DIVERSITY WITHOUT DISCOVERY: WHY CHAIN PERTURBATION FAILS TO EXPLORE IN LATENT TEST-TIME SCALING

Anonymous authors
Paper under double-blind review

ABSTRACT

Parallel test-time scaling (PTTS) improves reasoning by sampling multiple chain-of-thought (CoT) traces and aggregating them, marginalizing over the reasoning paths the model inherently finds plausible. Various SOTA latent-reasoning models instead evolve continuous hidden states under deterministic dynamics, following one trajectory per input with no native randomness to sample from. Recent work therefore perturbs the latent chain at test time and presents the resulting answer diversity as exploration potential in continuous space. To test this, we fairly compare latent perturbation methods with standard explicit reasoning on a shared backbone and find that, for unconstrained latent thoughts, i.e., thoughts not restricted to the embedding space, aggregating perturbed latent chains brings no gain over the no-sampling baseline, whereas explicit reasoning turns the same budget into accuracy gains. Our formal analysis demonstrates that explicit reasoning marginalizes over the trajectory distribution stochastically exhibited by the model, while a perturbed state comes from no such distribution, so its vote carries no belief of the model and no guarantee of favoring the correct answer. Empirically, the added diversity comes predominantly from borderline samples and, unlike with explicit reasoners, a correct answer reached through perturbation carries no gain in confidence. As such, even a frozen latent chain, perturbed at most once at the answer step, can be calibrated to reproduce or outperform every full-chain perturbation method we tested at no additional reasoning-token cost.

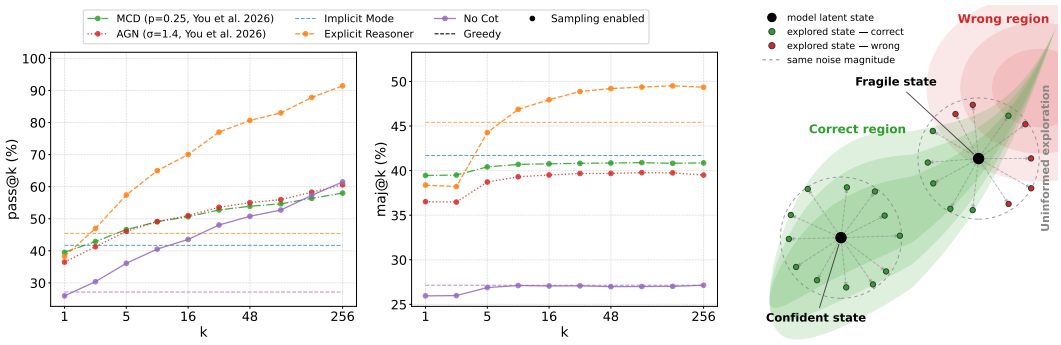
1 INTRODUCTION

Large Language Models (LLMs) incorporate inference-time compute to solve complex tasks by deliberating reasoning steps in plain language (Zhang et al., 2025a). Parallel test-time scaling (PTTS) further amplifies this inference-time budget into multiple independent reasoning traces, which are then aggregated or ranked, translating computational budget into problem-solving gains (Wang et al., 2023; Cobbe et al., 2021).

Recent work (Hao et al., 2025; Shen et al., 2025; Suleymanzade et al., 2026; Zhu et al., 2025) shows that this multi-step inference can instead unfold inside continuous hidden states. In these models, a loop of latent thoughts replaces explicit token generation as the intermediate reasoning steps. Such models generate the latent chain by deterministic continuous dynamics with no decoder sampling at each step, so it carries no token-level distribution. This eliminates the native mechanism to sample or rank alternative reasoning paths that PTTS in the explicit setting relies on (Wang et al., 2023). We primarily study models with *unconstrained* thoughts, free of the embedding space, and extend our analysis to *vocabulary-anchored* models, where each thought is a weighted combination of token embeddings (Zhang et al., 2025c; Tan et al., 2025; Deng et al., 2026) (Appendix G.1).

A growing line of work compensates for these deterministic dynamics by injecting heuristic cumulative perturbations, Monte-Carlo dropout (MC-dropout) and additive Gaussian noise (AGN), at every step of the latent chain (You et al., 2026); we call this *full-chain perturbation*. You et al. (2026) report that the two mechanisms exhibit "distinct exploration dynamics" in latent space, and together with Wang et al. (2026c) observe pass@k rising with the sampling budget. These works frame the perturbed latent chain as the continuous-space analogue of an explicit CoT trace, with every perturbed

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.



(a) (b)

Figure 1: **(a)** pass@k (left) and maj@k (right) for CODI on GPT-2. Full-chain perturbation and a No-CoT baseline raise pass@k, but only explicit reasoning lifts maj@k above its greedy accuracy. **(b)** Uninformed perturbation maintains confident states but displaces borderline, fragile ones.

chain treated as an independent trace that follows a different line of reasoning (You et al., 2026; Wang et al., 2026c;b). Aggregating enough of these samples should, in principle, let PTTS work in the latent setting like its explicit counterpart.

This analogy has not been tested at the level of the reasoning itself. Given the uninterpretability of latent reasoning states (Aswal et al., 2026), it is not clear whether perturbation genuinely explores different reasoning trajectories or merely surfaces as answer diversity with no reasoning-level exploration behind it, since a rising pass@k curve cannot tell the two apart. Existing evidence remains indirect: perturbed chains are observed to be geometrically indistinguishable and lack the inductive biases needed to deliberate among them (Wang et al., 2026c).

We demonstrate that the pass@k gains of full-chain perturbation do not carry over to the majority vote. In the representative case of Figure 1a, pass@k keeps rising with the sample budget under both methods (You et al., 2026; Wang et al., 2026c) and passes the greedy accuracy of the same model within a few samples, reflecting the diversity that perturbation adds. However, the majority vote over the same samples does not follow this trend and saturates at or below the no-sampling baseline, despite the substantial additional reasoning tokens. The same holds for every unconstrained latent method we tested (Section 3). Under comparable settings, explicit reasoning behaves differently, as its majority vote rises above its own greedy accuracy. We complement this with a simple No-CoT baseline that answers directly with temperature sampling, and observe the same pattern as latent perturbation. The pass@k gains of full-chain perturbation therefore cannot by themselves be read as reasoning-level exploration.

This contrast points to a possible structural limitation of unconstrained-thought models. We hypothesize that perturbation does not explore alternative reasoning paths, since a larger sampling budget would otherwise be expected to improve the majority vote, as it does for explicit CoT. We argue that explicit reasoning samples from a belief over reasoning paths, whereas the latent chain is trained primarily to reach the correct answer along a single deterministic trajectory. There is then no distribution over alternative chains to sample from. We investigate this in our formal analysis (Section 4) and show that perturbing the hidden state, even at a single step, changes the output distribution according to the geometry of the output head under the noise, rather than an internal belief over reasoning paths. Accordingly, a correct answer discovered by perturbation is unlikely to be explained by the model likelihood.

Our empirical results are consistent with this account (Sections 3.3 and 3.4). A correct answer reached through perturbation carries no measurable gain in confidence, whereas a correct answer discovered by resampling explicit reasoning does. The answers that perturbation changes are largely those on which the unperturbed chain was already borderline and low-confidence (Figure 1b), so much of the diversity it adds is spurious. Following our formal analysis, the propagated effect of full-chain perturbation appears largely unnecessary, as it can be reproduced with a frozen chain and a single calibrated perturbation at the answer generation phase. This suggests that full-chain perturbation guides no internal exploration in the latent space.

Our contributions are as follows.

- We show that every full-chain perturbation method we tested adds diversity without discovery for unconstrained thoughts: pass@k rises while the majority vote stays at the greedy accuracy.
- We compare latent and explicit PTTS under fair settings, on models that reason in either mode on one backbone. The same model succeeds at exploration in the explicit mode but fails in the latent mode.
- We show that correct answers reached through latent perturbation carry no gain in confidence, unlike those found by explicit reasoning.
- We characterize perturbation formally and show that its outcome is set by the geometry of the weights under the noise, so it carries no guarantee of favoring the correct answer.
- Guided by this analysis, we show that a single calibrated perturbation at the answer generation phase reproduces full-chain perturbation.

2 Preliminaries and Related Work

2.1 Explicit reasoning and parallel test-time scaling

Let $x_{1:Q}$ be the question tokens, $E(\cdot)$ the token embedding applied element-wise, and f_θ the transformer forward function. Generation extends this sequence, so for $t \geq Q$ we write $x_{1:t}$ for the question followed by the tokens produced so far. Standard autoregressive generation computes hidden states and a token distribution at every step:

$$h_t = f_\theta\big(E(x_{1:t})\big), \qquad \mathbb{P}_\theta(x_{t+1} \mid x_{1:t}) = \mathrm{Softmax}(W h_t/\tau), \tag{1}$$

with W the output head and τ the decoding temperature. An explicit CoT trace comprises R reasoning tokens $r_{1:R}$ followed by A answer tokens $y_{1:A}$. Writing $\oplus$ for sequence concatenation and $u_{1:R+A} = r_{1:R} \oplus y_{1:A}$, a trace is a sample from the factorized distribution $\mathbb{P}_\theta(u_{1:R+A} \mid x_{1:Q}) = \prod_{t=1}^{R+A} \mathbb{P}_\theta(u_t \mid x_{1:Q}, u_{1:t-1})$, which represents the belief of the model over reasoning paths. Writing x for the question, $r = r_{1:R}$ for a path and a for the final answer after normalization, this belief is the trajectory law $\rho = \mathbb{P}_\theta(r \mid x)$, and marginalizing it gives the answer distribution $\mathbb{P}_\theta(a \mid x) = \sum_r \mathbb{P}_\theta(r \mid x)\,\mathbb{P}_\theta(a \mid x, r)$ that Section 4 builds on. Sampling N such traces $\{u^{(j)}\}_{j=1}^N$ supplies the diversity that PTTS exploits (Zhang et al., 2025a), aggregating them by majority vote, best-of-N under a learned reward model, or beam search (Cobbe et al., 2021).

2.2 Latent reasoning

Latent reasoning replaces the discrete token-level feedback loop with a continuous one. We write z_i for the latent state at step i and reserve h_t for the hidden states of Eq. (1). A beginning-of-thought (BOT) marker opens the latent loop, after which the model feeds its own hidden state back as the next input embedding rather than sampling a token, through a *thought head* π, for $i = 1, \ldots, T$:

$$h_{i+1} = f_\theta\big(E(x_{1:Q}) \oplus E(BOT) \oplus z_{1:i}\big), \qquad z_{i+1} = \pi(h_{i+1}). \tag{2}$$

Coconut (Hao et al., 2025) takes π to be the identity, so $z_i = h_i$, whereas CODI and MUX (Shen et al., 2025; Suleymanzade et al., 2026) learn an MLP. In CoLaR (Tan et al., 2025) and ReGuLaR (Wang et al., 2026a) the thought head is generative, so the next thought is sampled, $z_{i+1} \sim \pi(\cdot \mid h_{i+1})$, rather than computed. An end-of-thought (EOT) marker closes the loop after T steps and returns the model to text space, where the answer is decoded from the last hidden state as in Eq. (1). The chain length T is typically only a fraction of the length R of an explicit trace. No softmax is applied at any of the T latent steps, so for a deterministic thought head the chain is a deterministic function of $x_{1:Q}$ and that sampling mechanism is unavailable.

We categorize latent reasoning models by the space their thoughts occupy: *vocabulary-anchored* thoughts are weighted combinations of token embeddings (Zhang et al., 2025c), whereas *unconstrained* thoughts are hidden states free of the embedding space. The two groups behave differently under perturbation (Appendix G.1). We focus on unconstrained thoughts, which offer a significantly higher representational bandwidth (Zhu et al., 2025).

The training methodology of some of these models gives rise to a further property. CODI (Shen et al., 2025) introduced an alignment loss that trains a shared backbone to distill the hidden states of explicit reasoning into the latent chain. As a consequence, the multi-task trained checkpoint retains its explicit mode, and the BOT and EOT markers switch it into latent reasoning. We call such a model *dual-mode*, since one checkpoint reasons in either the explicit or the latent mode. This ability is not a design goal of latent reasoning models, and SIM-CoT and MUX (Wei et al., 2025; Suleymanzade et al., 2026) extend the recipe and inherit it. We exploit it to compare both modes on one backbone under comparable settings. Appendix C.1.1 describes the dual-mode models we use, and Appendix C.1.2 extends the categorization above into a full taxonomy.

2.3 Full-chain perturbation

To recover sampling diversity without a token-level distribution, recent latent PTTS work injects stochasticity into the latent autoregression itself (You et al., 2026; Wang et al., 2026c). Two mechanisms have emerged, both applied at every latent thought step. MC-dropout keeps dropout active at inference, sampling a binary mask $m^{(j)} \sim \mathrm{Bernoulli}(p)$ over the weights $\theta^{(j)}$, so each sample j draws a different weight configuration:

$$h_{i+1}^{(j)} = f_{\theta^{(j)}}\big(E(x_{1:Q}) \oplus E(BOT) \oplus z_{1:i}^{(j)}\big), \qquad z_{i+1}^{(j)} = \pi\big(h_{i+1}^{(j)}\big). \tag{3}$$

AGN instead adds isotropic Gaussian noise directly to the latent thought:

$$\varepsilon_i^{(j)} \sim \mathcal{N}(0, \sigma^2 I), \qquad z_i^{(j)*} = z_i^{(j)} + \varepsilon_i^{(j)}, \qquad h_{i+1}^{(j)} = f_\theta\big(E(x_{1:Q}) \oplus E(BOT) \oplus z_{1:i}^{(j)*}\big). \tag{4}$$

Wang et al. (2026b) keep the same per-step injection but learn the perturbation adaptively at each step, in place of a fixed heuristic.

