AHEAD OF THE EVIDENCE: CERTIFYING PREMATURE COMMITMENT IN CHAINS OF THOUGHT

Anonymous authors
Paper under double-blind review

ABSTRACT

A chain of thought is increasingly read as evidence of how a model reached its answer. The worry that the model already knew before it finished reasoning is easy to state and hard to test, because "already" has no reference point: early relative to what? We supply one. On tasks whose correct procedure can be written as a short program, the program's state after each step fixes how far the answer is pinned down, a prescribed trajectory that no rule reading only that state can exceed. A chain that is really carrying out the procedure is one such reader, so it cannot have settled sooner. Earliness thereby becomes a testable claim rather than a suspicion. We locate a model's commitment causally, by transplanting the hidden state from one step of a different problem into the chain and asking whether the answer follows the transplant. Causal Lookahead Probing (CLP) fits no parameters, so a positive reading cannot be the artifact of a learned mapping that has proved fatal to probe-based causal analyses. On transformers compiled from the program itself, where the computation is known in advance, the method recovers that trajectory to within 0.006 at every step. Across four such tasks and nine checkpoints up to 8B, reasoning-distilled models commit up to $2.21\times$ earlier than that state could license on three of them; on the task that licenses commitment latest, the gap to the undistilled parent persists at equal accuracy. Wherever the bound has room to bind, their undistilled parents stay within it, as does every base, instruction-tuned and state-space control. Their chains are, in these cases, transcripts of a decision already made.

1 INTRODUCTION

A chain of thought (CoT) is not a stylistic flourish; it changes what a transformer can compute. A constant-depth model is confined to TC^0, the constant-depth threshold circuits (Merrill & Sabharwal, 2024); the same architecture, emitting intermediate tokens, reaches P, each token supplying a serial state unit the fixed-depth network lacks (Feng et al., 2023; Nye et al., 2021). Reasoning models now exploit this at scale (Guo et al., 2025), and humans and downstream systems read the resulting traces as evidence of how a decision was reached (Korbak et al., 2025). If the serial computation lives in the chain, that is where we should be able to watch it happen.

But the chain can be hollow. Transformers solve algorithmic tasks as well with reasoning tokens replaced by meaningless filler (Pfau et al., 2024), and delayed prediction helps even when the delay is contentless (Goyal et al., 2024): the extra tokens buy computation, not necessarily disclosure. Nor need contentful tokens be operative. Models adopt answers from features they never verbalize, then rationalize them (Turpin et al., 2023), on naturally worded prompts without injected bias (Arcuschin et al., 2025); reinforcement learning raises verbalization only to a plateau (Chen et al., 2025).

So the trace may not be the computation, and we would like to test that. The first approach tests it from inputs and outputs alone, following the counterfactual criterion of Jacovi & Goldberg (2020): truncate the chain (Lanham et al., 2023), corrupt a step (Ye et al., 2026), or unlearn the parameters a step relies on (Tutek et al., 2025), and see whether the answer moves. But Bentham et al. (2024) showed that a leading metric of this kind largely re-encodes accuracy and answer-letter bias. These tests detect that something is wrong; they cannot locate what.

Looking inside seems to locate it. Decoding intermediate layers with the logit and tuned lenses shows the eventual answer emerging well before the final one (nostalgebraist, 2020; Belrose et al., 2023). A

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.

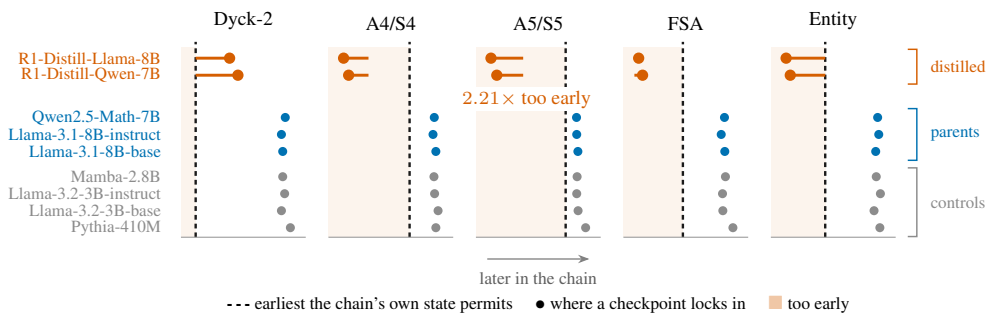


Figure 1: Where each checkpoint locks in its answer, against the earliest step at which the chain's own state permits (Section 5.4). On the three tasks in the middle the two reasoning-distilled checkpoints sit far inside the unlicensed region while every other checkpoint sits past the line; on DYCK-2 the answer is fixed so early that no model can outrun it, which is what a calibration case should look like.

single hidden state also anticipates tokens several positions ahead (Pal et al., 2023). On reasoning models a probe reads the final answer off the last state before the chain even begins (Cox et al., 2026; Boppana et al., 2026). But as Vig et al. (2020) observed of representational methods generally, they measure whether information is present, not whether the model uses it.

