DOES LATENT CHAIN-OF-THOUGHT ACTUALLY REASON? A SINGLE, UNIFYING TAXONOMY FOR LATENT REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Latent chain-of-thought replaces written reasoning steps with continuous hidden states, giving models additional steps to compute before answering. It remains unclear what computation these steps actually perform. High task accuracy and decodable intermediate answers do not establish that a model uses its latent chain to reach an answer. We study this question across twenty-one model-task configurations covering three training recipes, three reasoning tasks, and two backbones. We combine chain deletion and content replacement with probes and trajectory analysis to test whether the chain affects the answer, what it encodes, and how its states change. On logical tasks, COCONUT often carries an answer that is already readable before the chain begins. Its states settle early, and deleting the chain leaves predictions almost unchanged. Per-step supervision produces an ordered layout of reasoning steps and can make the model depend on the chain. Yet even an ordered chain can act as a placeholder, with deletion hurting performance while content replacement leaves answers unchanged. On arithmetic tasks, all tested recipes show dependence on chain content. These results show why readable latent states alone are insufficient evidence of reasoning. We introduce a taxonomy that separates what a chain represents from how the model uses it. Seen this way, earlier reports no longer compete: placeholder tokens, collapsed superposition and step-dependent causal roles are each consistent with a different kind of chain, each obtained on a different task, training recipe or measurement. Once a study says which property it tested, whether the model uses the chain, what the chain holds, or how its state moves, claims about latent reasoning can be compared.

1 INTRODUCTION

Chain-of-thought gives a language model intermediate steps to work through before answering. Latent reasoning methods aim to do this without writing out every step. In the methods we study, the model feeds continuous hidden states back as inputs, forming a latent chain between the question and the answer (Hao et al., 2025; Shen et al., 2025; Wei et al., 2026). Each position in this chain, which we call a slot, provides an additional computation step. But what do these steps actually do? A model may use them to build an answer, or it may already have enough information to answer after reading the question. Distinguishing these cases matters for understanding the benefit of latent steps and how to train models to use them. Task accuracy alone cannot tell us where the answer was formed, and decoding an intermediate result from a state does not show that the model uses it (Elazar et al., 2021). Prior studies report several behaviors, including slots that act as placeholders (Zhang et al., 2025), candidate superposition that collapses or goes unused after fine-tuning (Rizvi-Martel et al., 2026), and different causal roles across steps (Li et al., 2026). These studies use different tasks, training recipes and measurements, leaving open how their observations fit together.

We study how latent chains contribute to answering across tasks and training recipes within a common experimental framework. Our grid covers twenty-one model-task configurations across ProsQA, PrOntoQA and GSM8k, using GPT-2 and Llama-3.2-1B-Instruct. We compare COCONUT, which recycles hidden states without direct supervision of slot content, CODI, which aligns latent and written reasoning at the answer position, and SIM-COT-COCONUT, which adds per-step supervision to COCONUT. The last comparison holds the inference mechanism and slot budget fixed. Continued-

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.

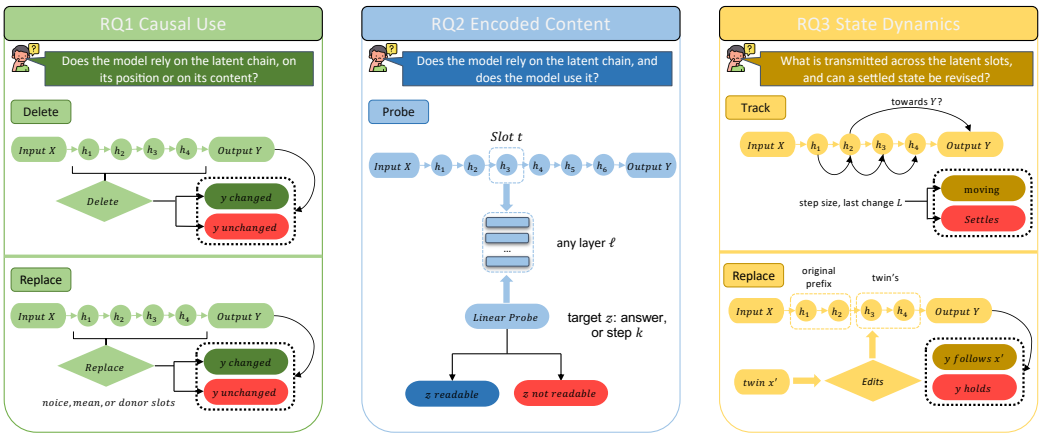

Figure 1: **The three questions and the five tests, drawn on one latent chain.** RQ1 deletes the slots or replaces their contents and reads the answer. RQ2 probes one slot at one layer against a permuted-label null. RQ3 tracks the state across slots and edits the chain mid-way. Answering all three lets us separate what a chain represents from how the model uses it, and place the resulting behaviors in a single taxonomy.

training controls help separate the effect of per-step supervision from that of extra training. On this grid, we ask three questions about causal use, encoded content and state dynamics (Figure 1).

RQ1: Does the model rely on the latent chain, on its positions or on its contents? We first test whether the model depends on the chain by deleting its slots. Where prior causal tests apply a single corruption (deletion (Dilgren & Wiegreffe, 2026)or noise(Cui et al., 2026)) each, we pair deletion with three replacements (noise, the dataset mean, and another problem's vectors) plus a same-base training-time control that switches off the per-step loss (Section 4). This isolates a placeholder pattern a single corruption cannot: on one backbone's logical-task cell, deletion costs accuracy yet no replacement moves an answer, while on most content-used cells another problem's slots hurt more than deleting the chain outright. Deleting the unsupervised recipe's chains changes at most 3 answers on the logical tasks but hurts accuracy on arithmetic. Per-step supervision can establish reliance on a chain where the same base model had little.

RQ2: What is written in the latent chain, and does the model use it? A chain that does not affect the answer may still contain information about it. Prior probing has asked one question at a time, whether the answer is decodable early(Rizvi-Martel et al., 2026; Li et al., 2026)or whether a slot superposes candidates(Rizvi-Martel et al., 2026; Aswal et al., 2026); we probe for the final answer, the candidates and each reasoning step, at every slot and layer, and read each profile against its RQ1 verdict (Section 5). On the logical tasks, unsupervised chains carry an answer that is already readable at the question's final hidden state, before the chain begins, and hold the path spread across slots rather than in order; no slot holds both candidates. Per-step supervision lays the steps out one per slot, and that ordered layout appears even in a chain whose contents can be replaced without changing an answer. Placed beside RQ1, this gives the paper's central observation: on the logical tasks the chains with the richest readable content are ones the model does not rely on, and an ordered layout can sit in a chain whose contents are never consulted.

RQ3: What is transmitted across the latent slots, and can a settled state be revised? Knowing what individual slots contain leaves open how the state evolves between them. State movement has been described by trajectory geometry and answer commitment(Duraipandian et al., 2026; Aswal et al., 2026; Li et al., 2026) but not decided against a null for whether a chain settles; we turn it into a decision across all recipes and both backbones, and add a mid-chain edit that splices a twin problem's content into the chain (Section 6). On the logical tasks, chains without per-step supervision settle within the first slots, while per-step chains keep changing to the last slot, though the movement is directed toward the answer on only one task. The edit, a test the prior studies do not include, shows that revising an answer is set by the backbone, not the mechanism or the supervision: one backbone increasingly ignores later edits while the other follows them. State movement and the ability to revise an answer thus describe different aspects of the chain.

These results show why a single measurement gives an incomplete account of latent reasoning. A chain can carry a readable answer without affecting the output, or encode ordered steps while serving only as a set of extra positions under the tested interventions. We combine the three measurements in a taxonomy that distinguishes six types of latent chain, while leaving unresolved cases unclassified (Section 7.1). Its value is that the scattered reports above no longer appear to make conflicting claims about latent chains in general. Placeholder behaviour, collapsed superposition, and step-dependent causal roles are each consistent with a different kind of chain, arising under different tasks, training recipes, or measurements. By stating what each test establishes about a chain, the taxonomy explains why earlier findings seem to disagree and clarifies which property—use, content, or dynamics—a claim about latent reasoning addresses.

2 Related Work

Latent reasoning methods. Several approaches give language models more computation before an answer without requiring a full written chain. Pause and filler tokens provide extra positions without explicit reasoning content (Goyal et al., 2024; Pfau et al., 2024), while recurrent-depth models apply additional computation within the network (Geiping et al., 2025). Other methods train models to internalize written reasoning or compress it into continuous representations (Deng et al., 2024; Cheng & Van Durme, 2024). We focus on methods that feed hidden states back as inputs to form a chain of latent slots. COCONUT gradually replaces written steps with recycled hidden states during training (Hao et al., 2025). CODI uses self-distillation to align the latent pathway with written reasoning at the answer position (Shen et al., 2025). SIM-COT adds direct supervision of individual slots through an auxiliary decoder (Wei et al., 2026). These training signals offer different ways to shape a latent chain. Comparing COCONUT with its per-step supervised variant lets us test how slot supervision changes the chain while keeping its inference mechanism fixed.

Interpreting latent states. Prior work has looked for recognizable reasoning structure inside continuous states. Hao et al. (2025) interpret vocabulary projections of COCONUT states as evidence of breadth-first search, and Zhu et al. (2025) study how continuous thoughts can represent multiple search paths. Work on CODI and SIM-COT examines whether intermediate reasoning content can be recovered from latent states (Shen et al., 2025; Wei et al., 2026). These interpretations depend on how the model is trained. For example, Rizvi-Martel et al. (2026) find that candidate superposition can collapse after fine-tuning. Analyses of state transitions also provide ways to study how reasoning evolves across steps (Yu et al., 2025). Such measurements describe the information and structure present in a trajectory, but their relevance to the answer needs separate evidence. This distinction follows a broader lesson from probing: a target can be predictable from a representation without being used for the model's task (Ravichander et al., 2021; Elazar et al., 2021). We pair content and trajectory measurements with interventions on the same model and task configurations.

Causal use of latent chains. Studies of written chain-of-thought use interventions to test whether reasoning traces affect model answers (Lanham et al., 2023). For latent chains, Dilgren & Wiegreffe (2026) study both trace readability and necessity, finding that removing latent steps can preserve performance on logical tasks. Zhang et al. (2025) report insensitivity to content interventions and describe a placeholder role, while Li et al. (2026) find different causal roles across steps and information flow between nonadjacent positions. Closely related to our study, Aswal et al. (2026) combine matched controls, causal interventions and trajectory analysis. They show that observable reasoning patterns can also appear in controls, and that behavioral influence can be concentrated in a small subspace even when the full states appear nearly static.

3 Experimental Setup

3.1 Latent-Chain Formulation

Given a question x, a latent-reasoning model with parameters θ produces T continuous states before decoding an answer. We index these slots from zero to match the recorded trajectories. Let $\mathbf{h}_t^{(\ell)} \in \mathbb{R}^d$ be the hidden state at layer ℓ of slot t, and let $\mathbf{h}_t$ denote its final-layer state. The latent chain and answer distribution can be written as

$$\mathbf{h}_t = F_\theta(x, \mathbf{h}_{0:t-1}), \qquad p_\theta(y \mid x, \mathbf{h}_{0:T-1}). \tag{1}$$

Here F_θ includes the model-specific mapping from a hidden state to the next input. This is the identity map for COCONUT and includes a projection network for CODI. The model can attend to the question as well as earlier slots, so the notation does not assume that each state depends only on the preceding one. We write $y^\star$ for the gold answer and z_k for step k of an annotated reasoning path, where available. Probe targets z are defined for each task in RQ2.

3.2 MODELS, TASKS, AND CONTROLS

We compare three training recipes on GPT-2 and Llama-3.2-1B-Instruct. COCONUT recycles hidden states without a direct loss on slot content (Hao et al., 2025). CODI aligns the hidden state at the answer position with that of its written reasoning pathway (Shen et al., 2025). SIM-COT-COCONUT adds the per-step supervision of SIM-COT to COCONUT: an auxiliary decoder reconstructs written step k from slot k during training and is discarded at inference (Wei et al., 2026). All main-grid models use $T = 6$ slots. Table 2 summarizes the input mapping, supervision and checkpoint coverage of each recipe.

The tasks cover proof search, syllogisms and arithmetic. ProsQA asks which of two candidate concepts follows from a set of facts and rules (Hao et al., 2025). PrOntoQA asks yes/no questions about logical implications (Saparov & He, 2023). GSM8k contains arithmetic word problems with numerical answers (Cobbe et al., 2021).

For COCONUT and CODI, we use the public checkpoints of Dilgren & Wiegreffe (2026) on both backbones. We also include a second independently trained public COCONUT/ProsQA checkpoint on GPT-2. We train SIM-COT-COCONUT on all three tasks and both backbones at the matched slot budget. On ProsQA, a control on each backbone continues the same COCONUT base for the same training schedule with the auxiliary loss weighted to zero. These controls separate the effect of per-step supervision from extra training. The grid contains twenty-one cells, where a cell is one checkpoint evaluated on one task. Checkpoint sources and training details are in App. A. The Llama-3.2-1B-Instruct/PrOntoQA per-step checkpoint was selected by a saved sweep before interventions, as described in App. E.1.

The COCONUT and SIM-COT-COCONUT comparison holds the inference mechanism and slot budget fixed. Comparisons involving CODI change both mechanism and supervision, so they do not isolate either factor. The two backbones differ in scale and instruction tuning as well as model family. We report them separately and interpret their differences as backbone effects.

3.3 EVALUATION PROTOCOL

Every cell is evaluated on the same frozen set of $N = 300$ held-out problems for its task. Public checkpoints are checked against their reported accuracy before analysis. Our trained checkpoints are anchored to the base models they continue. Split differences and parity checks are reported in App. A.

We keep the model parameters fixed during evaluation. An intervention $\mathcal{I}$ changes the latent computation during the forward pass, giving the clean and intervened predictions

$$\hat{y} = \mathrm{Decode}_\theta(x), \qquad \hat{y}^{\mathcal{I}} = \mathrm{Decode}_\theta(x; \mathcal{I}). \tag{2}$$

Deletion skips latent slots. Content replacement substitutes the vectors fed into the latent loop while keeping the slot positions, and subsequent states are computed under that intervention. Mid-chain edits use the separate protocol described in RQ3. We compare predictions on the same problems and report both accuracy and changes in individual answers, since unchanged average accuracy can hide changes in opposite directions.