3 Comparing latent and explicit exploration

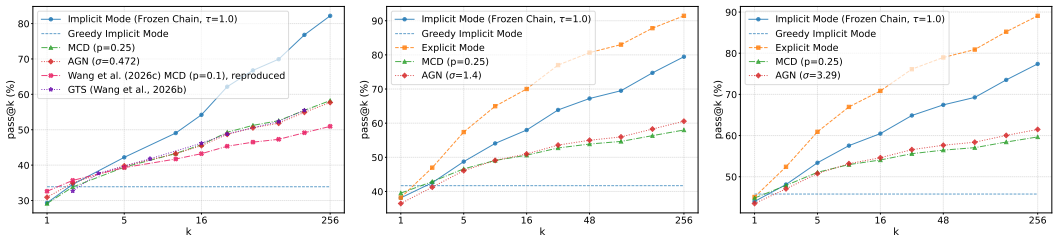
We examine how exploration by full-chain perturbation behaves next to exploration by explicit reasoning. We first extend the comparison of Figure 1a to a broader set of sampling strategies and latent reasoning models. We then inspect the confidence of the predicted answers before and after exploration in both reasoning paradigms (Section 3.3). Finally, based on our observations of chain perturbation behavior, we ask what the perturbed chain contributes, by testing whether a single calibrated perturbation at the answer generation phase reproduces full-chain perturbation (Section 3.4).

3.1 Parallel sampling strategies

In our experimental setup, each strategy draws N parallel samples per question, and the strategies differ in where stochasticity is introduced. Given these differences, we characterise each strategy by how its reasoning-token cost grows with N.

- *Explicit CoT* samples N independent traces in the explicit mode (Eq. (1)), consuming $O(N\bar{R})$ reasoning tokens, with $\bar{R}$ the average trace length. For a fair comparison with the latent strategies, these traces come from the same dual-mode model rather than a separately trained explicit-CoT model. This strategy is not available for Coconut, which has no explicit mode.
- *Full-chain MC-dropout* and *full-chain AGN* regenerate the chain for every sample, with an independent perturbation at each latent step (You et al., 2026; Wang et al., 2026c) (Eqs. (3) and (4)), consuming $O(TN)$ latent tokens.
- *Frozen chain* runs the latent chain of Eq. (2) once in the implicit mode and resamples only the answer decoder, by enabling temperature sampling of the answer. It consumes $O(T)$ latent tokens for any N, a constant budget. We use it as a trivial baseline against full-chain perturbation.

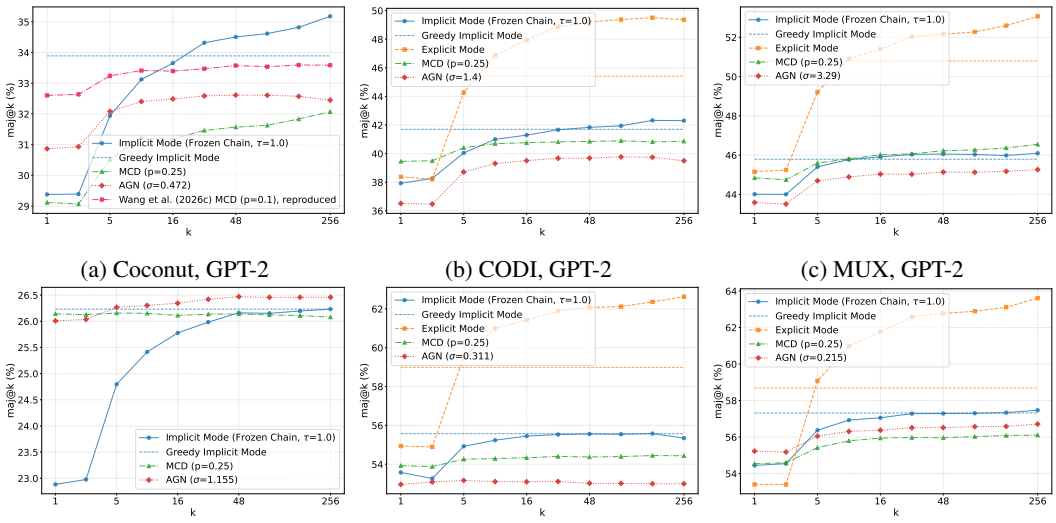
Following You et al. (2026) and Wang et al. (2026c), the perturbed strategies decode the answer greedily, so the perturbation is their only source of stochasticity. The AGN scale is set in proportion to the magnitude of the latent thoughts of each model, and all noise levels are given in Appendix F. We evaluate all strategies by pass@k and maj@k (definitions in Appendix A), and compare maj@k against greedy decoding as a single-sample reference.



(a) Coconut, GPT-2 (b) CODI, GPT-2 (c) MUX, GPT-2

(d) Coconut, Llama-3.2-1B (e) CODI, Llama-3.2-1B (f) MUX, Llama-3.2-1B

Figure 2: pass@k as the sample budget k grows, for the sampling strategies of Section 3.1. Coconut is not dual-mode, so it has no explicit-mode curve.



(a) Coconut, GPT-2 (b) CODI, GPT-2 (c) MUX, GPT-2

(d) Coconut, Llama-3.2-1B (e) CODI, Llama-3.2-1B (f) MUX, Llama-3.2-1B

Figure 3: maj@k for the same models and strategies as Figure 2.

3.2 Latent perturbation adds diversity but not discovery

Figure 2 compares the frozen chain with full-chain MC-dropout and AGN (Section 3.1) on Coconut, CODI and MUX, each on GPT-2 and Llama-3.2-1B, for sample budgets up to $k = 256$. The Coconut and CODI panels also include the results reported by Wang et al. (2026c) and Wang et al. (2026b). Resampling the decoder over one frozen chain keeps covering new questions, whereas both perturbation methods grow far more slowly. As k grows, the frozen chain leads in every panel, at the constant reasoning cost of a single chain. Perturbing the chain is therefore not needed for the pass@k gains reported for latent PTTS (You et al., 2026; Wang et al., 2026c).

Figure 3 extends the maj@k panel of Figure 1a to the same models. Only reasoning in the explicit space lifts maj@k clearly above its own greedy accuracy. Every latent strategy stays within about one point of the greedy accuracy of its model, whether it samples the decoder or perturbs the chain.

Table 1: First-answer-token entropy (nats) before and after exploration, for draws whose answer flipped (Llama-3.2-1B, 64 draws per question). AUC is how well the greedy entropy ranks which draws flip, with 0.5 carrying no signal. Bold marks an entropy drop in Δ and, in the AUC columns, a latent value above the explicit CoT value for the same model. Full breakdown in Appendix D.

		Flipped to wrong				Flipped to correct			
Model	Noise	Before	After	Δ	AUC	Before	After	Δ	AUC
CODI	MC-dropout	1.263	1.546	+0.283	**0.91**	1.444	1.540	+0.096	**0.60**
CODI	AGN	1.177	1.607	+0.430	**0.90**	1.472	1.701	+0.229	**0.59**
MUX	MC-dropout	1.230	1.861	+0.631	**0.87**	1.520	1.641	+0.121	**0.63**
MUX	AGN	1.377	1.751	+0.374	**0.89**	1.649	1.662	+0.012	**0.59**
SIM-CoT	MC-dropout	1.544	1.926	+0.383	**0.91**	1.597	1.721	+0.124	**0.61**
SIM-CoT	AGN	1.755	2.010	+0.254	**0.93**	1.681	1.808	+0.127	**0.59**
CODI	Explicit CoT	0.483	0.928	+0.445	0.73	0.881	0.567	**−0.314**	0.57
MUX	Explicit CoT	0.554	1.043	+0.488	0.73	1.000	0.597	**−0.403**	0.56
SIM-CoT	Explicit CoT	0.451	1.026	+0.575	0.72	0.917	0.576	**−0.340**	0.58

The frozen chain matches full-chain perturbation on the majority vote and leads it on pass@k, at the cost of a single chain, whereas full-chain perturbation pays for a new chain per sample without a gain in either. A larger sampling budget therefore widens coverage for latent reasoning, while its majority vote stays at the greedy accuracy.

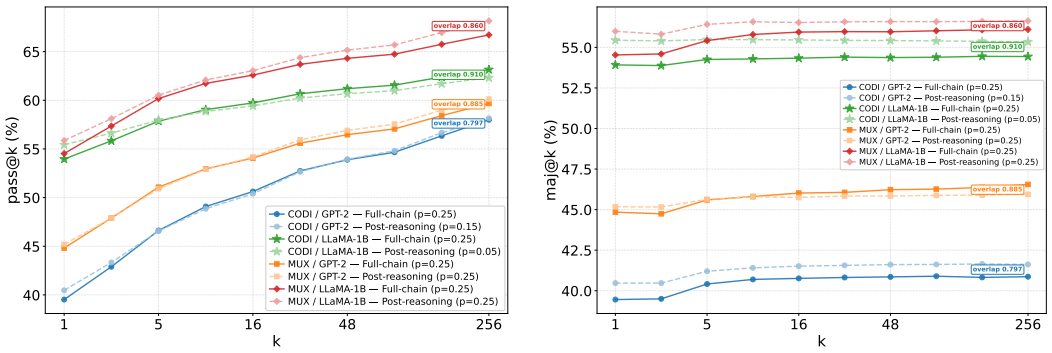
SIM-CoT and LOTUS follow the same pattern, with additional results in Appendix G. The vocabulary-anchored models CoLaR and Latent-SFT behave differently, with full-chain perturbation ahead of the frozen chain. Perturbing explicit CoT in the same way produces a similar pattern, which places these models closer to explicit reasoning than to unconstrained thoughts (Appendix G.1). Our formal analysis accounts for this distinction. Under a simplifying assumption, perturbing a vocabulary-anchored thought approximates sampling a token from the trained next-token distribution of the model, whereas an unconstrained thought has no such reading (Appendix B, Eq. (30)). We therefore focus on unconstrained thoughts, which we choose for their higher representational bandwidth (Zhu et al., 2025).

3.3 Confidence of discovered answers

We use the entropy of the first answer token as the confidence of a prediction. For each draw we compare the entropy of the greedy prediction with that of the perturbed one. We extend this protocol to explicit reasoning so that the two sampling paradigms can be compared on equal terms. In the explicit mode, the greedy trace provides the reference prediction and temperature sampling provides the exploration.

Table 1 reports the first-answer-token entropy of draws whose answer flipped. Comparing the correct answers discovered with those lost separates the two paradigms. Under explicit CoT, entropy rises when a correct answer is lost and falls by 33% on average when one is discovered. Under latent noise, entropy rises in both cases, by 4% on average for a discovered answer. The discovered correct answers therefore carry no confidence gain and remain unexplained by confidence. Across all four outcome buckets (Appendix D), latent perturbation raises entropy on average, so the answers it produces are less confident than the ones the clean chain produces.

The AUC column reports how often a draw about to flip already carried a higher greedy entropy than a draw of the same greedy correctness that kept its answer. Under latent noise, a draw that leaves a correct answer is separated from one that keeps it with AUC 0.87 to 0.93, against 0.72 to 0.73 under explicit sampling. The answers that latent perturbation changes are therefore mostly borderline answers of the greedy chain, which is the structural constraint we argue for: a perturbation with no access to the reasoning reaches whatever already sits near the decision boundary. For the answers that exploration repairs, the greedy entropy is a weaker predictor in both modes, at 0.56 to 0.63, although the latent values stay above the explicit ones in every pair. This is likely because a question the greedy chain answers wrongly admits one correct answer among many wrong alternatives, so reaching it is rare and the greedy entropy gives little indication of when it happens. Combined with



(a) (b)

Figure 4: pass@k **(a)** and maj@k **(b)** of full-chain MC-dropout against the single calibrated perturbation at the first answer token.

the entropy increase under latent noise, a repaired answer in the latent mode is not distinguishable by confidence from the answers the same noise gets wrong.

3.4 One calibrated perturbation reproduces the whole chain

We now ask what the propagated noise contributes. If perturbing every latent step moved the chain onto a different line of reasoning, then a perturbation that never touches the chain should not be able to imitate it. The *single calibrated perturbation*, which we introduce, keeps the chain untouched and perturbs the model once in the answer generation phase, at the token where the explicit and latent modes are aligned during training (Shen et al., 2025), which is the position of the first answer token, consuming $O(T)$ latent tokens. We apply this experiment only with MC-dropout on the model weights, since the latent thoughts that AGN perturbs are absent at the answer generation phase. We calibrate its rate so that its pass@k curve matches that of full-chain MC-dropout, and test whether it also reproduces the fine-grained answer statistics of full-chain MC-dropout through the overlap of their answer distributions on each question q,

$$O_q = \sum_a \min\big(P_q(a), Q_q(a)\big), \tag{5}$$

where P_q and Q_q are the empirical answer frequencies of the two strategies. For a reliable evaluation, we normalise O_q by its bootstrap-estimated ceiling under finite sampling, and report 95% bootstrap confidence intervals over questions.

Figure 4 compares full-chain MC-dropout with the single calibrated perturbation of Section 3.1 on CODI and MUX, each on GPT-2 and Llama-3.2-1B. Every calibrated rate almost matches the pass@k growth of its target over the whole sample budget, spending far fewer tokens to do so (Figure 4a). The two strategies also track each other on the majority vote, where both stay at the greedy accuracy (Figure 4b). To check that the predictions behind those curves are also distributed similarly, we compare the answer distributions of the two strategies question by question. Their noise-adjusted overlap (Eq. (5)) reaches 0.80 to 0.91: the two strategies agree not only on how many questions are solved but also on which answers each question receives. The calibrated rate equals the full-chain rate for MUX on both backbones ($0.25 \to 0.25$), but is lower for CODI ($0.25 \to 0.15$ / $0.25 \to 0.05$), which is likely related to MUX's per-thought supervision (Suleymanzade et al., 2026), making its answer sensitive to noise at a single step as much as across the whole chain.

4 Characterizing exploration dynamics

We now formalize the findings of Section 3, with proofs in Appendix B. We restrict the analysis to the first answer token for simplicity, which we expect to be a reasonable proxy for the short numeric answers of math reasoning (Cobbe et al., 2021). The output head W maps the final state h to the logits $s = Wh$. These logits give the clean distribution $p_0 = \mathrm{Softmax}(s)$ with its corresponding

greedy answer $g = \arg\max_a s_a$. Each strategy induces a distribution q over answers, and the two metrics read two different features of it. Let a^* denote the correct answer and g_q the mode of q. For k independent draws on one question,

$$\text{pass@}k = 1 - \big(1 - q(a^*)\big)^k, \qquad \text{maj@}k \xrightarrow[k\to\infty]{} \mathbb{I}\{g_q = a^*\}, \tag{6}$$

where the limit follows from the consistency of majority voting (Cordero-Encinar & Duncan, 2025). Coverage is set by the mass $q(a^*)$ and the vote by the mode g_q. The mass on the correct answer can therefore grow while the mode stays on a wrong answer.