Interventions answer that by changing the state rather than reading it. Causal mediation analysis, activation patching, and circuit discovery locate components that carry a behavior (Vig et al., 2020; Stolfo et al., 2023; Wang et al., 2023), and causal abstraction asks whether a network implements a hypothesized algorithm (Geiger et al., 2025). Distributed alignment search fits that hypothesis by gradient descent, learning a rotation $R \in O(d)$ under which subspace interventions match interventions on the hypothesized variable (Geiger et al., 2024). But a rotation flexible enough to find an algorithm anywhere finds one in randomly initialized weights too, so a positive result says nothing about the trained model (Sutter et al., 2025). Restricting the map to be linear is one response, and linear representations do have formal and empirical support (Park et al., 2024; Burns et al., 2023); but the restriction is a modeling choice that decides the verdict without being tested. An intervention that fits nothing escapes the choice entirely: causal tracing patches activations directly (Meng et al., 2022), and the choice of what to patch against is itself a free parameter (Heimersheim & Nanda, 2024).

None of this says how early is too early. Committing early is only suspect if the task has not yet given the answer away: when three symbols already determine it, settling at the fourth is correct. A reference is missing: for each step, how much the task has revealed so far. The gap has stayed open because closing it needs three things at once: a task whose correct procedure can be written down exactly, an intervention that fits no parameters, and a way to put the two on one axis. Each exists separately; they have not been combined. We supply the reference from the task itself. For a task whose procedure can be written as a short program, the state that program has built after k steps fixes how well any decision rule could identify the answer. A chain that is carrying out the procedure is one such rule, so it cannot do better (Figure 1). We call the resulting curve the **prescribed trajectory** $\mathrm{CD}_P(k)$ (the subscript reads *prescribed*), and measure a model against it with **commitment depth** $\mathrm{CD}_M(k)$ (*measured*): transplant the step-k residual stream from a different problem, an interchange intervention in the sense of Geiger et al. (2021), and record how often the final answer follows. We estimate CD_M with **Causal Lookahead Probing** (CLP), which fits no parameters and so lies outside the construction that makes alignment search vacuous. Our contributions are as follows.

- **Normative reference.** We prove that any executor reading only the chain's own state satisfies $\mathrm{CD}_M(k) \leq \mathrm{CD}_P(k)$, a bound that is task-specific but algorithm-independent. Premature commitment becomes a certifiable claim rather than a suspicion (Section 3.3; proof in Section C).
- **Measurement that fits nothing.** CLP estimates $\mathrm{CD}_M(k)$ with no learned alignment, so a positive reading cannot be a fitting artifact. To our knowledge no prior causal measure of commitment timing has this property (Section 4; Section C.6).
- **Finding.** On the three tasks where the envelope leaves room, both reasoning-distilled checkpoints cross the bound while their undistilled parents and every other control stay within it, at matched

final-answer accuracy. On transformers compiled from the program, the estimator returns the trajectory exactly, so the separation is a property of the models, not of the measurement (Section 6).

2 Preliminaries and Problem Setup

2.1 Chain-of-Thought Computation

Let $\mathcal{M}$ be an autoregressive language model with L transformer layers and model dimension d. Write $\mathbf{h}_k^{(L)} \in \mathbb{R}^d$ for the last-layer *residual stream*, the per-position state each layer reads and updates, at the position that emits the k-th generated token; it is the only activation this paper intervenes on.

