BENCHMARKING SAMPLE REASONING QUALITY FROM INTERNAL GEOMETRIC DYNAMICS IN LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Understanding how a large language model processes a reasoning sample requires more than inspecting its final answer. Prior work studies token-wise or layer-wise dynamics, but typically follows one representation at a time or delegates the trajectory to an opaque classifier. We instead propose CUBE, which operationalizes sample-level reasoning quality through the joint geometric dynamics of all tokens across layers. A supervised calibration stage learns one shared, low-dimensional, outcome-discriminative coordinate map; CUBE then summarizes each trajectory with three interpretable signatures: the Detour Ratio, Convergence Ratio, and Early-Step Consistency. We demonstrate two uses. Once calibrated, CUBE-EVAL scores a supplied reasoning text without generating or checking an answer for that text, while CUBE-SELECTOR uses a strong model's geometry to rank post-training examples. Across multiple model families and scales, we find that the signatures remain informative about answer correctness, and the selected subsets improve downstream accuracy over the reported selection baselines. Code and data will be released for future research.

1 INTRODUCTION

Large language models (LLMs) have demonstrated remarkable reasoning abilities, from resolving real-world GitHub issues (Jimenez et al., 2024) to making mathematical discoveries (Romera-Paredes et al., 2024) and conducting autonomous chemical research (Boiko et al., 2023). Yet how they reason internally remains poorly understood. Such an understanding is key to both improving and evaluating LLM reasoning: it would guide how we select training samples (Albalak et al., 2024), design training procedures and architectures, and build trustworthy benchmarks. Today, however, most attempts to judge the reasoning behind a given sample rely on the model's surface output, its final outcome or chain of thought, and thus lack any genuine account of the computation that produced it. We argue that this is not enough: an outcome or a chain of thought does not faithfully reflect the quality of reasoning, and we must instead look into how the reasoning dynamics evolve inside the model. Intuitively, many different reasoning trajectories can arrive at the same final state, yet they are not of equal quality: one may drive almost straight to the answer while another wanders, doubles back, and stumbles into it. The final answer, the signal most criteria rely on, cannot tell these apart, so how to assess the intrinsic quality of reasoning, rather than merely its outcome, has become a bottleneck for the field. In this work, we use *reasoning quality* as an operational term for three properties of the internal process—directness, relative stabilization, and the consistency of early update scales. We test the criterion validity of this operationalization through its associations with answer correctness and controlled failure types.

> *It is good to have an end to journey toward; but it is the journey that matters, in the end.*
>
> —Ursula K. Le Guin

To answer this question, a growing body of work seeks the answer inside the model. These efforts differ mainly in the axis along which they measure evolution. One line tracks token-wise evolution along the generated chain of thought: TRACED measures the progress and stability of the hidden-state trajectory (Jiang et al., 2026), "Flowing Logics" models reasoning as a flow of

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

step-level embeddings (Zhou et al., 2026), and MI-Peaks locates informative thinking tokens via mutual information (Qian et al., 2025). A complementary line tracks depth-wise evolution across layers: the Tuned Lens decodes each layer's intermediate prediction in vocabulary space (Belrose et al., 2023), intrinsic-dimension analyses show representations expanding and then compressing with depth (Joshi et al., 2025), and belief-state geometry can be distributed across the residual streams of multiple layers (Shai et al., 2024). "Truth as a Trajectory" spans both axes, feeding the layer-wise displacements of all tokens to a learned sequence classifier (Damirchi et al., 2026). However, these methods either follow a single representation at each step or layer, or feed activation changes to an opaque classifier; none offers an interpretable account of how the configuration of all tokens is jointly reshaped across layers.

To fill this gap, we propose CUBE, which characterizes LLM reasoning dynamics through the lens of solving a Rubik's cube. We represent a sample at each layer by the joint configuration of all its tokens, projected into a shared low-dimensional, outcome-discriminative coordinate space learned on a labeled calibration set. Across layers, these configurations form the sample's reasoning trajectory. Viewing each layer as one move toward a solved configuration (Joyner, 2008), we introduce three signatures: the Detour Ratio, Convergence Ratio, and Early-Step Consistency. Together they measure whether the trajectory advances directly, stabilizes relative to the model's typical depth profile, and avoids unusually small updates in its early layers. Once the shared subspace is fit on a small labeled calibration set, scoring a new sample reads only the model's internal trajectory, with no outcome label or answer check, which makes the assessment directly useful in practice. We therefore turn it into two representative applications: an evaluator that scores a supplied reasoning text from its internal trajectory, and a selector that uses a strong model to pick high-quality training samples. We summarize our contributions as follows.

- We propose CUBE, a framework that operationalizes reasoning quality from internal dynamics rather than from the outcome alone. It tracks how the configuration of all tokens evolves across layers and summarizes the trajectory with three geometric signatures: the Detour Ratio, Convergence Ratio, and Early-Step Consistency.
- We develop two applications of CUBE: CUBE-EVAL, which scores supplied reasoning texts after supervised calibration from their internal trajectories, and CUBE-SELECTOR, which uses the reasoning dynamics of a strong model to select high-quality samples for post-training.
- We study four model families across mathematics, code generation, and multi-hop question answering. The resulting associations with correctness are model and domain dependent. An idealized discrete-depth model links directness to directional signal-to-noise ratio and relative stabilization to depth-wise contraction.

2 RELATED WORK

We review two lines of related work: reasoning dynamics in LLMs and data selection for LLM training. Since the former is more closely related to our method, we discuss it here and defer the latter to Appendix B.2. A growing body of work treats reasoning not as a single terminal prediction but as a process that unfolds inside the network, and seeks to characterize how the internal state evolves. These characterizations differ primarily along the axis on which evolution is measured. One line studies token-wise evolution along the generated chain of thought: TRACED projects final-layer hidden states onto a learned subspace and characterizes the resulting trajectory by its progress and stability (Jiang et al., 2026), "Flowing Logics" models reasoning as a flow of step-level embeddings whose velocity is governed by logical statements (Zhou et al., 2026), MI-Peaks identifies thinking tokens as information peaks via mutual information (Qian et al., 2025), and other work decomposes chain-of-thought computation into sparse-autoencoder features (Chen et al., 2026). A complementary line studies depth-wise evolution across layers or inference-time recurrence: the Tuned Lens decodes intermediate predictions layer by layer (Belrose et al., 2023), intrinsic-dimension analyses show that representations expand and then compress across layers as models reach a decision (Joshi et al., 2025), belief-state work finds that transformers linearly represent the geometry of belief states in their residual stream, sometimes distributed across layers (Shai et al., 2024), and latent reasoning models reason in continuous or depth-recurrent latent spaces (Hao et al., 2025; Altabaa et al., 2025), although probing such models reveals limited evidence of an interpretable latent chain of thought (Lu et al., 2025). "Truth as a Trajectory" spans both axes, feeding the layer-wise displacements of

all tokens to a learned sequence classifier (Damirchi et al., 2026). Appendix B.1 summarizes these methods in Table 7 and contrasts them with CUBE.

3 METHODS

This section presents the methodological framework of CUBE. Section 3.1 first formalizes reasoning dynamics at the sequence level. Building on this formulation, we introduce the Detour Ratio, Convergence Ratio, and Early-Step Consistency in Section 3.2. Section 3.3 then establishes their theoretical interpretation under a depth-wise stochastic model.

3.1 PRELIMINARIES AND NOTATION

Reasoning Dynamics in LLMs. Prior work differs in how it construes reasoning dynamics in LLMs. We summarize these characterizations in Table 7 of Appendix B.1, organized by the axis along which evolution is measured and the space in which the evolving state is read out. We argue that the quality of reasoning is reflected not only in the terminal state but also in how the reasoning geometry is transformed and transported across layers. Intuitively, this process can be viewed as follows: ***an LLM progressively refines its understanding of the text as computation unfolds across layers.*** Motivated by this view, we first formalize reasoning dynamics in LLMs in Definition 3.1.

Definition 3.1 (Reasoning Dynamics)**.** *Let M_θ be a transformer with L transformer blocks and width D. Index the embedding state by $\ell = 0$ and the block outputs by $\ell = 1, \ldots, L$. For the i-th reasoning sample x_i with T_i tokens, let $h_{i,t}^{(\ell)} \in \mathbb{R}^D$ denote the hidden state at token t and layer ℓ. The reasoning dynamics of M_θ on x_i is the depth-ordered sequence of per-layer geometry matrices $\Gamma_i = \big(\mathbf{Z}_i^{(0)}, \mathbf{Z}_i^{(1)}, \ldots, \mathbf{Z}_i^{(L)}\big)$, where each $\mathbf{Z}_i^{(\ell)} = \big[z_{i,0}^{(\ell)}, \ldots, z_{i,T_i-1}^{(\ell)}\big]^\top \in \mathbb{R}^{T_i \times k}$ collects the k-dimensional token coordinates $z_{i,t}^{(\ell)} \in \mathbb{R}^k$ at layer ℓ, which lie in a low-dimensional reasoning subspace.*

Low-Dimensional Reasoning Subspace. Following learned-readout methods (Jiang et al., 2026; Sun et al., 2026), we seek one low-dimensional, outcome-discriminative map shared by every layer. Let $U \in \mathbb{R}^{D \times V}$ and $b_{\text{out}} \in \mathbb{R}^V$ be the unembedding parameters. We form $o_{i,t}^{(\ell)} = h_{i,t}^{(\ell)} U + b_{\text{out}}$ and learn $B \in \mathbb{R}^{V \times k}$ from labeled calibration states with a supervised contrastive loss. Calibration samples exclude each scored sample; in the correlational analysis they are the current outer-training fold. The shared coordinates are $z_{i,t}^{(\ell)} = o_{i,t}^{(\ell)} B \in \mathbb{R}^k$.

This model-tied parameterization induces the effective map $A_{\text{eff}} = UB \in \mathbb{R}^{D \times k}$, while b_{out} cancels from layer differences. When U has full row rank, varying B spans all linear maps from $\mathbb{R}^D$ to $\mathbb{R}^k$; discrimination comes from the supervised fit of B. Appendix A.4 gives the loss and Algorithm 1; Section 4.3 compares alternative spaces.

3.2 LAYER-TRAJECTORY SIGNATURES OF REASONING QUALITY

Inspired by Rubik's-cube solving strategies, we design three geometric signatures to characterize whether an LLM refines its understanding directly, stabilizes relative to its usual depth profile, and avoids unusually small early updates. Figure 1 previews these signatures through direct and roundabout solution paths, consistent and uneven early moves, and early and late stabilization.

Given the sequence-level trajectory Γ_i, we treat each $\mathbf{Z}_i^{(\ell)}$ as a point in $\mathbb{R}^{T_i \times k}$. For each $\ell = 0, \ldots, L-1$, the layer increment is $\Delta\mathbf{Z}_i^{(\ell)} = \mathbf{Z}_i^{(\ell+1)} - \mathbf{Z}_i^{(\ell)}$, with magnitude $s_\ell^{(i)} = \|\Delta\mathbf{Z}_i^{(\ell)}\|_F$. The Frobenius norm measures the joint displacement of all token coordinates and provides the common basis for the three signatures.