We build the analysis on one property of a transform. We call a transform *mode-preserving* when the distribution it induces keeps g as its mode. We first identify simple transforms that have this property. We then show that perturbation is not guaranteed to have it, and that when it moves the mode, nothing guarantees a move toward the correct answer. We start with the simplest case, noise added directly to the logits, and show that it cannot raise the vote above greedy accuracy.

Proposition 1 (Logit noise is mode-preserving)**.** *Adding noise to the logits of the first answer token is mode-preserving whenever the noise law is permutation invariant and has an everywhere-positive density. Permutation invariance means that relabelling the answers leaves the law of the noise unchanged, so the noise favors no answer. Temperature sampling is one such transform. Dividing the logits by a temperature keeps their order, and sampling at that temperature is equivalent to adding independent Gumbel noise to the scaled logits and taking the argmax (Jang et al., 2017). Under any mode-preserving transform, maj@k converges to the greedy accuracy (Propositions 4 and 5, Eq.* (14)*).*

Resampling the decoder of a frozen chain therefore widens coverage but cannot raise the vote above greedy accuracy. Section 3.2 reports this pattern, with pass@k growing with k while maj@k saturates at greedy accuracy. Even under such noise, the greedy answer is not equally stable on every question. The smaller the gap between its logit and that of its closest competitor, the higher the floor on the probability that the noise displaces it (Proposition 6). This is consistent with the borderline answers of Section 3.3.

Proposition 2 (Hidden-state noise follows the head geometry)**.** *Let $\tilde{h} = h + \varepsilon$ with $\varepsilon \sim \mathcal{N}(0, \Sigma)$, decoded greedily. Isotropic noise, $\Sigma = \sigma^2 I$, perturbs every direction of the hidden space by the same amount, so it favors no direction there. The output head removes this symmetry. The logit noise $W\varepsilon$ then has covariance $\sigma^2 W W^\top$. The noise on answer a has variance $\sigma^2 \|w_a\|^2$, with w_a the row of W for a, and the noise on two answers is correlated through the overlap of their rows. Different answers therefore receive noise of different size, and the mode of the induced distribution can differ from g. That mode is set by the geometry of W and the noise law rather than by the distribution the model was trained to predict, so nothing ties it to a^*. The expected log-probability of every answer drops by the same amount, so no answer gains support (Eqs.* (11)*,* (12) *and* (16)*).*

Proposition 2 covers the simplest case, noise added to the state that enters the head. Full-chain perturbation injects noise at every latent step instead, so the perturbed signal passes through further computation before it reaches the head. The conclusion that nothing ties the mode to the correct answer extends to this case, and to any other perturbation scheme. At the first answer token the whole chain reaches the decision only through the state entering the head, so a perturbation scheme is summarized by the law it induces there, whatever that law is (Proposition 7). The number of injection points and their positions enter only through that law. Within this first-token setting, nothing in that law refers to the correct answer, so the perturbation carries no guarantee of favoring it.

A single injection and a full chain of injections therefore differ only in the law they induce on the final state. Whether a single injection can reproduce the full chain is an empirical question, and Section 3.4 finds that it can once its strength is calibrated. Neither scheme keeps the protection of Proposition 5. That protection requires the noise that reaches the logits to be permutation invariant, and any map the noise passes through, such as W itself, breaks this in general (Remark 2).

Proposition 3 (Only explicit reasoning samples a belief)**.** *Explicit CoT samples its trace from the trajectory law $\mathbb{P}_\theta(r \mid x)$ of the model, so its vote converges to the mode of the marginal*

$$\mathbb{P}_\theta(a \mid x) = \sum_r \mathbb{P}_\theta(r \mid x)\, \mathbb{P}_\theta(a \mid x, r),$$

which can differ from g (Eq. (26)). An unconstrained latent chain is a deterministic function of the question, so its trajectory law is a point mass. Its samples come either from the frozen chain or from a perturbation law over the T noise draws. Training shapes the trajectory law of explicit CoT, since generation and answer decoding share θ. The perturbation law is instead fixed by a single hyperparameter, p or σ, before decoding, so training on correct answers has no channel through which to reach it (Remark 3, Eq. (27)).

Explicit reasoning thus samples from a belief over reasoning paths, whereas full-chain perturbation supplies variety with no belief behind it. These results hold for unconstrained thoughts. We complement the analysis for vocabulary-anchored thoughts. For Latent-SFT, a latent thought lives in the vocabulary simplex, and a simplifying assumption extends this reading to CoLaR. For both models, a perturbation either nudges the thought toward semantically similar tokens (Eq. (29)) or samples a relevant token from the next-token distribution of the model. The first keeps the thought among the tokens the model already weighs. The second recovers the marginal of explicit reasoning (Eqs. (30) and (31)).

Together, these statements account for the findings of Section 3. Decoder resampling is mode-preserving, so its vote saturates at greedy accuracy, which is the flat maj@k of Section 3.2 (Proposition 1). A perturbation carries no guarantee of favoring the correct answer, which is consistent with the absence of a confidence gain for the answers it discovers in Section 3.3 (Propositions 2 and 7). Explicit reasoning samples from a trajectory law that training shapes, and latent perturbation does not, which separates the two modes throughout (Proposition 3). Since only the induced law matters, a single injection can reproduce the full chain whenever it induces a similar law, which is what Section 3.4 finds (Proposition 7).

5 Conclusion

In this work, we examined perturbation-based methods for latent PTTS on models with unconstrained thoughts. Our investigation shows that the diversity these methods add does not translate into consistent discovery. Compared with explicit reasoning on the same backbone, the correct answers that perturbation discovers carry no gain in confidence and remain unexplained by the confidence of the model. Our formal analysis offers an explanation. Explicit reasoning samples from a trajectory law that training shapes, whereas a perturbation induces an answer distribution fixed by the geometry of the weights under the noise, with no guarantee of favoring the correct answer. Accordingly, a single calibrated perturbation at the answer generation phase reproduces full-chain MC-dropout, which suggests that propagating the noise through the chain adds no exploration. Our study covers math reasoning on GSM8K, and extending it to other tasks is left for future work. More broadly, our results suggest that latent exploration may need to be induced during training, for instance with multiple parallel trajectories, so that the model learns a distribution to sample from.

AI Use Statement

We used generative AI assistants in a limited, non-research capacity: for language choice, paraphrasing, and proofreading of author-written text, and to help explore the related literature. All research contributions, including the problem formulation, theoretical results, experimental design, implementation, and experiments, were conceived, carried out, and verified by the authors. Every reference surfaced with AI assistance was read and checked against its original source before being cited. The authors take full responsibility for the final content of this work.

Ethics Statement

This work analyzes the test-time behavior of existing latent reasoning models on GSM8K, a publicly available benchmark of grade-school math word problems. The data contain no personally identifiable information, and the work involves no human subjects. All models and checkpoints are used under their original licenses. We identify no potential harms, conflicts of interest, or fairness, privacy, or legal compliance concerns arising from this work.

Reproducibility Statement

All theoretical results are stated and proved in Appendix B. The evaluation metrics are defined in Appendix A. Implementation details, including the models, checkpoints, chain lengths, decoding settings, sample budgets, and seeds, are given in Appendix E. The noise calibration procedure is given in Appendix F, and the full entropy analysis in Appendix D. Anonymized code to reproduce all experiments is available at `https://anonymous.4open.science/r/Div-wo-Dis`.

References

Aleksei Arzhantsev, Otmane Sakhi, and Nicolas Chopin. Self-Consistency via Marginal Sharpening. arXiv preprint, 2026.

Darpan Aswal, Thomas Palmeira Ferraz, Yongxin Zhou, and Maxime Peyrard. Observable patterns are not explanations: A causal-geometric analysis of latent reasoning models, 2026. URL http://arxiv.org/abs/2606.12689. arXiv:2606.12689 [cs.CL].

Natasha Butt, Ariel Kwiatkowski, Ismail Labiad, Julia Kempe, and Yann Ollivier. Soft tokens, hard truths, 2025. URL https://arxiv.org/abs/2509.19170. arXiv:2509.19170.

Shuochen Chang, Tong Bai, Xiaofeng Zhang, Qianli Ma, Qingyang Liu, Zhaohe Liao, Yibo Miao, and Li Niu. Unlocking the black box of latent reasoning: An interpretability-guided approach to intervention. In Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics, pp. 34019–34032, 2026. doi: 10.18653/v1/2026.acl-long.1568.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating Large Language Models Trained on Code. 2021.

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

Paula Cordero-Encinar and Andrew B. Duncan. Certified Self-Consistency: Statistical Guarantees and Test-Time Training for Reliable Reasoning in LLMs. arXiv preprint, 2025.

Yingqian Cui, Zhenwei Dai, Bing He, Zhan Shi, Hui Liu, Rui Sun, Zhiji Liu, Yue Xing, Jiliang Tang, and Benoit Dumoulin. How do latent reasoning methods perform under weak and strong supervision?, 2026. URL http://arxiv.org/abs/2602.22441. arXiv:2602.22441 [cs.AI].

Jingcheng Deng, Liang Pang, Zihao Wei, Shicheng Xu, Zenghao Duan, Kun Xu, Yang Song, Huawei Shen, and Xueqi Cheng. LLM latent reasoning as chain of superposition, 2026. URL http://arxiv.org/abs/2510.15522. arXiv:2510.15522 [cs.CL].

Ying Fan, Anej Svete, and Kangwook Lee. Bridging the gap between latent and explicit reasoning with looped transformers, 2026. URL http://arxiv.org/abs/2606.31779. arXiv:2606.31779 [cs.LG].

Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In International Conference on Machine Learning, 2016.

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, 2025. URL http://arxiv.org/abs/2412.06769. arXiv:2412.06769 [cs].

Eric Jang, Shixiang Gu, and Ben Poole. Categorical Reparameterization with Gumbel-Softmax. In International Conference on Learning Representations, 2017.

Michael Rizvi-Martel, Guillaume Rabusseau, and Marius Mosbach. The illusion of superposition? A principled analysis of latent thinking in language models, 2026. URL https://arxiv.org/abs/2604.06374. arXiv:2604.06374.

Zhenyi Shen, Hanqi Yan, Linhai Zhang, Zhanghao Hu, Yali Du, and Yulan He. CODI: Compressing chain-of-thought into continuous space via self-distillation, 2025. URL http://arxiv.org/abs/2502.21074. arXiv:2502.21074 [cs.CL].

Ayhan Suleymanzade, Halil Alperen Gozeten, Michael Bronstein, İsmail İlkan Ceylan, and Jinwoo Kim. MUX: Continuous reasoning via multiplexed tokens, 2026. URL https://arxiv.org/abs/2607.18264.

Wenhui Tan, Jiaze Li, Jianzhong Ju, Zhenbo Luo, Ruihua Song, and Jian Luan. Think silently, think fast: Dynamic latent compression of LLM reasoning chains, 2025. URL https://arxiv.org/abs/2505.16552. arXiv:2505.16552.

Fanmeng Wang, Haotian Liu, Guojiang Zhao, Hongteng Xu, and Zhifeng Gao. ReGuLaR: Variational latent reasoning guided by rendered chain-of-thought, 2026a. URL https://arxiv.org/abs/2601.23184. arXiv:2601.23184.

Minghan Wang, Ye Bai, Thuy-Trang Vu, Ehsan Shareghi, and Gholamreza Haffari. GTS: Inference-time scaling of latent reasoning with a learnable Gaussian thought sampler, 2026b. URL https://arxiv.org/abs/2602.14077. arXiv:2602.14077.

Minghan Wang, Thuy-Trang Vu, Ehsan Shareghi, and Gholamreza Haffari. Towards inference-time scaling for continuous space reasoning. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), Findings of the Association for Computational Linguistics: ACL 2026, pp. 26842–26856, San Diego, California, United States, July 2026c. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.1338. URL https://aclanthology.org/2026.findings-acl.1338/.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In International Conference on Learning Representations, 2023. URL https://arxiv.org/abs/2203.11171.

Xilin Wei, Xiaoran Liu, Yuhang Zang, Xiaoyi Dong, Yuhang Cao, Jiaqi Wang, Xipeng Qiu, and Dahua Lin. SIM-CoT: Supervised implicit chain-of-thought, 2025. URL https://arxiv.org/abs/2509.20317. arXiv:2509.20317.

Runyang You, Yongqi Li, Meng Liu, Wenjie Wang, Liqiang Nie, and Wenjie Li. Parallel test-time scaling for latent reasoning models, 2026. URL http://arxiv.org/abs/2510.07745. arXiv:2510.07745 [cs].

Qiyuan Zhang, Fuyuan Lyu, Zexu Sun, Lei Wang, Weixu Zhang, Wenyue Hua, Haolun Wu, Zhihan Guo, Yufei Wang, Niklas Muennighoff, Irwin King, Xue Liu, and Chen Ma. A survey on test-time scaling in large language models: What, how, where, and how well?, 2025a. URL http://arxiv.org/abs/2503.24235. arXiv:2503.24235 [cs].

Yuyi Zhang, Boyu Tang, Tianjie Ju, Sufeng Duan, and Gongshen Liu. Do latent tokens think? A causal and adversarial analysis of chain-of-continuous-thought, 2025b. URL http://arxiv.org/abs/2512.21711. arXiv:2512.21711 [cs.CL].

Zhen Zhang, Xuehai He, Weixiang Yan, Ao Shen, Chenyang Zhao, Shuohang Wang, Yelong Shen, and Xin Eric Wang. Soft thinking: Unlocking the reasoning potential of LLMs in continuous concept space, 2025c. URL http://arxiv.org/abs/2505.15778. arXiv:2505.15778 [cs.CL].

Rui-Jie Zhu, Tianhao Peng, Tianhao Cheng, Xingwei Qu, Jinfa Huang, Dawei Zhu, Hao Wang, Kaiwen Xue, Xuanliang Zhang, Yong Shan, Tianle Cai, Taylor Kergan, Assel Kembay, Andrew Smith, Chenghua Lin, Binh Nguyen, Yuqi Pan, Yuhong Chou, Zefan Cai, Zhenhe Wu, Yongchi Zhao, Tianyu Liu, Jian Yang, Wangchunshu Zhou, Chujie Zheng, Chongxuan Li, Yuyin Zhou, Zhoujun Li, Zhaoxiang Zhang, Jiaheng Liu, Ge Zhang, Wenhao Huang, and Jason Eshraghian. A survey on latent reasoning, 2025. URL http://arxiv.org/abs/2507.06203. arXiv:2507.06203 [cs].

