

Decoding Self-Knowledge from Latent Reasoning Models for Selective Prediction and Adaptive Routing

Zhiqi Li¹, Yichi Zhang¹, Cunjie He¹, Ziheng Hua¹,
Maolin Wang³, Zenglin Xu^{1,2}

¹Fudan University, ²Shanghai Academy of AI for Science,

³City University of Hong Kong

{lizq26@m., zenglinxu@}fudan.edu.cn

Abstract

Latent reasoning models compress chain-of-thought into hidden states, but whether these representations retain usable self-knowledge about correctness remains unclear. We study two compression-based paradigms, Latent-SFT and CoLaR, and find a sharp distinction between problem-level and trajectory-level self-knowledge. A lightweight attention probe predicts which problems the model will solve (AUC 0.805/0.853) and transfers across datasets and compression ratios, yet cannot reliably rank rollouts of the same problem; distribution-matched retraining does not recover this signal. This distinction maps directly to downstream use: selective prediction improves accuracy by 20–25pp at 50% coverage, and adaptive routing outperforms either solver while reducing token cost, whereas latent best-of-N yields negligible gains despite available oracle headroom. The problem-level signal also persists on a 7B math checkpoint but vanishes on StrategyQA. These results identify an algorithm–paradigm boundary for exploiting self-knowledge in latent reasoning. Code is available at <https://github.com/Arthas777/latent-probe>.

1 Introduction

Latent reasoning models compress explicit chain-of-thought (Wei et al., 2022) into sequences of hidden states, reducing inference cost while preserving reasoning capability (Hao et al., 2024; Tan et al., 2025; Deng et al., 2025; Goyal et al., 2024). Do these compressed representations retain *self-knowledge*, internal signals of whether the model’s reasoning is likely to be correct? Du et al. (2026) answer affirmatively for Huginn-3.5B (Geiping et al., 2025), a *recurrent* latent model with stochastic initialization, where a trained classifier enables best-of-N selection over rollouts and gains +5.9pp on GSM8K. Huginn’s architecture, however, differs fundamentally from *compression-*

based paradigms that deterministically compress explicit reasoning into shorter latent sequences (Deng et al., 2024). Whether such paradigms also encode exploitable self-knowledge, and whether the same algorithmic approach applies, remains open.

We study two compression-based paradigms: Latent-SFT (Deng et al., 2025), which represents latent reasoning as vocab-space soft tokens at a fixed compression ratio, and CoLaR (Tan et al., 2025), which uses continuous hidden states with dynamic termination. On both, the signal exists but is mechanistically different from prior work. A lightweight attention probe reaches AUC 0.805 on Latent-SFT and 0.853 on CoLaR, with the same architecture working across both paradigms and zero-shot transfer across datasets and compression ratios (GSM8k→SVAMP). Crucially, the signal is *problem-level*, not *trajectory-level*: the probe discriminates which problems the model will solve, with pooled AUC ranging from 0.755 to 0.975 across configurations, but cannot rank rollouts of the same problem, where within-problem AUC stays between 0.510 and 0.680. Distribution-matched retraining on stochastic rollouts does not recover trajectory-level discrimination.

This dichotomy predicts which downstream applications succeed and which fail (Figure 3). Problem-level decisions work: selective prediction at 50% coverage improves accuracy by 20–25pp, and adaptive routing between latent and explicit CoT reaches 52.5% accuracy while saving 38% of tokens, exceeding both pure solvers. Trajectory-level decisions do not: reproducing LTO’s best-of-N algorithm (Du et al., 2026) on our paradigms yields at most +0.3pp, even though oracle ceilings of +3.8–10.2pp show that better rollouts exist. The same architecture generalizes to a 7B checkpoint from a different model family: the problem-level signal persists on two math benchmarks but falls to chance on a commonsense domain. Together

with LTO’s positive result on Huginn, these observations delineate a *paradigm-conditional viability boundary*: whether latent self-knowledge is practically exploitable depends on the algorithm–paradigm pairing rather than signal strength alone, and problem-level applications generalize more broadly across compression-based systems.

2 Related Work

Latent reasoning and test-time compute. Chain-of-thought prompting (Wei et al., 2022) and self-consistency (Wang et al., 2023) established that explicit intermediate reasoning improves LLM performance, motivating latent alternatives that preserve quality at reduced token cost. Several paradigms compress explicit CoT into latent representations: Coconut (Hao et al., 2024) uses hidden states as latent tokens, CoLaR (Tan et al., 2025) applies dynamic compression with KL-aligned distribution matching, Latent-SFT (Deng et al., 2025) distills reasoning into vocab-space superposition tokens at a fixed ratio, and pause tokens (Goyal et al., 2024) add latent computation before generation. Huginn-3.5B (Geiping et al., 2025) takes a different route, coupling recurrent depth with stochastic initialization, building on adaptive computation (Graves, 2016; Dehghani et al., 2019). Concurrent analyses include causal studies of continuous thought (Zhang et al., 2025b) and depth-accuracy failure modes (Sahoo et al., 2026). Prior work focuses on the accuracy–efficiency tradeoff; our interest is what these models internally “know” about their own reasoning quality.

Confidence estimation and hidden-state probing. Selective prediction (El-Yaniv and Wiener, 2010; Geifman and El-Yaniv, 2017) requires reliable confidence scores, yet last-token probabilities are poorly calibrated on reasoning tasks (Kadavath et al., 2022; Guo et al., 2017). Hidden-state probing (Conneau et al., 2018; Belinkov, 2022) has been applied to factual knowledge (Burns et al., 2023), truthfulness (Azaria and Mitchell, 2023), and representation structure (Zou et al., 2023). Recent work probes *explicit* reasoning models: Zhang et al. (2025a) show that DeepSeek-R1 hidden states encode correctness signals usable for early exit, LYNX (Akgul et al., 2025) applies conformal prediction to the same end, and ReProbe (Ni et al., 2025) trains a probe for best-of-N selection over explicit traces. Our work targets *compression-based*

latent reasoning, which is architecturally distinct: explicit cue tokens are absent, and standard token-level confidence is either unreliable, as in Latent-SFT, or undefined, as in CoLaR.

Routing, adaptive computation, and LTO.

Adaptive computation (Graves, 2016; Schuster et al., 2022; Raposo et al., 2024) allocates computation at the token level within a single forward pass, while model routing (Chen et al., 2023; Ong et al., 2024) directs inputs to different models based on predicted difficulty. Our setting differs from both: we route between *reasoning strategies* of the same base model at the problem level, exploiting solver complementarity. Most closely related is LTO (Du et al., 2026), which trains a classifier on Huginn-3.5B’s latent thoughts for best-of-N trajectory selection, relying on inter-rollout diversity from stochastic initialization. We reproduce LTO’s algorithm on compression-based paradigms, show that it fails there (§7), and propose routing and selective prediction as alternatives matched to the problem-level signal structure. Inference-time interventions on discrete-CoT models, such as Soft Thinking (Zhang et al., 2025c), SwiReasoning (Shi et al., 2026), and LaTER (Li et al., 2026), operate intra-trajectory via soft embeddings or entropy-based mode switching; they are complementary to our problem-level analysis of compression-trained latent paradigms.