Directness or Roundaboutness. The Detour Ratio, formalized in Definition 3.2, captures how directly an LLM refines its understanding across layers. For a difficult sample, successive layers may repeatedly revise earlier interpretations, producing substantial internal motion but limited net

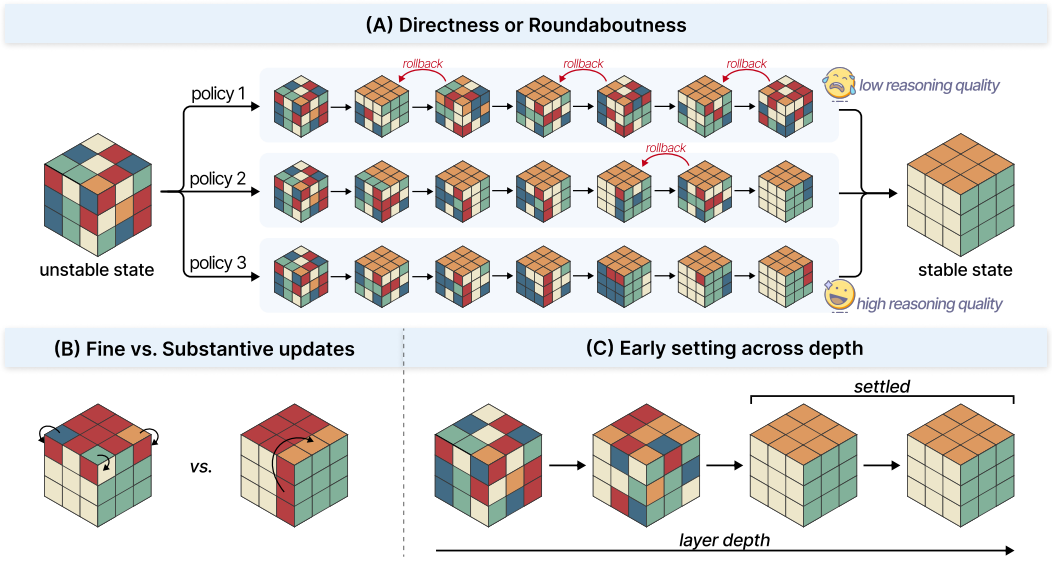


Figure 1: **Rubik's-cube intuition for the layer-trajectory signatures.** **(A)** Indirect or mutually cancelling updates lengthen the path to the same endpoint, lowering the Detour Ratio. **(B)** Consistent vs. uneven early update scales, summarized by Early-Step Consistency. **(C)** Early settling reduces late-layer motion, raising the Convergence Ratio.

progress. This resembles a roundabout Rubik's-cube strategy that undoes earlier moves or revisits configurations instead of advancing consistently toward the solution. The ratio captures this distinction by comparing net endpoint displacement with accumulated motion.

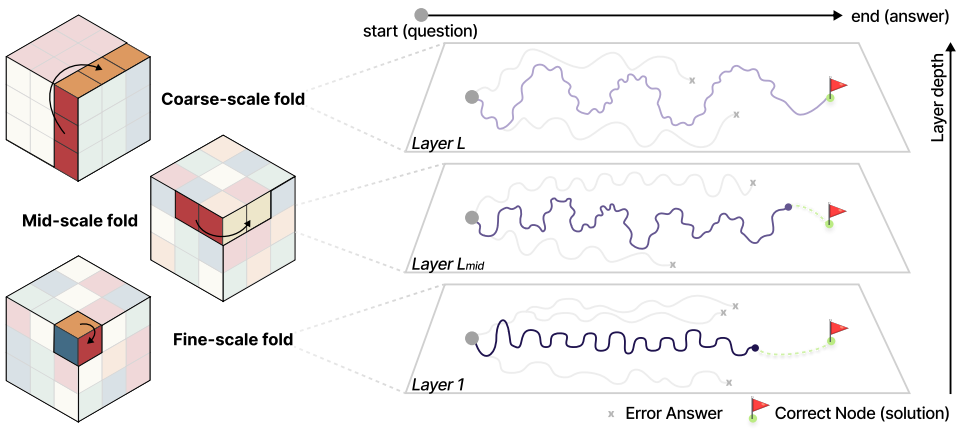
Definition 3.2 (Detour Ratio). *For nonzero endpoint displacement, the Detour Ratio is* $\mathcal{D}_i = \|\mathbf{Z}_i^{(L)} - \mathbf{Z}_i^{(0)}\|_F / \left(\sum_{\ell=0}^{L-1} s_\ell^{(i)}\right) \in (0, 1]$, *the ratio of the direct first-to-last displacement to the sequence-level path length. Equality holds precisely when all nonzero increments are positively collinear.*

Accordingly, values near one indicate direct refinement, whereas smaller values indicate increasingly roundabout updates.

Stabilization or Self-Doubt. Directness alone does not reveal whether the trajectory settles near the output. In Rubik's-cube solving, a strategy may move broadly toward the solution yet still require substantial corrections in its final moves. The Convergence Ratio, formalized in Definition 3.3, compares early-layer motion with late-layer motion to distinguish early settling from persistent late correction.

Definition 3.3 (Convergence Ratio). *Let* $w = \max(1, \lfloor L/3 \rfloor)$. *We use the first* w *transitions,* $\mathcal{W}_E = \{0, \ldots, w-1\}$, *and the* w *transitions immediately before the final one,* $\mathcal{W}_L = \{L-w-1, \ldots, L-2\}$. *The Convergence Ratio is* $\mathcal{C}_i = \left(|\mathcal{W}_E|^{-1} \sum_{\ell \in \mathcal{W}_E} s_\ell^{(i)}\right) / \left(|\mathcal{W}_L|^{-1} \sum_{\ell \in \mathcal{W}_L} s_\ell^{(i)}\right)$, *provided the late-layer mean motion is nonzero.*

The final transition is excluded from the late window because some extraction pipelines apply a terminal normalization only to the last returned state, making that transition unlike the block-to-block updates. This convention is specific to $\mathcal{C}_i$; $\mathcal{D}_i$ retains the endpoint transition by definition. Absolute levels also reflect a model-wide depth profile: in our measured trajectories, late updates are typically larger than early ones. We therefore compare $\mathcal{C}_i$ within the same model and extraction convention, where a larger value indicates earlier relative stabilization.



(a) multi-scale cube folding (b) layer-wise latent path folding

Figure 2: Reasoning as multi-scale folding. (a) A reasoning step is a Rubik's-cube move that transforms the configuration at different levels, from a single cell to a block to a whole face. (b) The same trajectory evolves on the latent concept space: stacking layers from bottom to top, fine high-frequency wiggles in early layers are progressively reorganized into a smaller number of coarse bends.

Consistency of Early Update Scales. As illustrated in Figure 2, Rubik's-cube strategies combine coarse moves that reorganize large parts of the configuration with fine moves that make local adjustments. Early-Step Consistency, formalized in Definition 3.4, measures how often a trajectory's first-third updates are not far below its own typical early update. Because the reference is sample-relative, this signature describes lower-tail regularity, not an absolute amount of computation.

Definition 3.4 (Early-Step Consistency)**.** *Using w and $\mathcal{W}_E$ from Definition 3.3, let $\widetilde{s}_i^E = \mathrm{median}_{\ell \in \mathcal{W}_E} s_\ell^{(i)}$. With the fixed threshold $\alpha = 0.7$, Early-Step Consistency is $\mathcal{F}_i = w^{-1} \sum_{\ell \in \mathcal{W}_E} \mathbb{1}[s_\ell^{(i)} \geq \alpha \widetilde{s}_i^E] \in [1/2, 1]$.*

Values near one mean that few early updates fall substantially below the sample's median early scale; smaller values indicate a heavier lower tail of unusually small updates. Because the reference scale is sample-relative, uniformly tiny steps also yield $\mathcal{F}_i = 1$.

3.3 Theoretical Analysis

Following the discrete-depth setup of Jiang et al. (2026), we use an idealized Gaussian increment model to isolate two properties of the signatures. Each layer update contains a component in one shared direction and an isotropic component, while a positive depth profile allows update magnitudes to grow or shrink across layers.

Theorem 3.1 gives two comparative statements. ***For a fixed depth profile, the Detour Ratio increases monotonically with directional signal-to-noise ratio; for a profile with sample-specific contraction, the Convergence Ratio increases relative to the corresponding model-wide depth baseline.*** These are the directness and relative-stabilization components of reasoning quality. A higher directional signal-to-noise ratio places more of each layer update along one shared direction, so a larger share of the accumulated motion becomes net displacement and the trajectory refines more directly. Sample-specific contraction reduces late-layer motion relative to the model's ordinary depth profile, so the trajectory stabilizes earlier than that baseline.

Theorem 3.1 (Directional signal and relative stabilization). *Under Assumption D.1, set* $w = \max(1, \lfloor L/3 \rfloor)$, $A_i = \sum_{\ell=0}^{L-1} a_{i,\ell}$, *and* $\rho_i = \sum_{\ell=0}^{L-1} a_{i,\ell}^2 / A_i^2$. *As* $d_i = T_i k \to \infty$ *with* L *fixed,*

$$\mathcal{D}_i = \sqrt{\frac{m_i^2 + \rho_i \sigma_i^2}{m_i^2 + \sigma_i^2}} + o_p(1) \quad \textit{and} \quad \mathcal{C}_i = \frac{\sum_{\ell \in \mathcal{W}_E} a_{i,\ell}}{\sum_{\ell \in \mathcal{W}_L} a_{i,\ell}} \big(1 + o_p(1)\big).$$

For fixed $\{a_{i,\ell}\}$, $\sigma_i > 0$, *and* $\rho_i < 1$, *the leading term of* $\mathcal{D}_i$ *increases strictly with* m_i/σ_i. *If* $a_{i,\ell} = \lambda_\ell q_i^\ell$ *with* $q_i \in (0, 1]$, *set* $\mathcal{C}_0 = (\sum_{\ell \in \mathcal{W}_E} \lambda_\ell)/(\sum_{\ell \in \mathcal{W}_L} \lambda_\ell)$. *Then* $\mathcal{C}_i/\mathcal{C}_0 \geq q_i^{-(L-2w)}(1 + o_p(1))$, *with a strict leading-order bound when* $q_i < 1$ *and* $L \geq 3$.

Intuitively, aligned components add in the endpoint displacement, whereas independent high-dimensional components add mainly in squared norm. The amount of alignment therefore changes $\mathcal{D}_i$ continuously. Likewise, raw $\mathcal{C}_i$ combines sample-specific contraction with the model's ordinary depth profile, giving it a within-model or baseline-normalized stabilization interpretation. Early-Step Consistency is scale-invariant by construction and records the lower tail of a sample's early step-size distribution. The complete assumptions and proof are in Appendix D.

4 Experiments

This section presents our experiments. Section 4.1 describes the setup, with full configurations deferred to Appendix A; Section 4.2 reports the main results; and Section 4.3 provides further analyses.

4.1 Experimental Setup

Models. For the correlational analysis, we evaluate Qwen3-8B (Qwen Team, 2025), Llama-3.1-8B (Llama Team, 2024), GLM-4-9B (GLM Team et al., 2024), and Hunyuan-7B (Sun et al., 2024); Qwen3-8B is evaluated in both thinking and non-thinking modes. Smaller checkpoints are used only in the scaling check. CUBE-EVAL uses Qwen3-8B and Llama-3.1-8B. CUBE-SELECTOR fine-tunes two target models, Qwen3-8B and Hunyuan-7B, with DeepSeek-V4-Flash as the strong model. The hallucination analysis in Appendix C.2 uses the same strong model. Table 4 lists each checkpoint together with its architecture, parameter count, inference mode, and experiment assignment.

Benchmarks. We evaluate three reasoning categories: mathematical reasoning on MATH-500 (Lightman et al., 2024) and AIME 2024 (Mathematical Association of America, 2024); code generation on LiveCodeBench (Jain et al., 2024) and BigCodeBench (Zhuo et al., 2025); and multi-hop question answering on MoreHopQA (Schnitzler et al., 2024) and FanOutQA (Zhu et al., 2024). For FanOutQA, we use its 310-example public development split in the closed-book setting because the 724-example test split has no public answer labels. The primary metric is exact match for both mathematics benchmarks, pass@1 for both code benchmarks, normalized exact match for MoreHopQA, and loose string accuracy for FanOutQA. We macro-average the two benchmark scores within each category for model-level summaries.

4.2 Main Results

We first assess the validity of the geometric signatures by analyzing their association with reasoning correctness across model families and domains in Section 4.2.1. We then examine CUBE's performance in two concrete applications in Section 4.2.2: CUBE-EVAL for scoring supplied solutions from their internal trajectories, and CUBE-SELECTOR for post-training data selection.

4.2.1 Correlational Analysis of Geometric Signatures

To test whether CUBE's geometry carries information about reasoning correctness, we use a pipeline that cleanly separates response generation, geometric measurement, and downstream prediction. Each benchmark question is rendered with the model's chat template to elicit a reasoning trace and final answer, while the token-wise, layer-wise hidden states and binary correctness label are cached during generation. From these cached trajectories, we compute the Detour Ratio $\mathcal{D}_i$, Convergence

Table 1: **Cross-validated correctness prediction from the three geometric signatures.** For each model and domain, we fit a logistic regression jointly on standardized $\mathcal{D}$, $\mathcal{C}$, and $\mathcal{F}$. We report cross-validated AUC and the fold mean of each standardized coefficient in units of 10^{-1}, with one standard deviation given as a subscript. Stars mark coefficients whose five per-fold estimates differ from zero in a two-sided one-sample t-test ($^{*}p < 0.05$, $^{**}p < 0.01$, $^{***}p < 0.001$). **Bold** marks the best AUC within a domain and underline the runner-up.