A Evaluation metrics

For a question with ground-truth answer a^* we draw N samples, decode an answer $a^{(j)}$ from each, and count the correct ones as $c = \sum_{j=1}^{N} \mathbb{I}\{a^{(j)} = a^*\}$. We report the unbiased pass@k estimator of Chen et al. (2021) and the self-consistency score maj@k of Wang et al. (2023):

$$\mathrm{pass@}k = \mathbb{E}_{\mathcal{D}}\left[1 - \binom{N-c}{k} \Big/ \binom{N}{k}\right], \qquad \mathrm{maj@}k = \mathbb{E}_{\mathcal{D}}\left[\mathbb{I}\{\mathrm{mode}(a^{(1)}, \ldots, a^{(k)}) = a^*\}\right], \tag{7}$$

where the expectation is over the evaluation set $\mathcal{D}$ and $\mathrm{mode}(\cdot)$ returns the most frequent answer among the k samples. The two credit different things: pass@k counts a question as solved when any one of the k samples is correct, whereas maj@k requires the most frequent answer among them to be correct.

B Sampling, perturbation, and the voting limit

In this section we formulate the mechanics of each sampling strategy of Section 3.1: what distribution it draws the answer from, and what a vote over k samples can reach under it. We consider the first answer token, both for simplicity and because it already carries most of a math-reasoning answer (Cobbe et al., 2021), the same convention used for entropy in Section 3.3. Answers are compared after normalization, so they range over a finite set.

Let x be a question and a^* its correct answer. The answer is decoded from a state h through the output head of Eq. (1). We write $s = Wh$ for the logits at that state and $p_0 = \mathrm{Softmax}(s)$ for the clean distribution over answers. The greedy answer is

$$g = \arg\max_a \, s_a, \tag{8}$$

which we assume unique, and we write $\mathrm{Acc}_{\mathrm{greedy}}$ for its accuracy over the evaluation set. Throughout, q is the distribution over answers that one run of a strategy returns. We write g_q for the mode of q and V_k for the most frequent answer among k draws.

Every strategy is one distribution over answers. Whatever a strategy does internally, it returns one answer per run. It is therefore described by the q it induces, and the two metrics read two different features of that single object. Coverage is decided by the mass $q(a^*)$ on the correct answer. The voting limit is decided by the mode g_q, which we also assume unique. For k independent draws, the two metrics take the forms of Eq. (6) per question, with $\mathrm{maj@}k = \Pr\left(V_k = a^*\right)$ before the limit. These are the population forms of the estimators in Eq. (7). The limit of $\mathrm{maj@}k$ is the consistency of majority voting, which we take as given from Cordero-Encinar & Duncan (2025). Strategies differ only in the q they induce. That is what we compute below.

B.1 Temperature keeps the greedy answer as the mode

Sampling the decoder of a frozen chain at temperature τ draws from $q_\tau(a) = \mathrm{Softmax}(s/\tau)_a$. Dividing by $\tau > 0$, exponentiating and normalizing are monotonically increasing maps. None of them changes the order of the logits:

$$s_i > s_j \iff q_\tau(i) > q_\tau(j). \tag{9}$$

Proposition 4 (Temperature preserves the mode)**.** *For every $\tau > 0$ the mode of q_τ is the greedy answer, $g_\tau = g$, and*

$$\mathrm{maj@}k \xrightarrow[k\to\infty]{} \mathrm{Acc}_{\mathrm{greedy}}. \tag{10}$$

Proof. By Eq. (9), q_τ ranks the answers in the same order as s, so its largest value sits at g. Averaging Eq. (6) over the evaluation set gives Eq. (10). □

Resampling the decoder of a frozen chain therefore converges on the answer greedy decoding already returns. This holds at any temperature. What converges is the vote itself: each new draw is still a sample from the same q_τ.

B.2 Perturbation induces a different distribution

A perturbation replaces the state h by a random state $\tilde{h}$ and decodes greedily from it. We take the perturbation to be centered on the clean state, $\mathbb{E}[\tilde{h}] = h$. This holds for Gaussian noise added to the vector entering the output head. It also holds for dropout applied at that same point with the usual rescaling (Gal & Ghahramani, 2016). Full-chain perturbation does not meet this condition, and Remark 2 gives its scope.

Every draw makes a hard decision. What decides q is how often the noise lands in the region where a wins, not how well a scores on average. Writing $\tilde{s} = W\tilde{h}$ for the perturbed logits,

$$q_{\mathrm{pert}}(a) = \Pr\Big(a = \arg\max_b \tilde{s}_b\Big), \qquad \mathbb{E}\big[\tilde{s}\big] = s. \tag{11}$$

The mean logits are those of the clean state. The mode of q_{pert} can still differ from the answer they rank first,

$$\arg\max_a \mathbb{E}\big[\tilde{s}_a\big] \;\neq\; g_{\mathrm{pert}} \quad \text{in general,} \tag{12}$$

because each draw decides before anything is averaged. Take $\tilde{h} = h + \varepsilon$ with $\varepsilon \sim \mathcal{N}(0, \Sigma)$. The perturbed logits then carry covariance $W\Sigma W^\top$, so g_{pert} is a property of the output-head geometry under the noise law. It is not a property of the distribution the model was trained to predict. Perturbation carries no guarantee of concentrating the vote on the correct answer.

Coverage and the voting limit read different features of q_{pert} and can move in opposite directions. The mass $q_{\mathrm{pert}}(a^*)$ can grow while the mode sits on a wrong answer. Figures 2 and 3 demonstrate this distinction between coverage and the majority vote, with pass@k rising while maj@k stays flat.

Remark 1 (Which way the limit moves is a measurement)**.** *Over the evaluation set, write*

$$D = \Pr\big(g = a^*,\ g_{\mathrm{pert}} \neq a^*\big), \qquad R = \Pr\big(g \neq a^*,\ g_{\mathrm{pert}} = a^*\big)$$

for the questions perturbation damages and the questions it repairs. Partitioning on whether greedy was correct gives

$$\lim_{k\to\infty} \mathrm{maj@}k\big|_{\mathrm{pert}} = \mathrm{Acc}_{\mathrm{greedy}} - D + R. \tag{13}$$

A limit below greedy accuracy means $D > R$. That is a measurement rather than a consequence of the noise being centered. A flat curve over the budgets we run does not establish it.

B.3 The space the noise enters decides the distribution it induces

Perturbation is not the opposite of sampling. Temperature sampling is itself a perturbation, by the Gumbel-max identity (Jang et al., 2017),

$$\arg\max_a\{s_a + \tau G_a\} \sim q_\tau, \qquad G_a \overset{\text{iid}}{\sim} \mathrm{Gumbel}(0,1). \tag{14}$$

What separates the two is not the presence of randomness. It is the distribution the randomness induces over answers, and the space the noise acts in decides that law.

Proposition 5 (Exchangeable logit noise keeps the greedy answer on top)**.** *Let $A = \arg\max_a(s_a + \eta_a)$, where the law of η is invariant under permutation of its coordinates and has an everywhere-positive density. Then $s_i > s_j \Rightarrow \Pr(A{=}i) > \Pr(A{=}j)$, so the mode of the induced distribution is g.*

Proof. Fix i and j with $s_i > s_j$. Swap the two coordinates η_i and η_j of a draw. The swap leaves the law of η unchanged. It also turns every draw on which j wins into one on which i wins, since i carries the larger logit. Every outcome that favors j therefore has a partner that favors i. Outcomes on which i wins both before and after the swap have no such partner, and the positive density gives them positive probability. The inequality is therefore strict. □

Proposition 6 (Smaller margin, higher floor on flip probability)**.** *Let g be the greedy answer and g' its closest competitor, $\gamma = s_g - s_{g'} \geq 0$. Under the hypothesis of Proposition 5, exchangeability gives $\eta_a - \eta_g$ the same distribution for every competitor $a \neq g$, since a permutation carrying any one pair of coordinates onto any other leaves the joint law of η unchanged. Write F for its cumulative distribution function, $F(t) = \Pr(\eta_a - \eta_g \leq t)$, the same F regardless of a. Then*

$$\Pr(g_{\text{pert}} \neq g) \;\geq\; 1 - F(\gamma), \tag{15}$$

and $1 - F$ is non-increasing, since F is non-decreasing.

Proof. If $\eta_{g'} - \eta_g > \gamma$ then $s_{g'} + \eta_{g'} > s_g + \eta_g$, which alone removes g from the arg max regardless of every other coordinate. So $\{\eta_{g'} - \eta_g > \gamma\} \subseteq \{g_{\text{pert}} \neq g\}$, and taking probabilities gives Eq. (15). □

A smaller clean margin therefore forces a higher floor on the flip probability: states with a closer competitor are provably at least as vulnerable to perturbation as states with a comfortable lead. Two things this does not show. First, the margin is close to a definition of confidence, so the result is close to a restatement of what being near the decision boundary already means; the content is that the bound is exact and monotone. Second, the argument is symmetric in correctness: it bounds how often g is displaced, not whether the state noise moves into is right or wrong. It does not by itself explain why perturbation converts correct answers to wrong ones more often than the reverse (Eq. (13)). In other words, that asymmetry is an empirical property of this model and dataset, not a consequence of margin alone.

Noise added independently to the logits satisfies the exchangeability condition at any scale. Mean-zero randomness that never sees a^* therefore does not displace the greedy answer on its own. Being uninformative about correctness is not what makes a perturbation harmful. Noise on the hidden state does not satisfy the condition. It reaches the logits as $W\varepsilon$, whose covariance $W\Sigma W^\top$ is not permutation invariant in general. The two cases differ in the space the noise acts in. This analysis concerns noise on the logits alone, and it does not settle by itself how models with unconstrained thoughts compare with models with vocabulary-anchored thoughts; a later subsection raises that comparison to the level of the trajectory, where it belongs. The next subsection asks a narrower question first, one that only needs the single-state analysis above.

B.4 Expected log-score: an exact gap that does not settle the vote

Perturbation's effect on an answer's log-probability can be characterized directly, without deciding which answer wins the vote. Let $p_{\tilde{h}} = \mathrm{Softmax}(\tilde{s})$ be the predictive distribution of a perturbed draw. Expanding the KL divergence between p_0 and $p_{\tilde{h}}$ and using $\mathbb{E}[\tilde{s}] = s$ gives, for every answer a,

$$\mathbb{E}\big[\log p_{\tilde{h}}(a)\big] = \log p_0(a) - \mathbb{E}\big[\mathrm{KL}(p_0 \| p_{\tilde{h}})\big] \;\leq\; \log p_0(a). \tag{16}$$

A centered perturbation can therefore only lower the expected log-probability of an answer, by exactly an expected KL divergence from the clean predictive distribution. The log-softmax read-out is concave in h, so Eq. (16) is its Jensen gap written in exact form.

Write $L(z) = \log \sum_b e^{z_b}$ for the log-sum-exp of a logit vector z, so that $\mathrm{Softmax}(z)_a = e^{z_a - L(z)}$ and $\log \mathrm{Softmax}(z)_a = z_a - L(z)$ for every answer a. Applying this to the perturbed logits gives $\log p_{\tilde{h}}(a) = \tilde{s}_a - L(\tilde{s})$. Taking expectations and using $\mathbb{E}[\tilde{s}_a] = s_a$,

$$\mathbb{E}\big[\log p_{\tilde{h}}(a)\big] = s_a - \mathbb{E}\big[L(\tilde{s})\big]. \tag{17}$$

Applying the same identity to the clean logits gives $\log p_0(a) = s_a - L(s)$, so $s_a = \log p_0(a) + L(s)$. Substituting into Eq. (17),

$$\mathbb{E}\big[\log p_{\tilde{h}}(a)\big] = \log p_0(a) - \Big(\mathbb{E}\big[L(\tilde{s})\big] - L(s)\Big). \tag{18}$$

It remains to identify the bracketed term. Expanding the KL divergence directly,

$$\mathrm{KL}(p_0 \| p_{\tilde{h}}) = \sum_b p_0(b)\big[\log p_0(b) - \log p_{\tilde{h}}(b)\big] \tag{19}$$

$$= \sum_b p_0(b)\big[(s_b - L(s)) - (\tilde{s}_b - L(\tilde{s}))\big] \tag{20}$$

$$= \sum_b p_0(b)(s_b - \tilde{s}_b) - L(s) + L(\tilde{s}), \tag{21}$$

using $\sum_b p_0(b) = 1$ to carry the two terms that do not depend on b through the sum. Taking expectations, with p_0 and s fixed and $\mathbb{E}[\tilde{s}_b] = s_b$ for every b,

$$\mathbb{E}\big[\mathrm{KL}(p_0 \| p_{\tilde{h}})\big] = \sum_b p_0(b)\big(s_b - s_b\big) - L(s) + \mathbb{E}\big[L(\tilde{s})\big] = \mathbb{E}\big[L(\tilde{s})\big] - L(s). \tag{22}$$

This is exactly the bracketed term above. Substituting,

$$\mathbb{E}\big[\log p_{\tilde{h}}(a)\big] = \log p_0(a) - \mathbb{E}\big[\mathrm{KL}(p_0 \| p_{\tilde{h}})\big],$$

which is the equality in Eq. (16). The inequality holds because KL divergence is never negative.

Two consequences matter. First, the deficit is the same for every answer. Writing $\kappa = \mathbb{E}[\mathrm{KL}(p_0 \| p_{\tilde{h}})]$, Eq. (16) gives $\mathbb{E}[\log p_{\tilde{h}}(a)] = \log p_0(a) - \kappa$ for every a, and κ does not depend on a. Subtracting the same constant from every value in a set does not change which one is largest, so $\arg\max_a \mathbb{E}[\log p_{\tilde{h}}(a)] = \arg\max_a \log p_0(a)$. Since $\log$ is strictly increasing and $p_0 = \mathrm{Softmax}(s)$ is order-preserving (Eq. (9)), this equals $\arg\max_a s_a = g$:

$$\arg\max_a \mathbb{E}\big[\log p_{\tilde{h}}(a)\big] = \arg\max_a s_a = g. \tag{23}$$

Averaging log-scores over perturbations leaves the greedy answer on top. This is not a claim about the mode of q_{pert} itself: voting does not average log-scores, it counts hard decisions. Eq. (12) is where the two diverge: the expected log-scores can rank g first while the vote converges on a different answer.