A *chain of thought* on a problem instance is the sequence of tokens $x_1, \ldots, x_T$ that $\mathcal{M}$ generates after the prompt and before the answer is read. The *step index* $k \in \{1, \ldots, T\}$ counts generated tokens, and $x_{1:k}$ is the prefix through step k. The model's *answer* is $a = \kappa(\mathcal{M}(x_{1:T})) \in \mathcal{A} \cup \{\perp\}$, where the fixed readout κ greedily decodes the answer line and applies a task-specific deterministic parser that maps the decoded text to an element of the finite set $\mathcal{A}$ or to the invalid symbol $\perp$; invalid, non-answer, truncated, and malformed completions count as non-hits under the estimator. The set $\mathcal{A}$ is small and task-specific: group elements, binary labels, automaton states, or entity locations. To probe commitment we adopt *interchange interventions*: $\mathcal{M}$ runs on a *base* instance $\mathbf{b}$ and a *source* instance $\mathbf{s}$, drawn independently from the task distribution, and at step k we *transplant* the source run's residual stream into the base run. We write $a(\mathbf{s})$ for the correct answer to $\mathbf{s}$, as against the model's own answer a. Whether the answer follows the source is what we measure.

2.2 Prescribing an Algorithm: C-RASP$^{\text{CoT}}$

Judging premature commitment requires a reference algorithm that makes explicit how much of the answer it has determined after each symbol it reads. We take that algorithm from C-RASP (Yang & Chiang, 2024), the counting variant of RASP (Weiss et al., 2021) that compiles into future-masked transformers at any input length and carries a depth hierarchy (Yang et al., 2025): a program operates on an input string $w \in \Sigma^+$ over the task alphabet, and its *prefix counting* primitive tracks quantities such as bracket depth or a running group product. What the program has determined after each symbol is therefore explicit. Tracr (Lindner et al., 2023) compiles RASP programs into executors whose computation is fixed in advance. We treat the resulting check as a floor a valid estimator must pass, not a full validation (primitives in Section B.1).

A standard C-RASP program reads its entire input at once, but a CoT emits one symbol at a time and conditions on what it has already written. We therefore use C-RASP$^{\text{CoT}}$, the autoregressive reading of a C-RASP program in which the program consumes the prefix $x_{1:k}$ available at step k and may read its own previous outputs as input to later steps. The superscript marks that reading. Semantics are in Section B, with an equivalence to standard C-RASP that preserves depth; here the program is a map π from prefixes to a distribution over $\mathcal{A}$.

Write R_k for the state π has built after k steps; what matters is the distribution it leaves over the answer, $\Pr[a(\mathbf{s}) \mid R_k(\mathbf{s})]$: a program that resolves the answer early makes this distribution concentrate early, one that must read the whole input leaves it near uniform until late.

2.3 Faithfulness as a Causal Property

A CoT is *faithful* when the steps it emits are the steps the model uses to reach its answer (Jacovi & Goldberg, 2020): a claim about computation, not about the plausibility of the text. Behavioral and representational tests each fall short (Section 1), so we read it causally: a CoT has *committed* at step k if replacing the state there with one from a different problem redirects the output, and has not if the answer survives. The property is indexed by step: a chain has a commitment profile, not a verdict.

3 Commitment Depth and the Prescribed Trajectory

This section makes the causal reading of Section 2.3 quantitative: one trajectory records when a model commits, the other states when committing is warranted.

3.1 COMMITMENT DEPTH

Fix a model $\mathcal{M}$ and draw a *base* instance $\mathbf{b}$ and a *source* instance $\mathbf{s}$ independently from the task distribution. Independence matters: conditioning on the two answers differing would change the quantity being estimated, so we draw freely and let same-answer pairs contribute nothing.

Definition 1 (Commitment depth)**.** *Run $\mathcal{M}$ on $\mathbf{b}$. At CoT step k, replace the last-layer residual stream at the step-k position with the value it takes when $\mathcal{M}$ is run on $\mathbf{s}$. Let $\mathcal{M}$ complete the chain autoregressively and read the answer through κ. Write $T(k)$ for the probability that this answer equals $a(\mathbf{s})$. For a task component whose answer set has m equiprobable labels, the* commitment depth *is that probability rescaled so that guessing scores zero and perfect redirection scores one:*

$$\mathrm{CD}_M(k) \;=\; \frac{T(k) - 1/m}{1 - 1/m}. \tag{1}$$

The rescaling makes the quantity comparable across tasks: without it a model that had learned nothing would score $1/2$ on a binary task and $1/120$ on S_5. Centering puts chance at zero, so CD_M takes values in $[-1/(m-1),\, 1]$ and small negative estimates are the expected sampling noise around zero rather than an anomaly. Tasks mixing component sizes are normalized within component, then combined with fixed weights. The replacement is applied at the selected step and the model then continues autoregressively (Section D.3). A high CD_M means the step-k state already carries the answer; near zero it does not, and the transplant washes out. Throughout, "commit" means exactly this, that the state causally determines the final answer under our intervention; whether that determination could have been revised by machinery the transplant also replaces is not distinguished.