Model / setting	CV AUC	$\widehat{\beta}_{\mathcal{D}}$ ($\times 10^{-1}$)	$\widehat{\beta}_{\mathcal{C}}$ ($\times 10^{-1}$)	$\widehat{\beta}_{\mathcal{F}}$ ($\times 10^{-1}$)
Mathematical reasoning				
Qwen3-8B (Think)	0.520	$+0.18_{\pm 1.23}$	$+0.32_{\pm 0.84}$	$+0.35_{\pm 0.73}$
Qwen3-8B (NoThink)	**0.753**	$+0.75^{**}_{\pm 0.41}$	$+0.39^{*}_{\pm 0.35}$	$+1.08^{***}_{\pm 0.43}$
Llama-3.1-8B	<u>0.740</u>	$+1.46^{***}_{\pm 0.56}$	$+0.36^{**}_{\pm 0.24}$	$+0.58^{**}_{\pm 0.35}$
Hunyuan-7B	0.618	$+0.52_{\pm 0.66}$	$+0.33_{\pm 0.75}$	$+0.70_{\pm 0.84}$
GLM-4-9B	0.690	$+0.60^{*}_{\pm 0.59}$	$+0.86^{**}_{\pm 0.60}$	$+0.67^{**}_{\pm 0.43}$
Code generation				
Qwen3-8B (Think)	0.643	$+0.06_{\pm 1.10}$	$+0.50_{\pm 1.01}$	$+0.22_{\pm 1.26}$
Qwen3-8B (NoThink)	**0.882**	$+0.72^{**}_{\pm 0.47}$	$+2.00^{***}_{\pm 0.90}$	$+0.65^{**}_{\pm 0.42}$
Llama-3.1-8B	<u>0.788</u>	$+0.80_{\pm 0.86}$	$+1.65^{**}_{\pm 0.81}$	$+0.92^{**}_{\pm 0.58}$
Hunyuan-7B	0.661	$+0.37_{\pm 0.76}$	$-0.34_{\pm 0.53}$	$+0.89^{**}_{\pm 0.64}$
GLM-4-9B	0.781	$+0.56_{\pm 0.70}$	$+1.46^{**}_{\pm 0.83}$	$+0.81^{**}_{\pm 0.62}$
Multi-hop QA				
Qwen3-8B (Think)	0.537	$+0.22_{\pm 0.37}$	$+0.26_{\pm 0.42}$	$+0.14_{\pm 0.43}$
Qwen3-8B (NoThink)	0.603	$-0.22_{\pm 1.35}$	$+0.07_{\pm 1.32}$	$+0.14_{\pm 0.87}$
Llama-3.1-8B	<u>0.621</u>	$-0.01_{\pm 0.84}$	$+1.27^{**}_{\pm 0.93}$	$+0.12_{\pm 0.98}$
Hunyuan-7B	0.607	$+0.48_{\pm 0.63}$	$+0.08_{\pm 0.43}$	$+0.65^{*}_{\pm 0.68}$
GLM-4-9B	**0.674**	$+0.50_{\pm 0.64}$	$+1.17^{*}_{\pm 1.06}$	$+0.45_{\pm 1.03}$

Ratio $\mathcal{C}_i$, and Early-Step Consistency $\mathcal{F}_i$, requiring neither another model forward pass nor access to the raw question text. For each model–domain pair, a logistic regression receives only these three signatures and predicts correctness. We use stratified five-fold cross-validation, fitting the projection B, feature standardization, and regression parameters on each training fold before applying them to its held-out fold. Table 1 reports out-of-fold AUC and the fold-wise mean and standard deviation of each coefficient.

The joint signatures yield above-chance AUC point estimates in every reported cell, but the strength of the association is not consistent across settings. It is strongest for non-thinking mathematics and code, where Qwen3-8B reaches AUC 0.753 and 0.882 and Llama-3.1-8B reaches 0.740 and 0.788. Each row is fit separately within one model configuration, and the Think–NoThink contrast shows that the usefulness of these signatures also depends on inference mode. Overall, Table 1 shows model-, domain-, and mode-dependent associations with correctness.

4.2.2 Applications: Reasoning Evaluation and Data Selection

Having established that the geometry contains a signal associated with answer correctness, we explore two practical uses of the resulting operational profile. CUBE-EVAL scores supplied solution texts from their depth-wise trajectories and aggregates those scores for model comparison, whereas CUBE-SELECTOR uses a strong model's calibrated geometry to rank training samples.

CUBE-EVAL: scoring supplied solutions from internal trajectories. CUBE-EVAL scores how a model processes a supplied text from its depth-wise internal trajectory; the model generates no response, and no answer is extracted or checked during scoring. Each input is a self-contained declarative paragraph that DeepSeek-V4-Pro constructs from a benchmark QA sample and its reference solution. For example, *"A rectangle has perimeter 30 and length twice its width. What is its area?"* becomes *"A rectangle with perimeter 30 and length twice its width has width 5, length 10, and therefore area 50."* Because the paragraph already states the answer, CUBE-EVAL measures how the model processes a given solution rather than whether it can find one. The cross-fitted calibration data provide the outcome labels used to learn B and the random-forest scorer (Appendix A.4), and aggregate scores are compared with accuracy on the original benchmarks.

We aggregate the sample scores on the mathematics and multi-hop QA benchmarks and compare them with accuracy on the original benchmarks. Across model–benchmark cells, the CUBE-EVAL score has a positive rank correlation of $\rho = 0.55$ with accuracy (Figure 3). While Table 1 establishes the sample-level signal on generated solutions, this aggregate signal persists when the model only reads solution-containing declarative text rather than QA-formatted input or its own generated answer. The eight model–benchmark aggregates provide a model-ranking check for declarative inputs.

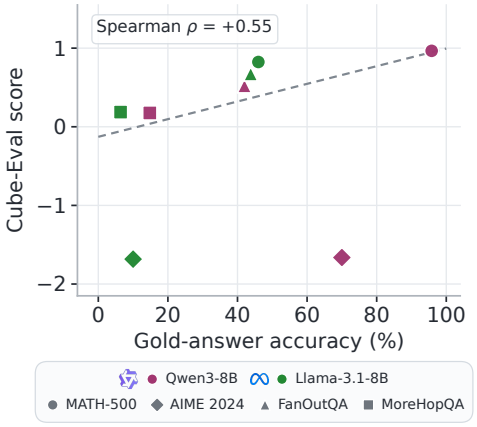


Figure 3: **CUBE-EVAL score versus answer-based evaluation.** Each point is one of 8 model–benchmark cells from Qwen3-8B and Llama-3.1-8B. The score is the predicted score of a random forest trained on the geometric signatures (Table 5).

CUBE-SELECTOR: post-training sample selection. Intuitively, ***a sample that a strong model struggles to semantically refine across its layers is a poor training example.*** From a random pool of 20,000 Open-R1 examples (Hugging Face, 2025b), each method selects exactly 6,000 (a fixed 30% budget). The strong model first solves problems from a disjoint Open-R1 validation split, treating incorrect trajectories as negative examples and correct trajectories as positive examples. We learn B from these labeled trajectories and fit a random-forest classifier on $\mathcal{D}$, $\mathcal{C}$, and $\mathcal{F}$. The fitted classifier scores the candidate trajectories, and CUBE-SELECTOR keeps the 6,000 highest-scoring candidates. Every subset then undergoes the same LoRA fine-tuning recipe with three seeds (Table 6). Table 2 compares it with random selection, perplexity, DEITA (Liu et al., 2024), and LESS (Xia et al., 2024) on MATH-500, OlympiadBench (He et al., 2024), and AIME 2024 (protocol in Appendix A.4). Its reported means exceed the strongest comparator by 1.1 points for Qwen3-8B and 1.0 point for Hunyuan-7B. The advantage is consistent across all six model–benchmark cells rather than being driven by a single task. Because CUBE scores are computed from the strong model before either target model is fine-tuned and without using evaluation labels, the gains indicate that the selected subset transfers across target families and benchmarks.

Table 2: **Fixed-budget post-training selection.** Mean downstream accuracy (%) over three seeds; every method uses the same 20K pool, 6K subset, and optimization. Avg. is the unweighted benchmark mean.

Model	Selector	MATH-500	Olympiad	AIME'24	Avg.
Qwen3-8B	Random	90.4	58.5	54.2	67.7
	Perplexity	89.4	59.5	55.0	68.0
	DEITA	91.8	59.8	55.6	69.1
	LESS	92.2	60.3	56.1	69.5
	CUBE-SELECTOR	**93.0**	**61.3**	**57.4**	**70.6**
Hunyuan-7B	Random	86.2	57.3	50.0	64.5
	Perplexity	85.7	57.8	50.4	64.6
	DEITA	87.3	58.6	51.2	65.7
	LESS	87.9	59.1	51.8	66.3
	CUBE-SELECTOR	**88.7**	**60.1**	**53.0**	**67.3**

4.3 ANALYSIS

In this subsection, we further analyze the reliability of CUBE: we test its sensitivity to the representation-space configuration and separately trace how conventional answer accuracy scales within and across families. Appendix C.2 further examines whether trajectory geometry distinguishes qualitatively different reasoning failures that correctness alone conflates.

Representation-space configuration ablation. On a 20% Qwen3-8B Think tuning subset, we use five folds to compare raw and unembedding-transformed trajectories with learned maps at $k \in$

$\{4, 8, 16, 32, 64\}$. Figure 4 shows that the learned map outperforms both baselines on mathematics and code. On mathematics, performance is stable for $k \in \{4, 8, 16\}$ before declining at $k = 32$ and 64, while every learned map remains above the two fixed representations. Code shows the same broad advantage: it peaks at $k = 4$ and stays above both baselines across all tested dimensions. Multi-hop QA is more selective, reaching a pronounced maximum at $k = 16$, whereas the smaller and larger maps fall near or below the fixed representations. The effect of dimension is therefore domain dependent rather than monotonic. The selected $k = 16$ lies on the high-performing range for mathematics and code while capturing the unique multi-hop QA peak, making it the strongest shared operating point across the three domains. Notably, even the four-dimensional learned map exceeds both 4096-dimensional fixed representations on mathematics and code, so the gain cannot be attributed to retaining more coordinates. Together, these results suggest that correctness-relevant variation is concentrated in a compact subspace rather than spread uniformly across the residual stream. They also favor selecting the projection dimension on held-out calibration data instead of treating it as a universal architectural constant.

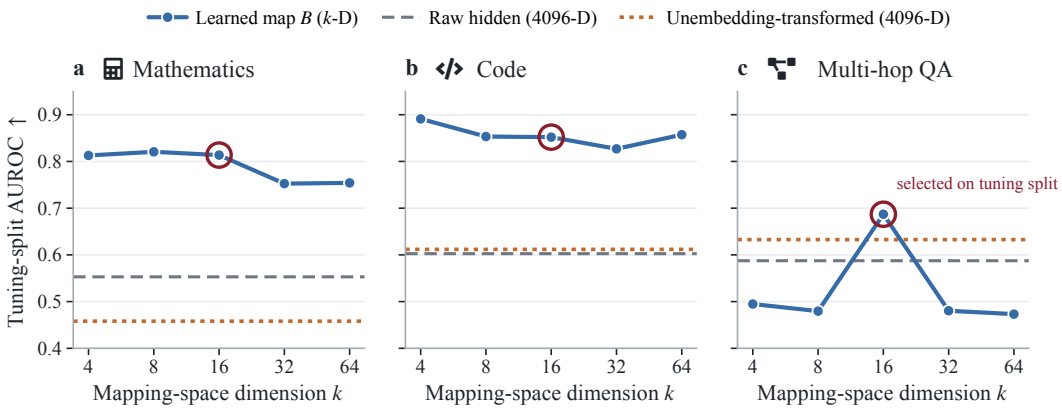


Figure 4: **Representation-space ablation on the Qwen3-8B Think 20% tuning subset.** Panels show per-domain macro-average five-fold out-of-fold AUROC; circles mark the selected $k = 16$.