Probes are trained and evaluated with folds split by problem, so states from one problem do not appear in both training and test folds. Probe readouts are compared with permuted-label nulls, and trajectory statistics with the nulls specified in RQ3. Each test reports its minimum detectable effect (MDE). Effects too small to resolve are kept separate from bounded null results. Confirming and falsifying criteria were recorded before the first forward pass of each experiment, with later amendments retained alongside the originals. The following sections define the measurements for each RQ, and the appendices give their settings and per-cell results.

Table 1: **Deleting the chain and replacing its contents** on the 3×3 grid of both backbones, with the outcome of each test.

		Delete the chain				Replace the contents (Δ accuracy)			
Task	**Recipe**	C_0	C_1	helped/hurt	verdict	noise	mean	donor	verdict
GPT-2									
GSM8k	COCONUT	0.333	0.103	74/5	consulted	−0.243	−0.220	−0.307	content used
	CODI	0.430	0.263	69/19	consulted	−0.370	−0.283	−0.380	content used
	SIM-COT-COCONUT	0.360	0.080	85/1	consulted	−0.290	−0.263	−0.333	content used
ProsQA	COCONUT	0.973	0.973	0/0	inert	0.000	0.000	0.000	*not reached*
	CODI	0.853	0.850	10/9	unresolved	−0.787[a]	0.000	0.000	*not reached*
	SIM-COT-COCONUT	0.993	0.800	58/0	consulted	−0.047	−0.073	−0.077	content used
PrOntoQA	COCONUT	0.990	0.993	0/1	inert	+0.003	+0.003	+0.003	*not reached*
	CODI	0.933	0.940	0/2	inert	−0.020	+0.003	+0.003	*not reached*
	SIM-COT-COCONUT	1.000	0.827	52/0	consulted	−0.157	−0.440	−0.507	content used
Llama-3.2-1B-Instruct									
GSM8k	COCONUT	0.370	0.253	59/24	consulted	−0.060	−0.047	−0.080	content used
	CODI	0.553	0.360	67/9	consulted	−0.423	−0.247	−0.373	content used
	SIM-COT-COCONUT	0.390	0.307	56/31	consulted	−0.093	−0.070	−0.087	content used
ProsQA	COCONUT	0.983	0.987	0/1	inert	0.000	0.000	0.000	*not reached*
	CODI	0.983	0.940	13/0	consulted	0.000	0.000	0.000	placeholder
	SIM-COT-COCONUT	0.997	0.077	276/0	consulted	0.000	0.000	0.000	placeholder
PrOntoQA	COCONUT	0.990	0.987	1/0	inert	+0.003	0.000	0.000	*not reached*
	CODI	0.940	0.937	1/0	inert	0.000	0.000	+0.003	*not reached*
	SIM-COT-COCONUT	0.993	0.990	2/1	inert	0.000	0.000	0.000	*not reached*

[a] An off-manifold effect: random noise breaks the readout, while the dataset mean and donor vectors move no answer (Δ=0, App. B.3).

4 RQ1: DOES THE MODEL RELY ON THE LATENT CHAIN, ON ITS POSITIONS OR ON ITS CONTENTS?

A latent chain gives the model extra computation before answering, but its presence does not establish that the model needs it. Even when deleting the chain hurts performance, the loss could reflect missing computation steps rather than missing problem-specific content. We therefore test two forms of dependence: whether the model needs the latent slots, and whether its answers are sensitive to the vectors supplied at those slots.

Interventions and readouts. For each cell, we evaluate the same 300 problems with the model parameters and question held fixed. Deletion skips every latent slot and decodes the answer. This tests whether the chain is necessary under that intervention, but removes both its contents and its computation steps. Substitution keeps the slot positions and replaces the vectors fed into the loop with norm-matched Gaussian noise, the dataset-mean vector, or recorded vectors from another problem at the same positions. Subsequent states are recomputed under the intervention. These tests measure how replacement inputs affect the remaining computation and the answer. Noise can move states away from their natural distribution, so noise sensitivity alone is not evidence that the model uses problem-specific content. Mean and donor replacements provide complementary tests, with direction-by-scale controls on selected cells (App. B.4). Pairing deletion with substitution separates two patterns a single corruption cannot: cells where a donor's contents are more disruptive than removing the chain outright, and cells that resist deletion but are also immune to any replacement (the placeholder signature reported below).

We report intact and deleted accuracy (C_0, C_1), prediction changes, and paired counts of problems *helped* by the chain (correct only when intact) and *hurt* (correct only when deleted). These counts distinguish unchanged predictions from changes in opposite directions that cancel in the average. In Table 1, *consulted* means deletion reduces accuracy beyond the cell's MDE, and *inert* means at most 3 answers change. Other deletion outcomes are unresolved. Among consulted cells, *content used* denotes sensitivity to replacement beyond the MDE, while *placeholder* denotes no prediction changes under any tested replacement. Smaller unresolved effects are not assigned to either outcome. Full thresholds and per-cell tests are in App. B. An inert outcome does not prove that the chain has no role in the intact forward pass, just as replacement sensitivity does not by itself establish step-by-step reasoning.

Finding 1: Dependence on the chain varies across tasks. On GPT-2, deleting COCONUT's chain leaves ProsQA accuracy unchanged at 0.973, but reduces GSM8k accuracy from 0.333 to 0.103 (Table 1). The same logical-versus-arithmetic split appears on Llama-3.2-1B-Instruct. CODI also depends on the chain on GSM8k and has little deletion effect on PrOntoQA. Its ProsQA behavior differs by backbone. On GPT-2, deletion loses 10 correct answers and gains 9, leaving the direction unresolved despite the small average gap. On Llama-3.2-1B-Instruct, deletion causes a clear loss. These results show that the training recipe alone does not specify whether the chain is needed.

Finding 2: Per-step supervision can establish dependence on the chain. On GPT-2, SIM-COT-COCONUT depends on its chain on both logical tasks, where the COCONUT base has little deletion effect. On ProsQA, accuracy falls from 0.993 to 0.800, with 58 correct answers lost. The continued-training control remains inert, supporting an effect of the per-step loss beyond extra training time (App. B.1). Truncation provides a further check: in every consulted GPT-2 cell, accuracy rises monotonically as more slots are retained (App. B.2). The Llama-3.2-1B-Instruct/ProsQA per-step model also depends on its chain, while its continued-training control has an unresolved deletion effect below its MDE. This pattern is not universal. The selected Llama-3.2-1B-Instruct/PrOntoQA per-step checkpoint remains inert, and the Llama-3.2-1B-Instruct/GSM8k per-step cell has no matched training-time control.

Finding 3: Dependence on slots does not always imply sensitivity to their contents. All tested GSM8k recipes on both backbones are sensitive to content replacement. Donor vectors can be more harmful than removing the chain altogether. For GPT-2 COCONUT/GSM8k, the accuracy change is -0.307 under donor replacement, compared with -0.230 under deletion. This supports dependence on the supplied vectors beyond the presence of extra slots. The ordering is not required for content sensitivity. On GPT-2 SIM-COT-COCONUT/ProsQA, donor replacement hurts less than deletion, even though the replacement effect is detectable. Other donor-versus-deletion contrasts remain bounded by their MDE (App. B.3). Across the eight content-used cells (Table 6), donor replacement is more harmful than deletion in five, less harmful in one, and undetermined in two.

The Llama-3.2-1B-Instruct/ProsQA CODI and per-step cells show a different pattern. Deletion hurts performance, but no tested replacement changes an answer, including a different-answer donor at twice the natural norm. This is placeholder-like behavior, consistent with the benefit of extra positions studied by Goyal et al. (2024); Pfau et al. (2024). It is bounded by the tested interventions and their sensitivity, since no replacement moves an answer in any Llama-3.2-1B-Instruct/ProsQA cell. The model also retains access to the question during replacement, so this result does not imply that its later computation uses no problem information.

The model's dependence on its latent chain varies with task, supervision and backbone. Deletion and replacement distinguish chains that can be skipped, chains whose supplied vectors affect the answer, and chains that show dependence only when the slots are removed. These tests establish behavioral effects under intervention. RQ2 next examines what information the states contain, including in chains that are not necessary to answer.

5 RQ2: WHAT IS WRITTEN IN THE LATENT CHAIN, AND DOES THE MODEL USE IT?

RQ1 showed which chains the model reads, but not what they contain. An unused chain may still encode information, and the main accounts make different predictions: superposition puts multiple candidate answers in the same slot (Hao et al., 2025; Zhu et al., 2025); stepwise reasoning lays the path out in order, one step per slot (Wei et al., 2026; Deng et al., 2024); and computed-at-the-question reasoning predicts that the answer is already present before the chain begins (Pal et al., 2023; Yang et al., 2024).

Setup. We fit a linear probe $p_{t,\ell}$ to predict a target z from $\mathbf{h}_t^{(\ell)}$. Targets include the final answer concept and gold path step z_k on ProsQA, and a magnitude bin of the numerical answer on GSM8k. We report final-layer readouts by slot here and the full layer sweep in App. C.2. Probe accuracy is measured on held-out problems using the cross-validation protocol of Section 3.3. A target is called readable only when its accuracy exceeds the permuted-label null by more than the cell's MDE. We do not compare amounts of readable content across backbones. The probe of record uses 30 principal components followed by logistic regression. On ProsQA, we also fit a full-dimension ridge probe and

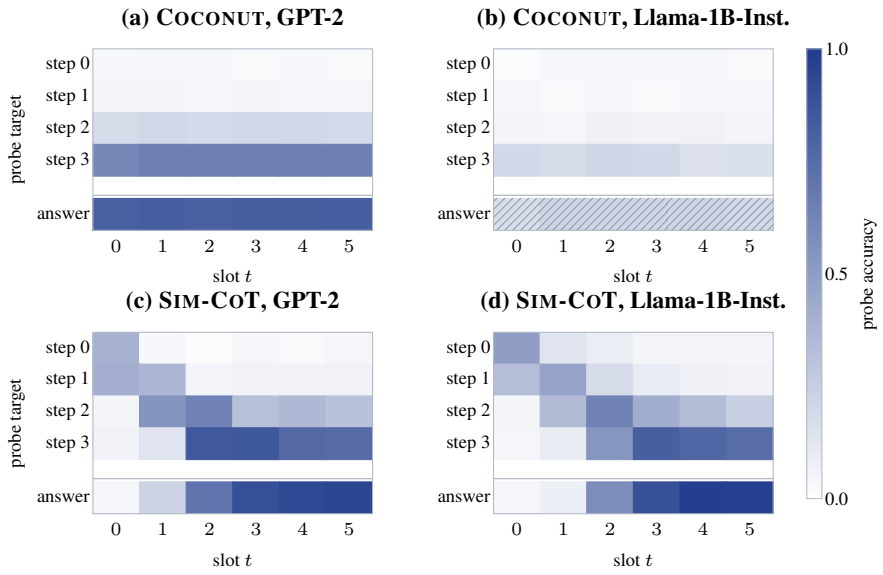

Figure 2: **Linear probe accuracy across slots** on ProsQA. Upper rows: path steps $k \in \{0, \ldots, 3\}$; bottom row: final answer; shared colour scale. Probes read each slot's final-layer state, the vector fed back as the next input (layer 12 on GPT-2, 16 on Llama-1B-Inst.). Hatched: projection artefact of the record probe on the Llama-1B-Inst. COCONUT answer row (App. C.1).

state an unqualified finding only where both agree. A nonlinear probe at the same points provides a further check (largest gain 0.0067). These readouts test access to information in a state. They do not establish that the model uses that information to answer.

Finding 4: The unsupervised chain carries an answer that is already readable before the chain begins; per-step supervision distributes computation across the slots. The answer rows of Figure 2 show two regimes on both backbones. Inert COCONUT chains read the answer from the first slot as well as from the last, while per-step chains start near chance and build the answer monotonically toward the last slot. The hatched Llama-3.2-1B-Instruct COCONUT row is a projection artefact of the probe of record; the ridge probe reads 0.760 from slot 0. The answer is already readable at the question's final hidden state, before any latent slot is written, on both independent trainings (App. C.1): the unsupervised chain is a carrier rather than a workspace, consistent with RQ1's inert verdict and with prior work showing that answers form during the question pass (Pal et al., 2023; Yang et al., 2024). On GSM8k the answer's magnitude becomes more readable along the chain in the GPT-2 COCONUT-mechanism cells and on the Llama-3.2-1B-Instruct per-step twin (App. C.3).

Finding 5: Per-step supervision installs an ordered layout, with step k written into slot k. In the supervised panels, the step rows are diagonal on both backbones: step k becomes readable at slot k; early steps fade along the chain, while later steps and the answer persist to the last slot. In both probe arms, no node off the gold path is readable, even when off-path nodes are matched to on-path nodes in graph degree (App. C.4). The training-time control, trained equally long without the per-step loss, shows no ordered path (App. C.4). This ordered layout is not restricted to chains the model consults: the Llama-3.2-1B-Instruct SIM-COT-COCONUT/ProsQA cell shows the same diagonal structure while RQ1 found its contents replaceable without changing the answer (Section 4).

Finding 6: The unsupervised chain stores the path uniformly, not step by step: the chains with the richest readable content are inert. The unsupervised step rows are flat. Early steps are not linearly readable from any slot, and the readable steps read about equally from every slot, on both backbones and on an independent training, the opposite of the frontier expansion a search account predicts. Nor does any ProsQA cell hold both candidates at once: the highest both-readable rate in either arm is 0.113, against a bar of 30% (App. C.5). Answer-aligned chains hold little of the path at all (App. C.4).

Together, these results show that content follows the training objective: unsupervised chains carry a finished answer across the slots, and per-step supervision lays the path out one step per slot.