3.2 THE PRESCRIBED TRAJECTORY

Committing early is only suspect relative to what the task has revealed, so we set against CD_M a normative trajectory derived from the reference program π. The question it answers is concrete: *after π has executed k steps on the source instance, how well could any decision rule identify the source answer?* A faithful chain cannot beat that, because a faithful chain is such a rule.

Let $R_k(\mathbf{s})$ be the state of π after k steps on $\mathbf{s}$, and let $\beta(k) = \mathbb{E}\max_a \Pr[a(\mathbf{s}) = a \mid R_k(\mathbf{s})]$ be the accuracy of the Bayes-optimal guess from that state.

Definition 2 (Prescribed trajectory)**.** *Put the Bayes accuracy on the same scale as Equation* (1)*:*

$$\mathrm{CD}_P(k) \;=\; \frac{\beta(k) - 1/m}{1 - 1/m}, \tag{2}$$

combined across components with the same fixed weights.

Both endpoints follow immediately: at $k = 0$ the program has read nothing, the best guess is the prior $1/m$, and $\mathrm{CD}_P(0) = 0$; once the state determines the answer, $\beta = 1$ and $\mathrm{CD}_P = 1$. For the four tasks with an exact program, $\beta(k)$ is computed by enumerating the reachable program states and their answer posteriors, so the trajectory is exact rather than estimated (Section C).

The two indices count different things: the program's counts program steps, one symbol read each, while the model's counts generated tokens. We compare them by normalizing both to $[0, 1]$. The identification is exact for the program and approximate for a model whose token budget per step varies; Section I.6 re-runs the analysis on the program's native operation axis.

3.3 THE FAITHFULNESS BOUND

Theorem 1 (Faithfulness bound)**.** *Suppose the patched model's answer depends on the source only through $R_k(\mathbf{s})$, the state the program has licensed by step k: that is, it is a function of the base context, that state, and sampling randomness independent of the source (Definition 5). Then $\mathrm{CD}_M(k) \le \mathrm{CD}_P(k)$ for every k.*

The bound is the optimum over decision rules, so the argument is immediate. Under the hypothesis the patched answer is some function f of $\big(\mathbf{b}, R_k(\mathbf{s})\big)$; its probability of equalling $a(\mathbf{s})$ is therefore at most that of the best such function, which is $\beta(k)$ by definition. Rescaling both sides by Equation (1)

preserves the inequality. Section C states the hypothesis as an intervention-compatible causal abstraction and gives the componentwise aggregation.

The bound assumes nothing about answer-set size, uses no link function and appeals to no complexity class: it states what optimal decisions from a given state can achieve, and is tight by construction. It also bounds every executor reading only the chain's own state rather than fingerprinting one algorithm, which matters because trained models realize idiosyncratic solutions (Zhong et al., 2023). A violation therefore is certified rather than suspected: not that the chain looks unfaithful, but that no executor reading only that state could have committed that early from the evidence read so far.

3.4 Quantifying Premature Commitment

Definition 3 (Unfaithfulness functional and commitment step)**.** *The* unfaithfulness functional *is the largest signed gap by which a model exceeds the bound,*

$$U(\mathcal{M}, \pi) \;=\; \max_{k} \big[\mathrm{CD}_M(k) - \mathrm{CD}_P(k)\big]_+, \tag{3}$$

and the commitment step *is the first step at which the model is more likely than not to have committed,* $k^* = \min\{k : \mathrm{CD}_M(k) \geq \frac{1}{2}\}$.

A violation is summarized by how far, and how early, the model outruns its evidence. We report $U(\mathcal{M}, \pi)$ as the headline measure of how far a model commits ahead of its evidence, and k^*, which locates commitment on the step axis, for matched-accuracy comparisons across model families. Figure 1 illustrates the two trajectories and the gap that U measures.

4 Causal Lookahead Probing

4.1 The Intervention

$\mathrm{CD}_M(k)$ is a probability over the task distribution, so it must be estimated from finitely many pairs. Causal Lookahead Probing (CLP) is that estimator, and Algorithm 1 states the procedure. Averaging the N flip indicators and applying the prior centering of Equation (1) yields $\widehat{\mathrm{CD}_M}(k)$; sampling design and variance are in Section G.