5 Limitations

CUBE reads hidden states and the unembedding matrix, which makes it difficult to apply to closed models served only through APIs. Our theory also relies on simplifying assumptions about depth-wise dynamics that trained transformers need not satisfy exactly. The controlled failure analysis separates response classes but does not compare quality among correct solutions. The post-training comparison reports three-seed aggregate means without seed-level dispersion, and the pooled appendix analysis lacks sample counts.

6 Conclusion

Inspired by the Rubik's cube, which can be solved by strategies of different quality, we operationalized three aspects of an LLM's internal process rather than judging it only by the final answer. CUBE follows all-token configurations across layers and measures how much motion becomes net progress (Detour Ratio), when motion occurs relative to a model's depth profile (Convergence Ratio), and whether early update scales avoid an unusually small lower tail (Early-Step Consistency). Their associations with correctness vary substantially by model, domain, and inference mode; in particular, the thinking configuration is near chance on mathematics and multi-hop QA. A separate analysis suggests that the same profile can distinguish some answer–reasoning mismatches. Finally, we explored two uses: CUBE-EVAL scores a supplied solution after supervised calibration from its internal geometry, and CUBE-SELECTOR uses a strong model's calibrated geometry to rank post-training data.

7 Ethics Statement

This work follows the ICLR Code of Ethics. It uses public datasets and involves no human subjects, animal experiments, or personally identifiable information. We follow the datasets' stated licenses and usage guidelines.

8 Reproducibility Statement

Appendix A documents the benchmarks (Table 3), the model checkpoints (Table 4), the construction of the reasoning subspace (Algorithm 1), the cross-validation protocol, and the data-selection, fine-tuning, and evaluation recipe of CUBE-SELECTOR. Appendix D states the assumptions and complete proofs of the theoretical results. Dataset releases, decoding, extraction, calibration, and numerical precision are specified explicitly, and every benchmark is publicly available. Given the extracted states and fitted maps, signature computation is deterministic. Code and data will be released with the anonymous artifact.

9 AI Usage Statement

During manuscript preparation, we used large language models to polish wording and check English grammar, spelling, and word choice. The authors developed the research ideas, methodology, theoretical analysis, and experimental design, reviewed all suggested edits, and take responsibility for the paper's content. The paper separately describes uses of language models in data construction and experimental analysis.

References

Alon Albalak, Yanai Elazar, Sang Michael Xie, Shayne Longpre, Nathan Lambert, Xinyi Wang, Niklas Muennighoff, Bairu Hou, Liangming Pan, Haewon Jeong, Colin Raffel, Shiyu Chang, Tatsunori Hashimoto, and William Yang Wang. A survey on data selection for language models. *arXiv preprint arXiv:2402.16827*, 2024.

Awni Altabaa, Siyu Chen, John Lafferty, and Zhuoran Yang. Unlocking out-of-distribution generalization in transformers via recursive latent space reasoning. *arXiv preprint arXiv:2510.14095*, 2025.

Tianyi Bai, Ling Yang, et al. Efficient pretraining data selection for language models via multi-actor collaboration. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025*, 2025.

Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*, 2023.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pp. 41–48, 2009.

Daniil A. Boiko, Robert MacKnight, Ben Kline, and Gabe Gomes. Autonomous chemical research with large language models. *Nature*, 624(7992):570–578, 2023.

Ernie Chang, Yang Li, et al. Automixer: Checkpoint artifacts as automatic data mixers. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025*, 2025.

Xi Chen, Aske Plaat, and Niki van Stein. How does chain of thought think? mechanistic interpretability of chain-of-thought reasoning with sparse autoencoding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2026. arXiv:2507.22928.

Nicholas E. Corrado, Julian Katz-Samuels, Adithya M. Devraj, Hyokun Yun, Chao Zhang, Yi Xu, Yi Pan, Bing Yin, and Trishul Chilimbi. Automixalign: Adaptive data mixing for multi-task preference optimization in llms. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025*, 2025.

Hamed Damirchi, Ignacio Meza De la Jara, Ehsan Abbasnejad, Afshar Shamsi, Zhen Zhang, and Javen Qinfeng Shi. Truth as a trajectory: What internal representations reveal about large language model reasoning. *arXiv preprint arXiv:2603.01326*, 2026.

GLM Team, Aohan Zeng, Bin Xu, et al. ChatGLM: A family of large language models from GLM-130B to GLM-4 all tools. *arXiv preprint arXiv:2406.12793*, 2024.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space. In *Conference on Language Modeling*, 2025. arXiv:2412.06769.

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

Hugging Face. Math-Verify: A robust mathematical expression evaluation system, 2025a. URL `https://github.com/huggingface/Math-Verify`.

Hugging Face. Open R1: A fully open reproduction of DeepSeek-R1, January 2025b. URL `https://github.com/huggingface/open-r1`.

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

Yaning Jia, Chunhui Zhang, Xingjian Diao, Xiangchi Yuan, Zhongyu Ouyang, and Soroush Vosoughi. What makes a good curriculum? disentangling the effects of data ordering on llm mathematical reasoning. *arXiv preprint arXiv:2510.19099*, 2025.

Xinyan Jiang, Ninghao Liu, Di Wang, and Lijie Hu. Beyond scalars: Evaluating and understanding LLM reasoning via geometric progress and stability. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. arXiv:2603.10384.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. SWE-bench: Can language models resolve real-world GitHub issues? In *International Conference on Learning Representations*, 2024.

Abhinav Joshi, Divyanshu Bhatt, and Ashutosh Modi. Geometry of decision making in language models. In *Advances in Neural Information Processing Systems*, 2025. arXiv:2511.20315.

David Joyner. *Adventures in Group Theory: Rubik's Cube, Merlin's Machine, and Other Mathematical Toys*. Johns Hopkins University Press, 2nd edition, 2008.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Jisu Kim and Juhwan Lee. Strategic data ordering: Enhancing large language model performance through curriculum learning. *arXiv preprint arXiv:2405.07490*, 2024.

Xize Liang, Lin Yang, Jie Wang, Yiyang Lu, Runyu Wu, Hanzhu Chen, and Jianye Hao. Boosting multi-domain fine-tuning of large language models through evolving interactions between samples. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*. OpenReview.net, 2025.

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

Qian Liu, Xiaosen Zheng, Niklas Muennighoff, Guangtao Zeng, et al. Regmix: Data mixture as regression for language model pre-training. In *Proc. ICLR*, 2025.

Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. In *International Conference on Learning Representations*, 2024.

Llama Team. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Wenquan Lu, Yuechuan Yang, Kyle Lee, Yanshu Li, and Enqi Liu. Latent chain-of-thought? decoding the depth-recurrent transformer. In *First Workshop on the Application of LLM Explainability to Reasoning and Planning at COLM*, 2025. arXiv:2507.02199.

Zheheng Luo, Xin Zhang, et al. Velocitune: A velocity-based dynamic domain reweighting method for continual pre-training. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025*, 2025.

Mathematical Association of America. 2024 AIME I and AIME II. MAA American Mathematics Competitions, 2024.

Chen Qian, Dongrui Liu, Haochen Wen, Zhen Bai, Yong Liu, and Jing Shao. Demystifying reasoning dynamics with mutual information: Thinking tokens are information peaks in LLM reasoning. In *Advances in Neural Information Processing Systems*, 2025. arXiv:2506.02867.

Qwen Team. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M. Pawan Kumar, Emilien Dupont, Francisco J. R. Ruiz, Jordan S. Ellenberg, Pengming Wang, Omar Fawzi, Pushmeet Kohli, and Alhussein Fawzi. Mathematical discoveries from program search with large language models. *Nature*, 625(7995):468–475, 2024.

Julian Schnitzler, Xanh Ho, Jiahao Huang, Florian Boudin, Saku Sugawara, and Akiko Aizawa. MoreHopQA: More than multi-hop reasoning. *arXiv preprint arXiv:2406.13397*, 2024.

Adam S. Shai, Sarah E. Marzen, Lucas Teixeira, Alexander Gietelink Oldenziel, and Paul M. Riechers. Transformers represent belief state geometry in their residual stream. In *Advances in Neural Information Processing Systems*, 2024. arXiv:2405.15943.

Mustafa Shukor, Louis Bethune, Dan Busbridge, David Grangier, Enrico Fini, Alaaeldin El-Nouby, and Pierre Ablin. Scaling laws for optimal data mixtures. *arXiv preprint arXiv:2507.09404*, 2025.

Daouda Sow, Herbert Woisetschläger, et al. Dynamic loss-based sample reweighting for improved large language model pretraining. In *The Thirteenth International Conference on Learning Representations, ICLR 2025*, 2025.

Lihao Sun, Hang Dong, Bo Qiao, Qingwei Lin, Dongmei Zhang, and Saravan Rajmohan. LLM reasoning as trajectories: Step-specific representation geometry and correctness signals. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics*, 2026. arXiv:2604.05655.

Xingwu Sun, Yanfeng Chen, Yiqing Huang, et al. Hunyuan-Large: An open-source MoE model with 52 billion activated parameters by tencent. *arXiv preprint arXiv:2411.02265*, 2024.

Anvith Thudi, Evianne Rovers, Yangjun Ruan, Tristan Thrush, and Chris J. Maddison. Mixmin: Finding data mixtures via convex minimization. In *Forty-second International Conference on Machine Learning, ICML 2025*. OpenReview.net, 2025.

Jiachen T. Wang, Tong Wu, Dawn Song, Prateek Mittal, and Ruoxi Jia. GREATS: online selection of high-quality data for LLM training in every iteration. In *Advances in Neural Information Processing Systems 38, NeurIPS 2024*, 2024.

Alexander Wettig, Aatmik Gupta, Saumya Malik, and Danqi Chen. Qurating: Selecting high-quality data for training language models. In *International Conference on Machine Learning*, pp. 52915–52971. PMLR, 2024.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. LESS: Selecting influential data for targeted instruction tuning. In *International Conference on Machine Learning*, 2024.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy S Liang, Quoc V Le, Tengyu Ma, and Adams Wei Yu. Doremi: Optimizing data mixtures speeds up language model pretraining. In *Advances in Neural Information Processing Systems*, volume 36, pp. 69798–69818. Curran Associates, Inc., 2023a.

Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy Liang. Data selection for language models via importance resampling. In *Advances in Neural Information Processing Systems*, volume 36, pp. 34201–34227, 2023b.

Wanyun Xie, Francesco Tonin, and Volkan Cevher. Chameleon: A flexible data-mixing framework for language model pretraining and finetuning. In *Forty-second International Conference on Machine Learning, ICML 2025*. OpenReview.net, 2025.

Yang Yu, Kai Han, Hang Zhou, Yehui Tang, Kaiqi Huang, Yunhe Wang, and Dacheng Tao. LLM data selection and utilization via dynamic bi-level optimization. In *Forty-second International Conference on Machine Learning, ICML 2025*, 2025a.

Zichun Yu, Spandan Das, and Chenyan Xiong. Mates: Model-aware data selection for efficient pretraining with data influence models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 108735–108759. Curran Associates, Inc., 2024.

Zichun Yu, Fei Peng, Jie Lei, Arnold Overwijk, Wen tau Yih, and Chenyan Xiong. Group-level data selection for efficient pretraining. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025b.

Wenxuan Zhou, Ravi Agrawal, Shujian Zhang, Sathish Reddy Indurthi, Sanqiang Zhao, Kaiqiang Song, Silei Xu, and Chenguang Zhu. WPO: enhancing RLHF with weighted preference optimization. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024*, pp. 8328–8340. Association for Computational Linguistics, 2024.

Yufa Zhou, Yixiao Wang, Xunjian Yin, Shuyan Zhou, and Anru R. Zhang. The geometry of reasoning: Flowing logics in representation space. In *International Conference on Learning Representations*, 2026. arXiv:2510.09782.

Andrew Zhu, Alyssa Hwang, Liam Dugan, and Chris Callison-Burch. FanOutQA: A multi-hop, multi-document question answering benchmark for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 18–37, 2024.