6 RQ3: What Is Transmitted Across the Latent Slots, and Can a Settled State Be Revised?

RQ2 showed what each slot contains. RQ3 asks whether that content is merely carried forward or transformed across slots; only the second uses the recurrence that gives a latent chain its capacity for serial computation.

Setup. We measure step size as the cosine distance between consecutive states. To locate the last change in a discrete state sequence, we follow the state-transition approach of Yu et al. (2025):

$$s_t = c\big(P\bar{\mathbf{h}}_t\big)\,, \qquad L = \max\{t : s_t \neq s_{t-1}\}. \tag{3}$$

Here $\bar{\mathbf{h}}_t$ is the mean hidden state across layers, P is a shared random projection, and c is a K-means clustering fitted jointly to the model's latent and written-step states. This gives all positions a common set of state labels without fitting them to the answer. Chains with no transition are counted separately. A cross-problem null replaces slots with those of other problems at the same positions. A chain settles early when its last change L precedes its null L^0 by at least 0.5 slots under all 5 clustering seeds. The reading is unresolved when the null reaches the last slot (App. D.1). The along-answer ratio r measures the share of a step along a fitted answer direction. We call movement answer-directed only when it exceeds a label-permuted refit null (App. D.4). Step size alone is descriptive.

For mid-chain edits, we retain the first i slots of the original problem and use an edited twin with a different gold answer for the remaining computation. The recycling-model driver splices in the twin's recorded suffix, while the CODI driver recomputes later slots. We therefore use this battery to measure the answer response, not to infer re-planning from post-edit trajectories. The adaptive rate $A(i)$ is the fraction of answers that follow the twin, restricted to problems solved on the twin from scratch. Backbone comparisons use matched problems and are reported separately (App. D.5).

Finding 7: On logical tasks, unsupervised and answer-aligned chains settle early; per-step supervision sustains state movement to the last slot and, on ProsQA, toward the answer. Both the discrete reading and the step size show early settling (App. D.2; Figure 3): settled chains halt state changes up to three slots before their null and remain nearly stationary from the first step. Conversely, per-step chains continue moving to the end. Directional analysis reveals that this movement rarely turns toward the answer: in 11 of 15 cells with a usable answer direction, each step's along-answer component stays at or below its null. Per-step ProsQA is the one exception confirmed under both probe arms; the other three are not confirmed under both arms (App. D.4). Even there, the state's identity alone never predicts the answer (Figure 2). As RQ2 implies, a chain whose answer is already readable at its first input has no reason to evolve, whereas per-step supervision forces a new readable step at each slot. On GSM8k the Llama-3.2-1B-Instruct per-step twin settles early like the unsupervised chains, so the split is a finding about the logical tasks (App. D.3).

Finding 8: Whether a mid-chain edit revises the answer is set by the backbone, not by the mechanism or the supervision. Disrupting settled states mid-chain reveals a split by backbone. On GPT-2, answers remain locked to the original prefix: the adaptive rate falls at each later edit point in every consulted cell, directly measuring premature commitment (Wu et al., 2026). On Llama-3.2-1B-Instruct, the answer follows the edit: $A(i)$ stays high under the same edits on matched problems (Figure 4). A held answer on GPT-2 reflects a readout anchored in the prefix slots; on Llama-3.2-1B-Instruct, the same readout follows the twin.

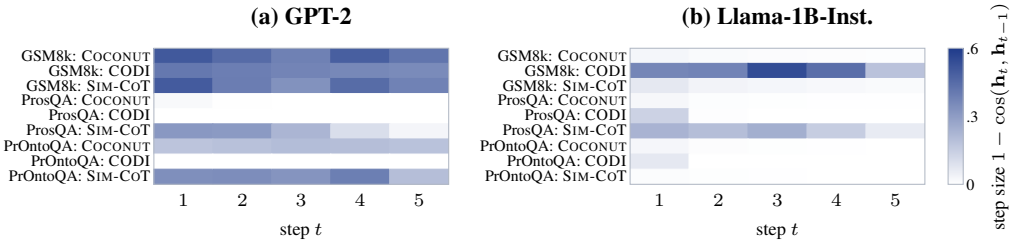

Figure 3: **Step size along the chain,** measured as $1 - \cos(\mathbf{h}_t, \mathbf{h}_{t-1})$, for every grid cell on **(a)** GPT-2 and **(b)** Llama-3.2-1B-Instruct, on the layer-mean state $\bar{\mathbf{h}}_t$ of Eq. 3. Shared colour scale; darker is a larger step size.

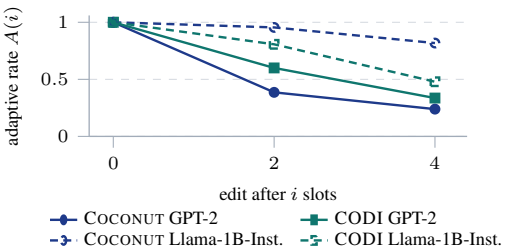

Figure 4: **Answers after a mid-chain edit.** Adaptive rate $A(i)$, the fraction of answers that follow an edit made after i slots, for GPT-2 (solid) and Llama-3.2-1B-Instruct (dashed) on 480-twin sets; every backbone gap exceeds chance after Holm correction (Table 19). After the edit only the prefix belongs to the original problem (App. D.5).

Together, these results show that the recurrence is largely unused for serial computation. Answer aligned chains transmit a state fixed at the question; per-step supervision keeps the state moving, though only on ProsQA is that movement confirmed as answer-directed.

7 DISCUSSION

7.1 A TAXONOMY OF LATENT CHAINS

The three RQs describe different aspects of the same chains. We now combine them to summarize the behaviors observed in our grid (Figure 5). Deletion and substitution determine the use category. Content probes and state dynamics describe the profile within that category. Readability does not override an intervention result, and cells with unresolved measurements are left unclassified.

Inert chains are called *carriers* when they carry a readable answer and settle early, or *empty* when little is readable under our probes. Among chains with the placeholder signature, *bare placeholders* settle early with little content, while *laid-out placeholders* contain ordered steps and continue changing. Content-used chains are *assembling* when they show an ordered step layout and sustained state changes, or *unstructured* when no such layout is recovered. These names summarize the measured profiles. In particular, "assembling" does not establish that slots causally correspond decoded step. The full decision rules, thresholds and per-cell assignments are in App. F.

Where the cells land. The taxonomy (Figure 5) places cells in all six leaves. Every inert chain we profiled is either a carrier or empty. Both placeholders occur on Llama-3.2-1B-Instruct/ProsQA, while content-consulted chains are assembling on the logical tasks and unstructured on GSM8k. The full census, node thresholds, and three worked examples appear in App. F (Table 22).

Each leaf predicts an intervention not used to define it. This makes the taxonomy predictive rather than descriptive: carriers are unaffected by corruption, placeholders remain immune even to a different-answer donor at twice the natural norm, and the only chain helped by donor thoughts is classified as assembling (App. E.2).

Reading a latent trace certifies representation, not computation. Our results reconcile prior findings once we distinguish what a chain represents from what the model uses. The placeholder signature reported by Zhang et al. (2025) appears in two cells, whereas COCONUT is a carrier on our logical tasks. The collapsed superposition reported by Rizvi-Martel et al. (2026) is consistent with what carriers hold, while the staged causal structure and non-local routing of Li et al. (2026) match our consulted GSM8k leaf.

8 CONCLUSION

We identify three separate properties of latent chains. First, the model relies on a chain only when the task requires it or per-step supervision installs that reliance. Second, the chains that encode the most are the ones the model does not use: their answers are already readable before the chain begins, and the slots only carry them. Third, most chains transmit a fixed state from the third slot onward; on the logical tasks, only per-step supervision keeps the state changing, and on ProsQA that movement is directed toward the answer. The three properties are thus separate, and a claim about latent reasoning should say which of them it concerns.

USE OF LARGE LANGUAGE MODELS

In this work, we used generative AI tools to implement analysis code, to provide feedback on the experimental design and on the interpretation of results, and to help draft the paper, its figures and its LaTeX.

REFERENCES

Darpan Aswal, Thomas Palmeira Ferraz, Yongxin Zhou, and Maxime Peyrard. Observable patterns are not explanations: A causal-geometric analysis of latent reasoning models. *arXiv preprint arXiv:2606.12689*, 2026.

Jeffrey Cheng and Benjamin Van Durme. Compressed chain of thought: Efficient reasoning through dense representations. *arXiv preprint arXiv:2412.13171*, 2024.

Karl Cobbe et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

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? *arXiv preprint arXiv:2602.22441*, 2026.

Yuntian Deng, Yejin Choi, and Stuart Shieber. From explicit CoT to implicit CoT: Learning to internalize CoT step by step. *arXiv preprint arXiv:2405.14838*, 2024.

Connor Dilgren and Sarah Wiegreffe. Are latent reasoning models easily interpretable? *arXiv preprint arXiv:2604.04902*, 2026.

Sabari Iyyappan Duraipandian, Shreya Sanjay Boyane, Manju Nagesh, Jerome Francis, Archana Vaidheeswaran, and Kevin Zhu. Interpreting latent CoT reasoning as dynamical systems. In *ICML 2026 Workshop on Foundations of Deep Generative Models (FoGen)*, 2026. arXiv:2607.09698.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. Amnesic probing: Behavioral explanation with amnesic counterfactuals. *Transactions of the Association for Computational Linguistics*, 9:160–175, 2021.

Jonas Geiping, Sean McLeish, Neel Jain, John Kirchenbauer, Siddharth Singh, Brian R. Bartoldson, Bhavya Kailkhura, Abhinav Bhatele, and Tom Goldstein. Scaling up test-time compute with latent reasoning: A recurrent depth approach. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Sachin Goyal, Ziwei Ji, Ankit Singh Rawat, Aditya Krishna Menon, Sanjiv Kumar, and Vaishnavh Nagarajan. Think before you speak: Training language models with pause tokens. In *International Conference on Learning Representations (ICLR)*, 2024.

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

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, et al. Measuring faithfulness in chain-of-thought reasoning. *arXiv preprint arXiv:2307.13702*, 2023.

Zirui Li, Xuefeng Bai, Kehai Chen, Yizhi Li, Jian Yang, Chenghua Lin, and Min Zhang. Dynamics within latent chain-of-thought: An empirical study of causal structure. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, 2026.

Koyena Pal, Jiuding Sun, Andrew Yuan, Byron C. Wallace, and David Bau. Future lens: Anticipating subsequent tokens from a single hidden state. In *Proceedings of the 27th Conference on Computational Natural Language Learning (CoNLL)*, 2023.

Jacob Pfau, William Merrill, and Samuel R. Bowman. Let's think dot by dot: Hidden computation in transformer language models. In *First Conference on Language Modeling (COLM)*, 2024.

Abhilasha Ravichander, Yonatan Belinkov, and Eduard Hovy. Probing the probing paradigm: Does probing accuracy entail task relevance? In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pp. 3363–3377, 2021.

Michael Rizvi-Martel, Guillaume Rabusseau, and Marius Mosbach. The illusion of superposition? a principled analysis of latent thinking in language models. *arXiv preprint arXiv:2604.06374*, 2026.

Abulhair Saparov and He He. Language models are greedy reasoners: A systematic formal analysis of chain-of-thought. In *International Conference on Learning Representations (ICLR)*, 2023.

Zhenyi Shen, Hanqi Yan, Linhai Zhang, Zhanghao Hu, Yali Du, and Yulan He. CODI: Compressing chain-of-thought into continuous space via self-distillation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 677–693, 2025.

Xilin Wei, Xiaoran Liu, Yuhang Zang, Xiaoyi Dong, Yuhang Cao, Jiaqi Wang, Xipeng Qiu, and Dahua Lin. SIM-CoT: Supervised implicit chain-of-thought. In *International Conference on Learning Representations (ICLR)*, 2026.

Junda Wu, Yuxin Xiong, Xintong Li, Sheldon Yu, Zhengmian Hu, Tong Yu, Rui Wang, Xiang Chen, Jingbo Shang, and Julian McAuley. CTRLS: Chain-of-thought reasoning via latent state transition. In *Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2026.

Sohee Yang, Elena Gribovskaya, Nora Kassner, Mor Geva, and Sebastian Riedel. Do large language models latently perform multi-hop reasoning? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024.

Sheldon Yu, Yuxin Xiong, Junda Wu, Xintong Li, Tong Yu, Xiang Chen, Ritwik Sinha, Jingbo Shang, and Julian McAuley. Explainable chain-of-thought reasoning: An empirical analysis on state-aware reasoning dynamics. In *Findings of the Association for Computational Linguistics: EMNLP*, 2025.

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. *arXiv preprint arXiv:2512.21711*, 2025.

Hanlin Zhu, Shibo Hao, Zhiting Hu, Jiantao Jiao, Stuart Russell, and Yuandong Tian. Reasoning by superposition: A theoretical perspective on chain of continuous thought. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

A Experimental details

This appendix specifies the grid (A.1–A.2), the parity and pre-registration rules every run obeyed (A.3, A.7), and the three instruments: interventions (A.4), probes (A.5) and mid-chain edits (A.6).

A.1 Models and the grid

SIM-CoT-Coconut runs the identical inference computation as Coconut; the two differ only in what supervised the slots. SIM-CoT-Coconut's auxiliary decoder reconstructs written step k from slot k during training only, so the inference computation is Coconut's; the training-time control therefore isolates the supervision signal from the extra epochs. CODI differs from both in mechanism and supervision, so every comparison involving it is descriptive. Crossing mechanism with task removes the confound of the usual comparison, Coconut on a logical task against CODI on arithmetic. ProsQA asks which of two offered concepts follows from a subject, so a two-candidate probe is well posed; GSM8k's computed answers make the probe target a magnitude bin with edges frozen from the training split. The Llama-3.2-1B-Instruct twins are larger and instruction-tuned, which is why they replicate the battery but are never pooled with GPT-2.

Table 2: **The model axis.** Input to each slot, what supervises the slots, the slot budget and the grid cells per model. "ours": trained by us; †: trained at COCONUT's slot budget (the released SIM-COT-COCONUT checkpoint with 10 slots is the parity anchor, App. E.2).