3 Experimental Setup

Latent reasoning paradigms. We study two compression-based paradigms built on Llama-3.2-1B-Instruct (Team, 2024). **Latent-SFT** (Deng et al., 2025) compresses explicit CoT into vocab-space soft tokens at a fixed compression ratio r ; we use $r=2$ and $r=4$. **CoLaR** (Tan et al., 2025) produces continuous hidden states with dynamic termination governed by a compression factor $c=5$. These two paradigms cover the dominant representational classes for compression-based latent reasoning; Table 1 summarizes their trajectory lengths and accuracies. For scale and family generalization we additionally evaluate a 7B checkpoint from a different family, Qwen2.5-Math-7B-Latent-SFT (Yang et al., 2024; Deng et al., 2025), using the same probe architecture (§5).

Datasets. Our primary evaluation uses **GSM8k-Aug**, derived from GSM8K (Cobbe et al., 2021), with **SVAMP** (Patel et al., 2021) as the arithmetic

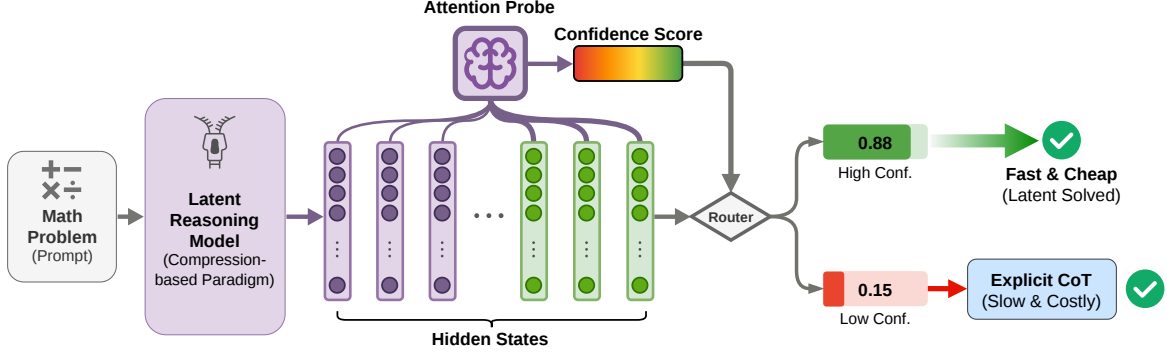


Figure 1: Overview of our approach. A lightweight attention probe extracts a confidence score from the hidden-state trajectory of a compression-based latent reasoning model. A router uses this score to select fast latent inference when confidence is high and explicit chain-of-thought when confidence is low, improving accuracy while reducing token cost.

cross-dataset transfer testbed. The 7B evaluation adds **MATH-500** (Lightman et al., 2023) and **StrategyQA** (Geva et al., 2021). All reported AUC values measure binary classification of correct vs. incorrect model outputs.

Probe architecture. We train an **AttentionConfidenceHead**: a lightweight attention-pooling module over the latent trajectory followed by an MLP classifier, following the probing methodology of prior work (Conneau et al., 2018; Belinkov, 2022). The probe takes the last-layer hidden states $\{h_1, \dots, h_T\}$ and outputs a scalar confidence score. Training uses binary cross-entropy on correctness labels under nested cross-validation (§5), with all CV splits at the problem level so that no problem appears in both train and test folds.

Key metrics. **Pooled AUC** measures cross-problem discrimination over all problems. **Within-problem AUC** averages per-problem AUC over rollouts and measures trajectory-level discrimination; it is computed over problems containing at least one correct and one incorrect rollout, for which per-problem AUC is defined (3–20% of problems, depending on configuration). **Oracle utilization** is the LTO gain divided by the oracle best-of-N gain.

Unless otherwise noted, all reported probe AUCs use 5-fold nested cross-validation at the problem level on GSM8k-Aug ($N=1319$), with mean $\pm$ std over independent CV seeds.

Paradigm	Representation	T (mean)	Acc.
Latent-SFT(2)	Vocab softmax	12.9	50.0%
Latent-SFT(4)	Vocab softmax	7.6	45.0%
CoLaR ($c=5$)	Continuous $\mathbb{R}^{2048}$	5.8	28.1%

Table 1: Latent reasoning paradigms studied. All share the Llama-3.2-1B-Instruct backbone. Representation indicates the form of latent states: vocab-space soft tokens for Latent-SFT, continuous hidden vectors for CoLaR. T is the mean trajectory length in latent steps.

4 The Signal Exists But Not Where Naïve Methods Look

We motivate the trained probe in two steps: hand-crafted scalar signals, while informative, are not robust across paradigms (§4.1), and a single closed-form normalization partially resolves this instability while leaving a substantial gap that only learned representations can close (§4.2).

4.1 Hand-Crafted Signals Exist But Are Not Robust

Length direction is paradigm-dependent; raw entropy is length-confounded. Trajectory length T carries strong correctness signal, but its *direction* reverses across paradigms (Figure 2). CoLaR’s dynamic termination yields short T for correct solutions, with AUC 0.750, while Latent-SFT’s fixed compression shows the opposite direction, with AUC 0.331–0.346. A length-based classifier trained on one paradigm therefore *anti-predicts* on the other. Raw spectral entropy, a natural paradigm-agnostic candidate, correlates inversely with correctness (AUC 0.356–0.409) because of a Simpson’s paradox: incorrect samples

cluster at high T , mechanically inflating entropy. Length is the primary confound, with $\rho = -0.386$ to correctness, and any robust scalar must account for it.

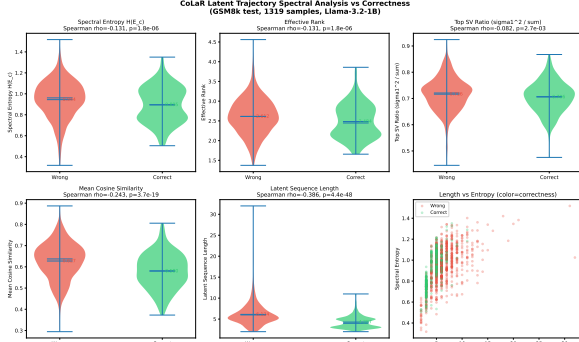


Figure 2: Spectral feature distributions by correctness on CoLaR (GSM8k test, $N=1319$, Llama-3.2-1B). Bottom-right: Length vs. Entropy scatter colored by correctness.

4.2 LENS: A Closed-Form Cross-Paradigm Baseline

The diagnosis above identifies two problems: length direction reverses across paradigms, and raw entropy is confounded by length. We resolve both with a single normalization, defining LENS (Length-normalized ENTropy of Singular values):

$$\text{LENS}(x) = \frac{H(\text{singular values of } [\alpha_1; \dots; \alpha_T])}{\log T}, \quad (1)$$

where $[\alpha_1; \dots; \alpha_T] \in \mathbb{R}^{T \times d}$ is the matrix of latent hidden states along the trajectory for input x , $H(\cdot)$ is the Shannon entropy of the normalized singular value distribution, and T is the trajectory length.