Expected log-score is therefore a statement about predictive support, and it settles neither metric. Eq. (16) transfers neither to $\mathbb{E}[p_{\tilde{h}}(a^*)]$ nor to $q_{\mathrm{pert}}(a^*)$, the quantity that sets coverage. The distribution q and its mode remain what decide pass@k and maj@k. Eq. (16) is consistent with the entropy increases of Table 1 without deriving them. We do not use entropy as a primary diagnostic.

Proposition 7 (A perturbation scheme is summarized by the law it induces)**.** *Fix the weights θ and a question x. Let*

$$\tilde{h} = \Phi_\theta(x, \varepsilon), \qquad \varepsilon \sim D_\eta,$$

where Φ_θ is the forward pass of Eq. (2) *followed by the answer phase, ε collects every noise draw the scheme makes, and D_η is a law fixed by a single scale η, the rate p of Eq.* (3) *or the deviation σ of Eq.* (4)*. Write $q_{\mathrm{pert}}(a) = \Pr\big(a = \arg\max_b (W\tilde{h})_b\big)$ for the first answer token. Then q_{pert} is determined by θ, x, W and η. The number of injection points and their positions enter only through the law of $\tilde{h}$, and a^* enters nowhere.*

Proof. The forward pass is deterministic given its inputs, so $\tilde{h}$ is a measurable function of ε and the law of $\tilde{h}$ is the image of D_η under $\Phi_\theta(x, \cdot)$. The first answer token is decoded from $\tilde{h}$ alone through Eq. (1), so q_{pert} is a functional of that image law and of W. None of Φ_θ, D_η and W takes a^* as an argument. □

The statement holds for MC-dropout and AGN, for any number of latent steps, and whether or not the perturbation shifts the mean of the state. Explicit reasoning differs in the law it samples from. That law is $\mathbb{P}_\theta(r \mid x)$, which training shapes, whereas D_η is fixed by one hyperparameter that never enters training (Remark 3).

Remark 2 (Scope)**.** ***Idealization gap.*** *Eqs.* (11) *and* (16) *are exact for a perturbation added directly to the vector that enters the output head, with no computation in between. None of the perturbations we run meet that condition exactly. The single calibrated perturbation of Section 3.1 and full-chain*

MC-dropout and AGN (You et al., 2026) (Eqs. (3) *and* (4)*) all inject inside the forward pass, at the weights or at z_i, and the perturbed signal passes through further layers before it reaches the head.*

The mean is not preserved. *Eqs.* (11) *and* (16) *are stated for noise entering the output head directly, and we do not claim them for full-chain perturbation, where the noise passes through further computation. By Proposition 7, whatever the further computation does to the mean, the result is fixed by the weights, the question and the noise scale, and none of the three sees a^*. Even a perturbation with unchanged mean logits carries no guarantee on the vote (Eq.* (12)*), so a shifted one carries none either.*

Exchangeability does not survive compounding. *A single linear map already breaks the permutation invariance Proposition 5 needs, since $W\Sigma W^\top$ is not permutation invariant in general. The same holds for the Jacobian of each latent step. Nothing about how these maps are trained aligns them to undo one another's asymmetry. Composing one such map per latent step only compounds that loss of structure and gives no route back to it. Full-chain perturbation therefore has, if anything, less reason than a single-step injection to keep Proposition 5's protection, matching the entropy increase and mode-shifts full-chain perturbation produces in practice without by itself deriving their size.*

Matched once calibrated. *Despite this idealization gap, the two strategies do produce closely matched answer distributions once calibrated (Section 3.4).*

B.5 Resampling the trajectory removes the ceiling

The state-level distinction of the sections above raises the same question one level up, at the level of the trajectory itself. Proposition 4 describes a frozen chain. The ceiling it gives comes from freezing. Suppose instead that every sample redraws the trajectory r from a law ρ. The answer distribution is then a mixture,

$$q_\rho(a) = \mathbb{E}_{r\sim\rho}\big[\,\mathbb{P}_\theta(a \mid x, r)\,\big]. \tag{24}$$

Greedy decoding follows one trajectory. The mixture averages over many, and its mode need not be the mode of the trajectory greedy follows: this is exactly why self-consistency improves over greedy decoding in practice (Wang et al., 2023; Arzhantsev et al., 2026), an improvement that would be impossible if the two modes always agreed. Once the chain is redrawn, maj@k is no longer bounded by greedy accuracy. What replaces that ceiling depends on which distribution ρ is.

Remark 3 (A deterministic chain has no marginal to recover)**.** *A mixture carries information only when the model itself puts mass on more than one trajectory. Explicit CoT does. A deterministic latent chain does not. Eq.* (2) *makes $z_{1:T}$ a function of the question. The trajectory law of the model is therefore a point mass on the single chain $z^0_{1:T}$ it computes, and marginalizing returns the single-chain answer distribution unchanged,*

$$\mathbb{P}_\theta(a \mid x) = \mathbb{P}_\theta\big(a \mid x,\, z^0_{1:T}\big). \tag{25}$$

Two options are then exhaustive. Sampling from the model means sampling the frozen chain. That is the case Proposition 4 covers, and it cannot pass greedy. Sampling anything else leaves $\mathbb{P}_\theta$ behind, and that is what full-chain perturbation does, made precise below in Eq. (27)*. The mechanism that lets self-consistency exceed greedy accuracy is therefore unavailable to these models rather than merely weaker.*

Explicit reasoning and unconstrained thoughts differ in exactly this respect. *Explicit CoT* samples tokens directly from the model's own distribution, with no temperature or truncation. Its ρ is the trained distribution $\mathbb{P}_\theta(\cdot \mid x)$ itself, so the mixture collapses into a marginal of the model,

$$q_\rho(a) = \sum_r \mathbb{P}_\theta(r \mid x)\,\mathbb{P}_\theta(a \mid x, r) = \mathbb{P}_\theta(a \mid x). \tag{26}$$

Voting converges to $\arg\max_a \mathbb{P}_\theta(a \mid x)$, the answer the model holds most probable once the reasoning is marginalized out (Wang et al., 2023; Arzhantsev et al., 2026). Greedy decoding returns the answer of a single trajectory. The two answers can differ, which is what allows self-consistency to exceed greedy accuracy. This benefit relies on autoregressive sampling: ρ needs a trajectory of more than one token to place mass on more than one chain at all.

Unconstrained thoughts fall under Eq. (25): the chain is deterministic, $z_{i+1} = \pi(h_{i+1})$, and the model defines no distribution over trajectories at all. Diversity has to be supplied from outside, by

perturbation, injected fresh at every one of the T latent steps: a new dropout mask m_i for MC-dropout, a new Gaussian draw ε_i for AGN, so one trajectory is governed by the whole sequence $m_{1:T}$ or $\varepsilon_{1:T}$. Write $\hat{a}(\xi) = \arg\max_a \mathbb{P}_\theta(a \mid x, z_{1:T}(\xi))$ for the hard decision one such sequence ξ produces. MC-dropout (Eq. (3)) and AGN (Eq. (4)) each marginalize with their own law over these sequences: the probability that a draw's hard decision lands on a is the expectation, under that law, of the indicator that it does,

$$q_{\mathrm{mcd}}(a) = \mathbb{E}_{m_{1:T}\sim\nu_{\mathrm{drop}}}\big[\mathbb{I}\{\hat{a}(m_{1:T}) = a\}\big], \qquad q_{\mathrm{agn}}(a) = \mathbb{E}_{\varepsilon_{1:T}\sim\nu_{\mathrm{gauss}}}\big[\mathbb{I}\{\hat{a}(\varepsilon_{1:T}) = a\}\big], \quad (27)$$

where ν_{drop} is the law of T independent $\mathrm{Bernoulli}(p)$ masks and ν_{gauss} the law of T independent $\mathcal{N}(0, \sigma^2 I)$ draws, both fixed by a single hyperparameter (p or σ) independently of θ that appears nowhere in training. Eq. (26) performs the same operation for explicit CoT: it certifies a limit for q_{mcd} and q_{agn} just as firmly as for explicit reasoning, yet that limit carries far less.

Whether the law can favor a^*. The difference between Eq. (26) and Eq. (27) is whether the marginalizing law depends on θ at all. For explicit CoT, $\rho = \mathbb{P}_\theta(\cdot \mid x)$ is exactly what training raises toward a^*, since generation and answer decoding share θ. Nothing plays that role for ν_{drop} or ν_{gauss}: each is fixed by p or σ before decoding, with no channel through which a^* or θ's training could enter it. Whichever answer a mode shift under Eq. (27) reaches is decided by the network's geometry around $s(x)$, not by which answer is correct. Full-chain perturbation is therefore not a latent form of explicit reasoning for these models: it supplies variety where explicit reasoning supplies belief.

Extension: vocabulary-anchored thoughts. The comparison above concerns unconstrained thoughts. Models whose thought lives in vocabulary space align with the mechanism of explicit CoT. Latent-SFT builds each thought as a weighted average of embedding rows through the same head that would decode explicitly (Eq. (34)), and those weights are the model's own trained next-token distribution at that point,

$$z_{i+1} = \mathbb{E}_{v\sim\bar{p}_{i+1}}\big[E(v)\big], \qquad \bar{p}_{i+1}(v) = \mathrm{softmax}_v\big(Wh_{i+1}\big)\Big|_{v\in\mathcal{T}_{i+1}} = \mathbb{P}_\theta(v \mid x, z_{1:i}). \quad (28)$$

The clean thought is therefore the mean of a trained distribution over real tokens rather than an arbitrary point. Additive noise on such a thought can be read without further assumptions. We score each token v by $E(v)^\top \tilde{z}_{i+1}$. With tied input and output embeddings, as in Llama-3.2-1B, this is the logit the output head of the model would assign to v if it read $\tilde{z}_{i+1}$ directly. For $\tilde{z}_{i+1} = z_{i+1} + \delta$ with $\delta \sim \mathcal{N}(0, \sigma^2 I)$, the score splits into a fixed part and a Gaussian part,

$$E(v)^\top \tilde{z}_{i+1} = E(v)^\top z_{i+1} + E(v)^\top \delta, \qquad E(v)^\top \delta \sim \mathcal{N}\big(0,\, \sigma^2 \|E(v)\|^2\big).$$

Subtracting the scores of the top token v and a rival u gives the probability that the rival overtakes it,

$$\Pr\big(u \text{ overtakes } v\big) = \Phi\left(-\frac{\big(E(v) - E(u)\big)^\top z_{i+1}}{\sigma\,\|E(v) - E(u)\|}\right), \quad (29)$$

where Φ is the standard normal distribution function. The fraction is the distance from z_{i+1} to the boundary between v and u, in units of σ. A token u in $\mathcal{T}_{i+1}$ is part of the mixture, so z_{i+1} lies close to that boundary and small noise swaps the two often. An unrelated token lies far from it and rarely wins. A perturbation therefore nudges the thought toward the tokens it already mixes and their close neighbours in embedding space. Under the common reading that embedding similarity tracks semantic similarity, these are semantically similar tokens. For the other reading we take one simplifying assumption, that perturbation replaces the mean with a draw from the same distribution,

$$\tilde{z}_{i+1} \;\approx\; E(v_{i+1}), \qquad v_{i+1} \sim \bar{p}_{i+1}, \quad (30)$$

which formalizes the reading of a vocabulary-anchored thought as a relaxed token that perturbation turns into a sampled one (Zhang et al., 2025c; Butt et al., 2025). What licenses that reading is a property of the space rather than of the noise. Every point in the convex hull of the embedding rows denotes a mixture of real tokens, and reading such mixtures is what the input layer does, so a perturbed thought remains interpretable. CoLaR and Latent-SFT train for that decodability directly (Tan et al., 2025; Deng et al., 2026). An unconstrained thought has no such reading. Table 6 places it 7 to 113 times outside the embedding scale, and by Remark 3 the only points its own recursion produces are a point mass per question, so an arbitrary displacement of it has no such interpretation. Under Eq. (30)

a perturbed chain draws $v_{1:T}$ from the model's own trained distribution, so the answer distribution it induces is

$$q_{\mathrm{vocab}}(a) = \mathbb{E}_{v_{1:T}\sim\mathbb{P}_\theta(\cdot|x)}\big[\mathbb{P}_\theta(a \mid x, v_{1:T})\big] = \mathbb{P}_\theta(a \mid x), \tag{31}$$

which is Eq. (26) itself. Perturbing a vocabulary-anchored thought therefore recovers the marginal explicit CoT computes, and with it the mechanism Remark 3 denies an unconstrained chain. The assumption carries the weight here. Gaussian noise does not land on embedding rows exactly, and at $\varepsilon = 0$ no draw happens at all, leaving the single frozen chain Latent-SFT generates deterministically.

CoLaR enters on the same terms under one further assumption, that its thought lies in the same vocabulary convex hull, which its own training methodology supports. Appendix G reports results consistent with Eq. (31) for both models. Perturbation raises pass@k well above the frozen chain, and maj@k rises above greedy by up to 9.3 points, while every unconstrained model stays within 1.3 points of its own greedy accuracy at every k. Explicit CoT responds to the same noise the same way. That rise is small next to what sampling a full explicit trace achieves, but Eq. (31) is what permits it at all: the marginal it recovers carries no greedy ceiling, by the same argument Remark 3 uses to deny one to an unconstrained chain. Eq. (27) offers no such route.

Remark 4 (The Bayesian reading of dropout does not recover it)**.** *MC-dropout is motivated as approximate inference over weights (Gal & Ghahramani, 2016). One might therefore read q_{pert} as a posterior predictive rather than an injected distribution. Two things block that reading here. The marginalization would run over parameter uncertainty rather than over reasoning trajectories, and it is trajectories that Eq.* (26) *aggregates. The single calibrated perturbation's dropout rate is also tuned so that its pass@k curve matches full-chain perturbation (Section 3.4) rather than inherited from training. That breaks the correspondence with any posterior. AGN on the latent thought carries no such reading at all.*

Eq. (14) and Proposition 5 describe the idealized case, noise that respects the discrete structure of the alternatives it acts on. Explicit CoT reaches a version of that protection at the trajectory level through belief, since its ρ is the distribution the model was trained to produce (Eq. (26)). An unconstrained thought reaches neither: its ρ is not a belief, so Eq. (11) is what decides the vote. Appendix G reports this empirically.