Model	Input to slot $k{+}1$	What supervises the slots	Slots	Cells (GPT-2; Llama-1B-Inst. twins)
COCONUT	last hidden state at slot k, recycled as the next input embedding	nothing beyond the answer loss	6	GSM8k, ProsQA ($\times 2$), PrOntoQA; all three twinned
continued (control)	identical to COCONUT	nothing; base continued for SIM-COT-COCONUT's schedule, auxiliary loss $= 0$	6	ProsQA (ours; Llama-1B-Inst. twin ours)
SIM-COT-COCONUT	identical to COCONUT	per-step content: an auxiliary decoder reconstructs written step k from slot k; discarded at inference	$6^{\dagger}$	GSM8k, ProsQA, PrOntoQA (all ours); ProsQA and GSM8k twins (ours); PrOntoQA twin at a swept checkpoint (ours)
CODI	recycled state through a projection network	answer-position alignment: self-distillation from the model's own written pathway	6	GSM8k, ProsQA, PrOntoQA; all three twinned

Table 3: **What each contrast identifies.** What is held fixed and what varies in the four contrasts the paper draws; only the first two vary a single factor.

Contrast	Held fixed	Varies	Identifies
COCONUT vs. SIM-COT-COCONUT	mechanism, base weights, slot budget (6), tasks, evaluation set	slot supervision, plus the training it entails	supervision, once the control rules out training time
COCONUT vs. its continued control	all of the above and the training schedule	training time alone	whether continuation alone changes the chain
CODI vs. either recycling model	tasks, slot budget, evaluation set	mechanism and supervision jointly	neither factor alone; any grouping is a hypothesis
GPT-2 cell vs. Llama-1B-Inst. twin	mechanism, task, evaluation set	base-model family: scale, instruction tuning, fine-tuning recipe	the checkpoint family, not scale

A.2 WHAT EACH CONTRAST IDENTIFIES

Table 3 lists the four contrasts the paper draws, what each holds fixed and what it therefore identifies; only the first two vary a single factor.

A.3 PARITY GATES

No analysis runs on a checkpoint before it reproduces its published accuracy. Judgement is against the published spread rather than a point value, because independent reproductions of the same cell differ by up to several points across groups. Where our number falls outside tolerance, the difference is diagnosed and documented rather than absorbed by widening the tolerance.

A documented protocol difference. Our frozen 300-problem evaluation prefixes run above the published numbers on ProsQA, by amounts that grow as the reference model gets weaker. Running *our* scorer on the original authors' full split reproduces their table to within 0.002 (for example 0.818 against a published 0.816), which localises the difference to the split rather than to scoring. All internal comparisons share the same frozen prefix, so the difference does not affect any within-paper contrast; it is recorded so that absolute accuracies are not read as disagreements with prior work.

A.4 INTERVENTION CONDITIONS

Every corruption condition replaces the vector fed at a slot position and leaves the rest of the pipeline untouched. Semantics are within-loop: the substituted vector enters the recycling loop, so downstream slots recompute from it.

- **Noise**: an isotropic Gaussian draw, unit-normalised per position and rescaled to ε_p, the mean natural slot-vector norm at that position over the evaluation set, with the seed fixed per (model, problem). That per-position mean norm is the *only* matched moment: the direction is uniform on the sphere, so the slot distribution's mean vector and covariance are not reproduced, and every noise vector at position p carries exactly ε_p rather than the natural spread of norms.
- **Dataset mean**: the mean slot vector over the evaluation set.
- **Donor**: the recorded vector of a different problem from the same dataset at the same slot position.

Magnitude is reported in between-problem units, bracketed by a same-problem control at 0.000 and a different-problem control at ≈ 1.0. An injection audit verifies that the conditions are routed through an identical code path and differ only in the substituted content, and records how far each substitute is turned from the vector it replaces. Representations in these models are strongly anisotropic, so the dataset-mean and donor substitutes stay nearly parallel to natural ones (cosine 0.98–1.00 across the audited cells) and are not trivially out of distribution. Noise is the deliberate exception: a uniform direction in d dimensions is near-orthogonal to what it replaces (cosine $|\cdot| < 0.002$), so it is matched in magnitude alone. The noise condition therefore probes the *removal* of slot content, whereas the mean and donor conditions probe its *replacement* by confidently wrong content — a distinction the noise-versus-mean asymmetry of the grid turns on. The audit is run on COCONUT/ProsQA and its training-time control; the construction it verifies is shared by every cell.

A.5 PROBES AND NULLS

Probes are fit with PCA to 30 dimensions followed by logistic regression under five-fold cross-validation over problems, never over positions within a problem. Nulls permute labels 200 times and refit. Each cell reports the minimum effect it could have detected, and cells that could not have resolved the effect of interest are excluded from verdicts rather than counted as nulls. Every claim is additionally checked with a training-free vocabulary projection, which shares no fitting artefacts with the probe.

Probe configuration. Probes are fitted for every slot t and layer ℓ against a permuted-label null; on ProsQA the probe of record (PCA to 30 components, logistic regression) is paired with a full-dimension ridge arm and a verdict is stated alone only where both agree. The probe's sensitivity ceiling falls with hidden size d, so no cross-backbone comparison of how much a chain holds is made. The superposition test scores each candidate with its own binary probe and counts problems on which both read from the same slot against a bar fixed before measurement at 30%. The layout test asks whether the gold path is present, localised, ordered and specific to the path rather than to graph neighbourhood, with a degree-matched off-path control.

Chain-decoding method notes. The gold path is parsed from the recorded trace rather than recovered by breadth-first search, which agrees on only 227/300 problems when several paths reach the target; an edge-set verification passes 300/300. The off-path control is degree-matched because a random off-path node is typically a leaf, so an unmatched contrast would score graph topology rather than path membership. Three cells returned accuracies identical to sixteen digits, which the shared cross-validation folds explain: at this cell count the probability of at least one three-way tie is ≈ 0.83.

A.6 MID-CHAIN EDITS: GATES, ENLARGEMENT AND SPECIFICITY

1. **Edit validity.** The pre-edit answer must be provably unreachable in the edited problem, verified by replaying the annotated solution. 0.810 of the original 300 constructed GSM8k twins survive (0.640 on ProsQA).
2. **Manipulation check.** The swap must actually displace the states, measured against a noise floor and a permutation null.
3. **Ceiling anchor.** Analysis is restricted to problems the model solves on the edited problem from scratch, so that failure to adapt is distinguished from inability to solve.
4. **Specificity.** An answer-preserving twin provides the control; the reported quantity is the difference between flip and control displacement against its own null.

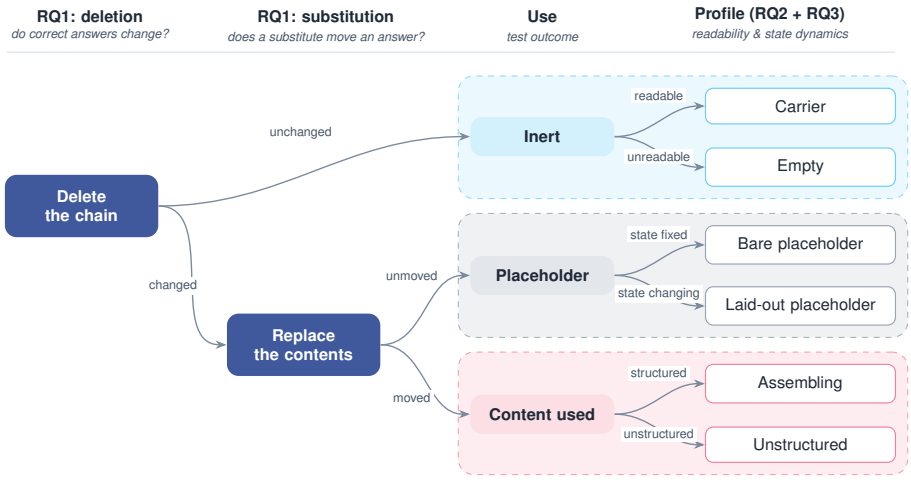

Figure 5: **A taxonomy based on the three RQs.** Deletion and content replacement distinguish causal-use outcomes. Readability and state movement describe the profile within each outcome. Unresolved cases are not assigned to a leaf.

5. **Vacuity.** On cells where deletion changes no answer, adaptation rates restate inertness rather than measuring re-exploration, and are reported as undefined for that purpose.

The backbone comparison was enlarged once, under a protocol frozen before the new twins were scored: the target of 480 twins was set by power calculation, the evaluation set extended to a byte-identical prefix superset of the original, both mechanism families enlarged together, and N fixed in advance with no second enlargement permitted. The enlargement moved the projection-family intersection from 65 to 140 paired problems and its discordant counts from 18 and 23 to 39 and 38; gaps that had reached 83%–86% of their minimum detectable effect before the enlargement cleared it after. The $i{=}0$ position is excluded from the Holm family, since adaptation there is 1.000 by construction.

Specificity controls for edit injection. Flip-minus-control displacement runs $+0.25$ to $+0.40$ against null 95th percentiles of 0.04–0.07; the effect is specific to answer-changing edits, not to editing.

A.7 Pre-registration

Confirming and falsifying readings are written into the run directory before the first forward pass of each experiment. Where a result required a reading not anticipated by the registration, the amendment is recorded in the same directory alongside the original, rather than replacing it.

A.8 Overview of Research Questions

B RQ1 supplement: deletion and substitution

This appendix holds the per-cell record behind Section 4: the controls (B.1), the truncation curves (B.2), the substitution contrasts with their minimum detectable effects and paired tests (B.3) and the direction-by-scale control (B.4). GPT-2 and Llama-3.2-1B-Instruct are reported apart and never pooled.

B.1 Replication and training-time controls

The second training agrees with the primary COCONUT/ProsQA cell on every condition within 0.010 (Section 4); the loss-off controls separate the effect of per-step supervision from that of training time, since they share the per-step twin's schedule and differ only in the auxiliary loss. The GPT-2 controls are as inert as the cell they continue; the Llama-3.2-1B-Instruct control moves answers both ways below its MDE (0.031) and is unresolved.

Table 4: **The three control cells under deletion**, in the format of Table 1. The second training is an independent COCONUT/ProsQA run on GPT-2; the loss-off controls continue the checkpoint of each backbone for the per-step twin's schedule with the auxiliary loss weighted to zero (Section 3.2).

Control	Backbone	C_0	C_1	helped/hurt	verdict
2nd training	GPT-2	0.973	0.977	0/1	inert
loss off	GPT-2	0.970	0.973	0/1	inert
loss off	Llama-1B-Inst.	0.877	0.850	16/8	unresolved

B.2 TRUNCATION

Table 5: **Accuracy by slots kept.** k=0 is full deletion (C_1), *all* is the intact chain (C_0); the interior conditions keep two and four slots (App. **??**).

Cell	k=0	k=2	k=4	**all**	**shape**
GPT-2 — the supervision grid					
COCONUT/GSM8k	0.103	0.197	0.260	0.333	monotone
COCONUT/ProsQA	0.973	0.970	0.973	0.973	flat
COCONUT/PrOntoQA	0.993	0.993	0.993	0.990	flat
SIM-COT-COCONUT/GSM8k	0.080	0.203	0.270	0.360	monotone
SIM-COT-COCONUT/ProsQA	0.800	0.960	0.993	0.993	monotone
SIM-COT-COCONUT/PrOntoQA	0.827	0.857	0.913	1.000	monotone
CODI/GSM8k	0.263	0.287	0.387	0.430	monotone
CODI/ProsQA	0.850	0.867	0.853	0.853	flat
CODI/PrOntoQA	0.940	0.937	0.940	0.933	flat
Llama-3.2-1B-Instruct — the backbone twins					
COCONUT/GSM8k	0.253	0.367	0.373	0.370	saturates early
CODI/GSM8k	0.360	0.487	0.453	0.553	dips at k=4
SIM-COT-COCONUT/GSM8k	0.307	0.383	0.387	0.390	saturates early

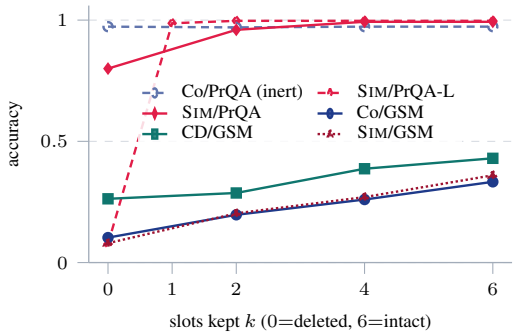

Figure 6: **Truncation curves.** Accuracy against the number of slots kept (k=0 deletion, k=6 intact); the Llama-3.2-1B-Instruct per-step ProsQA twin (dashed) adds the one-slot point.

Truncation keeps the first k slots and inserts the end-of-thought marker, $y^{(k)} = g_\theta(x, \mathbf{h}_{0:k-1})$, so k=0 is deletion and k=T the intact chain. On GPT-2 the last two slots of every consulted GSM8k cell still carry 24%–32% of the deletion gap, so the dependence is graded with no early saturation; on the Llama-3.2-1B-Instruct per-step ProsQA twin one slot restores accuracy from 0.077 to 0.987 and every further slot moves at most three answers, and the per-step GSM8k twin recovers 0.92 of its gap with two slots. On the Llama-3.2-1B-Instruct backbone the GSM8k curves lose monotonicity—the flat tail means the model has finished early, not that thinking hurts. On GPT-2 every inert cell is flat at all depths and every consulted cell rises monotonically, so the graded shape tracks reliance rather than architecture; CODI/GSM8k's first two slots buy 14% of its deletion gap.

B.3 SUBSTITUTION: PER-CELL CONTRASTS

Across the eighteen cells of Table 6 the tally is 5 donor-worse-than-deletion, 7 not applicable, 3 bounded (the released ten-slot cell is a fourth, bounded on the frozen grid and resolved at N=900; App. E.2) and 3 with the donor above deletion (two inversions and one placeholder); the two significant outcomes are opposite in direction and are never pooled. In every content-used GSM8k

Table 6: **Substitution, per cell.** Change in accuracy against the intact chain when every slot is replaced by noise, by the dataset-mean vector or by another problem's recorded thoughts (donor), and the donor-versus-deletion contrast. Discordant pairs read donor-worse / deletion-worse; p is McNemar. GPT-2 and Llama-3.2-1B-Instruct are never pooled.