Dividing by $\log T$ removes the length confound from entropy and absorbs the paradigm-dependent length direction. After normalization, higher per-step representational diversity consistently indicates correctness, regardless of whether the paradigm produces short-correct or long-correct trajectories, and the correlation turns positive across all three settings (Table 7, Appendix B).

LENS achieves $\text{AUC} \geq 0.70$ everywhere, ranging from 0.702 to 0.754, making it the strongest closed-form scalar baseline we evaluated. Yet the partial correlation of LENS with correctness given length remains modest, at 0.125–0.307, indicating that LENS captures only part of the available signal. The remaining gap motivates learning directly from the hidden-state sequence (§5), where atten-

tion over the full trajectory can exploit structure inaccessible to any single scalar.

5 Trained Probe: Method, Performance, and Transfer

We establish three properties of the trained probe: it substantially exceeds closed-form and feature-engineering baselines on both paradigms (§5.1), it transfers zero-shot across datasets and compression ratios, applies without architectural modification across paradigms, and remains effective on a larger model family (§5.2), and the signal it captures is not answer-encoding, exceeds prompt-derived difficulty, and concentrates near the trajectory endpoint (§5.2).

5.1 Architecture and Main Results

Architecture. The AttentionConfidenceHead applies self-attention over the T trajectory steps, pools the result, and passes it through a two-layer MLP that outputs a scalar confidence score. Attention pooling handles variable-length trajectories naturally. Training minimizes binary cross-entropy on correctness labels over 1,319 labeled trajectories from GSM8k-Aug, using 5-fold nested cross-validation at the problem level. The head has roughly 0.4M parameters on the 1B backbone; details are in Appendix A.

Setting	LENS	Probe	Δ
LSFT GSM8k	0.713	0.805±0.011	+0.092
CoLaR GSM8k	0.650	0.853±0.004	+0.203

Table 2: Main probe results on GSM8k-Aug, comparing the trained AttentionConfidenceHead against LENS, a parameter-free closed-form baseline given by spectral entropy normalized by trajectory length. Seeds: 6 runs for Latent-SFT, 4 for CoLaR.

Main results. The trained probe reaches AUC 0.805 on Latent-SFT and 0.853 on CoLaR, well above the LENS baseline (Table 2). The asymmetric improvement, +0.092 versus +0.203, reflects a representational difference: Latent-SFT’s vocabulary simplex is already well captured by hand-crafted entropy features, whereas CoLaR’s continuous hidden states contain richer learnable structure that scalar statistics cannot reach. Standard token-level confidence (last-token argmax, verbalized elicitation) and feature-engineering baselines all underperform substantially; the full comparison is in Appendix C.

A sampling-based baseline gives a further point of reference. Over 8 rollouts per problem, the fraction of rollouts agreeing with the majority answer predicts whether the majority-vote answer is correct at AUC 0.786. This targets majority-vote correctness rather than per-rollout correctness, so we treat it as a comparison point rather than a matched contrast to the probe’s per-rollout AUC; it shows that the problem-level signal is also reachable through repeated sampling, at roughly $8\times$ the forward passes of the single-pass probe.

Why attention? Sequence-aware processing is critical. A pointwise MLP that processes each step independently barely exceeds LENS, by +0.007 on Latent-SFT and +0.096 on CoLaR, while attention across trajectory steps provides the dominant improvement (Table 8, Appendix B). Attention therefore captures information distributed across trajectory steps that is missed by pointwise statistics.

5.2 Generalization and Signal Characterization

Cross-dataset transfer. The probe trained on GSM8k transfers zero-shot to SVAMP, reaching AUC 0.810 on Latent-SFT and 0.785 on CoLaR. The LENS baseline reaches 0.478 and 0.646 on the same transfer. Both datasets are arithmetic, so we treat this as cross-dataset transfer within a domain rather than cross-domain generalization.

Cross-paradigm and cross-compression-ratio. The architecture transfers across paradigms, while probe weights transfer across compression ratios. The probe trained on Latent-SFT($r=4$) is applied zero-shot to Latent-SFT($r=2$) for routing (§6.2) with no retraining, validating cross- r weight transfer. On CoLaR the same head applies without architectural modification but is trained under the identical protocol rather than weight-transferred, so paradigm transfer here is architectural.

Scale and domain generalization. To test whether the problem-level signal depends on the 1B backbone, we apply the same probe architecture to Qwen2.5-Math-7B-Latent-SFT (Yang et al., 2024; Deng et al., 2025), a 7B checkpoint from a different model family, under the same nested-CV protocol (Table 3). On the two math benchmarks the signal is decodable, with pooled AUC 0.701 on GSM8k-Aug and 0.636 on MATH-500; both are above chance and mutually consistent, though weaker than the 1B results. Because base accu-

racy is high, selective prediction is correspondingly strong, reaching 94.9% and 79.1% at 50% coverage. On the commonsense benchmark StrategyQA the signal is absent, with pooled AUC 0.506 and no selective-prediction gain.

Benchmark	Domain	Base acc.	Pooled AUC	Selective@50
GSM8k-Aug	Arithmetic	88.3%	0.701±0.018	94.9% (+6.6pp)
MATH-500	Competition math	71.4%	0.636±0.027	79.1% (+7.7pp)
StrategyQA	Commonsense	58.2%	0.506±0.010	58.3% (+0.0pp)

Table 3: Problem-level probe on Qwen2.5-Math-7B-Latent-SFT, 6-seed 5-fold nested CV. The signal is decodable on math and absent on a commonsense domain. Selective@50 reports accuracy on the top-50% most confident problems under the probe.

Not answer encoding. A natural concern is that the probe implicitly decodes the answer (Belinkov, 2022). An answer-prediction probe on the same last-step hidden state reaches only 25.8% token-level top-1 accuracy on Latent-SFT and 10.2% on CoLaR, with numeric exact match at 0.0% on both. The model thus “knows when it is right” without exposing “what is right” in a directly readable form.

Beyond prompt-derived difficulty. A prompt-only MLP probe, reading the last hidden state before latent generation, achieves AUC 0.769 on Latent-SFT and 0.728 on CoLaR. The latent trajectory contributes +0.036 and +0.125 of AUC above this prompt-derived difficulty, with paired bootstrap 95% CIs excluding zero. The downstream gap is larger: prompt-only routing gains +0.5pp over pure CoT, versus +2.0pp for the full-trajectory probe (§6.2).

Endpoint dominance. Both paradigms concentrate signal near the trajectory endpoint, though with a mechanistic difference. On Latent-SFT, the final step alone recovers about 98% of full-trajectory AUC and masking it costs 0.107 AUC, so the endpoint is load-bearing. On CoLaR, the final step recovers about 95% of full-trajectory AUC, but masking it costs only 0.009 AUC and the last three steps already recover the full signal, so CoLaR’s confidence is distributed across the final steps rather than pinned to one state (Table 12, Appendix F). Either way, the final representation summarizes the reasoning outcome rather than the step-by-step trajectory dynamics, foreshadowing the problem-level nature analyzed in §7.