Terry Yue Zhuo, Minh Chien Vu, Jenny Chim, Han Hu, Wenhao Yu, Ratnadira Widyasari, Imam Nur Bani Yusuf, Haolan Zhan, Junda He, Indraneil Paul, et al. BigCodeBench: Benchmarking code generation with diverse function calls and complex instructions. In *International Conference on Learning Representations*, 2025.

Appendix Contents

A Experimental Details 14
A.1 Datasets 14
A.2 Baselines 15
A.3 Model Configuration 16
A.4 Implementation and Evaluation Protocol 16

B More Related Work 19
B.1 Comparison with Reasoning-Dynamics Characterizations 19
B.2 Data Selection Strategies for LLMs 20

C More Results 21
C.1 Pooled Correlation Analysis 21
C.2 Geometry and Hallucinated Reasoning 21
C.3 Scaling on Mathematical Reasoning 22

D More Theoretical Analysis and Proofs 23
D.1 Discrete Depth-Wise Increment Model 23
D.2 Proof of Theorem 3.1 23
D.3 Corollaries for Signature Interpretation 24
D.4 What Early-Step Consistency Measures 24

E Notation 26

A Experimental Details

A.1 Datasets

We validate our geometric signatures on six benchmarks that cover three reasoning categories: mathematical reasoning, code generation, and multi-hop question answering. Table 3 reports the evaluated size, scoring metric, and prompt-length statistics of each benchmark. We use the public releases of MATH-500, AIME 2024, LiveCodeBench v6, BigCodeBench, MoreHopQA, and FanOutQA. In the correlational analyses, stratified five-fold cross-validation ensures that the subspace, standardization, and predictor are fit on the outer-training partition and frozen for the held-out fold. Thus, no sample's label enters the pipeline that scores it, although the same benchmark pool supplies training examples. FanOutQA uses its 310-example public development split in the closed-book setting because its test labels are not public. The post-training evaluation additionally uses 675 English, text-only competition-mathematics problems derived from OlympiadBench (He et al., 2024).

Input construction. Each record is converted to a single-turn prompt using the evaluated model's official chat template. The task content is held fixed across models; only tokenization and model-specific control tokens differ. We remove duplicate identifiers and exact duplicate prompts before evaluation, preserve the original problem statement, and do not truncate examples to equalize length. Padding positions and chat-template control tokens are excluded from token-level aggregates. The reported prompt lengths count whitespace-delimited words before model-specific tokenization. For generated-response analyses, we draw one completion with temperature 0.6, top-p 0.95, and a limit of $\min(32{,}768, \text{remaining context})$ new tokens under random seed 0. For mathematics and QA, the common task suffix asks the model to reason step by step and then end with "Final answer:";

Table 3: Evaluation benchmarks used in our experiments, grouped by reasoning category. #Eval is the number of evaluated samples, and the last two columns report the average and maximum prompt length in words. †Used only for the post-training evaluation in Table 2.

Category	Benchmark	#Eval	Metric	Avg len	Max len
Math	MATH-500	500	EM Acc.	32	206
	AIME 2024	30	EM Acc.	54	123
	OlympiadBench†	675	EM Acc.	53	711
Code	LiveCodeBench v6	1,055	pass@1	273	847
	BigCodeBench (Instr.)	1,140	pass@1	99	442
Multi-hop	MoreHopQA	1,118	Norm. EM	247	844
	FanOutQA (dev)	310	Loose acc.	18	37

code uses "Final code:" followed by one executable Python block. The measured reasoning span ends immediately before that delimiter. If the delimiter is absent, including because generation reaches the length limit, we retain the entire generated span for geometry but assign an incorrect outcome because no parseable answer was produced. Qwen thinking mode is enabled only for the configuration explicitly labeled "Think"; all other runs use the checkpoint's ordinary chat mode.

Outcome labels and scoring. For mathematical reasoning, we extract the final answer, normalize superficial formatting, and compare it with the reference by exact match. Code tasks use one generated completion per problem and the benchmark's official execution harness, giving pass@1. For multi-hop QA, we use normalized exact match for MoreHopQA and loose string accuracy for FanOutQA. These scores define correctness and enter all cross-benchmark aggregates. Geometry is computed before reading these labels; labels are used only for the correlational analysis, fitting the supervised subspace on the training fold, and conventional benchmark evaluation.

For Table 1, the mathematics-domain analysis pools the 30 AIME 2024 items with the 500 MATH-500 items before splitting. A benchmark-only statistic is omitted whenever either outcome class is absent or the fit is otherwise unidentified. Where benchmark scores are aggregated for model-level summaries, we macro-average them so that the larger benchmark does not dominate through sample count. The scaling curve in Appendix C.3 instead uses a micro-average over all 530 mathematics items.

A.2 Baselines

We compare CUBE with baselines that differ in both the information they consume and the cost at which that information is obtained. All selection methods receive the same candidate pool and return exactly the same number of examples. A method may therefore change *which* examples are used, but not the amount of post-training data or the optimization budget.

Data-selection baselines. **Random** draws a uniform subset without replacement, with a different draw for each of the three seeds. **Perplexity** ranks candidates by the target model's own perplexity, the exponentiated mean token-level negative log likelihood over response tokens with prompt and padding positions masked, and keeps the highest-perplexity candidates. **DEITA** (Liu et al., 2024) scores each candidate with the released DEITA complexity and quality scorers and adds candidates in descending order of the product of the two scores, skipping any candidate whose cosine similarity to an already selected example exceeds 0.9 under the target model's mean-pooled final-layer embeddings. **LESS** (Xia et al., 2024) follows its default configuration: a LoRA warmup on a random 5% of the pool for four epochs, 8,192-dimensional random projections of per-example gradients, and ranking by gradient similarity to the 2,500-example validation split of Appendix A.4, which is disjoint from every evaluation benchmark. This compact set covers uninformed, loss-based, quality-and-diversity, and gradient-influence selection.

Representation ablation. **Raw** computes the three signatures directly from the 4096-dimensional all-token hidden trajectory, whereas **Unembedding-transformed** first applies the square root of the

Table 4: **Model configurations and experiment assignments.** Architecture and unique-parameter counts follow official statements; tied embeddings are counted once. L, D, D_{ff}, H_q/H_{kv}, D_h, V, and Ctx. denote transformer blocks (equivalently, layer transitions), hidden size, feed-forward width, query and key–value heads, head dimension, vocabulary size, and native context length, respectively. Target and Strong denote CUBE-SELECTOR roles. Corr., Eval, Scal., and Hall. indicate use in the correlational analysis, CUBE-EVAL, scaling check, and hallucination analysis, respectively; see Table 1, Section 4.3, and Appendices C.2 and C.3. *Total / activated parameters per token. †Per-expert width; each MoE layer has 256 routed experts and one shared expert, with six routed experts active per token. ‡The thinking and non-thinking modes share weights; Corr. evaluates each mode separately, the Selector and the Figure 4 tuning sweep use thinking mode, and CUBE-EVAL and the scaling check use non-thinking mode.

		Architecture							Experiments				
Checkpoint	**Params.**	L	D	D_{ff}	H_q/H_{kv}	D_h	V	**Ctx.**	**Corr.**	**EVAL**	**Scal.**	**Selector**	**Hall.**
Qwen3													
Qwen3-0.6B	0.60B	28	1024	3072	16/8	128	151.9K	32K	–	–	✓	–	–
Qwen3-1.7B	1.72B	28	2048	6144	16/8	128	151.9K	32K	–	–	✓	–	–
Qwen3-8B (think / non-think)‡	8.19B	36	4096	12288	32/8	128	151.9K	32K	✓	✓	✓	**Target**	–
Llama 3													
Llama-3.2-1B-Instruct	1.24B	16	2048	8192	32/8	64	128.3K	128K	–	–	✓	–	–
Llama-3.2-3B-Instruct	3.21B	28	3072	8192	24/8	128	128.3K	128K	–	–	✓	–	–
Llama-3.1-8B-Instruct	8.03B	32	4096	14336	32/8	128	128.3K	128K	✓	✓	✓	–	–
Hunyuan													
Hunyuan-0.5B-Instruct	0.54B	24	1024	3584	16/8	128	120.8K	256K	–	–	✓	–	–
Hunyuan-1.8B-Instruct	1.79B	32	2048	6144	16/4	128	120.8K	256K	–	–	✓	–	–
Hunyuan-7B-Instruct	7.50B	32	4096	14336	32/8	128	128.2K	256K	✓	–	✓	**Target**	–
GLM													
GLM-4-9B-Chat	9.40B	40	4096	13696	32/2	128	151.6K	128K	✓	–	–	–	–
DeepSeek													
V4-Flash (MoE)	284B / 13B*	43	4096	2048†	64/1	512	129.3K	1M	–	–	–	**Strong**	✓

frozen unembedding Gram matrix. We compare these fixed baselines with the shared projection B, learned independently at each $k \in \{4, 8, 16, 32, 64\}$. The data splits, contrastive objective, signature definitions, and downstream classifier are held fixed.

A.3 MODEL CONFIGURATION

Model usage differs by experiment. The correlational analysis uses the largest listed Qwen, Llama, Hunyuan, and GLM checkpoints; CUBE-EVAL uses Qwen3-8B and Llama-3.1-8B, and its scaling check uses three sizes each of Qwen3, Llama 3, and Hunyuan. CUBE-SELECTOR distinguishes the fine-tuned target models from the strong model, DeepSeek-V4-Flash, which the hallucination analysis also uses. DeepSeek-V4-Pro serves only as a text generator for the declarative rewrites of CUBE-EVAL and the response variants of Appendix C.2; we never read its hidden states, so it is not listed in Table 4. Table 4 records the exact checkpoints, their architectures and parameter counts, selector roles, and experiment assignments. Thinking and non-thinking modes are treated as separate configurations because they induce different computation traces even when they share weights. All checkpoints are evaluated without parameter updates unless the experiment explicitly concerns post-training data selection.

A.4 IMPLEMENTATION AND EVALUATION PROTOCOL

Hidden-state extraction. Models run in evaluation mode. We retain the residual stream of every non-padding token in the reasoning sequence at every transformer block. For CUBE-EVAL, this covers the fixed reasoning paragraph; for CUBE-SELECTOR, it covers the trajectory generated while the strong model solves each OpenR1 problem; for the other generated-solution analyses, it covers the chain-of-thought preceding the answer segment defined in Appendix A.1. We apply no additional normalization: the states are the layer-wise hidden states, so the deepest returned state includes the architecture's native terminal normalization. The final transition remains part of the endpoint and path used by $\mathcal{D}$ but is omitted from the early/late comparison in $\mathcal{C}$. Hidden states are extracted once

Algorithm 1 Construction of the Reasoning Subspace

Require: $\{h_{i,t}^{(\ell)}\}$, $\{y_i\}$, (U, b_{out}), k, τ, η, and E

Ensure: Projection B and coordinates $\{\mathbf{Z}_i^{(\ell)}\}$

Step 1: Construct the shared output-space dataset

$\mathcal{D}_{\text{out}} \leftarrow \{(h_{i,t}^{(\ell)} U + b_{\text{out}}, y_i)\}_{i,\ell,t}$

Initialize $B \in \mathbb{R}^{V \times k}$

Step 2: Learn a low-rank outcome-discriminative map

for $e = 1, \ldots, E$ **do**

  for each label-balanced $\mathcal{B} \subset \mathcal{D}_{\text{out}}$ **do**

    $z_a \leftarrow o_a B, \quad \bar{z}_a \leftarrow z_a / \|z_a\|_2$ for all $(o_a, y_a) \in \mathcal{B}$

    $B \leftarrow B - \eta \nabla_B \mathcal{L}_{\text{con}}$

  end for

end for

Step 3: Generate shared reasoning coordinates

for each sample i and layer ℓ **do**

  $\mathbf{Z}_i^{(\ell)} \leftarrow [(h_{i,t}^{(\ell)} U + b_{\text{out}}) B]_{t=0}^{T_i - 1}$

end for

return B and $\{\mathbf{Z}_i^{(\ell)}\}$

and cached. The reference implementation uses PyTorch and Transformers; dense checkpoints run in bfloat16, while DeepSeek-V4-Flash uses its released mixed-precision weights. Cached states and subspace fitting use float32, and projected coordinates and signature accumulation use float64.