Cell	noise	mean	donor	donor vs. deletion
GPT-2 — the supervision grid				
COCONUT/GSM8k	-0.243	-0.220	-0.307	donor worse 27/4, p=0.000
SIM-COT-COCONUT/GSM8k	-0.290	-0.263	-0.333	donor worse 20/4, 0.053>0.048
SIM-COT-COCONUT/PrOntoQA	-0.157	-0.440	-0.507	donor worse 124/24, 0.333>0.116
CODI/GSM8k	-0.370	-0.283	-0.380	donor worse 73/9, p=0.000
SIM-COT-COCONUT/ProsQA	-0.047	-0.073	-0.077	**donor above deletion**: 14/49, 0.117>0.074
CODI/ProsQA	-0.787	0.000	0.000	bounded (9/10; needs N>30000)
COCONUT/ProsQA	0.000	0.000	0.000	not applicable (inert under deletion)
COCONUT/PrOntoQA	+0.003	+0.003	+0.003	not applicable (inert under deletion)
CODI/PrOntoQA	-0.020	+0.003	+0.003	not applicable (inert under deletion)
Llama-3.2-1B-Instruct — the backbone twins				
CODI/GSM8k	-0.423	-0.247	-0.373	donor worse 66/12, p=0.000
COCONUT/GSM8k	-0.060	-0.047	-0.080	bounded (needs N≈1800; pool 1319)
SIM-COT-COCONUT/GSM8k	-0.093	-0.070	-0.087	bounded (44/43, p=1.000; needs N≈271000; pool 1319)
CODI/ProsQA	0.000	0.000	0.000	donor above deletion: 0/13, p=0.000
COCONUT/ProsQA	0.000	0.000	0.000	not applicable (inert under deletion)
COCONUT/PrOntoQA	+0.003	0.000	0.000	not applicable (inert under deletion)
CODI/PrOntoQA	0.000	0.000	+0.003	not applicable (inert under deletion)
SIM-COT-COCONUT/ProsQA	0.000	0.000	0.000	donor above deletion: no substitute moves an answer (placeholder)
SIM-COT-COCONUT/PrOntoQA	0.000	0.000	0.000	not applicable (inert under deletion)

cell and on PrOntoQA, another problem's thoughts cost more than no thoughts, the ordering a format account of ablation damage cannot produce.

Sensitivity of the substitution readout on Llama-3.2-1B-Instruct/ProsQA. No substitute has moved an answer in any Llama-3.2-1B-Instruct/ProsQA cell, so the two placeholder verdicts bound the range the harness can express on that backbone. The injection is audited to reach the model, the identical code path moves 24 answers on the GPT-2 twin, and the maximal override (a donor with a different gold answer at twice the natural norm) moves 0 of 300.

The triage. Nine donor-versus-deletion contrasts that previously read as unresolved separate into three situations. Six were never underpowered: deletion had already established their chains inert, so the contrast's presupposition—that there is content whose corruption could matter—fails, and they are not applicable at any N. Three are bounded, a stronger statement than unresolved: the COCONUT-Llama-3.2-1B-Instruct/GSM8k contrast would need N≈1800 against a total task pool of 1319, CODI/ProsQA's 9/10 split would need N>30000, and the ten-slot cell is bounded on the frozen grid. That last one was fixable and was fixed: re-run at a pre-registered N=900 it resolves to donor worse than deletion (52/18, p=0.00006; gap 0.038 against MDE 0.026). The per-step Llama-3.2-1B-Instruct/GSM8k twin, entered last, is bounded in the extreme: its 44/43 discordant pairs give a gap of 0.003 against an MDE of 0.092, so every substitute costs about what deletion costs and resolving the ordering would need N≈271000 against a pool of 1319.

The inversion. Under per-step supervision the two logical tasks disagree about what the chain carries. PrOntoQA's donor transplant is devastating (donor change -0.507, worse than noise at -0.157), while ProsQA's transplant sits significantly *above* deletion: a donor's thoughts beat having none, and deleting costs 0.193 where corrupting costs at most -0.077. Same supervision, same architecture: one task's chain carries content, the other's mostly carries computation.

B.4 THE DIRECTION-BY-SCALE CONTROL

Two anomalies are confirmed real rather than artifactual. CODI/ProsQA does not read its chain for content—a donor changes zero answers—yet requires whatever occupies the slots to lie on the manifold it was trained against (-0.787 at either norm); its Llama-3.2-1B-Instruct twin drops even that requirement, a clean placeholder signature (deletion changes 22 answers, 13 of them from correct to wrong, while noise, mean and donor change zero, and the identical code path moves 252 answers on CODI-Llama-3.2-1B-Instruct/GSM8k). And SIM-COT-COCONUT/PrOntoQA's dataset-mean thought is far more damaging than a random one at both magnitudes: a random vector is

Table 7: **Direction-by-scale** 2×2 on the pre-registered cell list, noise seed reused. Columns: random direction at the noise norm ε and at the norm of the dataset-mean vector; mean direction at its own norm and at ε.

Cell	rand@ε	rand@$\|\|\mu\|\|$	mean@$\|\|\mu\|\|$	mean@ε	**reading**
GPT-2 — the supervision grid					
CODI/ProsQA	-0.787	-0.790	0.000	0.007	direction
CODI/GSM8k	-0.370	-0.320	-0.283	-0.297	direction
COCONUT/GSM8k	-0.243	-0.257	-0.220	-0.223	mixed (all four within 0.037)
SIM-COT-COCONUT/PrOntoQA	-0.157	-0.150	-0.440	-0.440	direction
Llama-3.2-1B-Instruct — the backbone twins					
CODI/GSM8k	-0.423	-0.420	-0.247	-0.247	direction
SIM-COT-COCONUT/GSM8k	-0.093	-0.080	-0.070	-0.070	mixed (all four within 0.023)

uninformative, the mean vector is actively misleading—shaped like a valid thought, pointing nowhere in particular. On the per-step Llama-3.2-1B-Instruct/GSM8k twin the four deltas lie within 0.023 of each other, half that cell's MDE, so direction and scale cannot be separated there; its content-used verdict rests on the mean and donor substitutes, which do not depend on the attribution.

C RQ2 SUPPLEMENT: PROBING

This appendix holds the probe record behind Section 5: the answer through depth and by slot (C.1), the position-by-layer sweep (C.2), accumulation and strength on GSM8k (C.3), the layout funnel and the cells held out of the body (C.4) and the superposition test (C.5). Probe configuration is in App. A.

C.1 THE ANSWER THROUGH DEPTH AND BY SLOT

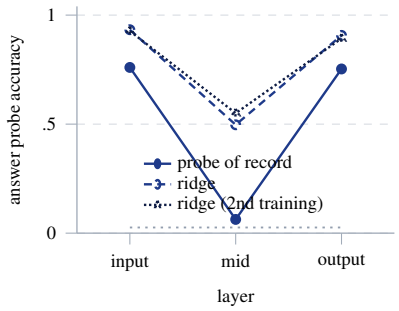

Figure 7: **The unsupervised chain's answer through the stack** at the first slot (COCONUT/ProsQA, GPT-2), under both probe arms and on the second training.

The answer is present at the slot's input layer, the question's final hidden state, weaker mid-stack and re-linearised at the output, under both probe arms and on the second training (Figure 7).

Numbers behind Section 5. Through depth the unsupervised ProsQA chain's answer reads 0.760 at the embedding layer, 0.063 at its mid-stack trough and 0.753 at the output under the probe of record, and 0.933, 0.497 and 0.907 under ridge. By slot, the same chain runs from 0.597 to 0.607 (0.847 to 0.813 under ridge) while the supervised chain runs from unreadable at its first two slots to 0.963 (0.973) at the last. The peak-to-mean ratio of node decodability across slots is 1.16 on the GPT-2 unsupervised chain and 1.17 on its Llama-3.2-1B-Instruct twin.

C.2 POSITION-BY-LAYER SWEEP

No cell on any ProsQA row reaches the pre-registered bar of 30% in either arm (0 cells); the largest two-candidate rate is 0.090 under the probe of record and 0.113 under ridge (SIM-COT-COCONUT), a move between two sub-threshold bands.

Table 8: **Position × layer sweep** under both probe arms. *cleared*: cells above their own permutation null; ≥5×: cells at five times chance; *peak*: the strongest cell and its location. Numeric rows use chance 0.200 and MDE 0.238; concept rows 0.026 and 0.042. The ridge arm exists on ProsQA only. The last layer column (12 on GPT-2, 16 on Llama-1B-Inst.) is the representation Figure 2 reports.

Cell	Task / arm	cleared	≥5×	peak	at
GPT-2					
COCONUT	ProsQA, record	78/78	71	**0.820**	p2 L0
	ProsQA, ridge	78/78	78	**0.927**	p0 L0
COCONUT′	ProsQA, record	78/78	66	**0.823**	p5 L1
	ProsQA, ridge	78/78	78	**0.933**	p0 L0
SIM-COT-COCONUT	ProsQA, record	56/78	54	**0.967**	p5 L7
	ProsQA, ridge	55/78	54	**0.987**	p5 L10
CODI	ProsQA, record	27/78	0	0.100	p0 L1
	ProsQA, ridge	20/78	7	0.177	p2 L1
CODI	GSM8k	78/78	—	0.417	p4 L12
SIM-COT-COCONUT (10 sl.)	GSM8k	130/130	—	0.650	p9 L5
Llama-3.2-1B-Instruct					
COCONUT	ProsQA, record	102/102	86	0.217	p4 L7
	ProsQA, ridge	102/102	102	**0.797**	p2 L2
CODI	ProsQA, record	63/102	6	0.220	p1 L7
	ProsQA, ridge	70/102	32	0.410	p1 L6

Table 9: **Content accumulation across slots on GSM8k,** target magnitude of the final answer (chance 0.200, MDE 0.238): first-slot and last-slot accuracy and their difference.

Cell	Mechanism	first	last	Δ	Profile
GPT-2					
COCONUT	recycling	0.310	0.587	**+0.277**	builds
SIM-COT-COCONUT (6 sl.)	recycling + per-step	0.320	0.553	**+0.233**	builds
SIM-COT-COCONUT (10 sl.)	recycling + per-step	0.320	0.607	**+0.287**	builds
CODI	distillation	0.407	0.377	-0.030	flat (p=0.452)
Llama-3.2-1B-Instruct					
COCONUT	recycling	0.370	0.383	+0.013	flat (p=0.736)
CODI	distillation	0.393	0.380	-0.013	flat (p=0.737)
SIM-COT-COCONUT (6 sl.)	recycling + per-step	0.363	0.487	0.123	builds at the MDD (0.112; App. E.1)

Table 10: **Per-cell strength on GSM8k** at the best slot, target magnitude of the final answer (chance 0.200, MDE 0.238). Best-slot selections are uncorrected maxima.

Cell	Supervision	slot	accuracy
GPT-2			
COCONUT	none	p5	0.540
SIM-COT-COCONUT (6 sl.)	per-step	p5	0.540
SIM-COT-COCONUT (10 sl.)	per-step	p8	0.637
CODI	aligned	p4	0.417
Llama-3.2-1B-Instruct			
COCONUT	none	p5	0.387
CODI	aligned	p2	0.410
SIM-COT-COCONUT (6 sl.)	per-step	p5	0.447

C.3 Accumulation and strength on GSM8k

Accumulation appears in every GPT-2 recycling cell (first-to-last gains +0.23 to +0.29, $p < 0.0001$) and, at the readout's own resolution, on the Llama-3.2-1B-Instruct per-step twin; in no other; with single, confounded off-diagonal cells the interaction is a hypothesis (Table 9). Best-slot and best-family selections in Table 10 are uncorrected maxima: a replicate scores lower at the selected slot on 19/30 rows, and fold-rule instability has mean 0.026 (p90 0.054), which the spread of 0.250 survives. Cross-backbone amounts are not compared (App. **??**).

C.4 Layout and held-out cells

Table 11: **Layout on ProsQA: cells held out of the body and the chain funnel** under both probe arms (record / ridge). *elig./both*: eligible slots and the pooled both-readable rate at the adjudicating slot; *present*: path cells present of 24; *local.*: localisation; *off-path*: live off-path cells. COCONUT′ is the independently trained replication checkpoint.

Cell	Superv.	elig.	both	present	local.	off-path	Chain verdict
GPT-2							
COCONUT (primary)	none	6/6	0.080 / 0.090	15/24 / 23/24	—	0/24 / 0/24	present, not localised (both arms)
COCONUT′ (replication)	none	6/6	0.100 / 0.067	18/24 / 24/24	—	0/24 / 0/24	present, not localised (both arms)
SIM-COT-COCONUT	per-step	5/6	0.067 / 0.083	18/24 / 20/24	✓	0/24 / 0/24	localised and ordered (both arms)
CODI	aligned	0/6 / 0/6	n/a	15/24 / 18/24	—	4/24 / 0/24	not localised; specificity probe-dependent
Llama-3.2-1B-Instruct							
COCONUT	none	6/6	0.017 / 0.060	11/24 / 12/24	—	0/24 / 0/24	present, not localised (both arms)
CODI	aligned	0/6 / 6/6	n/a / 0.023	13/24 / 17/24	—	0/24 / 2/24	not localised; specificity probe-dependent

The GPT-2 answer-aligned cell adjudicates the bar in neither arm; its Llama-3.2-1B-Instruct twin becomes adjudicable under ridge (sub-bar). Stage 4 is probe-dependent on the two answer-aligned cells and on no other, so their specificity is unresolved; the replication checkpoint agrees with the primary in both arms. The two answer-aligned cells are where the probe arm matters. Both fail localisation under both arms, with on-path accuracies of 0.026–0.107 against the supervised cell's 0.034–0.872. The degree-matched specificity control then flips with the arm in opposite directions: on GPT-2 4/24 off-path cells are live under the probe of record and 0/24 under ridge, on Llama-3.2-1B-Instruct 0/24 and 2/24. A verdict that depends on the probe family is a statement about the instrument until a second family agrees, so specificity on these cells is unresolved; "neighbourhood, not chain" is not a verdict on either. Both arms are linear, so "holds little" remains a statement about linear accessibility (Limitations); the non-linear control of App. **??** bounds how much that qualification can hide.

