IEEE Trans. Inf. Forensics Secur.* **2024**, *19*, 8009–8022. [CrossRef]