Algorithm 1 Causal Lookahead Probing

Require: model $\mathcal{M}$, one step k, component with m answers, pairs $\{(\mathbf{b}_n, \mathbf{s}_n)\}_{n=1}^{N}$ drawn independently

$c \leftarrow 0$
for $n = 1$ to N **do**
  run $\mathcal{M}$ on $\mathbf{s}_n$; cache $\mathbf{h}_k^{(L)}$
  run $\mathcal{M}$ on $\mathbf{b}_n$; at step k set $\mathbf{h}_k^{(L)} \leftarrow$ cached value
  complete the chain autoregressively; read $\hat{a} = \kappa(\cdot)$
  if $\hat{a} = a(\mathbf{s}_n)$ **then** $c \leftarrow c + 1$
  end if
end for
$\widehat{T} \leftarrow c/N$
return $\widehat{\mathrm{CD}_M}(k) = \dfrac{\widehat{T} - 1/m}{1 - 1/m}$

The design is deliberately rigid: the intervention transplants the entire last-layer residual stream at one step and leaves every parameter untouched, so a reading is a counterfactual on the model's own state rather than a readout of it (Section 2.3). The closest precedent for the mechanism is Pal et al. (2023), who transplant a hidden state into a different context to ask how many tokens ahead it determines. We ask a different question of the same operation: not how far ahead the state reaches, but at which step it stops being revisable, and whether that step is earlier than the task allows.

4.2 Immunity to the Non-Linear Representation Dilemma

The vacuity problem of Section 1 comes from the flexibility of a learned alignment: an expressive enough map scores perfectly on a model that computes nothing. CLP's alignment is the identity on the full residual stream, $\tau = \mathrm{id}$: no rotation, no subspace rank, no optimization of any kind.

Proposition 1 (Immunity to the dilemma)**.** *Because* CLP *fixes the alignment to the identity on the full last-layer residual stream and optimizes no alignment parameters, the family of expressive maps that drives the non-linear representation dilemma cannot be applied to* CLP*. A nonzero or bound-violating reading of* $\mathrm{CD}_M(k)$ *therefore cannot be manufactured by alignment flexibility.*

Section D proves this, and Section D.6 relates CLP to activation patching, distributed alignment search, layerwise mediation and forward-decoding probes: those ask where information is, or whether it can be read out; CLP asks whether altering it moves the answer. Section I.1 covers what a whole-state transplant gives up, variable-level attribution, and why the last layer is the place to read.

5 Experimental Setup

The tasks and models serve one end: a C-RASP$^{\text{CoT}}$ program, hence a prescribed trajectory, can be written down for every task, and final-answer accuracy is held fixed while reasoning training varies.

5.1 Tasks

Algebraic difficulty determines when a task's answer becomes determined, and therefore how much room a model has to commit early. Dyck-2 leaves almost none, so it tests whether the estimator reports spurious violations (Ebrahimi et al., 2020). The group word problems compute a product in a finite group from a generator sequence. A_4/S_4 admit shallow C-RASP$^{\text{CoT}}$ solutions and their trajectory rises at a moderate pace; the A_5/S_5 trajectory rises more slowly, so any model committing early must outrun it. That makes A_5/S_5 the sharpest test. Two further tasks supply ecological contrast: finite-state automaton (FSA) execution, in which the model reads a DFA and an input string and reports the final state (Zhang et al., 2025). The second is natural-language entity tracking: the model reads a short discourse describing objects changing location and reports where a queried entity ends up. Both reduce to the same state-tracking template (Li et al., 2025); construction details are in Section F. Each task comprises 1,000 evaluation instances whose input length is task-specific (Table 1: 8–32 bracket symbols for Dyck-2, 4–8 generators for the group tasks, 8–16 symbols for FSA, 3–5 sentences for Entity); generated chains range from 32 to 256 tokens depending on the task and model.

Task	Class	$\lvert\mathcal{A}\rvert$	CD_P rise	Tracr
Dyck-2	bounded-depth	2	early	✓
A_4, S_4	solvable group	12, 24	mid	✓
A_5, S_5	non-solv. (NC^1)	60, 120	late	✓
FSA exec.	regular	varies	mid	✓
entity track.	natural language	varies	late	approx.

Table 1: The five state-tracking tasks, ordered by algebraic difficulty. $\lvert\mathcal{A}\rvert$ is the answer-set size; "CD_P rise" is when the prescribed trajectory leaves the prior; the last column marks tasks with a Tracr-compiled ground-truth model; entity tracking compiles only an approximate program and is therefore excluded from the validity check of Section 6.1.