C.5 One candidate at a time

No chain holds both candidates at once. Each ProsQA candidate is scored with its own binary probe and the problems on which both read from the same slot are counted against a bar fixed before measurement at 30%. The highest rate at any slot on any cell is 0.100 under either arm, the test could have detected a rate far below the bar, and a layer sweep and a non-linear probe do not change this (Table 8). Where the question cannot be asked (CODI on GPT-2 has no readable answer, PrOntoQA is yes/no, GSM8k has no candidate list) the cell is not applicable rather than negative.

Table 12: **One candidate at a time.** Rate at which both offered ProsQA answers are readable from the same slot, at each cell's most generous eligible slot, under both probe arms, against a bar of 90 problems of 300; ceilings are the readout's measured sensitivity per arm.

				both		
Cell	**Superv.**	**elig.**	**slot**	**record**	**ridge**	**Verdict**
GPT-2 (ceiling 0.817 record, 1.000 ridge)						
COCONUT	none	6/6	p0	**0.080**	**0.090**	sub-bar, both arms
SIM-COT-COCONUT	per-step	5/6	p4	**0.067**	**0.083**	sub-bar, both arms
Llama-3.2-1B-Instruct (ceiling 0.627 record, 0.960 ridge)						
COCONUT	none	6/6	p4	**0.017**	**0.060**	sub-bar, both arms
CODI	aligned	0/6, 6/6	p0	n/a	**0.023**	sub-bar, ridge only

The largest rate observed is 30 problems (the replication checkpoint) and 0.090 under ridge. Each candidate is scored by its own binary probe, because a multinomial softmax couples the two scores and manufactures anti-correlation (-0.94 on synthetic data); co-occurrence null 0.0025, MDE 0.0072. The GPT-2 answer-aligned cell is not applicable.

The second arm, per slot. Table 13 gives the pooled both-readable rate at every slot under both arms, so the bar can be read slot by slot rather than only at the adjudicating one; a dash marks a slot at which the gold concept is not readable in that arm.

Table 13: **Pooled both-readable rate by slot,** probe of record / ridge, six ProsQA cells (bar 30%; ceilings 0.817 / 1.000 on GPT-2, 0.627 / 0.960 on Llama-3.2-1B-Instruct). A dash marks a slot at which the gold concept is not readable in that arm.

Cell	p0	p1	p2	p3	p4	p5
COCONUT (GPT-2)	0.080 / 0.090	0.077 / 0.057	0.063 / 0.073	0.063 / 0.063	0.063 / 0.070	0.063 / 0.070
COCONUT′ (GPT-2)	0.100 / 0.067	0.083 / 0.047	0.093 / 0.050	0.087 / 0.050	0.087 / 0.050	0.087 / 0.043
COCONUT (Llama-3.2-1B-Instruct)	0.010 / 0.060	0.007 / 0.037	0.010 / 0.043	0.013 / 0.060	0.017 / 0.053	0.010 / 0.060
SIM-COT-COCONUT (GPT-2)	– / –	0.007 / 0.027	0.050 / 0.083	0.053 / 0.063	0.067 / 0.060	0.067 / 0.083
CODI (GPT-2)	– / –	– / –	– / –	– / –	– / –	– / –
CODI (Llama-3.2-1B-Instruct)	– / 0.023	– / 0.010	– / 0.007	– / 0.007	– / 0.010	– / 0.010

D RQ3 SUPPLEMENT: DYNAMICS

This appendix holds the instrument settings and per-cell tables behind Section 6: the state machine (D.1), settling and transmission (D.2), consecutive-slot similarity (D.3), the answer-alignment reconciliation (D.4), mid-chain edits (D.5) and the remaining instruments (D.6).

D.1 STATE-MACHINE SETTINGS

Why this abstraction. The discrete reading exists because settling needs an event, not a threshold: the pre-registered band on the cosine misfires on a trajectory that stops moving (App. D.3), whereas the last slot at which a state label changes is defined per problem with no threshold, and it is the object the state-transition view of written reasoning already uses (Yu et al., 2025). Each design choice in Eq. 3 serves one requirement. The projection P is a shared random Gaussian map rather than a fitted one (PCA on the same states, or a probe direction) so that the state labels cannot be tuned toward the answer or the path; this is what makes the state machine independent of the RQ2 probe and licenses the reading that a state's identity does not by itself predict the answer (Finding 7). The map is shared across slots and across latent and written states, so a label means the same thing at every position, which is what makes a transition well defined; per-slot clustering would leave labels incomparable across slots. The clustering c is one K-means fitted jointly over the model's latent states and its written-step states, so a latent state change is measured in the same partition that the model's own written reasoning occupies; assigning latent states to their nearest written step would need step labels, which the unsupervised chains lack. Because K sets the granularity, every $K \in \{5, 8, 12\}$ is reported and none is privileged, and a verdict is stated only where all 5 clustering seeds agree.

One joint K-means is fitted over the pooled latent and written-step rows of a model; results are reported at every $K \in \{5, 8, 12\}$ and never averaged; 5 clustering seeds are run and a non-unanimous verdict is unresolved (146 of 153 verdicts are unanimous). Nulls replace slot t of one problem with slot t of another. The mid-chain edit keeps $\mathbf{h}_{0:i-1}$ and continues on a twin whose gold answer differs by one changed quantity; an output that keeps the original answer is *rigid*.

Instrument settings behind Section 6: the shared Gaussian map projects each layer-averaged state to 128 dimensions; cross-problem nulls use 200 draws; the effect floors are 0.5 slots for settling and 1.0 bit for sequence entropy. Of the 17 cells, 0 reach no-movement under the label-free test of App. D.6.

D.2 SETTLING AND TRANSMISSION

Figure 8 shows the last state change of every cell against its null; Table 14 gives the last-transition slot, its null and their difference for every cell on both backbones, and Table 15 the per-cell transmission verdicts behind Section 6.

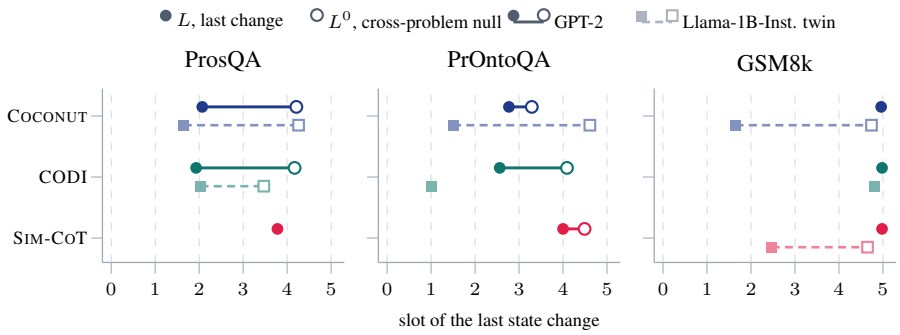

Figure 8: **Where the chain stops changing.** Slot of the last state change L (filled) against its cross-problem null L^0 (hollow) at $K{=}5$; the segment is the early-settling margin of Table 14, read against the half-slot floor. GPT-2: circles, solid; Llama-3.2-1B-Instruct twin: squares, dashed; pairs closer than 0.15 slots are drawn as one marker. On GSM8k the GPT-2 chains and their null both sit at the last slot, so that reading is unresolved; the Llama-3.2-1B-Instruct CODI/PrOntoQA twin has $L = L^0$ at the first slot and is unresolved. Dash in Table 14: value not emitted.

Table 14: **When the chain stops changing.** Slot of the last state transition at $K{=}5$ (mean over chains with at least one transition), the cross-problem null and their difference in slots; single-state chains counted apart. *ceiling*: the null itself sits at the last slot. Verdicts obey the 0.5-slot floor.

Cell	1-state	Last	Null	Δ	Verdict
ProsQA					
COCONUT	14	2.07	4.21	2.14	early
COCONUT (2nd)	113	1.51	4.46	2.94	early
COCONUT (Llama)	0	1.65	4.26	2.61	early
CODI	53	1.93	4.17	2.24	early
CODI (Llama)	0	2.03	3.47	1.44	early
SIM-COT	0	3.78	3.85	0.07	**keeps changing**
PrOntoQA					
COCONUT	6	2.77	3.29	0.52	early
COCONUT (Llama)	35	1.50	4.61	3.11	early
CODI	0	2.56	4.09	1.53	early
CODI (Llama)	0	1.00	1.00	0.00	unres.
SIM-COT	0	4.00	4.49	0.49	**keeps changing**
GSM8k					
COCONUT	0	4.96	4.96	0.00	unres., ceiling
SIM-COT	0	4.98	4.99	0.01	unres., ceiling
CODI	0	4.98	4.99	0.01	unres., ceiling
COCONUT (Llama)	254	1.65	4.74	3.09	early
CODI (Llama)	0	4.80	4.89	0.09	at null, ceiling
SIM-COT (Llama)	–	2.470	4.650	2.18	early

Table 15: **What is transmitted, per cell** (GPT-2 grid, $K{=}5$; Llama-3.2-1B-Instruct twins in Tables 14 and 17). L: slot of the last state change; $L^0{-}L$: slots before the cross-problem null (*settles* above the half-slot floor; *moves*: no earlier than the null; *ceiling*: null at the last slot). r: along-answer ratio under the probe of record. Movement: *not ans.*, below the null with no state lift; *unres.*, within the null band; *re-m.*, both conditions hold; *mixed*, one holds; *assembles*, along the answer without state lift; n/a, direction below its MDE. †: changes under the cross-problem null.

Cell	L	$L^0{-}L$	settling	r	movement
ProsQA					
COCONUT	2.07	2.14	settles	1.00	unres. / not ans.†
CODI	1.93	2.24	settles	1.08	unres.†
SIM-COT-COCONUT	3.78	0.07	**moves**	1.94	**assembles**†
PrOntoQA					
COCONUT	2.77	0.52	settles	0.67	unres.†
CODI	2.56	1.53	settles	—	n/a
SIM-COT-COCONUT	4.00	0.49	**moves**	0.69	not ans.†
GSM8k					
COCONUT	4.96	0.00	ceiling	0.81	not ans.
CODI	4.98	0.01	ceiling	0.80	unres.†
SIM-COT-COCONUT	4.98	0.01	ceiling	0.72	not ans.

D.3 CONSECUTIVE-SLOT SIMILARITY

Consecutive-slot similarity. On ProsQA the cosine is 0.981 for COCONUT and 1.000 for CODI at the first step and stays there, whereas the per-step chain climbs from 0.687 to 0.971 on GPT-2 and from 0.771 to 0.939 on Llama-3.2-1B-Instruct; PrOntoQA shows the same split. On GSM8k every GPT-2 recipe keeps moving, whereas on Llama-3.2-1B-Instruct the unsupervised and per-step chains settle from the first step (0.972, 0.930) and only CODI keeps moving (Figure 3); the sustained movement that per-step supervision installs is therefore established on the logical tasks and does not carry to the arithmetic twin. $\cos(\mathbf{h}_t, \mathbf{h}_{t-1})$ is computed per problem on the fed-back final-layer state and on the layer-mean state of Eq. 3, against a null that pairs slot t of one problem with slot $t-1$ of another at the same positions (a deterministic cyclic shift). The reading pre-registered for this instrument, "settled when the curve stays inside the null band", misfires on this statistic: a trajectory that stops moving has cosine near one and leaves the band upward, so the frozen reading calls most cells unsettled. It is reported unchanged in the run dirs together with the diagnostic "rises above the band", and Figure 3 shows the observed step sizes; no settling verdict in the body rests on the band test.

D.4 ANSWER ALIGNMENT: FITTING POINT AND NULLS

Reconciliation (T.4): fitting point and nulls. The answer direction of Section 6 was pre-registered at the first latent slot. At that slot it falls below its MDE on every cell that assembles its answer across slots—SIM-COT-COCONUT/ProsQA reads 0.030 there and 0.963 at the last slot, reproducing the position curve of Section 5 inside this instrument—so a not-applicable verdict derived from it describes the fitting point, not the model. The direction is therefore fitted at the last latent slot, fixed for all 17 cells before any exit was read (a per-cell best slot was refused; it would have turned every change into a gain). 18 of 69 cell$\times K$ verdicts moved: 9 gained a reading, 6 lost one (CODI/ProsQA on Llama-3.2-1B-Instruct, 0.070 $\rightarrow$ 0.037; CODI/PrOntoQA, 0.653 $\rightarrow$ 0.530, both below MDE at the last slot), and COCONUT/PrOntoQA weakened to 0.587, 1.38 standard errors above its MDE. Table 16 is the pre-registered first-slot table as originally read.

Two nulls are reported for the alignment condition. The pre-registered cross-problem null replaces a within-problem step with a between-problem one; because the answer direction separates problems by construction, that null is answer-aligned almost tautologically and biased toward DISTINCT (on SIM-COT-COCONUT/ProsQA its 5th percentile is 0.248, above the observed along-fraction of 0.170). The null the body reads permutes the answer labels and refits the direction with the same estimator at the same dimension; it is exact for both probe arms, lands near the earlier subspace-matched null on the probe of record, and its 95th percentile on the same cell is 0.094. Under it, DISTINCT requires the along-fraction to fall below the null's 5th percentile with no state lift; inside the band with no state lift is unresolved. 32 of 69 verdicts change between the two nulls; Table 17 gives every cell under both, with the ridge arm on ProsQA. The lifts behind the body's two exceptions: on SIM-COT-COCONUT/ProsQA the along-component is 1.94 times its null under the probe of record and 1.23 under ridge, with a state lift of -0.014 against an MDE of 0.0085; on the Llama-3.2-1B-Instruct COCONUT/PrOntoQA twin the state lift is 0.34. The 69 stored verdicts were recomputed from stored quantities with 0 mismatches. Both changes are logged in App. **??**.