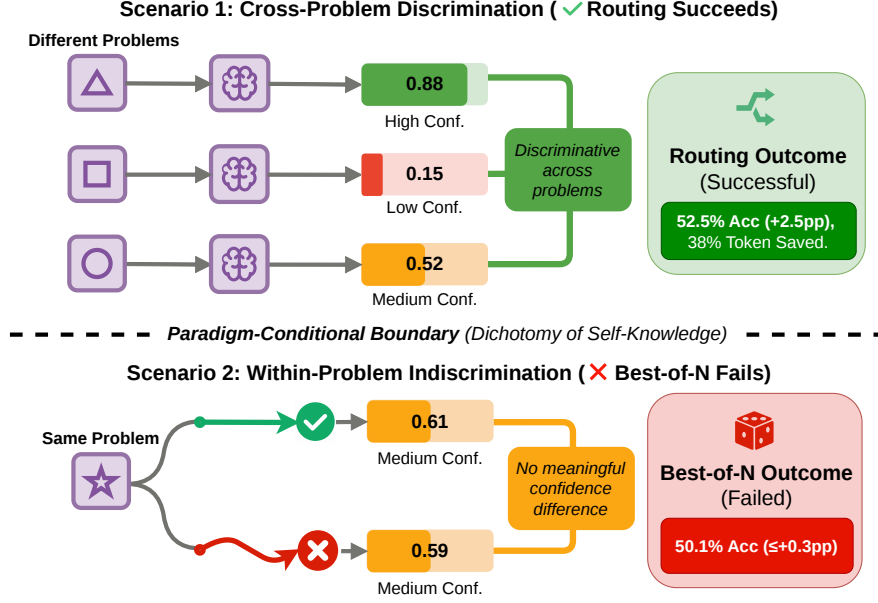


Figure 3: The dichotomy of self-knowledge. **Scenario 1:** the probe assigns well-separated scores to different problems, enabling routing. **Scenario 2:** rollouts of the same problem receive nearly identical scores regardless of outcome, causing best-of-N to fail.

6 Problem-Level Applications

The problem-level nature of the signal predicts which applications succeed. We evaluate two: selective prediction, which abstains on hard problems, and adaptive routing, which chooses between reasoning strategies.

6.1 Selective Prediction

Selective prediction abstains on low-confidence problems (El-Yaniv and Wiener, 2010; Geifman and El-Yaniv, 2017, 2019; Madhusudhan et al., 2024), and is the most direct application of a problem-level signal. We evaluate at 50% coverage, where the model answers only the top-50% most confident problems.

Setting	All	@50%	Δ
LSFT GSM8k	45.0%	69.7%	+24.7pp
LSFT SVAMP	55.6%	79.2%	+23.6pp
CoLaR GSM8k	28.1%	49.3%	+21.2pp
CoLaR SVAMP	47.2%	67.6%	+20.4pp

Table 4: Selective prediction at 50% coverage. All: accuracy on the full test set without abstention. @50%: accuracy on the top-50% most confident problems ranked by the probe.

Selective prediction improves accuracy by 20–25pp across all four settings with striking consistency (Table 4), from +20.4pp on CoLaR SVAMP,

the hardest cross-dataset setting, to +24.7pp on in-distribution Latent-SFT GSM8k. The uniformity is a direct consequence of the signal’s nature: the probe identifies which problems the model will solve, regardless of paradigm or dataset.

Coverage–accuracy tradeoff. The ordering extends smoothly across operating points. On Latent-SFT GSM8k, accuracy reaches 80.1% at 30% coverage and remains +14.3pp above the no-abstention baseline even at 70% coverage. The probe thus flags a long tail of confidently wrong problems rather than only extreme failures, making the abstention budget tunable to deployment cost.

6.2 Adaptive Inference Routing

Setup. Adaptive routing between reasoning strategies has been explored for cross-model settings (Chen et al., 2023; Ong et al., 2024; Dekoninck et al., 2024); we apply it to intra-model strategy selection. For each problem, we run latent reasoning with Latent-SFT(2), obtain a confidence score from the probe, and route: if the score exceeds threshold τ , we output the latent answer; otherwise we fall back to CoT-SFT. The amortized cost per problem is $\text{cost} = T_{\text{latent}} + \mathbb{I}[\text{conf} \leq \tau] \cdot T_{\text{CoT}}$, summing the latent forward pass and any CoT fallback. The probe was trained on Latent-SFT(4) and applied zero-shot to Latent-SFT(2), so rout-

ing simultaneously tests cross-compression-ratio transfer. The threshold is selected by 5-fold cross-validation with an argmax-accuracy criterion.

τ	Latent%	Acc.	Tokens	Saving
0.00 (pure latent)	100%	50.0%	12.9	84%
0.05	49%	52.8%	56.5	28%
0.10	37%	52.1%	65.4	17%
0.20	28%	51.1%	72.9	8%
1.00 (pure CoT)	0%	50.5%	78.8	0%
5-fold CV	59%	52.5$\pm$1.5%	49.1	38%

Table 5: Routing Pareto sweep on GSM8k-Aug. Latent% is the fraction of problems handled by latent reasoning; Tokens and Saving follow the cost definition in §6.2. The bottom row is the deployable operating point, with τ selected by 5-fold CV (mean $\pm$ std across folds).

Main results. Table 5 and Figure 4 present the routing Pareto frontier. At the CV-selected operating point, routing reaches 52.5% accuracy while saving 38% of tokens, exceeding both pure baselines. The full-test argmax optimum is only 0.7pp higher (53.2%), indicating small threshold-selection regret; at this diagnostic point, where τ is chosen using test labels, the paired gain over pure CoT is significant (McNemar $p=0.008$; 107 vs. 71 discordant problems; 95% CI [+0.7, +4.7]pp), while the CV-selected point itself is not (95% CI [−0.3, +3.9]pp). Among the routing baselines (Appendix C), the full-trajectory probe posts the largest CV gain, ahead of multi-feature logistic regression (+2.0pp vs. +1.4pp).

Mechanistic explanation: solver complementarity. Routing exceeds both baselines because the two solvers specialize. On the upper two-thirds of confidence, latent reasoning outperforms CoT by +3.9pp in each tertile; on the bottom third, CoT wins by +9.1pp. The probe therefore identifies each solver’s competence region. These bin-level differences are descriptive rather than individually significant, with only the low-confidence bin’s 95% CI excluding zero; the aggregate gain rests on the paired test above (Appendix F).

Cross-dataset and cross-paradigm validation. Under double transfer, cross-dataset and cross-compression-ratio, SVAMP routing reaches 64.0%, matching pure CoT’s 63.9% while saving tokens, and preserves Pareto dominance over random routing. On CoLaR, routing cannot beat CoT because the latent solver is far weaker (28.1% vs. 50.5%), so accuracy gains are precluded by the solver gap; the probe’s paradigm invariance is instead anchored

on selective prediction, which improves CoLaR accuracy by +21.2pp on GSM8k-Aug and +20.4pp on SVAMP (Table 4).