C Additional background

Investigating latent reasoning. A growing line of work probes what latent chains actually compute. Chang et al. (2026) report that latent vectors align geometrically with explicit CoT states and can be decoded back into reasoning steps through a trained linear map, that early thoughts carry most of the causal influence, and that a latent sequence transplanted from one question into another transfers a reasoning plan; they turn these findings into training-free, decode-time interventions that raise accuracy. Aswal et al. (2026) caution that observable structure is not evidence of use, since patterns taken as signs of internal reasoning also appear in controls, and find thought utilization to be graded rather than binary. Zhang et al. (2025b) report the complementary negative, that latent tokens of Coconut are largely insensitive to steering and favor shortcuts, and Rizvi-Martel et al. (2026) test the superposition hypothesis directly and find little evidence that trained models maintain several candidate solutions at once. These works and ours share the premise that a latent chain has to be examined from the inside. They ask what a single chain represents; we ask what many samples of it explore, so our object of study is PTTS rather than a single trajectory. Two consequences follow. Our dual-mode setting compares implicit and explicit states through the shared decoder they are trained against, with no fitted map between the two spaces, and our intervention is a diagnostic of that structure.

Supervision strength and shortcut behavior. Cui et al. (2026) relate the strength of supervision in latent reasoning to shortcut behavior. We define a shortcut as the preference of a model to produce its answer from the output head while ignoring its latent reasoning. In their analysis, per-thought supervision reduces this preference relative to a single, late answer-position signal. Shortcut behavior is therefore a matter of degree, set by the source of supervision. Our results measure this degree directly in unconstrained models. Moving the state at the distillation token changes the answers of SIM-CoT and MUX, both trained with per-thought supervision, far more than those of CODI. The answers of these two models thus depend more on their reasoning state and less on a shortcut.

Suleymanzade et al. (2026) further argue that one MUX thought can hold several candidate tokens and thereby explore in parallel within a single forward pass. Our pass@k and maj@k results concern exploration across independent samples, and by that measure the maj@k of every unconstrained latent method we test, MUX included, saturates at its greedy accuracy.

C.1 Latent reasoning models

C.1.1 Dual-mode latent models

CODI, SIM-CoT and MUX frame latent reasoning as knowledge distillation within a single model, where an explicit CoT pass supervises a latent one on a shared backbone. The teacher is supervised by maximum likelihood estimation (MLE) over both $r_{1:R}$ and $y_{1:A}$, whereas the student supervision is restricted to $y_{1:A}$. Both read out through the same output head W:

$$\mathcal{L}_{\text{teacher}} = -\sum_{t=1}^{R+A} \log \mathbb{P}_\theta\big(u_t \mid x_{1:Q}, u_{1:t-1}\big), \qquad \mathcal{L}_{\text{student}} = -\sum_{t=1}^{A} \log \mathbb{P}_\theta\big(y_t \mid x_{1:Q}, z_{1:T}, y_{1:t-1}\big). \tag{32}$$

A knowledge distillation term then aligns the two modes at the answer position reached by both traces, pulling the implicit-mode state $h_{\text{ans}}^{\text{student}}$ onto its explicit-mode counterpart $h_{\text{ans}}^{\text{teacher}}$:

$$\mathcal{L} = \mathcal{L}_{\text{teacher}} + \mathcal{L}_{\text{student}} + \frac{\lambda}{L}\sum_{\ell=1}^{L} d\big(h_{\text{ans},\ell}^{\text{student}}, h_{\text{ans},\ell}^{\text{teacher}}\big), \tag{33}$$

where d is a distance on hidden states and ℓ indexes the L transformer layers, so the two modes are aligned at every layer. Without the layer index, h_{ans} denotes the states of all layers together. We call the position at which d is applied the *distillation token*.

With this multi-task objective over a shared backbone, we observe that the trained model operates effectively in either mode, selected by the special tokens of Eq. (2), and that it additionally leaves the hidden states $h_{\text{ans}}^{\text{student}}$ and $h_{\text{ans}}^{\text{teacher}}$ aligned. We call such models *dual-mode*, and we use that shared position as the common ground for our investigation.

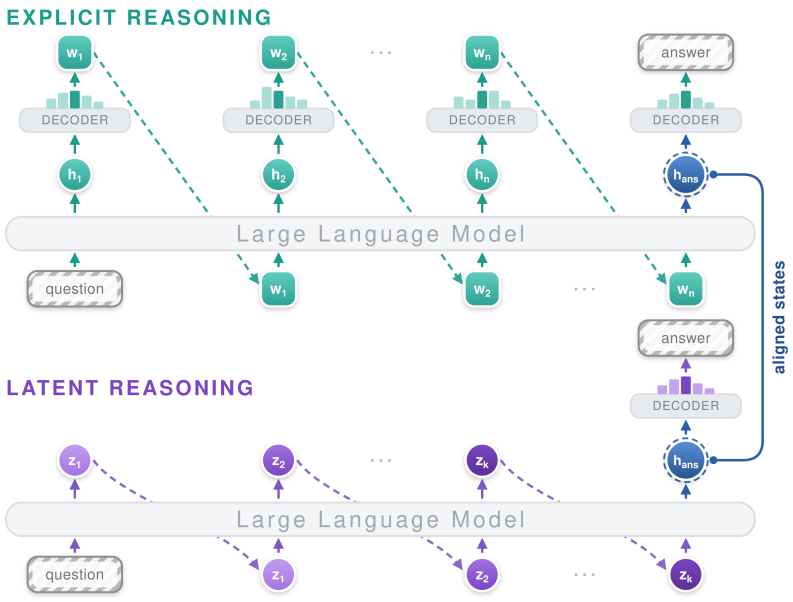


Figure 5: A *dual-mode* model. The explicit mode (top) decodes a word at every step and feeds it back, while the implicit mode (bottom) feeds its latent thought back directly. Both modes run on one backbone and decode the answer through the same output head at the distillation token.

C.1.2 TAXONOMY

Latent reasoning models differ in how the chain is supervised and how its length is controlled (Cui et al., 2026). Coconut (Hao et al., 2025) progressively replaces CoT steps with latent ones, and CODI (Shen et al., 2025) self-distills from an explicit-CoT teacher through a late, trajectory-level signal at the answer. SIM-CoT (Wei et al., 2025) attaches an auxiliary decoder to supervise individual thoughts, and MUX (Suleymanzade et al., 2026) instead supervises each thought through the shared output head against a chunk of the trajectory. LOTUS (Fan et al., 2026) processes latent tokens in blocks through looped transformer passes, aligning each block to its gold CoT step through the same head. CoLaR (Tan et al., 2025) supervises each thought against a compressed embedding of a chunk of CoT tokens, and ReGuLaR (Wang et al., 2026a) instead against a rendered image of the corresponding step; both sample the thought from a learned distribution and adapt T at inference. All of the other works fix T in advance and we adopt that setting throughout.

We organize the models discussed in Section 2.2 along four axes.

- **Where the thought lives.** Every thought is a vector of the same width, so the axis is what constrains it. *Unconstrained* models feed back the hidden state of Eq. (2), or a learned map of it, which is free to be any point in that space. *Vocabulary-anchored* models instead build the thought as a combination of token embeddings, or train it toward one, so it stays among the embeddings of real tokens and at their scale. Appendix F measures this scale for every model, and these measurements confirm the categorization.
- **Thought head π.** Identity, a learned deterministic map, or a distribution the thought is sampled from. For a generative head we write the sampled thought $\hat{z}_{i+1} \sim \pi(\cdot \mid h_{i+1})$, against the deterministic $z_{i+1} = \pi(h_{i+1})$ of Eq. (2); the hat marks that the thought is sampled. Only a deterministic head makes the chain a deterministic function of the input.
- **Supervision.** Every model is trained at minimum by next-token prediction, $\mathcal{L}_{\mathrm{NTP}}$ (Hao et al., 2025). Dual-mode models instead train one shared backbone jointly in both modes, and add the distillation term of Eq. (33), $\mathcal{L}_{\mathrm{KD}}$, which aligns their states at the answer position (Shen et al., 2025). Other models add a per-thought term, $\mathcal{L}_{\mathrm{step}}^{(\cdot)}$, applied at every latent step, with the superscript naming its target: an auxiliary decoder (dec), an invertible superposition of span tokens (mux), the gold step through the output head (head), a pooled token embedding (emb), a rendered step image (img), or a distilled latent chain matched by KL divergence (KL) (Wei et al., 2025; Suleymanzade et al., 2026; Fan et al., 2026; Tan et al., 2025; Wang et al., 2026a; Deng et al., 2026).
- **Chain length T.** Fixed in advance, or adapted at inference through a learned stop signal.

Each model below instantiates these axes in its own way; Table 2 summarises all of them together at the end of this section.

C.1.3 UNCONSTRAINED THOUGHTS

This group contains Coconut, CODI, SIM-CoT, MUX and LOTUS. We report our main results on these models.

Coconut (Hao et al., 2025). Coconut uses the identity thought head and feeds its last hidden state back as the next input embedding, $z_{i+1} = h_{i+1}$. A training curriculum replaces explicit CoT steps with latent ones stage by stage, trained with $\mathcal{L}_{\mathrm{NTP}}$ on the remaining language tokens.

CODI (Shen et al., 2025). CODI trains one backbone on the multi-task objective of Eq. (33). The explicit mode learns the full trace and the implicit mode learns only the answer (Eq. (32)). The distillation term aligns the two modes at the distillation token in every layer (Figure 5). The thought head π is an MLP.

SIM-CoT (Wei et al., 2025). SIM-CoT keeps the architecture and objective of CODI and adds per-thought supervision. An auxiliary decoder is trained to reconstruct each explicit CoT step from the corresponding latent thought. The decoder is discarded at inference, so SIM-CoT runs exactly like CODI at test time.

MUX (Suleymanzade et al., 2026). Similar to SIM-CoT, MUX extends CODI with per-thought supervision, applied through the shared output head W instead of an auxiliary decoder. The gold CoT is split into T spans. Each span is multiplexed into one target, a position-weighted superposition of the one-hot vectors of its tokens, and the latent thought at that step is trained to match it. The position weights make the map invertible, so the original span can be recovered from the target, which the authors call demultiplexing.

LOTUS (Fan et al., 2026). LOTUS loops a transformer over blocks of latent positions across several outer iterations, instead of feeding back one thought at a time. Each block is supervised against its gold CoT step through the output head W. Unlike CODI, whose distillation term ties two modes of the same model together, LOTUS is supervised by a separate, frozen explicit reasoner. Its latent states are therefore not trained to share a space with any explicit-mode states, and are not expected to align with them as those of dual-mode models are.

C.1.4 VOCABULARY-ANCHORED THOUGHTS

This group contains CoLaR, ReGuLaR and Latent-SFT. Each thought is a weighting over real tokens, as in the soft tokens of Zhang et al. (2025c), so it acts as a relaxed token. Perturbing it therefore resembles sampling tokens in explicit CoT, and Butt et al. (2025) use such noise on soft tokens as the exploration signal for reinforcement learning. Appendix G.1 reports their results.

CoLaR (Tan et al., 2025). The thought head of CoLaR predicts the mean and log-variance of a Gaussian, and the next thought is sampled from it at every step, $\hat{z}_{i+1} \sim \pi(\cdot \mid h_{i+1})$, in training and at inference. Each sample is then multiplied by a fixed constant that places it at the scale of the token embeddings (Table 6). Its target is the mean-pooled embedding of a chunk of CoT tokens, a point in the convex hull of the token embeddings; the chunk size, the compression factor, is requested at inference, and training samples it at random so that one model serves the whole range, offering dynamic compression. An entropy term in training keeps the Gaussian from collapsing to a point. Chain length is adaptive, since a stop decision is sampled from the output distribution at every step. Despite this generative head, which models each thought as a unimodal Gaussian, we observe that CoLaR behaves effectively deterministically (Appendix G.1).

ReGuLaR (Wang et al., 2026a). ReGuLaR keeps the generative head of CoLaR and replaces its target with a frozen vision encoding of the rendered CoT step, which a learned adapter projects to the hidden size and normalizes to the magnitude of the token embeddings. At inference it runs exactly like CoLaR.

Latent-SFT (Deng et al., 2026). The thought head of Latent-SFT is deterministic. Each thought is a weighted sum of token embeddings,

$$z_{i+1} = \sum_{v \in \mathcal{T}_{i+1}} \bar{p}_{i+1}(v)\, E(v), \tag{34}$$

where $\mathcal{T}_{i+1}$ holds the top-ranked tokens under $W h_{i+1}$ and $\bar{p}_{i+1}$ is the softmax over their logits. The weights are non-negative and sum to one, so the thought lies in the convex hull of the token embeddings, a stronger condition than matching their scale. Training has two stages. The first distills a compact latent chain from explicit CoT, and the second trains the model to generate that chain, including when to stop through a learned marker token. Chain length is therefore adaptive, as in CoLaR and ReGuLaR, but the stop decision is taken by arg max rather than sampled.

D ENTROPY FRAGILITY DETAIL

E IMPLEMENTATION DETAILS

All experiments use the full GSM8K test set of 1,319 questions (Cobbe et al., 2021) and run on a single NVIDIA H200 GPU. For a fair comparison with prior work, we use the checkpoints released by the original authors. CODI, SIM-CoT and MUX are evaluated on GPT-2 and Llama-3.2-1B-Instruct (Shen et al., 2025; Wei et al., 2025; Suleymanzade et al., 2026). Coconut on GPT-2 uses the

Table 2: Taxonomy of the models above (Table 6 gives the measured scale behind the two groups). Loss symbols follow the Supervision axis.