D.5 MID-CHAIN EDITS: COMMITMENT, BACKBONE COMPARISON AND DRIFT

Adaptation falls monotonically with injection position in every cell, on both mechanisms and all three supervision regimes (Table 18). Under identical edits on identical problems the larger instruction-tuned backbone re-computes where GPT-2 holds or corrupts, with the latent mechanism held fixed; every gap in Table 19 clears its own minimum detectable effect.

Post-edit trajectory drift. For an edit at slot i and every later slot, the cosine distance of the edited run's layer-mean state from the clean run on the unedited problem (d_{orig}) and from the clean run on the edited problem (d_{twin}) was computed per problem on the twin sets above, with three pre-registered readings (re-planning, replay, local attenuation). The instrument is uninformative on the recycling family. The edit driver keeps the original's first i slots and takes every later slot from the twin's own clean run, so d_{twin} is 0.000 at every post-edit slot on every COCONUT and SIM-COT-COCONUT cell, the re-planning rate of 1.000 and the contradicted replay reading follow from the

Table 16: **The pre-registered first-slot fit,** as originally read against the cross-problem null: probe accuracy of the answer direction at slot p0, the along-component as a ratio to the subspace-matched null, the state lift at $K{=}5$, and the verdict. Superseded by Table 17.

Cell	Probe	Ratio	Lift	Verdict
ProsQA				
COCONUT	0.597	0.78	+0.006	distinct
COCONUT′	0.507	1.04	+0.018	mixed
COCONUT (Llama)	0.103	1.11	+0.012	mixed
SIM-COT	0.030	—	—	n/a
CODI	0.047	0.79	-0.002	distinct
CODI (Llama)	0.070	0.72	-0.006	distinct
GSM8k				
COCONUT	0.507	0.76	0.000	distinct
SIM-COT	0.503	0.78	0.000	distinct
SIM-COT (10 sl.)	0.500	0.77	0.000	distinct
CODI	0.527	0.77	0.000	distinct
COCONUT (Llama)	0.523	0.80	+0.006	distinct
CODI (Llama)	0.537	0.84	0.000	distinct
PrOntoQA				
COCONUT	0.970	1.56	+0.013	mixed; re-meas. at $K{\geq}8$
CODI	0.653	1.73	-0.020	distinct
COCONUT (Llama)	0.967	2.18	+0.341	mixed
CODI (Llama)	0.917	0.61	-0.008	distinct
SIM-COT	0.527	—	—	n/a

Table 17: **Reconciliation at the last-slot fit under both nulls,** all K. *probe*: accuracy of the answer direction at the last slot; *along*: the along-component of consecutive-slot displacement; *ratio*: along over the refit-null mean; (i) refit / cross: whether the alignment condition fires under the label-permuted refit null and under the pre-registered cross-problem null; (ii): whether state identity lifts answer prediction above the majority prior by its MDE at $K{=}5/8/12$. *dist.* distinct, *mix.* mixed, *re-m.* re-measurement, *unr.* unresolved; n/a: direction below its MDE at the last slot.

Cell	arm	probe	along	ratio	(i) refit	(i) cross	(ii) 5/8/12	verdict, refit	verdict, cross
COCONUT/ProsQA	record	0.607	0.064	1.00	no	no	no/yes/yes	unr./mix./mix.	dist./mix./mix.
	ridge	0.813	0.033	0.49	no	no	no/yes/yes	dist./mix./mix.	dist./mix./mix.
COCONUT′/ProsQA	record	0.513	0.069	1.16	yes	no	yes/yes/yes	re-m. $\times 3$	mix. $\times 3$
	ridge	0.850	0.036	0.62	no	no	yes/yes/yes	mix. $\times 3$	mix. $\times 3$
COCONUT/ProsQA, Llama-3.2-1B-Instruct	record	0.210	0.052	1.21	yes	no	yes/no/yes	re-m./mix./re-m.	mix./dist./mix.
	ridge	0.720	0.041	0.82	no	no	yes/no/yes	mix./dist./mix.	mix./dist./mix.
SIM-COT-COCONUT/ProsQA	record	0.963	0.170	1.94	yes	no	no/no/no	mix. $\times 3$	dist. $\times 3$
	ridge	0.973	0.111	1.23	yes	no	no/no/no	mix. $\times 3$	dist. $\times 3$
CODI/ProsQA	record	0.043	0.041	1.08	no	no	no/yes/no	unr./mix./unr.	dist./mix./dist.
	ridge	0.030	—	—	—	—	no/yes/no	n/a	n/a
CODI/ProsQA, Llama-3.2-1B-Instruct	record	0.037	—	—	—	—	no/no/no	n/a	n/a
	ridge	0.027	—	—	—	—	no/no/no	n/a	n/a
COCONUT/GSM8k	record	0.603	0.051	0.81	no	no	no/no/no	dist. $\times 3$	dist. $\times 3$
SIM-COT-COCONUT/GSM8k (6 sl.)	record	0.613	0.039	0.72	no	no	no/no/no	dist. $\times 3$	dist. $\times 3$
SIM-COT-COCONUT/GSM8k (10 sl.)	record	0.660	0.095	1.09	no	no	no/no/no	unr. $\times 3$	dist. $\times 3$
CODI/GSM8k	record	0.547	0.043	0.80	no	no	no/no/yes	unr./unr./mix.	dist./dist./mix.
COCONUT/GSM8k, Llama-3.2-1B-Instruct	record	0.557	0.036	0.78	no	no	no/no/yes	dist./dist./mix.	dist./dist./mix.
CODI/GSM8k, Llama-3.2-1B-Instruct	record	0.567	0.075	0.90	no	no	no/no/yes	unr./unr./mix.	dist./dist./mix.
COCONUT/PrOntoQA	record	0.587	0.032	0.67	no	no	no/yes/yes	unr./mix./mix.	dist./mix./mix.
CODI/PrOntoQA	record	0.530	—	—	—	—	no/yes/yes	n/a	n/a
COCONUT/PrOntoQA, Llama-3.2-1B-Instruct	record	0.983	0.123	2.25	yes	no	yes/yes/yes	re-m. $\times 3$	mix. $\times 3$
CODI/PrOntoQA, Llama-3.2-1B-Instruct	record	0.913	0.020	0.33	no	no	no/no/no	unr. $\times 3$	dist. $\times 3$
SIM-COT-COCONUT/PrOntoQA	record	1.000	0.046	0.69	no	yes	no/no/no	dist. $\times 3$	mix. $\times 3$

Table 18: **Commitment on the GPT-2 grid:** adaptive and rigid rates by injection position i on ceiling-solvable problems. COCONUT and CODI rows use the enlarged 480-twin set, SIM-COT-COCONUT rows the original 300-twin set; $i{=}0$ rates are 1.000/0.000 by construction. Terminal labels are descriptive only.

Cell	usable n	adaptive $i{=}0/2/4$	rigid $i{=}0/2/4$	Terminal
COCONUT/GSM8k	111	1.000 / 0.405 / 0.225	0.000 / 0.459 / 0.532	rigid-dominant
SIM-COT-COCONUT/GSM8k	57	1.000 / 0.421 / 0.211	0.000 / 0.386 / 0.526	rigid-dominant
SIM-COT-COCONUT/ProsQA	64	1.000 / 0.750 / 0.250	0.000 / 0.250 / 0.750	rigid-dominant
CODI/GSM8k	153	1.000 / 0.582 / 0.327	0.000 / 0.072 / 0.268	corruption-dominant

Table 19: **The paired backbone comparison** on the intersection of the two models' ceiling subsets after the enlargement to 480 twins. *discordant*: problems adaptive on exactly one backbone (GPT-2-only / Llama-3.2-1B-Instruct-only); p is Holm-corrected McNemar.

Paired contrast	n	GPT-2	Llama-3.2-1B-Instruct	discordant	Holm p	gap vs MDE
COCONUT, i=2	88	0.386	0.955	1/51	0.0000	$0.568 > 0.234$
COCONUT, i=4	88	0.239	0.818	4/55	0.0000	$0.580 > 0.251$
CODI, i=2	140	0.600	0.807	5/34	0.0000	$0.207 > 0.133$
CODI, i=4	140	0.336	0.479	9/29	0.0017	$0.143 > 0.125$

construction rather than from the model, and d_{orig} is the distance between two clean trajectories. This is the constructional identity that blocked the edit-and-decode readout (App. **??**); it was recognised when the last cell entered reported the same rate on every row. The body therefore reads only the answer-side rate $A(i)$ from the edit battery, which is well defined on the spliced sequence. Drift on the CODI cells, whose driver recomputes the post-edit slots, is retained in the run dirs and read into no verdict. Nothing is pooled across backbones.

Induced revisiting after a mid-thought edit (T.5). A faithful hybrid that recomputes post-edit slots given a pinned prefix was validated against the independently stored ceiling anchors of Table 18 to three decimals (COCONUT 0.266/0.734; second training 0.234/0.766; CODI 0.305/0.033), with manipulation reach 1.000. On both COCONUT/ProsQA cells the answer is invariant to pinning zero, two or four slots of a conflicting twin's thoughts (0.266 throughout; 0.234 on the second training)—a fourth intervention type corroborating the inertness of Section 4—while CODI/GSM8k moves with pinning ($0.305 \to 0.206 \to 0.128$). The return statistic (a post-edit slot re-entering a pre-edit state, edited arm against the unedited twin arm, McNemar) reads NO RETURN on both COCONUT cells at i=4 (edited-arm returns 0.75–0.87 against 0.96–0.99) and UNRESOLVED at i=2 on the second training and on CODI (discordant counts below six). Scope: three cells with edit twins, within the trained slot budget, without an attention view. On inert cells it describes trajectory geometry, not reconsideration of an answer-relevant computation.

D.6 REVISITING, SCRIPTS AND FURTHER INSTRUMENTS

Table 20: **Revisiting at K=5.** Revisit rate, problems with at least one revisit (of 300), the cell's own shuffled-sequence baseline, the state verdict by MDE, the attention re-entry family, and whether the two families agree.

Cell	Rate	≥1	Base	State	Attn, agree
ProsQA					
COCONUT	0.000	0	0.234	no revisit	none, yes
COCONUT (2nd)	0.001	1	0.121	no revisit	none, yes
COCONUT (Llama)	0.000	0	0.271	no revisit	none, yes
SIM-COT	0.005	8	0.214	no revisit	detected, no
CODI	0.003	4	0.166	no revisit	detected, no
CODI (Llama)	0.001	2	0.203	no revisit	none, yes
PrOntoQA					
COCONUT	0.121	90	0.279	no revisit	detected, no
CODI	0.000	0	0.217	no revisit	detected, no
COCONUT (Llama)	0.000	0	0.144	no revisit	none, yes
CODI (Llama)	0.000	0	0.132	no revisit	none, yes
SIM-COT	0.000	0	0.267	no revisit	none, yes
GSM8k					
COCONUT	0.251	298	0.155	above baseline	none, no
SIM-COT	0.263	298	0.159	above baseline	none, no
CODI	0.678	299	0.377	above baseline	none (neg.), no
COCONUT (Llama)	0.000	0	0.037	no revisit	none (neg.), no
CODI (Llama)	0.335	290	0.267	above baseline	none (neg.), no

Revisiting. The two families disagree on 10 of 17 cells (the sixteen of Table 20 and the released ten-slot cell), where no single-family verdict is quoted. "No revisit" means below the baseline, not absent: COCONUT/PrOntoQA earns it with 90 revisiting problems. Consulted chains fall on both sides (4 do not revisit, 5 do), so revisiting is not a signature of use. Two further properties are read against the written chain of the same model (Table 20). Revisiting an earlier state occurs above baseline in every GPT-2 GSM8k chain and in no logical-task chain, latent or written, and consulted

chains fall on both sides, so it is not a signature of use; state sequences are less diverse than the null, but so are the written chains.

Scripts (T.3). Sequence entropy against the cross-problem null with the floor (deficit ≥ 1.0 bit and a repetition signature): over 51 cell$\times K$ combinations, 30 SCRIPT-LIKE, the rest LESS DIVERSE THAN NULL, positional, problem-specific or degenerate. The matched written-CoT control (per-side clustering at equal K; the latent sequence truncated to each problem's own written length) passes on COCONUT/ProsQA-Llama-3.2-1B-Instruct at every K (1 of 5 informative cells), inverts on 4 cells (COCONUT/ProsQA at $K{=}5$ on both trainings; CODI/GSM8k on both backbones at every K), and is not applicable where the matched length is below two steps. Every SCRIPT-LIKE verdict except the one that passes the control is therefore UNRESOLVED as a latent-specific claim.

Label-free movement. Mean consecutive-slot displacement against the cross-problem null, independent of cluster labels. COCONUT/GSM8k-Llama-3.2-1B-Instruct (254 single-state chains at $K{=}5$): step 17.4 against a null of 125.6, ratio 0.14, cosine 0.986 against 0.271. COCONUT/ProsQA, second training (113): ratio 0.15, cosine 0.998 against 0.960. CODI/ProsQA (53): ratio 0.35 with cosine 1.000—magnitude only, direction fixed. All other cells: at most 35 single-state chains and ratios 0.26–0.95. No cell reaches no-movement.

Markov order (T.6) and state overlap (T.7). A second-order model gains nothing over a first-order one on the logical tasks (residual below the shuffle null), while GPT-2 GSM8k chains are non-local at $K{=}5$ (residual 0.18–0.27 against null bands ≤ 0.14) and CODI/GSM8k at every K—another task split, K-dependent on two cells. Joint clusters are disjoint between the latent and written sides on 15 of 16 informative cell$\times K$ pairs, with the latent side using more effective states than the written one, so the latent chain is not a compression of the written one at the state level; overlap is partly by construction in a joint fit, and the effective-state comparison is the check.

E ADDITIONAL CHECKPOINTS

Two checkpoint families are not public releases: the per-step recipe we trained on Llama-3.2-1B-Instruct with its training-time control (E.1), and, outside the grid's slot budget, the publicly released ten-slot SIM-COT-COCONUT checkpoint (E.2).