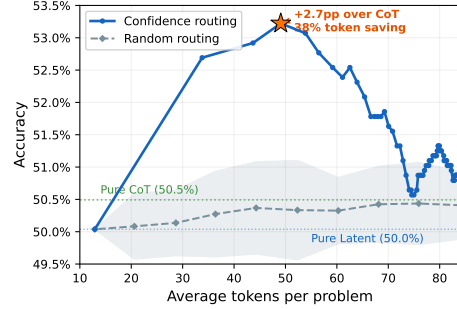


Figure 4: Routing Pareto frontier on Latent-SFT, GSM8k-Aug ($N=1319$, cross- r probe). The star is the full-test-set argmax-accuracy operating point ($\tau=0.03$): +2.7pp over pure CoT with 95% CIs [50.5, 56.0]% (accuracy) and [34.7, 40.7]pp (token saving); cost as in §6.2. Solid: confidence routing across τ . Dashed: random routing at latent probability $f \in \{0, 0.1, \dots, 1.0\}$ (shaded ± 1 std over 20 seeds). Confidence routing Pareto-dominates random.

7 Latent Best-of-N Is Paradigm-Conditional

We now test directly whether the self-knowledge signal supports trajectory-level selection, the setting Du et al. (2026) demonstrated on Huginn-3.5B and the analogue of best-of-N in explicit reasoning (Wang et al., 2023; Cobbe et al., 2021; Lightman et al., 2023).

7.1 LTO Best-of-N Reproduction

We reproduce LTO’s KL-regularized rejection sampling on our paradigms, using our trained probe as the reward model. For each problem we generate N rollouts and select using the LTO criterion, with inverse-temperature β scanned over [0.001, 5.0] and 200 repeated trials per setting.

LTO’s Algorithm 1 achieves $\leq +0.3$ pp on every compression-based configuration, despite oracle ceilings of +3.8–10.2pp showing that better rollouts exist (Table 6). On deterministic Latent-SFT(2), all $N=20$ rollouts are identical, so there is no rollout diversity to exploit and LTO returns the base accuracy unchanged. Even after injecting stochasticity, the probe cannot rank rollouts, and oracle utilization stays at 1.6–8.4%. Candidate diversity does exist: 3–20% of problems produce mixed outcomes at $N=8$ rollouts/problem (20.1% Latent-SFT($r=2$), 8.7% Latent-SFT($r=4$), 3.0% CoLaR), so the fail-

Method	Paradigm	Acc.	Δ	Oracle	Util.
Base	LSFT(2)	50.0	—	—	—
LTO	LSFT(2), det.	50.0	+0.0	0.0	—
LTO	LSFT(2), ε	50.1	+0.2	+10.2	1.6%
LTO	LSFT(4), ε	45.1	+0.3 ¹	+3.8	8.4%
LTO	CoLaR, σ	30.2	−0.5	+5.3	—
LTO ²	Huginn-3.5B	38.5	+5.9	—	—
Routing	LSFT(2)	52.5	+2.5	—	—

Table 6: LTO best-of-N reproduction on GSM8k-Aug ($N=20$ rollouts/problem). Δ is the gain over the single-rollout base; Oracle is the gain of oracle-best-of-N minus the base; Util. is Δ/Oracle . ¹Relative to the stochastic base (44.8%; deterministic LSFT(4) base 45.0%). ²From Du et al. (2026) (oracle not reported); Huginn’s within-problem behavior is inferred from their report, not measured by us.

ure is specifically within-problem ranking rather than absent diversity.

Weak trajectory selection also fails. Because the upper end of the within-problem range, 0.68 on Latent-SFT($r=4$), is not obviously near chance, we test whether weaker selectors can exploit it. On that configuration, neither majority voting over 3 rollouts nor probe-based best-of-3 improves over the 44.8% greedy baseline, both landing within 0.5pp of it, and even an oracle that selects any of 3 rollouts reaches only 46.6%. The available headroom of +1.8pp is itself small, so the upper-end signal does not support trajectory selection in practice (full sweep over $k \in \{3, 5, 8\}$ in Appendix D).

7.2 Intervention Test: The Signal Cannot Be Retrained

We test whether retraining on stochastic rollouts enables within-problem discrimination. Across five magnitudes of latent-space perturbation on CoLaR, with $\sigma \in \{1, \dots, 10000\}$, the oracle ceiling rises from +4.0pp to +11.5pp, confirming that diversity can be created, while within-problem AUC stays at 0.49–0.56. Distribution-matched retraining achieves pooled AUC 0.975, a near-perfect cross-problem ranking, but within-problem AUC of 0.510, barely above chance (Figure 6, Appendix F). Training thus cannot create trajectory-level signal that the architecture does not encode; the problem-level nature persists even when the training distribution matches the stochastic evaluation setting (Zou et al., 2023).

7.3 Cross-Configuration Universality

Within-problem AUC stays in the range 0.510–0.680 across all three configurations, spanning two paradigms, three compression ratios $r \in \{2, 4, \sim 5\}$, and trajectory lengths from 5.8 to 13.9 steps. In every case the range is insufficient for best-of-N (Table 10, Appendix D).

7.4 The 2×2 Paradigm-Algorithm Boundary

The central diagnostic is the contrast between the two signal levels. The probe decodes problem-level difficulty, with pooled AUC ranging from 0.755 to 0.975 across configurations, but does not differentiate rollouts of the same problem, where within-problem AUC ranges from 0.510 to 0.680. This yields a clear viability pattern. LTO’s best-of-N succeeds on the recurrent Huginn (+5.9pp; Du et al., 2026) but fails on compression paradigms ($\leq +0.3\text{pp}$), while our routing and selective prediction succeed precisely because they operate at the problem level. The Huginn comparison is indirect, since its within-problem behavior is inferred from LTO’s report rather than measured by us; the confounds of architecture, scale, and stochasticity are discussed in the Limitations.

8 Discussion

Our findings suggest a design principle: best-of-N (Snell et al., 2024; Tang et al., 2025) requires preserving rollout-level variation, while problem-level applications work with deterministic generation and pooled classifiers; verifier-based test-time scaling thus favors recurrent or stochastic architectures, and cost-sensitive abstention favors compression paradigms.

Why does compression erase trajectory-level signal? Objectives such as KL alignment and vocab-space distillation force the trajectory to match the *expected output* of explicit CoT, integrating over trajectory-internal variation. The endpoint concentration is the fingerprint: the final state encodes *whether the problem is solved*, not which path was taken. On Latent-SFT the endpoint is load-bearing, on CoLaR the signal spreads across the final steps, but the trajectory always collapses to an outcome summary. Huginn’s stochastic initialization preserves rollout-specific variation by design; the relevant distinction is thus not deterministic versus stochastic per se, but what variation the objective retains.

These findings yield a cheap diagnostic for new

latent paradigms: pooled and within-problem AUC over a few stochastic rollouts predict application viability without a full pipeline. High pooled with low within-problem admits selective prediction and routing but rules out best-of-N; high on both additionally admits trajectory selection; low pooled rules out all three. Self-knowledge is thus a structured property to be measured, not a scalar to be optimized.

Beyond this dichotomy, routing shows latent reasoning is not simply a cheaper CoT: stratified by confidence it outperforms CoT on the upper two-thirds of problems but loses on the lowest third, so quality is *redistributed* along difficulty rather than uniformly preserved (Hao et al., 2024; Deng et al., 2025; Tan et al., 2025). This complementarity lets routing exceed both pure solvers and suggests the probe encodes *solver-fit*—which reasoning mode suits a problem—rather than raw difficulty. SwiReasoning (Shi et al., 2026) similarly finds that purely latent decoding on discrete-CoT models diffuses probability mass and benefits from switching back to explicit reasoning.