Fold-wise subspace fitting and leakage control. In each outer split, B is fit on the training partition and frozen for the held-out fold. Training states are sampled in a label-balanced manner so that long responses and the majority outcome do not dominate the contrastive objective. Projected states are ℓ_2-normalized inside the loss; their unnormalized coordinates are used for geometric measurement. Early-stopping validation, feature standardization, and downstream regression use only the outer-training partition; held-out labels are accessed only for evaluation. For a minibatch $\mathcal{B}$, let $\bar{z}_a = z_a / \|z_a\|_2$ and $P(a) = \{p \in \mathcal{B} \setminus \{a\} : y_p = y_a\}$. The fitted map minimizes

$$\mathcal{L}_{\text{con}} = -\frac{1}{|\mathcal{B}|} \sum_{a \in \mathcal{B}} \frac{1}{|P(a)|} \sum_{p \in P(a)} \log \frac{\exp(\bar{z}_a^\top \bar{z}_p / \tau)}{\sum_{q \in \mathcal{B} \setminus \{a\}} \exp(\bar{z}_a^\top \bar{z}_q / \tau)}.$$

The main analyses use $k = 16$. Hyperparameters for learning B (τ, η, E, batch size, optimizer, initialization, and early-stopping rule) are $\tau = 0.07$, $\eta = 10^{-3}$, at most 20 epochs, and label-balanced batches of 256 sampled states. We sample at most 64 layer–token states from each calibration trajectory, initialize $B_{jk} \sim \mathcal{N}(0, 1/V)$, and optimize with AdamW without weight decay. A sample-level 10% split is used for early stopping with patience three and a minimum improvement of 10^{-6}; the base seed is 0 and fold f uses seed f. Figure 4 reports the sweep over $k \in \{4, 8, 16, 32, 64\}$ on a 20% Qwen3-8B Think tuning subset. We selected $k = 16$ on this subset before running the main cross-validation analysis on the full pool, including the tuning examples. Algorithm 1 summarizes the three-stage construction of the reasoning subspace described in Section 3.1.

Signature computation. We compute $\mathcal{D}$, $\mathcal{C}$, and $\mathcal{F}$ from the sequence-level coordinates exactly as defined in Section 3.2. The Convergence Ratio uses the first and last third of the available transitions and excludes the final transition. Early-Step Consistency uses only the first third of the transitions and counts steps at or above the fixed threshold $\alpha = 0.7$ times the median step length within that same early window. It is a relative lower-tail statistic and does not measure absolute step magnitude. Samples with zero endpoint displacement are undefined for $\mathcal{D}$ and are omitted from analyses involving that signature. Likewise, samples with zero mean motion in the late window are undefined for $\mathcal{C}$ and are omitted from analyses involving that signature.

Predictive and statistical evaluation. The main correlational analysis groups samples by model and domain and uses stratified five-fold cross-validation. Within every fold, the projection B, the

Table 5: **Random-forest scorer for CUBE-EVAL.** This classifier is the final sample scorer.

Model	Implementation	Key parameters	Role
Random forest	scikit-learn	100 trees, maximum depth 5, random seed 42	Primary scorer

mean and variance used to standardize each signature, and the joint unregularized logistic regression are fit only on the training partition. We report out-of-fold AUROC and the fold-wise mean and standard deviation of each coefficient; both quantities come from the same five fitted models. A fit is suppressed when the outcome is constant or otherwise unidentified. The hallucination analysis likewise uses stratified five-fold cross-validation and reports out-of-fold AUROC.

CUBE-EVAL. DeepSeek-V4-Pro rewrites each item of the mathematics and multi-hop QA benchmarks, together with its reference solution, into a self-contained declarative paragraph that states the final answer, and the evaluated model reads each paragraph in a single teacher-forced pass. The rewrite instruction is: "Using the problem and verified solution below, write one self-contained declarative paragraph that preserves the essential reasoning and explicitly states the final answer. Do not ask a question, use bullet points, or add meta-commentary." We decode DeepSeek-V4-Pro greedily with a 4,096-token limit and regenerate up to three times if the paragraph's answer fails the benchmark's normalizer and checker. The accepted corpus is then frozen across evaluated models.

For each evaluated model and domain, we reuse the five outer folds of the generated-solution analysis. In fold f, B is fit on the other four folds' generated trajectories and correctness labels, then applied to the held-out items' declarative paragraphs; consequently, a paragraph is never scored by a map trained on that item's label. The sample score is the predicted score of a random forest trained on the geometric signatures (Table 5). Figure 3 correlates the eight model–benchmark aggregates of these scores with gold-answer accuracy. The scaling curve in Appendix C.3 uses the same 530 pooled MATH-500 and AIME 2024 items but reports conventional answer accuracy rather than a geometric score.

CUBE-SELECTOR. *Data.* We begin with the default subset of OpenR1-Math-220K (Hugging Face, 2025b), remove duplicate normalized prompts, and decontaminate before sampling. Following the Open-R1 utility, text is lowercased and whitespace-collapsed; a candidate is removed if any word-level 8-gram in its problem appears in MATH-500, AIME 2024, or the 675-item OlympiadBench evaluation collection. From the survivors we draw 25,000 problems uniformly with seed 42, pair each with one verified-correct DeepSeek-R1 solution, and split them 8:1:1 into a 20,000-example candidate pool, a 2,500-example validation split, and a 2,500-example test split. The validation split is used to fit and tune the CUBE scorer; the test split is used only for debugging. Every selector chooses exactly 6,000 candidates (30%) from the same pool.

Labels and scoring. DeepSeek-V4-Flash first solves every validation problem in Think-High mode with one completion, temperature 1.0, top-p 1.0, at most 32,768 new tokens, and seed 0. A generated trajectory is labeled positive if its final answer matches the reference under Math-Verify (Hugging Face, 2025a) and negative otherwise. From these labeled OpenR1 trajectories, we learn B with Algorithm 1 ($k = 16$), compute $\mathcal{D}$, $\mathcal{C}$, and $\mathcal{F}$, and fit a random-forest classifier to distinguish correct from incorrect reasoning. Five-fold cross-validation within the validation split selects the scorer hyperparameters and early-stopping point; after selection, B and the random forest are refit on all 2,500 validation trajectories. The strong model then generates a trajectory for every candidate under the same protocol. The random forest's positive-class score is the candidate's CUBE score, and CUBE-SELECTOR keeps the 6,000 highest-scoring candidates. A low score marks a sample whose trajectory resembles the strong model's failures.

Fine-tuning. Every subset is trained with the same LoRA supervised fine-tuning recipe (Table 6). Examples use each model's official chat template with the reasoning trace inside its native thinking block, so Qwen3-8B is trained and evaluated in thinking mode. Data order and optimization budget are identical across selectors, and selection scores are computed before fine-tuning without any evaluation label.

Table 6: **LoRA supervised fine-tuning for CUBE-SELECTOR.** The same configuration is used for every selector and both target models. The global batch size is $8 \times 1 \times 8 = 64$. A maximum length of 16,384 tokens covers about 98% of OpenR1-Math (99th percentile ≈17,000 tokens). LoRA is applied to the query, key, value, and output projections of each attention block and to the gate, up, and down projections of each MLP.

Setting	**Value**
GPUs	8
Per-device batch size	1
Gradient accumulation steps	8
Epochs	1
Maximum sequence length	16,384
Learning rate	2×10^{-4}
Schedule	cosine
Warmup ratio	0.03
Weight decay	0
Optimizer	AdamW
LoRA rank r_{LoRA}	16
LoRA scaling α_{LoRA}	32
LoRA dropout	0.05
LoRA target modules	attention and MLP projections
Seeds	42, 43, 44
Precision	bf16
Gradient checkpointing	enabled
Data-loader workers	4
Logging interval	10 steps
Checkpoint interval	500 steps
Checkpoints retained	2

Evaluation. Each selector is fine-tuned and evaluated with three random seeds, and Table 2 reports their aggregate means. Decoding uses temperature 0.6, top-p 0.95, and at most 32,768 new tokens, and answers are checked with Math-Verify. MATH-500 and the 675-item OlympiadBench collection take one sample per problem per seed; AIME 2024, with only 30 problems, averages 16 samples per problem per seed.

Reproducibility checklist. The reference pipeline writes the checkpoint revision, tokenizer and chat template, benchmark version, prompt, normalized prediction, outcome label, selected token index, per-layer signature values, random seed, and split identifier.

B MORE RELATED WORK

B.1 COMPARISON WITH REASONING-DYNAMICS CHARACTERIZATIONS

Table 7 organizes the reasoning-dynamics characterizations reviewed in Section 2 by the axis along which evolution is measured and the space in which the evolving state is read out; all of them, like CUBE, read the residual stream. Most reduce each step or layer to a single point, such as a hidden state, a vocabulary distribution, or a mutual-information estimate, which says little about how the geometry of the whole token configuration is reshaped from one layer to the next. Truth as a Trajectory is the closest exception: it uses the layer-wise displacements of all tokens, but unrolls them into a single sequence for a learned LSTM classifier rather than characterizing their geometry (Damirchi et al., 2026). We instead keep the token configuration intact, follow it jointly across layers in a shared low-dimensional reasoning subspace, and summarize its motion with interpretable signatures. Under the caption's taxonomy, this is depth-wise rather than token-wise dynamics: all tokens form the state, but the evolution axis is layer depth. The tracked configuration is a point in $\mathbb{R}^{T_i \times k}$ rather than a subspace-valued state, and the construction passes through the vocabulary

Table 7: **Taxonomy of existing reasoning dynamics characterizations in LLMs.** Token-wise covers evolution along a generated sequence; depth-wise covers evolution across layers or inference-time recurrence loops. Subspace marks methods whose tracked object is itself a subspace or region; methods that project each step to a single point, such as TRACED, fall under Hidden State.

	Dynamical Axis		Readout Space			
Method	Token-wise	Depth-wise	Hidden State	Vocabulary	Sparse Feature	Subspace
TRACED	✓		✓			
Flowing Logics	✓		✓			
Truth as Traj.	✓	✓	✓			
MI-Peaks	✓		✓			
SAE Decomp.	✓				✓	
Step-Subspaces	✓					✓
Tuned Lens		✓		✓		
Geom. of Dec.		✓				✓
Belief State		✓				✓
COCONUT		✓	✓			
Recursive Latent		✓	✓			
Latent CoT		✓		✓		
Ours		✓	✓	✓		

parameterization before the learned projection; the corresponding checkmarks follow these literal definitions.

B.2 Data Selection Strategies for LLMs

Appendix B.1 characterizes reasoning as an evolving geometric object. The natural next question is which data to train on, a choice that is decisive for LLM quality. The overwhelming majority of this literature targets the pre-training stage and frames the problem as data mixing, seeking the optimal proportions of macroscopic sources such as web text, code, and academic prose, and moving beyond hand-tuned heuristics toward optimization-driven frameworks that Kaplan et al. (2020) motivate through scaling laws. Xie et al. (2023a) train a small proxy model to derive domain weights through distributionally robust optimization, after which Liu et al. (2025) recast mixing as regression over micro-models and Thudi et al. (2025) formulate it as convex minimization, further showing that mixtures found on small models remain scale-invariant. To curb retraining cost, Xie et al. (2025) reweight domains via leverage scores in a learned embedding space, Luo et al. (2025) adapt weights by learning velocity for domain-adaptive continual pre-training, and Shukor et al. (2025) establish scaling laws for the optimal mixture itself.