E.1 PER-STEP SUPERVISION ON LLAMA-3.2-1B-INSTRUCT

The per-step recipe was trained by us on Llama-3.2-1B-Instruct for ProsQA (the per-step twin of the GPT-2 SIM-COT-COCONUT/ProsQA cell), together with a training-time control that continues the same base for the same schedule with the auxiliary loss weighted to zero; the two share harness, step count, slot budget and evaluation items and differ only in the per-step decoder loss (39 run directories under one commit, $N{=}300$, seed 0). The PrOntoQA training saved 13 checkpoints; a sweep over all of them on 60 held-out problems, run before any intervention, found 8 emitting a single answer string, and selected checkpoint 9, the earliest with both answer classes present and sweep accuracy 1.000; the final checkpoint, which answers "True" on every problem ($C_0{=}0.497$, every intervention reading the same value by construction), is cited only as the degenerate reference. The GSM8k twin was trained last and has no training-time control; its deletion and substitution battery is in Tables 5, 6 and 7, and its profile is summarised below. Table 21 collects the ProsQA and PrOntoQA cells.

Three caveats bind these readings. On ProsQA both probe arms were run and two verdicts move between them, the emptiness of slot 0 and the depth reading, so those are stated with their arm; the ordered step-by-slot layout and its absence on the control are arm-invariant. No substitute has moved an answer in any Llama-3.2-1B-Instruct/ProsQA cell: the direction-by-scale $2{\times}2$, the donor, and a donor with a different gold answer at twice the natural per-position norm each move 0 of 300 on the per-step twin, the control and the CODI twin alike, while the same code moves 16, 24 and 23 answers on the GPT-2 twin under noise, mean and donor. The reading therefore bounds the whole range the harness can express rather than demonstrating the readout's sensitivity on this backbone. The PrOntoQA probes are not applicable: with a two-way answer the two binary probes are trained on complementary labels and the recovered sensitivity ceiling is zero for any model, and the answer

direction fitted at the last slot is at chance. On PrOntoQA no trajectory leaves its first state at $K{=}5$ or 8, so revisiting and settling are undefined there rather than absent.

The paired backbone comparison for the per-step family is descriptive only: the stored twin sets hold 53, 64 and 76 usable problems for the Llama-3.2-1B-Instruct twin, the GPT-2 twin and the control, no per-problem ceiling subset is stored, and the sets are far below the 480 used for the public pairs, so the intersection, discordant pairs and Holm p of Table 19 cannot be formed. On those sets the Llama-3.2-1B-Instruct per-step twin and its control follow every edit (1.000 at $i{=}4$) while the GPT-2 twin falls to 0.750 at $i{=}2$ and 0.250 at $i{=}4$. Since the twin and its control have identical recovery tables, the contrast is between backbones, not supervision.

The GSM8k twin's profile. Its answer magnitude is most readable at the last slot (2.233× chance) and rises from 0.363 at slot 0 to 0.487 at slot 5, a difference at the readout's own resolution (MDD 0.112; under stratified folds it falls just short), so the direction is reported and not the significance; the candidate readout is not applicable on GSM8k. Its state machine settles early (last change at slot 2.470 against a null of 4.650), never revisits a state (0.000 against a baseline of 0.292) and its consecutive-slot cosine sits above the null band from the first step (0.930 against 0.369), the same profile as the Llama-3.2-1B-Instruct COCONUT/GSM8k twin and unlike the GPT-2 per-step cells. Under a mid-chain edit it follows the edited problem (0.932 at $i{=}2$, 0.898 over the late positions, against a ceiling anchor of 1.000); the paired backbone comparison remains descriptive for the per-step family, and its post-edit drift statistic is degenerate for the same constructional reason as App. D.5 records, so no re-planning rate is quoted.

Table 21: **The Llama-3.2-1B-Instruct per-step cells** (ProsQA and PrOntoQA, $N{=}300$). Deletion: C_0, C_1, helped/hurt. Truncation: accuracy with one, two and four slots kept. Substitution: answers moved under each substitute. Content on ProsQA: gold readability at slots 0 and 5, probe of record / ridge (chance 0.026; ceilings 0.627 / 0.960), path cells live of 24. Dynamics: distinct states at $K{=}5$, slots settled before the null (floor 0.5), along-answer component at the last slot against the refit and cross-problem nulls.

	per-step, ProsQA	**control, ProsQA**	**per-step, PrOntoQA**
C_0 / C_1	0.997 / 0.077	0.877 / 0.850	0.993 / 0.990
helped / hurt	276 / 0	16 / 8	2 / 1
test	Holm $p{=}0.0006$	$\Delta{=}-0.027$, MDE 0.031	$\Delta{=}-0.003$, MDE 0.008
node 1	consulted	unresolved	inert (3 of 300 changed, within the rule-of-three bound)
kept 1 / 2 / 4 slots	0.987 / 0.997 / 0.997	0.860 / 0.863 / 0.867	— / 0.990 / 0.993
moved under substitution	0 of 300	0 of 300	—
node 2	placeholder (caveat)	—	—
answer at slot 0 → 5	0.070 → 0.923 / 0.457 → 0.777	0.250 → 0.217 / 0.587 → 0.497	n/a
both candidates (bar 30%)	0.083 / 0.103	0.043 / 0.077	n/a
path cells live / 24	23 / 24	12 / 17	n/a
layout	localised and ordered	neighbourhood, not chain	n/a
distinct states ($K{=}5$)	3.88	2.00	1.000
settled before null	0.106 (at null)	1.48 (early)	undefined
along-answer / refit / cross	0.165 / 0.099 / 0.191	0.097 / 0.078 / 0.181	at chance

E.2 THE RELEASED TEN-SLOT SIM-COT-COCONUT CHECKPOINT

The publicly released SIM-COT-COCONUT checkpoint (GSM8k, 10 slots) differs in slot budget from every other cell and is therefore kept out of the body; it serves two purposes here. It is the parity anchor for our six-slot SIM-COT-COCONUT training, and it is an independent-release replication of the step-supervised regime on GSM8k: deleted, its chain flips 122 of 300 answers (C_0 0.450, C_1 0.047); truncated, it rises monotonically (0.047, 0.177, 0.297, 0.353, 0.450 at $k = 0, 2, 4, 6, 10$); corrupted, a donor's thoughts cost -0.427 against noise -0.390, bounded on the frozen grid and

donor-worse at the pre-registered N=900 (52/18, p=0.00006; gap 0.038 against MDE 0.026). Two results exist only on this cell. Its commitment window spans the whole chain: adaptation 1.000, 0.603, 0.384, 0.178, 0.123 at $i = 0, 2, 4, 6, 8$, with the rigid rate rising to 0.726. And it is the one GSM8k cell with power to resolve the conditional-difficulty contrast: with an MDE of 0.087 it observed +0.119 ($p = 0.069$), 1.000 of 25 solvable problems of four or more steps needing the chain against 0.881 of 109 shallower ones—a powered null on decline, the apparent increase not itself established.

F THE CENSUS AND WORKED CLASSIFICATION OF THREE CELLS

F.1 CLASSIFICATION RULES

The following rules define the classification summarized in Section 7.1. They use the intervention results from RQ1 and the profiles from RQ2 and RQ3.

Node 1: delete the chain (RQ1). Remove every latent slot and re-read the answer. If at most 3 answers change, the rule-of-three bound on an absent effect at N=300, the model does not depend on the chain and the cell is **inert**. If the cell loses correct answers beyond its MDE, it is **consulted**. The test is the paired count of problems helped and hurt (Section 4); a cell whose answers move in both directions without a net direction is unresolved at this node.

Node 2: replace the contents (RQ1). For a consulted cell, retain every slot but replace its contents with vectors that carry no information about the problem: noise, the dataset mean, or another problem's recorded slots. If no substitute moves an answer, the model needs the slot positions but not their contents, and the cell is a **placeholder**: the filler-token regime of Pfau et al. (2024) and Goyal et al. (2024) realised by a trained chain. If substitutes move answers beyond the cell's MDE, the model consults the contents and the cell is **content used**. Whether another problem's slots then hurt more than deleting the chain is a reading within that outcome, not part of the test. A cell whose substitute effects lie within its MDE is unresolved here.

Profile: readability and state dynamics (RQ2, RQ3). The two nodes determine use, and the profile never overrides their verdict: a probe shows what is decodable, not what the computation depends on (Section 2). Within each use outcome, two readings describe the chain: *readability*, whether a linear probe can decode the answer or step-by-step path from the slots; and *state dynamics*, whether the chain's state continues to change across slots. The readings agree in every cell where both resolve, so Figure 5 labels each branch with the reading that names it; the definitions below state both. This agreement is an observation from a small grid, not a rule of the tree. An ordered layout entails a changing state because the slots hold different content. The readings could therefore diverge only when a state changes but nothing is readable; such a cell would be reported with both readings and would name no leaf, the same exit the nodes give an unresolved cell.

The six leaves. Use and profile jointly define six mutually exclusive chain types:

- **Carrier** (inert, readable): the answer is readable from the first slot and the discrete state settles early. The answer is already readable before the chain begins in the profiled cases.
- **Empty** (inert, unreadable): little is readable from any slot, the state stops changing early, and deleting the chain changes nothing.
- **Bare placeholder** (placeholder, state fixed): the slot positions are needed, little is readable, and the state stops changing early.
- **Laid-out placeholder** (placeholder, state changing): step k of the path is readable at slot k, and the state changes through the last slot, yet no substitute moves the answer.
- **Assembling** (content used, structured): step k is readable at slot k, the state changes through the last slot, and the contents affect the answer under substitution. This profile does not by itself assign a causal operation to each decoded step.
- **Unstructured** (content used, unstructured): the contents are consulted, but the probe reads no step-by-slot layout.

Table 22: **The twenty-one cells on the tree.** Node 1: the deletion verdict; node 2: the substitution verdict (consulted cells only); profile: the RQ2/RQ3 reading; the leaf follows from the tree. "—": not reached; "n/a": not applicable. The second COCONUT/ProsQA training and the two training-time controls are listed as cells of their own.

Backbone	Cell	Node 1	Node 2	Profile	Leaf
GPT-2	COCONUT/ProsQA	inert	—	readable, state fixed	carrier
GPT-2	COCONUT/ProsQA (2nd)	inert	—	readable, state fixed	carrier
GPT-2	COCONUT/PrOntoQA	inert	—	readable, state fixed	carrier
GPT-2	control/ProsQA	inert	—	state fixed, readability n/a	inert, unprofiled
GPT-2	CODI/PrOntoQA	inert	—	unreadable, state fixed	empty
GPT-2	CODI/ProsQA	unresolved	—	—	—
GPT-2	COCONUT/GSM8k	consulted	content	unstructured	unstructured
GPT-2	CODI/GSM8k	consulted	content	unstructured	unstructured
GPT-2	SIM-COT-COCONUT/GSM8k	consulted	content	unstructured	unstructured
GPT-2	SIM-COT-COCONUT/ProsQA	consulted	content	structured, state changing	assembling
GPT-2	SIM-COT-COCONUT/PrOntoQA	consulted	content	structured, state changing	assembling
Llama-1B-Inst.	COCONUT/ProsQA	inert	—	readable, state fixed	carrier
Llama-1B-Inst.	COCONUT/PrOntoQA	inert	—	readable, state fixed	carrier
Llama-1B-Inst.	CODI/PrOntoQA	inert	—	unreadable, state fixed	empty
Llama-1B-Inst.	CODI/ProsQA	consulted	placeholder	unreadable, state fixed	bare placeholder
Llama-1B-Inst.	SIM-COT-COCONUT/ProsQA	consulted	placeholder	structured, state changing	laid-out placeholder
Llama-1B-Inst.	CODI/GSM8k	consulted	content	unstructured	unstructured
Llama-1B-Inst.	COCONUT/GSM8k	consulted	content	unstructured	unstructured
Llama-1B-Inst.	SIM-COT-COCONUT/GSM8k	consulted	content	unstructured	unstructured
Llama-1B-Inst.	control/ProsQA	unresolved	—	—	—
Llama-1B-Inst.	SIM-COT-COCONUT/PrOntoQA	inert	—	state fixed, readability n/a	inert, unprofiled

F.2 CELL ASSIGNMENTS AND WORKED EXAMPLES

Table 23 applies the procedure of Section 7.1 to one cell per leaf, listing the statistic each node reads and the verdict it returns. Every value is the same macro-backed number that appears in the per-cell tables of App. B–D; the table only collects them in the order the tree consults them. The unit of the procedure is the cell, and each test is an intervention on the slot vectors of all 300 evaluation problems, so no per-problem annotation is involved; a reader who wishes to reproduce an assignment needs the deletion counts (Table 1), the corruption contrasts (Table 6), and, for the descriptive profile, the slot-0 answer probe (the answer row of Figure 2), the settling difference (Table 14) and the layout diagonal (Figure 2).

Table 23: **The classification procedure applied to one cell per leaf.** Node 1: intact and deleted accuracy, helped/hurt counts, verdict. Node 2 (consulted cells only): change in accuracy under noise, dataset mean and donor, the donor-versus-deletion contrast, verdict. Profile: slot-0 answer probe (chance 0.026), slots settled before the null at $K=5$ (floor 0.5), and whether step k is readable from slot k.

	COCONUT/ProsQA	CODI/ProsQA, Llama-1B-Inst.	SIM-COT-COCONUT/ProsQA
Node 1: delete the chain			
C_0 / C_1	0.973 / 0.973	0.983 / 0.940	0.993 / 0.800
helped / hurt	0 / 0	13 / 0	58 / 0
verdict	inert (rate bounded at 1%)	consulted	consulted
Node 2: replace the contents			
noise / mean / donor	—	0.000 / 0.000 / 0.000	-0.047 / -0.073 / -0.077
donor vs. deletion	—	no substitute moves an answer	donor above deletion by 0.117 (MDE 0.074)
verdict	—	placeholder	consulted for content
Profile (RQ2, RQ3)			
answer probe at slot 0	0.597	not readable (record)	0.030
settled before null	2.14 slots	1.44 slots	0.07 slots
step k in slot k	no (path whole in every slot)	no	yes
leaf	inert, carrier	placeholder	consulted, assembling