The 7B results delimit where problem-level self-knowledge is decodable: it persists, though weaker, on a larger model from a different family on math, and vanishes on a commonsense benchmark despite the model’s commonsense accuracy not being low. A plausible explanation is that decodable self-knowledge tracks competence in the model’s trained reasoning modality. A clean test would require a latent model trained on non-arithmetic CoT, which is not yet publicly available.

9 Conclusion

Compression-based latent reasoning retains decodable self-knowledge that is *problem-level, not trajectory-level*: a trained probe achieves strong pooled AUC while within-problem discrimination stays near chance, even under distribution-matched retraining. This dichotomy predicts application viability, selective prediction and adaptive routing succeeding where latent best-of-N fails. The paradigm-conditional boundary suggests that algorithm–paradigm pairing, not signal strength alone, determines whether latent self-knowledge is exploitable, and that problem-level methods generalize more broadly across compression-based systems.

10 Limitations

We study two compression-based paradigms on a single 1B backbone, mainly on arithmetic tasks; broader architectures (Coconut, CCOT) and larger scales remain untested. At 7B we measured only problem-level prediction, so the within-problem boundary is established only at 1B. The problem-level signal is at chance on a commonsense domain, so we do not claim generalization beyond math-heavy domains. Huginn’s within-problem behavior is inferred from LTO’s reported viability rather than measured by us, so the 2×2 boundary confounds architecture, scale, and stochasticity with paradigm class. Part of the probe signal reflects prompt-derived difficulty (§5.2), though latent processing contributes meaningfully. Routing gains depend on solver complementarity and shrink when one solver dominates. Finally, our decisions are per-problem; within-problem dynamic switching (Li et al., 2026) is complementary and outside our scope.

References

- Omer Faruk Akgul, Yusuf Hakan Kalayci, Rajgopal Kannan, Willie Neiswanger, and Viktor Prasanna. 2025. LYNX: Learning dynamic exits for confidence-controlled reasoning. *arXiv preprint arXiv:2512.05325*.
- Amos Azaria and Tom Mitchell. 2023. The internal state of an LLM knows when it’s lying. *arXiv preprint arXiv:2304.13734*.
- Yonatan Belinkov. 2022. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219.
- Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. 2023. Discovering latent knowledge in language models without supervision. In *International Conference on Learning Representations (ICLR)*.
- Lingjiao Chen, Matei Zaharia, and James Zou. 2023. FrugalGPT: How to use large language models while reducing cost and improving performance. *arXiv preprint arXiv:2305.05176*.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, and others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.
- Alexis Conneau, German Kruszewski, Guillaume Lample, and others. 2018. What you can cram into a single $\&\#\ast$ vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

- Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, and others. 2019. Universal transformers. In *International Conference on Learning Representations (ICLR)*.
- Jasper Dekoninck and others. 2024. A unified approach to routing and cascading for LLMs. *arXiv preprint arXiv:2410.10347*.
- Jingcheng Deng, Liang Pang, Zihao Wei, Shicheng Xu, Zenghao Duan, Kun Xu, Yang Song, Huawei Shen, and Xueqi Cheng. 2025. LLM latent reasoning as chain of superposition. *arXiv preprint arXiv:2510.15522*.
- Yuntian Deng, Yejin Choi, and Stuart Shieber. 2024. From explicit CoT to implicit CoT: Learning to internalize CoT step by step. *arXiv preprint arXiv:2405.14838*.
- Shrey Desai and Greg Durrett. 2020. Calibration of pre-trained transformers. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
- Hanwen Du, Yuxin Dong, and Xia Ning. 2026. Latent thinking optimization: Your latent reasoning language model secretly encodes reward signals in its latent thoughts. In *International Conference on Learning Representations (ICLR)*.
- Ran El-Yaniv and Yair Wiener. 2010. On the foundations of noise-free selective classification. *Journal of Machine Learning Research*, 11:1605–1641.
- Yonatan Geifman and Ran El-Yaniv. 2017. Selective classification for deep neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Yonatan Geifman and Ran El-Yaniv. 2019. SelectiveNet: A deep neural network with an integrated reject option. In *International Conference on Machine Learning (ICML)*.
- Jonas Geiping and others. 2025. Scaling up test-time compute with latent reasoning: A recurrent depth approach. *arXiv preprint arXiv:2502.05171*.
- Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. In *Transactions of the Association for Computational Linguistics (ACL)*.
- Sachin Goyal, Ziwei Ji, Ankit Singh Rawat, and others. 2024. Think before you speak: Training language models with pause tokens. *International Conference on Learning Representations (ICLR)*.
- Alex Graves. 2016. Adaptive computation time for recurrent neural networks. *arXiv preprint arXiv:1603.08983*.
- Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. In *International Conference on Machine Learning (ICML)*.
- Shibo Hao and others. 2024. Training large language models to reason in a continuous latent space. *arXiv preprint arXiv:2412.06769*.
- Saurav Kadavath and others. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.
- Xuan Li, Yining Wang, Yuchen Liu, Guanjun Liu, Delai Qiu, Shengping Liu, Jiaen Liang, Wei Huang, Jun Yu, and Junnan Zhu. 2026. LaTER: Efficient test-time reasoning via latent exploration and explicit verification. *arXiv preprint arXiv:2605.07315*.
- Hunter Lightman, Vineet Kosaraju, Yura Burda, and others. 2023. Let’s verify step by step. *arXiv preprint arXiv:2305.20050*.
- Nishanth Madhusudhan and others. 2024. Do LLMs know when to not answer? investigating abstention abilities of large language models. *arXiv preprint arXiv:2407.16221*.
- Jingwei Ni, Ekaterina Fadeeva, Tianyi Wu, Mubashara Akhtar, Jiaheng Zhang, Elliott Ash, Markus Leippold, Timothy Baldwin, See-Kiong Ng, Artem Shelmanov, and Mrinmaya Sachan. 2025. ReProbe: Efficient test-time scaling of multi-step reasoning by probing internal states of large language models. *arXiv preprint arXiv:2511.06209*.
- Isaac Ong, Amjad Almahairi, Vincent Wu, and others. 2024. RouteLLM: Learning to route LLMs with preference data. *arXiv preprint arXiv:2406.18665*.
- Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are NLP Models Really Able to Solve Simple Math Word Problems? In *Proceedings of NAACL*.
- David Raposo, Sam Ritter, Blake Richards, and others. 2024. Mixture-of-depths: Dynamically allocating compute in transformer-based language models. *arXiv preprint arXiv:2404.02258*.
- Subramanyam Sahoo, Aman Chadha, Vinija Jain, and Divya Chaudhary. 2026. When shallow wins: Silent failures and the depth-accuracy paradox in latent reasoning. *arXiv preprint arXiv:2603.03475*.
- Tal Schuster, Adam Fisch, Jai Gupta, and others. 2022. Confident adaptive language modeling. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Dachuan Shi, Abedelkadir Asi, Keying Li, Xiangchi Yuan, Leyan Pan, Wenke Lee, and Wen Xiao. 2026. SwiReasoning: Switch-thinking in latent and explicit for pareto-superior reasoning LLMs. *arXiv preprint arXiv:2510.05069*.
- Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*.
- Wenhui Tan, Jiaze Li, Jianzhong Ju, Zhenbo Luo, Ruihua Song, and Jian Luan. 2025. Think silently, think fast: Dynamic latent compression of LLM reasoning chains. *arXiv preprint arXiv:2505.16552*.