Model	Thought head	Deterministic	Training objective	T	Dual-mode
Unconstrained thoughts					
Coconut	identity	✓	$\mathcal{L}_{\mathrm{NTP}}$	fixed	✗
CODI	MLP	✓	$\mathcal{L}_{\mathrm{NTP}} + \mathcal{L}_{\mathrm{KD}}$	fixed	✓
SIM-CoT	MLP	✓	$\mathcal{L}_{\mathrm{NTP}} + \mathcal{L}_{\mathrm{KD}} + \mathcal{L}_{\mathrm{step}}^{\mathrm{dec}}$	fixed	✓
MUX	MLP	✓	$\mathcal{L}_{\mathrm{NTP}} + \mathcal{L}_{\mathrm{KD}} + \mathcal{L}_{\mathrm{step}}^{\mathrm{mux}}$	fixed	✓
LOTUS	residual	✓	$\mathcal{L}_{\mathrm{NTP}} + \mathcal{L}_{\mathrm{step}}^{\mathrm{head}}$	fixed	✗
Vocabulary-anchored thoughts					
CoLaR	Gaussian	✗	$\mathcal{L}_{\mathrm{NTP}} + \mathcal{L}_{\mathrm{step}}^{\mathrm{emb}}$	adaptive	✗
ReGuLaR	Gaussian	✗	$\mathcal{L}_{\mathrm{NTP}} + \mathcal{L}_{\mathrm{step}}^{\mathrm{img}}$	adaptive	✗
Latent-SFT	convex combination	✓	$\mathcal{L}_{\mathrm{NTP}} + \mathcal{L}_{\mathrm{step}}^{\mathrm{KL}}$	adaptive	✗

Table 3: First-answer-token entropy before/after perturbation (Δ in parentheses) for draws whose answer flipped, with 64 draws per question. Bold marks an entropy drop. Coconut has no explicit mode, so it appears in the latent rows only.

Model	Noise	Flipped to wrong	Flipped to correct
Llama-3.2-1B			
CODI	MC-dropout	$1.263/1.546$ $(+0.283)$	$1.444/1.540$ $(+0.096)$
CODI	AGN	$1.177/1.607$ $(+0.430)$	$1.472/1.701$ $(+0.229)$
MUX	MC-dropout	$1.230/1.861$ $(+0.631)$	$1.520/1.641$ $(+0.121)$
MUX	AGN	$1.377/1.751$ $(+0.374)$	$1.649/1.662$ $(+0.012)$
SIM-CoT	MC-dropout	$1.544/1.926$ $(+0.383)$	$1.597/1.721$ $(+0.124)$
SIM-CoT	AGN	$1.755/2.010$ $(+0.254)$	$1.681/1.808$ $(+0.127)$
CODI	Explicit CoT	$0.483/0.928$ $(+0.445)$	$0.881/0.567$ $(\mathbf{-0.314})$
MUX	Explicit CoT	$0.554/1.043$ $(+0.488)$	$1.000/0.597$ $(\mathbf{-0.403})$
SIM-CoT	Explicit CoT	$0.451/1.026$ $(+0.575)$	$0.917/0.576$ $(\mathbf{-0.340})$
GPT-2			
CODI	MC-dropout	$1.404/2.005$ $(+0.601)$	$1.899/1.995$ $(+0.095)$
CODI	AGN	$1.093/1.982$ $(+0.889)$	$1.980/2.128$ $(+0.148)$
MUX	MC-dropout	$1.097/1.436$ $(+0.339)$	$1.334/1.281$ $(\mathbf{-0.053})$
MUX	AGN	$0.922/1.445$ $(+0.523)$	$1.197/1.169$ $(\mathbf{-0.028})$
SIM-CoT	MC-dropout	$1.338/1.831$ $(+0.493)$	$1.732/1.771$ $(+0.039)$
SIM-CoT	AGN	$1.352/1.805$ $(+0.453)$	$1.799/1.771$ $(\mathbf{-0.028})$
Coconut	MC-dropout	$1.622/2.579$ $(+0.957)$	$2.130/2.644$ $(+0.514)$
Coconut	AGN	$1.910/2.556$ $(+0.646)$	$2.222/2.537$ $(+0.315)$
CODI	Explicit CoT	$0.473/0.994$ $(+0.521)$	$1.042/0.750$ $(\mathbf{-0.292})$
MUX	Explicit CoT	$0.461/0.875$ $(+0.414)$	$0.799/0.583$ $(\mathbf{-0.216})$
SIM-CoT	Explicit CoT	$0.501/0.967$ $(+0.466)$	$0.999/0.677$ $(\mathbf{-0.323})$

checkpoint of You et al. (2026), so that our perturbation results are directly comparable to theirs, and Coconut on Llama-3.2-1B uses the Coconut baseline released with CoLaR (Tan et al., 2025). LOTUS runs on Llama-3.2-3B-Instruct, and CoLaR and Latent-SFT run on Llama-3.2-1B-Instruct (Fan et al., 2026; Tan et al., 2025; Deng et al., 2026).

Each model keeps the chain length of its original work. CODI, SIM-CoT, MUX and Coconut use a fixed chain of $T = 6$ latent thoughts, and LOTUS runs six looped iterations. CoLaR decides its chain length adaptively, with about six thoughts per question on average at compression factor 5. Latent-SFT ends its chain on its own, after about 12 to 13 thoughts.

Every sampled decode, whether an explicit trace or the answer decoder over a frozen chain, uses temperature 1.0 and top-p 1.0, so the full distribution of the model is sampled. The frozen chain is computed once per question and shared by all its samples. For Coconut, the answer is decoded freely after the latent block, matching the reference evaluation of You et al. (2026). This setting lets Coconut continue some of its reasoning in the vocabulary space after its T latent steps. Each

Table 4: First-answer-token entropy before/after perturbation (Δ in parentheses) for draws whose answer did not change, with 64 draws per question. The small drift of these buckets is the reference for the flip rows of Table 3.

Model	Noise	Stayed correct	Stayed wrong
Llama-3.2-1B			
CODI	MC-dropout	$0.217/0.263$ $(+0.046)$	$1.841/1.893$ $(+0.052)$
CODI	AGN	$0.199/0.279$ $(+0.080)$	$1.843/1.940$ $(+0.097)$
MUX	MC-dropout	$0.218/0.306$ $(+0.088)$	$2.077/2.210$ $(+0.133)$
MUX	AGN	$0.221/0.288$ $(+0.067)$	$2.072/2.134$ $(+0.062)$
SIM-CoT	MC-dropout	$0.261/0.320$ $(+0.059)$	$2.054/2.143$ $(+0.089)$
SIM-CoT	AGN	$0.271/0.309$ $(+0.039)$	$2.053/2.123$ $(+0.070)$
CODI	Explicit CoT	$0.109/0.110$ $(+0.002)$	$1.153/1.170$ $(+0.017)$
MUX	Explicit CoT	$0.125/0.126$ $(+0.000)$	$1.285/1.270$ (-0.014)
SIM-CoT	Explicit CoT	$0.111/0.113$ $(+0.002)$	$1.200/1.221$ $(+0.021)$
GPT-2			
CODI	MC-dropout	$0.356/0.467$ $(+0.111)$	$2.529/2.722$ $(+0.193)$
CODI	AGN	$0.319/0.538$ $(+0.219)$	$2.531/2.802$ $(+0.271)$
MUX	MC-dropout	$0.233/0.268$ $(+0.035)$	$1.819/1.874$ $(+0.055)$
MUX	AGN	$0.211/0.279$ $(+0.069)$	$1.829/1.894$ $(+0.065)$
SIM-CoT	MC-dropout	$0.366/0.474$ $(+0.108)$	$2.572/2.705$ $(+0.133)$
SIM-CoT	AGN	$0.342/0.442$ $(+0.099)$	$2.573/2.612$ $(+0.039)$
Coconut	MC-dropout	$0.550/1.047$ $(+0.497)$	$3.001/3.366$ $(+0.365)$
Coconut	AGN	$0.565/0.825$ $(+0.260)$	$2.995/3.262$ $(+0.267)$
CODI	Explicit CoT	$0.127/0.138$ $(+0.011)$	$1.412/1.468$ $(+0.055)$
MUX	Explicit CoT	$0.083/0.090$ $(+0.007)$	$1.147/1.198$ $(+0.051)$
SIM-CoT	Explicit CoT	$0.128/0.135$ $(+0.008)$	$1.463/1.469$ $(+0.006)$

condition draws $N = 256$ samples per question ($N = 128$ for Latent-SFT, and for CoLaR under AGN), and pass@k and maj@k are computed for every $k \leq N$ (Appendix A).

Full-chain perturbation injects noise at every latent step and decodes the answer greedily, so the noise is its only source of stochasticity. MC-dropout uses $p = 0.25$ for unconstrained thoughts and $p = 0.03$ for vocabulary-anchored thoughts, and AGN uses the per-model σ of Table 6 (Appendix F). For LOTUS, noise enters between the looped block iterations, so it propagates through the loop as it does through an autoregressive latent chain. The single calibrated perturbation of Section 3.4 applies MC-dropout only at the answer generation phase. Its rate is chosen so that its pass@k curve matches full-chain MC-dropout, which gives $p = 0.15$ and 0.05 for CODI on GPT-2 and Llama-3.2-1B, and $p = 0.25$ for MUX on both.

The entropy analysis of Section 3.3 uses CODI, SIM-CoT and MUX on GPT-2 and Llama-3.2-1B-Instruct in both modes, and Coconut on GPT-2 in the latent mode. For this analysis, the answer cue of Coconut is forced, as for the other models, so that entropy is read at the same position. Entropy is measured on the first answer token, and the unperturbed reference is the greedy decode of each question in the corresponding reasoning mode, i.e., the unperturbed latent chain for the latent mode and the greedy explicit trace for the explicit mode.

F Noise calibration

Latent-thought scale. We measure the scale of the thought that each model feeds back as its per-dimension root mean square, $\mathrm{rms}(z) = \|z\|_2/\sqrt{d}$, averaged over 100 unperturbed GSM8K questions. For CODI, SIM-CoT and MUX we measure z after the learned projector, where the noise is added. Table 6 shows that this scale differs by nearly three orders of magnitude across models, so a single AGN level would perturb the models very unequally. Relative to the per-dimension root mean square of its own token embeddings, $\mathrm{rms}(E)$, an unconstrained thought is 7 to 113 times larger, whereas a vocabulary-anchored thought stays at 0.5 to 1 times that scale, consistent with the space in which each group operates.

Table 5: Significance of the entropy-fragility result: clean-state entropy separating the flipped-to-wrong bucket from stayed-correct (break) and flipped-to-correct from stayed-wrong (fix), draw-level AUC with a question-resampled bootstrap 95% CI in brackets (chance 0.5). % of draws is the share of the 64 draws per question that land in the bucket. Bold marks a latent AUC above that of the explicit CoT of the same model.

		Break		Fix	
Model	Noise	AUC	% of draws	AUC	% of draws
Llama-3.2-1B					
CODI	MC-dropout	**0.906** [0.883, 0.927]	2.8%	**0.600** [0.524, 0.667]	1.0%
CODI	AGN	**0.898** [0.870, 0.924]	4.0%	**0.591** [0.501, 0.675]	1.3%
MUX	MC-dropout	**0.870** [0.842, 0.894]	4.1%	**0.626** [0.574, 0.676]	1.4%
MUX	AGN	**0.885** [0.851, 0.913]	3.5%	**0.593** [0.542, 0.643]	1.3%
SIM-CoT	MC-dropout	**0.909** [0.889, 0.926]	3.3%	**0.607** [0.541, 0.676]	1.1%
SIM-CoT	AGN	**0.930** [0.912, 0.947]	2.5%	**0.588** [0.507, 0.659]	1.1%
CODI	Explicit CoT	0.727 [0.696, 0.756]	8.1%	0.570 [0.537, 0.604]	4.2%
MUX	Explicit CoT	0.731 [0.702, 0.760]	9.6%	0.559 [0.527, 0.594]	4.0%
SIM-CoT	Explicit CoT	0.722 [0.692, 0.748]	8.2%	0.575 [0.540, 0.610]	4.4%
GPT-2					
CODI	MC-dropout	**0.850** [0.822, 0.876]	4.6%	**0.646** [0.588, 0.704]	2.3%
CODI	AGN	**0.801** [0.773, 0.828]	8.2%	**0.630** [0.569, 0.688]	2.9%
MUX	MC-dropout	**0.869** [0.841, 0.895]	3.0%	**0.614** [0.564, 0.665]	2.1%
MUX	AGN	**0.832** [0.801, 0.861]	5.0%	**0.657** [0.604, 0.707]	2.5%
SIM-CoT	MC-dropout	**0.832** [0.800, 0.860]	3.9%	**0.692** [0.638, 0.740]	2.9%
SIM-CoT	AGN	**0.831** [0.801, 0.857]	4.6%	**0.674** [0.612, 0.728]	3.3%
Coconut	MC-dropout	0.825 [0.797, 0.852]	6.1%	0.696 [0.650, 0.746]	2.3%
Coconut	AGN	0.854 [0.830, 0.877]	4.5%	0.686 [0.642, 0.727]	2.1%
CODI	Explicit CoT	0.715 [0.688, 0.739]	11.4%	0.578 [0.547, 0.607]	4.1%
MUX	Explicit CoT	0.774 [0.746, 0.799]	9.1%	0.580 [0.548, 0.613]	3.6%
SIM-CoT	Explicit CoT	0.698 [0.668, 0.727]	11.2%	0.593 [0.565, 0.622]	3.9%

AGN level. We anchor on the level $\sigma = 1.4$ that You et al. (2026) use for CODI on GPT-2, and keep its ratio to the thought scale fixed across models, $\sigma = \alpha \cdot \mathrm{rms}(z)$ with $\alpha \approx 0.5$. The noise vector is then about half as long as the thought it perturbs, and Table 6 lists the resulting σ. We scale the noise to the magnitude of a single thought because parallel sampling explores within one question. A deterministic chain has no spread within a question, so the thought itself is the natural reference at that level. The spread across questions, $\mathrm{std}(z)$, instead measures how far apart the thoughts of different questions lie, a signal that exploration within one question is not meant to use. We still report it as a check on noise strength (last two columns of Table 6). Across the ten models where it could be measured (all but LOTUS), σ lies between 0.65 and 1.27 times $\mathrm{std}(z)$, with a median of 0.88, i.e., within about one standard deviation. The noise is therefore comparable to the natural variation between questions, and larger levels only lower pass@k (Appendix G.2).