A finer-grained line scores individual pre-training instances rather than whole domains. Building on static importance resampling and quality scoring of Xie et al. (2023b) and Wettig et al. (2024), the field has shifted toward dynamic, online mechanisms. Wang et al. (2024) greedily select high-quality batches in every iteration through a Taylor expansion of the loss, while Sow et al. (2025) reweight samples online by their loss to focus training on the most informative examples. Closer to influence-based reasoning, Yu et al. (2024) and Chang et al. (2025) estimate per-sample value from training artifacts, an idea that Yu et al. (2025b) extend to capture group-level interactions among examples. Still other work casts selection as bi-level optimization of a data-weighting model (Yu et al., 2025a) or reconciles heterogeneous criteria through multi-actor collaboration (Bai et al., 2025), and broader surveys map this design space (Albalak et al., 2024). Comparatively little attention has been paid to the post-training stage, where selection largely reduces to curriculum-style ordering of fine-tuning data (Bengio et al., 2009; Kim & Lee, 2024; Jia et al., 2025) or to reweighting task and preference data (Liang et al., 2025; Zhou et al., 2024; Corrado et al., 2025). Despite this breadth, these criteria rest overwhelmingly on external or outcome-level signals such as answer correctness, task difficulty, reward scores, and loss-based uncertainty, which reveal whether a sample is hard or useful in expectation but not whether it elicits meaningful reasoning in the model's current state, while the dynamic variants further incur heavy second-order or bi-level optimization overhead. We therefore depart from this paradigm by scoring each post-training sample through a strong model's calibrated

internal trajectory, yielding a state-dependent criterion that reads how the sample is processed rather than only whether it is hard, grounded in the reasoning-dynamics literature of Section 2.

C More Results

C.1 Pooled Correlation Analysis

Takeaway. *The pooled descriptive fit yields positive coefficients in mathematics and code, matching the predominant within-model direction.*

As an extension of the model-specific analysis in Table 1, we also pool samples across models within each reasoning domain and fit a three-factor logistic regression jointly on the standardized $\mathcal{D}$, $\mathcal{C}$, and $\mathcal{F}$ signatures. Table 8 reports the resulting coefficients and in-sample discrimination. The mathematics and code coefficients on $\mathcal{C}$ are positive, matching the predominant within-model direction in Table 1.

Table 8: **Pooled descriptive logistic regression of correctness on the three signatures.** Predictors are z-scored within the pooled sample. Each cell is the standardized coefficient ($\times 10^{-1}$; two-sided p); AUC is the in-sample discriminative accuracy of the three-factor model. Stars: $^{*}p < 0.05$, $^{**}p < 0.01$, $^{***}p < 0.001$.

Domain	$\mathcal{D}$	$\mathcal{C}$	$\mathcal{F}$	**AUC**
Math	$+1.021^{***}$	$+0.325^{***}$	$+0.232^{***}$	0.727
Code	$+0.340^{***}$	$+0.553^{***}$	$+0.341^{***}$	0.613
QA	$+0.203^{***}$	$+0.092$	$+0.008$	0.597

C.2 Geometry and Hallucinated Reasoning

Takeaway. *Fluent answer–reasoning mismatches are less roundabout than ordinary procedural errors, but show the strongest late-layer instability and the lowest early-step consistency. This distinct profile shows that internal geometry can separate unsupported conclusions from both sound reasoning and conventional procedural mistakes.*

Dataset and protocol. We construct OpenR1-CoT-Quality-10K from the training split of OpenR1-Math-220K (Hugging Face, 2025b), retaining one verified-correct chain of thought per problem. DeepSeek-V4-Pro then forms controlled response types: the unchanged solution, a refined correct rewrite, a correct but detoured rewrite, a rewrite containing a genuine reasoning error, and an otherwise coherent rewrite whose final answer contradicts or unjustifiably departs from its reasoning. A shared judge rubric first assesses answer correctness, reasoning validity, and solution quality; its labels are then merged into **Correct**, **Hallucinated** (fluent reasoning followed by an unsupported final answer), and **Other incorrect**. We probe DeepSeek-V4-Flash on the resulting collection and compute the three signatures from the full all-token, layer-wise trajectory defined in Section 3.1. The **Correct** aggregate combines unchanged source solutions with refined and deliberately detoured rewrites, whereas the two failure groups contain rewrites. Consequently, source and style are confounded with the three-way label. Merging the two correct rewrites also precludes a refined-versus-detoured comparison. Tables report aggregate group statistics and single-feature AUCs; the multifeature column reports five-fold means and standard deviations.

Results. The descriptive statistics in Table 9 reveal two different failure profiles. Ordinary incorrect responses follow the most roundabout paths, consistent with procedural errors accumulating throughout the solution. Hallucinated responses remain comparatively direct yet show the strongest late-layer motion and the lowest early-step consistency, suggesting locally coherent but unsettled processing near the unsupported answer. Five-fold standardized logistic regressions (Table 10) identify Early-Step Consistency as most informative for hallucinated versus correct responses and Con-

Table 9: **Layer-trajectory statistics by response group.** Each cell reports the group mean and one population standard deviation.

Signature	Correct	Hallucinated	Other incorrect
Roundaboutness $1/\mathcal{D}$	2.257 ± 0.390	2.405 ± 0.295	2.459 ± 0.368
Late-to-early motion $1/\mathcal{C}$	3.061 ± 0.396	3.236 ± 0.346	3.104 ± 0.317
Early-Step Consistency $\mathcal{F}$	0.666 ± 0.033	0.645 ± 0.022	0.652 ± 0.028

Table 10: **Discrimination from three layer-trajectory signatures.** Single-feature AUCs are oriented toward the positive class. The final column reports mean $\pm$ standard deviation across five folds.

Task	$\mathcal{D}$	$\mathcal{C}$	$\mathcal{F}$	All three
Hallucinated vs. Correct	0.644	0.673	**0.700**	0.720 ± 0.014
Hallucinated vs. Other incorrect	0.547	0.598	0.570	0.609 ± 0.015
Other incorrect vs. Correct	0.660	0.586	0.638	0.685 ± 0.006

vergence Ratio for hallucinated versus other incorrect responses; Detour Ratio is weak for the latter contrast. Their combination has the highest reported AUC in each comparison.

C.3 Scaling on Mathematical Reasoning

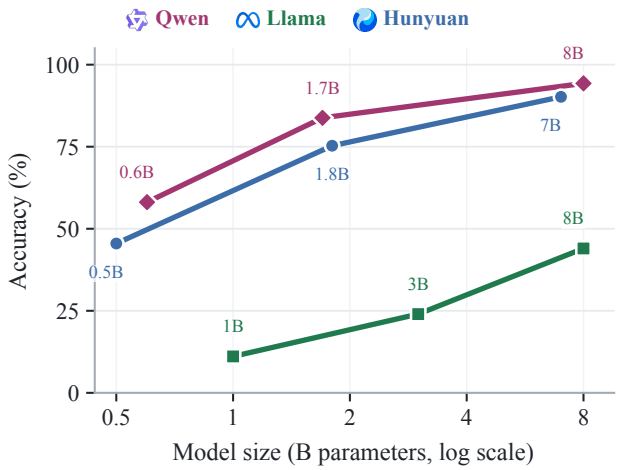


Figure 5: **Scaling on mathematical reasoning.** Micro-averaged gold-answer accuracy over the 530 pooled MATH-500 and AIME 2024 problems.

Figure 5 measures how mathematical accuracy scales across three checkpoints each of Qwen, Llama, and Hunyuan. Every checkpoint is scored on the same 530 MATH-500 and AIME 2024 items, and the plotted value is the micro-average, so MATH-500 carries weight $500/530$. Model-level summaries elsewhere macro-average the two benchmarks.

Within each family the curve increases with parameter count, and the increments differ. Qwen moves from 58.1% at 0.6B to 83.8% at 1.7B and 94.3% at 8B; Hunyuan moves from 45.5% at 0.5B to 75.3% at 1.8B and 90.2% at 7B. In both families the larger gain arrives on the first step. Llama moves from 11.1% at 1B to 24.0% at 3B and 44.0% at 8B, so its second step is the larger one, and its 8B checkpoint remains below Qwen-0.6B. Scale sorts checkpoints inside a family. At roughly 8B parameters the families stay apart: Qwen and Hunyuan are both above 90%, while Llama remains at 44.0%.

The curve establishes the checkpoints' gold-answer-accuracy ordering on the mathematics suite used throughout the paper; geometric signatures are analyzed separately.

D More Theoretical Analysis and Proofs

D.1 Discrete Depth-Wise Increment Model

For sample i, vectorize the sequence-level state as $v_i^{(\ell)} = \mathrm{vec}(\mathbf{Z}_i^{(\ell)}) \in \mathbb{R}^{d_i}$, where $d_i = T_i k$. This preserves the metric used by the signatures because $\|\Delta v_i^{(\ell)}\|_2 = \|\Delta \mathbf{Z}_i^{(\ell)}\|_F$.

Assumption D.1 (Discrete depth-wise increments)**.** *Let $g_0, \dots, g_{L-1} \overset{\text{iid}}{\sim} \mathcal{N}(0, I_{d_i})$. There are a unit direction u_i, nonnegative magnitudes m_i and σ_i with $m_i^2 + \sigma_i^2 > 0$, and positive depth weights $a_{i,0}, \dots, a_{i,L-1}$ such that*

$$\Delta v_i^{(\ell)} = a_{i,\ell}\left(m_i u_i + \frac{\sigma_i}{\sqrt{d_i}} g_\ell\right), \qquad \ell = 0, \dots, L-1.$$

For the relative-stabilization interpretation, we further factor $a_{i,\ell} = \lambda_\ell q_i^\ell$, where $\{\lambda_\ell > 0\}_{\ell=0}^{L-1}$ is a model-wide depth profile and $q_i \in (0, 1]$ is a sample-specific contraction or no-contraction factor.

A large m_i/σ_i means that the deterministic component dominates the update direction. Indeed, the deterministic part of layer ℓ is $a_{i,\ell} m_i u_i$, whereas σ_i controls the isotropic noise. Holding the depth weights fixed, increasing m_i/σ_i aligns the increments and moves $\mathcal{D}_i$ from its baseline $\sqrt{\rho_i}$ toward one. Separately, λ_ℓ specifies the model-wide depth profile: $q_i = 1$ follows that profile, while $q_i < 1$ attenuates later updates and yields $\mathcal{C}_i/\mathcal{C}_0 > 1$ at leading order for $L \geq 3$. Thus $\mathcal{D}_i$ measures directional coherence and $\mathcal{C}_i$ measures stabilization relative to the model-wide profile.

D.2 Proof of Theorem 3.1

In the remainder of this appendix, fix a sample i and suppress its index, writing d, $v^{(\ell)}$, u, m, σ, and a_ℓ for the corresponding quantities. There are exactly L observed increments $\Delta v^{(\ell)} = v^{(\ell+1)} - v^{(\ell)}$, $\ell = 0, \dots, L-1$, with lengths $s_\ell = \|\Delta v^{(\ell)}\|_2 = \|\Delta \mathbf{Z}_i^{(\ell)}\|_F$, so the analysis concerns the same sequence-level states and transitions used by all three signatures.

Proof of Theorem 3.1. Let $A = A_i = \sum_{\ell=0}^{L-1} a_\ell$ and $\rho = \rho_i = \sum_{\ell=0}^{L-1} a_\ell^2 / A^2$. We use two standard Gaussian bounds: for a unit vector u, $g \sim \mathcal{N}(0, I_d)$, and $0 < \epsilon < 1$,

$$\mathbb{P}\big(|\langle u, g\rangle| > \epsilon\sqrt{d}\big) \leq 2e^{-d\epsilon^2/2}, \qquad \mathbb{P}\big(\big|\|g\|_2^2/d - 1\big| > \epsilon\big) \leq 2e^{-d\epsilon^2/8}. \tag{1}$$

Each squared step norm expands as

$$s_\ell^2 = a_\ell^2\Big(m^2 + \frac{2m\sigma}{\sqrt{d}}\langle u, g_\ell\rangle + \frac{\sigma^2}{d}\|g_\ell\|_2^2\Big).$$

By equation 1 and a union bound over ℓ, $s_\ell = a_\ell\sqrt{m^2+\sigma^2}\,(1 + o_p(1))$ simultaneously for all ℓ, so the path length is $A\sqrt{m^2+\sigma^2}\,(1 + o_p(1))$. For the endpoint displacement, $G = \sum_{\ell=0}^{L-1} a_\ell g_\ell \sim \mathcal{N}(0, \rho A^2 I_d)$, and applying equation 1 to $G/(\sqrt{\rho}\,A)$ gives

$$\Big\|\sum_{\ell=0}^{L-1} \Delta v^{(\ell)}\Big\|_2^2 = A^2\Big(m^2 + \frac{2m\sigma}{A\sqrt{d}}\langle u, G\rangle + \frac{\sigma^2}{A^2 d}\|G\|_2^2\Big) = A^2\big(m^2 + \rho\sigma^2 + o_p(1)\big).$$

Dividing the endpoint displacement by the path length yields $\mathcal{D}_i = \sqrt{(m^2+\rho\sigma^2)/(m^2+\sigma^2)} + o_p(1)$.