Yung-Chen Tang, Pin-Yu Chen, and Andrea Cavallo. 2025. CarBoN: Calibrated best-of-N sampling improves test-time reasoning. *arXiv preprint arXiv:2510.15674*.

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

Xuezhi Wang, Jason Wei, Dale Schuurmans, and others. 2023. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations (ICLR)*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, and others. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, and others. 2024. Can LLMs express their uncertainty? an empirical evaluation of confidence elicitation in LLMs. *International Conference on Learning Representations (ICLR)*.

An Yang, Baosong Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Yang, Da Liu, Dayiheng Tu, and others. 2024. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*.

Anqi Zhang, Yulin Chen, Jane Pan, Chen Zhao, Aurojit Panda, Jinyang Li, and He He. 2025a. Reasoning models know when they’re right: Probing hidden states for self-verification. *arXiv preprint arXiv:2504.05419*.

Yuyi Zhang, Boyu Tang, Tianjie Ju, Sufeng Duan, and Gongshen Liu. 2025b. Do latent tokens think? a causal and adversarial analysis of chain-of-continuous-thought. *arXiv preprint arXiv:2512.21711*.

Zhen Zhang, Xuehai He, Weixiang Yan, Ao Shen, Chenyang Zhao, Shuohang Wang, Yelong Shen, and Xin Eric Wang. 2025c. Soft thinking: Unlocking the reasoning potential of LLMs in continuous concept space. *arXiv preprint arXiv:2505.15778*.

Andy Zou, Long Phan, Sarah Chen, and others. 2023. Representation engineering: A top-down approach to AI transparency. *arXiv preprint arXiv:2310.01405*.

A Experimental Protocol

Probe architecture details. The AttentionConfidenceHead projects each trajectory step to a 128-dimensional space, applies 4-head self-attention over the T steps with a padding mask, adds a LayerNorm, and mean-pools the attended states. A learned embedding of the trajectory length T is concatenated to the pooled representation, and a two-layer MLP with a sigmoid outputs the scalar confidence score. Total parameters are roughly

0.38M on the 1B backbone, where the input projection dominates; the 7B variant, whose input dimension is 3584 rather than 2048, scales to roughly 0.57M.

Training. Binary cross-entropy loss. Adam optimizer ($\text{lr}=1\text{e-}4$, weight decay $1\text{e-}5$). Batch size 64. Early stopping on validation AUC (patience 10). Training data: 1,319 trajectories from GSM8k-Aug with binary correctness labels, using 5-fold cross-validation at the problem level.

Nested cross-validation. Outer loop: 5 folds stratified by correctness. Inner loop: 4/5 train, 1/5 validation for early stopping. The final AUC is reported on the held-out outer fold. To quantify run-to-run variance, we repeat the full nested CV with different random seeds: 6 runs for Latent-SFT and 4 for CoLaR on the 1B backbone, and 6 seeds for each 7B setting.

7B evaluation. The 7B checkpoint is Qwen2.5-Math-7B-Latent-SFT, a math-specialized latent model from a different family, run with the official top-10 soft-token inference; we use a SymPy-based answer grader for GSM8k-Aug and MATH-500 and exact yes/no matching for StrategyQA. Trajectories are collected with generation budgets of 512 tokens for GSM8k-Aug and StrategyQA and 1024 for MATH-500, with truncation keeping the last T steps; the probe uses full-length trajectories ($\text{max_}T$ matching the budget). Data sizes are 1,319, 500, and 1,603 problems for GSM8k-Aug, MATH-500, and StrategyQA.

Data scope justification. We deliberately use only the GSM8k-Aug test set ($N=1,319$) under nested CV to keep probe evaluation unbiased and to ensure no correctness labels from the model’s training distribution leak into probe training. While a larger probe training set could raise absolute AUC, the consistency across nested-CV folds (std 0.004–0.011) indicates the signal is robust at this sample size.

Compute. All experiments run on a single NVIDIA A100 (80GB). Probe training takes about 5 minutes per fold, trajectory generation about 2 hours per paradigm on GSM8k-Aug, and the LTO reproduction, with 200 repeats over 11 β values, about 1 hour.

Artifact licenses. GSM8k (Cobbe et al., 2021) and SVAMP (Patel et al., 2021) are released under

the MIT License. Llama-3.2-1B-Instruct (Team, 2024) is released under the Llama 3.2 Community License Agreement. The Latent-SFT (Deng et al., 2025) and CoLaR (Tan et al., 2025) codebases are publicly available for research use. Our use of all artifacts is consistent with their intended research-only purposes.

B Hand-Crafted Signals and Architecture Ablation

Signal	CoLaR	LSFT(2)	LSFT(4)
Raw H_{spec}	0.409	0.356	0.391
Length T	0.750	0.331	0.346
LENS	0.754	0.702	0.714
Partial $\rho(\text{LENS} T)$	+0.125	+0.273	+0.307

Table 7: Scalar-signal AUC on GSM8k-Aug, $N=1319$. AUC <0.5 indicates an inverted signal.

Architecture	LSFT	CoLaR
LENS (closed-form)	0.713	0.650
MLP (pointwise)	0.720 $\pm$ 0.009	0.746 $\pm$ 0.006
Attention (ours)	0.805$\pm$0.011	0.853$\pm$0.004

Table 8: Architecture ablation on GSM8k-Aug; learned probes report mean $\pm$ std over nested-CV seeds (6 for Latent-SFT, 4 for CoLaR).

C Baseline Confidence Methods

We compare against standard confidence estimation approaches applicable to reasoning models. Last-token confidence, the argmax probability, reaches only AUC 0.527 on Latent-SFT and is *undefined* for CoLaR’s continuous hidden states; standard calibration methods (Guo et al., 2017; Desai and Durrett, 2020) do not apply to hidden-state paradigms, and verbalized confidence elicitation (Xiong et al., 2024) requires explicit token generation. A multi-feature logistic regression over length, entropy, and last-token reaches 0.738/0.770. The trained probe (0.805/0.853) dominates all of these, with the largest gains on CoLaR, where prompt and scalar features miss the most signal. A sampling-based baseline, the fraction of 8 rollouts agreeing with the majority answer, predicts majority-vote correctness at AUC 0.786; because its target is majority-vote rather than per-rollout correctness, we report it as a comparison point rather than a strictly matched contrast to the single-pass probe.

Table 9 compares these methods in the downstream routing setting under the same cost model and 5-fold CV threshold selection. The full-trajectory probe posts the largest CV-selected gain over pure CoT (+2.0pp), which is significant at the full-test diagnostic operating point (McNemar $p=0.008$; §6.2). Length-based routing hurts because Latent-SFT’s length–correctness direction is inverted.