MC-dropout rate. Following You et al. (2026), we fix the same dropout rate for all unconstrained thoughts, $p = 0.25$, and use $p = 0.03$ for vocabulary-anchored thoughts, selected by a rate sweep.

G Additional sampling results

G.1 Additional latent models

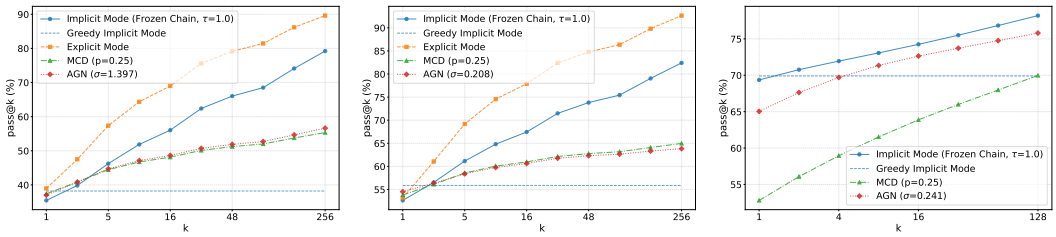
SIM-CoT and LOTUS show the same pattern as CODI, MUX and Coconut: the frozen chain leads on both pass@k and maj@k at every sample budget we test (Figures 6 and 7). This holds across training objectives (SIM-CoT shares CODI's architecture; LOTUS a looped block on a 3B backbone) and backbone sizes from 124M to 3B.

Vocabulary-anchored models. CoLaR samples every thought from its generative head, so it can perform PTTS on its own, without any perturbation. This head turns out to be effectively deterministic:

Table 6: Scale of the fed-back thought, $\mathrm{rms}(z)$, against the token embeddings of the same model, $\mathrm{rms}(E)$, and the resulting AGN level $\sigma = \alpha \cdot \mathrm{rms}(z)$ with $\alpha \approx 0.5$. For CODI, SIM-CoT and MUX, $\mathrm{rms}(z)$ is measured after the learned projector. The last two columns are the independent check described above, not part of the calibration.

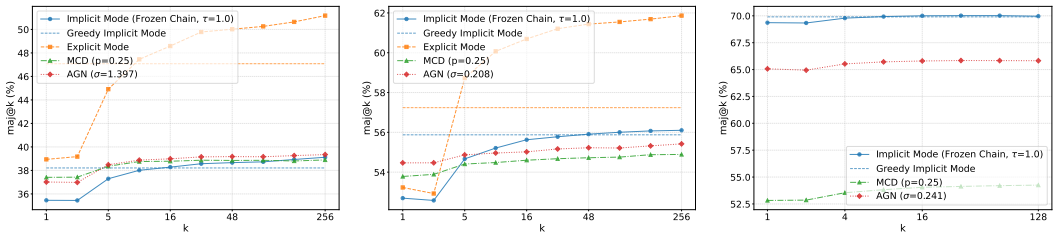
Model	Backbone	d	$\mathrm{rms}(z)$	$\mathrm{rms}(E)$	ratio	σ	$\mathrm{std}(z)$	$\sigma/\mathrm{std}(z)$
Unconstrained thoughts								
CODI	GPT-2	768	2.82	0.143	19.7×	1.400	1.36	1.03
CODI	Llama-3.2-1B	2048	0.626	0.0205	30.5×	0.311	0.353	0.88
SIM-CoT	GPT-2	768	2.81	0.143	19.7×	1.397	1.58	0.89
SIM-CoT	Llama-3.2-1B	2048	0.419	0.0205	20.4×	0.208	0.320	0.65
MUX	GPT-2	768	6.63	0.143	46.4×	3.290	2.59	1.27
MUX	Llama-3.2-1B	2048	0.433	0.0205	21.1×	0.215	0.301	0.71
Coconut	GPT-2	768	0.950	0.133	7.2×	0.472	0.513	0.92
Coconut	Llama-3.2-1B	2048	2.33	0.0205	113.3×	1.155	1.13	1.03
LOTUS	Llama-3.2-3B	3072	0.486	0.0196	24.8×	0.241	—	—
Vocabulary-anchored thoughts								
CoLaR	Llama-3.2-1B	2048	**0.0210**	0.0205	**1.0**×	0.0104	0.0141	0.74
Latent-SFT	Llama-3.2-1B	2048	**0.0110**	0.0205	**0.5**×	0.00546	0.0079	0.69

most questions return the same answer on every chain, and its pass@k curve stays close to flat. We therefore apply the same perturbation methods to the latent thought it generates, and we observe a different behavior for CoLaR and Latent-SFT (Figure 8). We run CoLaR at two compression factors. Factor 5, used throughout this appendix (Figure 8a), compresses the reasoning chain by a speed-up similar to that of the unconstrained models in Section 3.2. We also run factor 2 (Figure 8c; Appendix C.1 defines the compression factor), to show the pattern holds at both. At $k = 128$, full-chain perturbation exceeds the frozen chain by 32.6 to 41.0 points for CoLaR and 15.5 to 19.0 points for Latent-SFT (Figure 8e); their frozen chains stay close to flat, so resampling the decoder alone supplies little diversity. maj@k moves far less (Figures 8b, 8d and 8f), reaching at most 9.3 points above greedy, against 1.3 points for every unconstrained model. We test whether this reflects exploration over nearby vocabulary by applying the same perturbations to explicit reasoning. AGN adds noise to the input token embeddings, whereas MC-dropout keeps dropout active in the backbone at inference, which changes the next-token distribution of the model and hence the decoded token. Both perturbations likely stay within the language space, around semantically similar tokens: MC-dropout changes the decoded token discretely, whereas AGN shifts each embedding continuously toward nearby words. As in Section 3.1, we decode greedily, so perturbation is the only source of stochasticity. Explicit CoT responds the same way (Figures 8g to 8j): pass@k rises with the sample budget while maj@k stays near greedy. This is consistent with the vocabulary-anchored extension of Appendix B, where perturbing a vocabulary-anchored thought approximates sampling a token from the trained next-token distribution of the model (Eqs. (28) and (30)). This places CoLaR and Latent-SFT with the explicit reasoner rather than with the unconstrained models. This work instead focuses on reasoning in a continuous space, where the latent thought is free of the vocabulary and therefore more expressive (Zhu et al., 2025), and our primary results concern that setting.



(a) SIM-CoT, GPT-2 (b) SIM-CoT, Llama-3.2-1B (c) LOTUS, Llama-3.2-3B

Figure 6: pass@k for the additional unconstrained models, as in Figure 2.

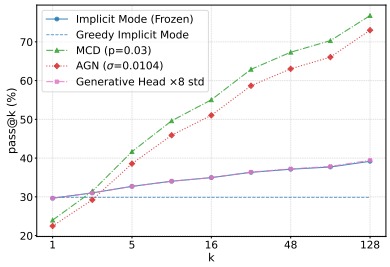


(a) SIM-CoT, GPT-2 (b) SIM-CoT, Llama-3.2-1B (c) LOTUS, Llama-3.2-3B

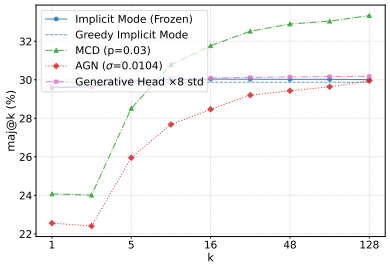
Figure 7: maj@k for the additional unconstrained models, as in Figure 3.

G.2 Larger noise scales

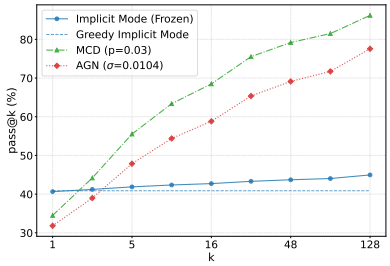
Figure 9 runs CODI on Llama-3.2-1B at 2× and 3× the calibrated noise level of Appendix F. Each step up lowers single-sample accuracy further, and the added diversity brings no consistent gain on either pass@k or maj@k (Figures 9a to 9d). The calibrated levels are therefore not conservatively low.



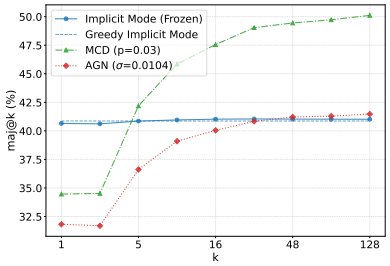
(a) pass@k, CoLaR (compression factor 5), Llama-3.2-1B



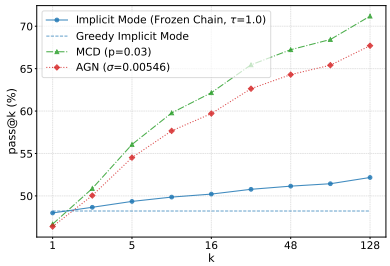
(b) maj@k, CoLaR (compression factor 5), Llama-3.2-1B



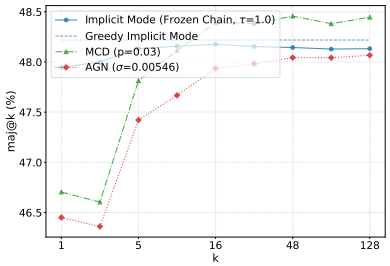
(c) pass@k, CoLaR (compression factor 2), Llama-3.2-1B



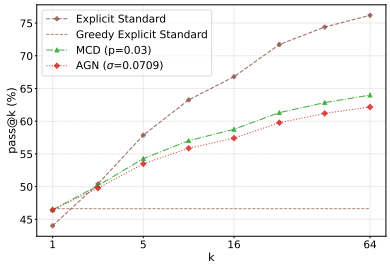
(d) maj@k, CoLaR (compression factor 2), Llama-3.2-1B



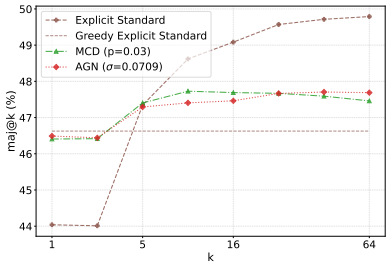
(e) pass@k, Latent-SFT, Llama-3.2-1B



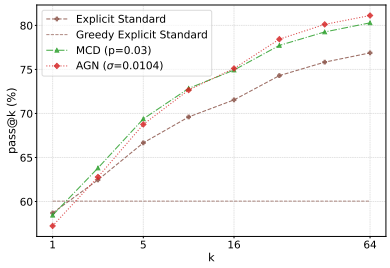
(f) maj@k, Latent-SFT, Llama-3.2-1B



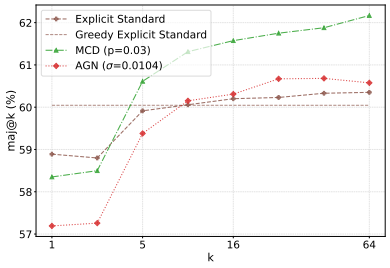
(g) pass@k, Explicit CoT, GPT-2



(h) maj@k, Explicit CoT, GPT-2

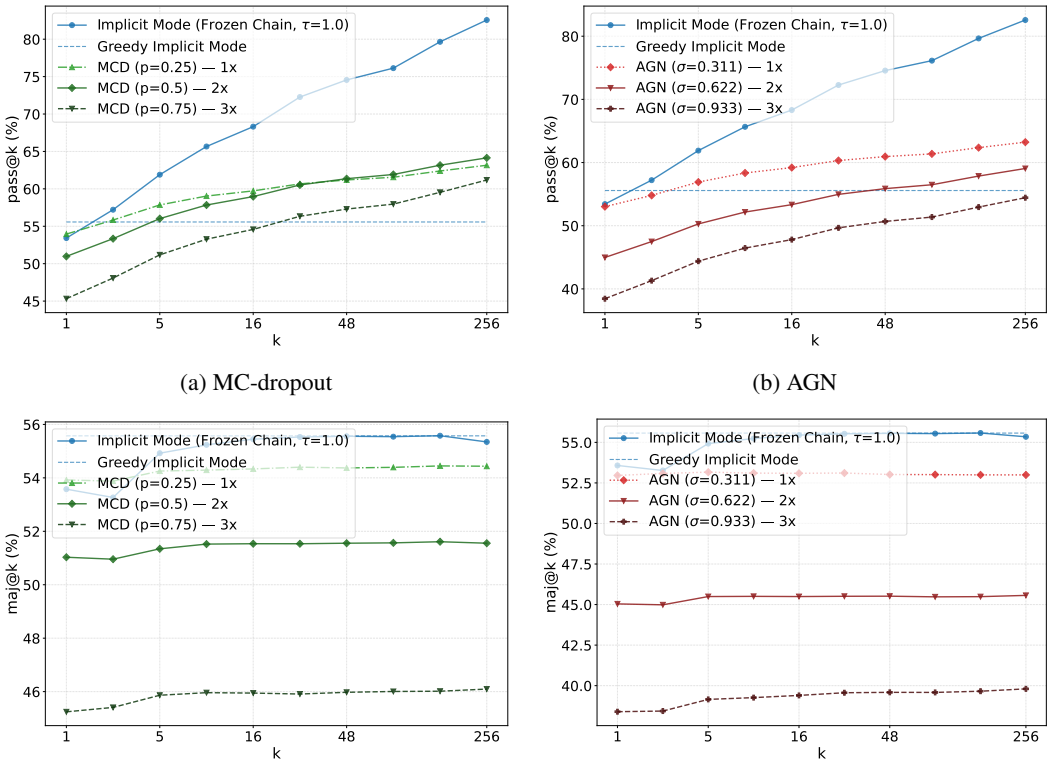


(i) pass@k, Explicit CoT, Llama-3.2-1B



(j) maj@k, Explicit CoT, Llama-3.2-1B

Figure 8: pass@k (left) and maj@k (right) under full-chain perturbation for the vocabulary-anchored models and for explicit CoT.



(a) MC-dropout

(b) AGN

(c) MC-dropout, maj@k

(d) AGN, maj@k

Figure 9: pass@k (top) and maj@k (bottom) of CODI on Llama-3.2-1B at $1\times$, $2\times$ and $3\times$ the calibrated noise level.