When $\sigma > 0$, set $r = m/\sigma$. The square of the leading term is $f(r^2) = (r^2+\rho)/(r^2+1)$, and $f'(r^2) = (1-\rho)/(r^2+1)^2$. Because all a_ℓ are positive, $\rho < 1$ whenever $L \geq 2$. Thus $\mathcal{D}_i$ increases strictly from $\sqrt{\rho}$ at $r = 0$ toward one. The same conclusion is immediate at the endpoints $\sigma = 0$ or $m = 0$.

For the Convergence Ratio, simultaneous concentration of the step lengths and the equal cardinalities of $\mathcal{W}_E$ and $\mathcal{W}_L$ give

$$\mathcal{C}_i = \frac{\sum_{\ell \in \mathcal{W}_E} a_\ell}{\sum_{\ell \in \mathcal{W}_L} a_\ell}\big(1 + o_p(1)\big).$$

Now let $a_\ell = \lambda_\ell q^\ell$ and use the shared-profile baseline $\mathcal{C}_0$ defined in Theorem 3.1. Since $q^\ell \geq q^{w-1}$ in $\mathcal{W}_E$ and $q^\ell \leq q^{L-w-1}$ in $\mathcal{W}_L$,

$$\frac{\mathcal{C}_i}{\mathcal{C}_0} \geq q^{-(L-2w)}\big(1 + o_p(1)\big).$$

For $q \leq 1$, the leading-order bound is at least one. When $q < 1$ and $L \geq 3$, $L - 2w \geq 1$, so the leading-order bound is strictly larger than one, proving the relative-stabilization statement. □

D.3 Corollaries for Signature Interpretation

> **Takeaway.** *Both signatures are profile-relative: the Detour Ratio ranges from $\sqrt{\rho_i}$ to one, while the Convergence Ratio signals contraction through $\mathcal{C}_i/\mathcal{C}_0$, not through an absolute threshold at one.*

Corollary D.1 (Profile-relative interpretation)**.** *Under the conditions of Theorem 3.1, denote the leading terms of $\mathcal{D}_i$ and $\mathcal{C}_i$ by $\mathcal{D}_i^\star$ and $\mathcal{C}_i^\star$, respectively. If $\sigma_i > 0$, $\rho_i < 1$, and $r_i = m_i/\sigma_i$, then*

$$(\mathcal{D}_i^\star)^2 = \frac{r_i^2 + \rho_i}{r_i^2 + 1}, \qquad r_i^2 = \frac{(\mathcal{D}_i^\star)^2 - \rho_i}{1 - (\mathcal{D}_i^\star)^2}.$$

Hence $\mathcal{D}_i^\star \in [\sqrt{\rho_i}, 1)$ and uniquely determines r_i once ρ_i is fixed. Under $a_{i,\ell} = \lambda_\ell q_i^\ell$,

$$q_i = 1 \implies \frac{\mathcal{C}_i^\star}{\mathcal{C}_0} = 1, \qquad q_i < 1,\ L \geq 3 \implies \frac{\mathcal{C}_i^\star}{\mathcal{C}_0} \geq q_i^{-(L-2w)} > 1.$$

Thus sample-specific contraction is identified by an increase over $\mathcal{C}_0$; without a restriction on $\mathcal{C}_0$, it does not imply $\mathcal{C}_i^\star > 1$.

Proof. Dividing the leading expression for $\mathcal{D}_i^2$ in Theorem 3.1 by σ_i^2 gives the first identity; solving it for r_i^2 gives the second. The stated range follows from $r_i \in [0, \infty)$. For $\mathcal{C}_i^\star$, substituting $q_i = 1$ yields $\mathcal{C}_i^\star = \mathcal{C}_0$, while the case $q_i < 1$ follows from the bound in Theorem 3.1 and $L - 2w \geq 1$. □

D.4 What Early-Step Consistency Measures

> **Takeaway.** *Early-Step Consistency is a scale-free lower-tail statistic: it decreases when early updates fall below a sample-relative threshold, not when their common scale changes.*

Proposition D.2 (Scale invariance and lower-tail interpretation)**.** *For a fixed sample, suppress its index. Let $s_0, \ldots, s_{w-1} \geq 0$, let $\widetilde{s}^E = \mathrm{median}_\ell\, s_\ell$, and let $\mathcal{F} = w^{-1} \sum_\ell \mathbb{1}[s_\ell \geq \alpha \widetilde{s}^E]$ for $0 < \alpha < 1$. Then $\mathcal{F} \in [1/2, 1]$ and is invariant under multiplying all steps by any $c > 0$. Moreover, $\mathcal{F} = 1$ exactly when no early step falls below $\alpha \widetilde{s}^E$.*

Proof. Let $I = \{0, \ldots, w-1\}$ and define

$$I_{\mathrm{med}} = \{\ell \in I : s_\ell \geq \widetilde{s}^E\}, \qquad I_\alpha = \{\ell \in I : s_\ell \geq \alpha \widetilde{s}^E\}.$$

By the defining property of the sample median, $|I_{\mathrm{med}}| \geq \lceil w/2 \rceil$. Since $s_\ell \geq 0$ implies $\widetilde{s}^E \geq 0$ and $0 < \alpha < 1$, we have $\alpha \widetilde{s}^E \leq \widetilde{s}^E$, hence $I_{\mathrm{med}} \subseteq I_\alpha$. Therefore

$$\mathcal{F} = \frac{|I_\alpha|}{w} \geq \frac{|I_{\mathrm{med}}|}{w} \geq \frac{\lceil w/2 \rceil}{w} \geq \frac{1}{2}.$$

The inequality $\mathcal{F} \leq 1$ follows from $I_\alpha \subseteq I$.

Now fix $c > 0$ and write $\widetilde{s}_c^E = \mathrm{median}_\ell(cs_\ell)$. Positive scaling preserves order, so $\widetilde{s}_c^E = c\widetilde{s}^E$. Consequently,

$$\frac{1}{w}\sum_{\ell\in I}\mathbb{1}[cs_\ell \geq \alpha\widetilde{s}_c^E] = \frac{1}{w}\sum_{\ell\in I}\mathbb{1}[cs_\ell \geq \alpha c\widetilde{s}^E] = \frac{1}{w}\sum_{\ell\in I}\mathbb{1}[s_\ell \geq \alpha\widetilde{s}^E] = \mathcal{F},$$

which proves scale invariance. Finally,

$$\mathcal{F} = 1 \iff |I_\alpha| = w \iff I_\alpha = I \iff \forall \ell \in I,\ s_\ell \geq \alpha\widetilde{s}^E,$$

which is equivalent to no early step falling below $\alpha\widetilde{s}^E$. □

Thus $\mathcal{F}_i$ measures occupancy above a sample-relative lower-tail threshold. A sequence of uniformly tiny steps and a sequence of uniformly large steps both attain one.

E NOTATION

Table 11: Notation grouped by its role in the data, model, optimization, geometric construction, and theoretical analysis.

Symbol	Type / Domain	Meaning
Data		
i, t, T_i	$\mathbb{N};\ \{0, \dots, T_i - 1\};\ \mathbb{N}_{>0}$	Sample index, token index, and length.
x_i, y_i	$\{1, \dots, V\}^{T_i};\ \{0, 1\}$	Sample and outcome label.
Model and Representation		
ℓ, L	$\{0, \dots, L\};\ \mathbb{N}_{>0}$	Layer-state index and number of transformer-block transitions; $\ell = 0$ is the embedding state.
D, V, k	$\mathbb{N}_{>0}$	Hidden, vocabulary, and subspace dimensions.
$h_{i,t}^{(\ell)}$	$\mathbb{R}^D$	Raw hidden state.
U, b_{out}	$\mathbb{R}^{D\times V};\ \mathbb{R}^V$	Unembedding parameters.
$o_{i,t}^{(\ell)}$	$\mathbb{R}^V$	Auxiliary output-space token state.
B	$\mathbb{R}^{V\times k}$	Shared projection.
A_{eff}	$\mathbb{R}^{D\times k}$	Effective hidden-to-subspace map UB.
$z_{i,t}^{(\ell)}$	$\mathbb{R}^k$	Token coordinate.
$\mathbf{Z}_i^{(\ell)}$	$\mathbb{R}^{T_i\times k}$	Per-layer token-coordinate matrix.
Γ_i	$(\mathbb{R}^{T_i\times k})^{L+1}$	Reasoning dynamics.
Optimization		
$\mathcal{L}_{\text{con}}$	$\mathbb{R}_{\geq 0}$	Supervised contrastive loss.
η, τ, E	$\mathbb{R}_{>0};\ \mathbb{R}_{>0};\ \mathbb{N}_{>0}$	Learning rate, temperature, and epochs.
Layer-Trajectory Signatures		
$\Delta\mathbf{Z}_i^{(\ell)}$	$\mathbb{R}^{T_i\times k}$	Consecutive-layer sequence increment.
$s_\ell^{(i)}$	$\mathbb{R}_{\geq 0}$	Frobenius norm of a sequence increment.
w	$\mathbb{N}_{>0}$	Window width $\max(1, \lfloor L/3 \rfloor)$.
$\mathcal{W}_E, \mathcal{W}_L$	$\{0, \dots, w-1\};\ \{L-w-1, \dots, L-2\}$	Early and late transition-index windows.
$\widetilde{s}_i^E$	$\mathbb{R}_{\geq 0}$	Median step length over $\mathcal{W}_E$.
α	$(0, 1)$	Sample-relative lower-tail threshold, fixed at 0.7.
$\mathcal{D}_i$	$(0, 1]$	Detour Ratio for nonzero endpoint displacement.
$\mathcal{C}_i$	$\mathbb{R}_{\geq 0}$	Early-to-late mean-motion ratio with final step excluded.
$\mathcal{F}_i$	$[1/2, 1]$	First-third Early-Step Consistency at $\alpha = 0.7$.
Theoretical Analysis		
$v_i^{(\ell)}, d_i$	$\mathbb{R}^{d_i};\ \mathbb{N}_{>0}$	Vectorized sequence state and its dimension $d_i = T_i k$.
$\Delta v_i^{(\ell)}$	$\mathbb{R}^{d_i}$	Consecutive-layer vectorized increment.
u_i	$\{u \in \mathbb{R}^{d_i} : \|u\|_2 = 1\}$	Unit shared direction.
m_i, σ_i	$\mathbb{R}_{\geq 0};\ \mathbb{R}_{\geq 0}$	Directional magnitude and isotropic-noise scale, with $m_i^2 + \sigma_i^2 > 0$.
r_i	$\mathbb{R}_{\geq 0}$	Directional signal-to-noise ratio m_i/σ_i when $\sigma_i > 0$.
$a_{i,\ell}, A_i$	$\mathbb{R}_{>0};\ \mathbb{R}_{>0}$	Depth weight and its sum $A_i = \sum_{\ell=0}^{L-1} a_{i,\ell}$.
ρ_i	$[1/L, 1]$	Profile concentration $\sum_\ell a_{i,\ell}^2 / A_i^2$; $\rho_i < 1$ for $L \geq 2$.
g_ℓ	$\mathbb{R}^{d_i}$	Independent noise with $g_\ell \sim \mathcal{N}(0, I_{d_i})$.
λ_ℓ, q_i	$\mathbb{R}_{>0};\ (0, 1]$	Model-wide depth-profile weight and sample-specific contraction/no-contraction factor.

Continued on the next page.

Table 11 continued.

Symbol	Type / Domain	Meaning
$\mathcal{C}_0$	$\mathbb{R}_{>0}$	Shared-profile baseline $\big(\sum_{\ell\in\mathcal{W}_E}\lambda_\ell\big)/\big(\sum_{\ell\in\mathcal{W}_L}\lambda_\ell\big)$.
$\mathcal{D}_i^\star, \mathcal{C}_i^\star$	$[\sqrt{\rho_i}, 1];\ \mathbb{R}_{>0}$	Leading terms of $\mathcal{D}_i$ and $\mathcal{C}_i$; $\mathcal{D}_i^\star < 1$ when $\sigma_i > 0$ and $\rho_i < 1$.