Method	AUC	CV Acc	Saving	Δ CoT
Pure CoT	—	50.5%	0%	—
Random ($f=0.59$)	0.500	50.2 $\pm$ 0.7%	43%	−0.3pp
Length T	0.329	49.6 $\pm$ 1.1%	17%	−0.9pp
Multi-feature LR	0.729	51.9 $\pm$ 1.5%	17%	+1.4pp
Prompt-only probe	0.769	51.0 $\pm$ 2.0%	23%	+0.5pp
Full traj. probe (ours)	0.805	52.5$\pm$1.5%	38%	+2.0pp

Table 9: Routing baselines under the same cost model and 5-fold CV threshold selection. Δ CoT is the accuracy gap vs. pure CoT.

D LTO Best-of-N Reproduction Details

We implement LTO’s algorithm following Du et al. (2026): for each problem x , generate N rollouts $\{z_1, \dots, z_N\}$, compute a reward $r_i = f(z_i)$ from the probe, and select $z^* = \arg \max_i r_i \cdot \exp(-\beta \cdot D_{\text{KL}}(\pi_{z_i} \| \pi_{\text{base}}))$. Because our paradigms produce deterministic rollouts, Latent-SFT using argmax decoding, we inject minimal noise: Gaussian perturbation of magnitude $\varepsilon=0.001$ to logits for Latent-SFT, and σ -scaled noise to hidden states for CoLaR.

Config.	r	Pooled	Within	LTO	Oracle
LSFT(2)	2	0.755	0.594	+0.2	+10.2
LSFT(4)	4	0.929	0.680	+0.3	+3.8
CoLaR	~ 5	0.975	0.510	−0.5	+5.3

Table 10: Within-problem AUC at $N=8$ rollouts/problem, averaged over problems with mixed outcomes (265/115/39 of 1,319 for Latent-SFT($r=2$)/($r=4$)/CoLaR); the LSFT(2) pooled-AUC discrepancy is explained below.

β -scan protocol. $\beta \in \{0.001, 0.005, 0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1.0, 2.0, 5.0\}$. For each β , 200 independent trials with different random seeds. The best β is selected by mean accuracy across trials.

LSFT(2) pooled AUC discrepancy. Table 10 reports LSFT(2) pooled AUC of 0.755, measured on ε -perturbed 8-rollout trajectories with the probe applied cross- r , trained on $r=4$. This differs from

Selector	$k=3$	$k=5$	$k=8$
Greedy (single rollout)	44.8%	44.8%	44.8%
Majority vote	44.3%	44.5%	44.4%
Best-of- k by probe	44.9%	45.1%	45.1%
Oracle (best of k)	46.6%	47.6%	48.6%

Table 11: Weak trajectory selection on Latent-SFT($r=4$) (highest within-problem AUC, 0.680); mean accuracy over 200 random size- k subsets from $N=8$ rollouts/problem.

the 0.805 ± 0.011 in Table 2, which uses deterministic single rollouts in-distribution. The degradation reflects calibration drift under cross- r out-of-distribution evaluation. Routing (§6.2) requires only coarse discrimination, which remains feasible at this pooled AUC, and the routing accuracy in Table 5 confirms this empirically.

E CoLaR Routing Pareto Frontier

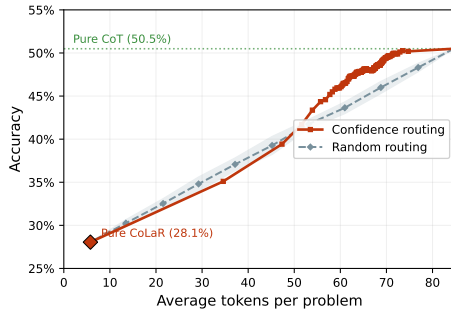


Figure 5: CoLaR routing Pareto frontier on GSM8k-Aug. With the latent solver far weaker than CoT, confidence routing approaches but never exceeds pure CoT. It is competitive with random routing at matched cost but does not Pareto-dominate it; the same probe architecture recovers the ranking signal on both paradigms, while achievable accuracy is bounded by the solver gap.

F Additional Results

Variant	Latent-SFT		CoLaR	
	AUC	% of full	AUC	% of full
Full trajectory	0.820	100%	0.856	100%
Last-3 steps	0.823	100%	0.854	99.8%
Last-1 step	0.806	98.3%	0.813	95.0%
Fixed-length	0.801	97.7%	0.858	100%

Table 12: Truncation analysis on GSM8k-Aug (single nested-CV run; the LSFT full-trajectory value 0.820 lies within the 0.805 ± 0.011 band of Table 2). Masking the endpoint costs 0.107 AUC on Latent-SFT but only 0.009 on CoLaR (§5.2).

Confidence bin	N	Latent	CoT	Δ
High (top 1/3)	440	72.0%	68.1%	+3.9pp [-0.5, +8.2]
Mid (middle 1/3)	440	55.8%	51.9%	+3.9pp [-0.7, +8.6]
Low (bottom 1/3)	439	22.4%	31.5%	-9.1pp [-13.9, -4.1]

Table 13: Solver complementarity by confidence tertile (95% bootstrap CIs on the gap); treated as descriptive, with significance anchored on the aggregate paired test (§6.2).

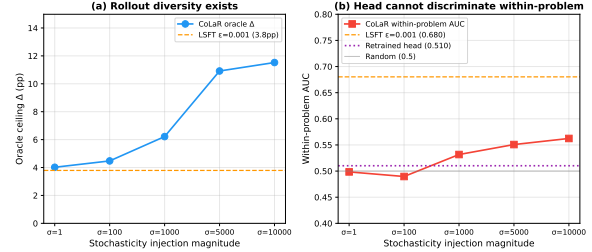


Figure 6: Intervention test on CoLaR. Left: the oracle ceiling rises with stochasticity. Right: within-problem AUC stays near chance across σ ; a distribution-matched retrained head reaches near-perfect pooled AUC while within-problem AUC remains at chance.

Quintile-level routing. The strongest single-bin routing advantage appears in quintile Q2, the second-highest confidence, where latent reasoning reaches 62.9% versus 56.8% for CoT, +6.1pp. The latent advantage peaks in the “confident but not trivial” regime: Q1 shows a smaller gap because both solvers handle the easiest problems well, and Q4–Q5 reverse direction as difficulty exceeds latent reasoning’s capacity. This non-monotone profile is consistent with the solver-fit interpretation, that the probe identifies which problems lie in latent reasoning’s competence region rather than ranking raw difficulty.

Risk-coverage analysis. Selective prediction degrades smoothly across operating points. Latent-SFT GSM8k accuracy traces 80.1%, 69.7%, and 59.3% at 30%, 50%, and 70% coverage, an approximately linear 10pp decay per 20pp of coverage. The confidence ordering thus remains informative across the full distribution rather than isolating a small tail of extreme failures, and the abstention budget is genuinely tunable: 30% coverage yields 80.1% accuracy, +35pp over the 45.0% no-abstention baseline, while 70% coverage still gains +14.3pp. The same monotone behavior holds on CoLaR and SVAMP, indicating that the curve shape reflects the signal structure rather than any single paradigm-dataset pairing.