5.2 Compiled Programs and Ground Truth

For each task we write a C-RASP$^{\text{CoT}}$ program π using the Boolean and counting primitives of Section 2.2, which fixes the state sequence R_k and hence $\mathrm{CD}_P(k)$ for that task. We then compile π with Tracr. Because the compiled model provably executes π, applying CLP to it should recover CD_P up to estimation error; this is our validity check. Section E gives the compilation details.

5.3 Models

We evaluate nine checkpoints chosen to separate reasoning-trace distillation from scale, architecture and generic instruction tuning as far as public matched checkpoints permit. Pythia (410M) and the base Llama-3.2 (3B) provide reasoning-free references, the instruction-tuned Llama-3.2 adds alignment training on the same pretrained base, and R1-Distill (8B) is the model under test. Two distillation families give the matched comparison: R1-Distill against its own Llama-3.1-8B base and instruct checkpoints, and R1-Distill-Qwen against Qwen2.5-Math, holding architecture, parameter count and pretraining corpus fixed within each family. Both children are supervised distillations of an RL-trained teacher, not RL-trained themselves. Mamba (2.8B) is an architectural control: state-space models face the same expressivity limits as transformers on state tracking (Merrill et al., 2024), so a difference here could not be architectural alone.

5.4 Metrics and Protocol

Each curve is evaluated at 24 equally spaced steps, on 2400 base–source pairs per checkpoint, task and seed, repeated over three seeds. We compare families on the instances both models answer correctly,

	DYCK-2 calibration		A_4/S_4 formal		A_5/S_5 formal		FSA formal		Entity extension	
Model	U	k^*	U	k^*	U	k^*	U	k^*	U	k^*
Non-distilled controls										
PYTHIA	0.004	0.277	0.002	0.482	0.000	0.791	0.000	0.567	0.000	0.705
LLAMA-3.2 (base)	0.000	0.263	0.000	0.489	0.000	0.752	0.007	0.536	0.000	0.684
LLAMA-3.2 (inst)	0.000	0.268	0.004	0.478	0.000	0.745	0.011	0.535	0.002	0.707
MAMBA	0.000	0.265	0.000	0.476	0.001	0.747	0.008	0.544	0.000	0.692
Matched parents of the distilled models										
LLAMA-3.1-8B (base)	0.009	0.265	0.000	0.482	0.000	0.751	0.003	0.542	0.000	0.690
LLAMA-3.1-8B (inst)	0.011	0.263	0.003	0.473	0.000	0.745	0.000	0.531	0.000	0.701
QWEN2.5-MATH	0.000	0.269	0.000	0.477	0.001	0.744	0.000	0.541	0.005	0.697
Reasoning-distilled										
R1-DISTILL	0.043	**0.182**	**0.463**†	**0.190**	**0.641**†	**0.302**	**0.307**†	**0.277**	0.287	0.367
R1-DISTILL-QWEN	0.033	**0.195**	**0.428**	**0.206**	**0.584**	**0.332**	**0.278**	**0.289**	0.322	0.382

Table 2: U and k^* for every checkpoint and task. The three formal tasks carry the certified bound; DYCK-2 resolves its answer so early that the envelope leaves little room for a violation and serves as a calibration case; Entity Tracking is a natural-language extension without an exact reference program. †Significantly greater than every non-distilled control on the formal tasks (Holm–Bonferroni over the 15-comparison primary family, all adjusted $p < 0.001$; Section O).

reading k^* at equal accuracy, with bootstrap confidence intervals on $\widehat{\mathrm{CD}}_M(k)$. Behavioral baselines run alongside CLP: early-answering (Lanham et al., 2023), its accuracy-controlled normalization (Bentham et al., 2024), the detection–extraction gap (Wang & Zhu, 2026), and answer/correctness probes. Sampling design, competitor reproductions, and compute are in Sections F and H.

6 RESULTS

6.1 THE MEASURE RECOVERS A KNOWN TRAJECTORY

Before reading any model result we check that CLP measures what it claims. On the Tracr-compiled executors, the prescribed trajectory lies inside the bootstrap 95% confidence interval (CI) of the estimated CD_M at all 24 bins of every formal task, the largest pointwise deviation being 0.006. On a model whose computation is known, the estimator returns that computation (Figure 2), so any separation reported below is a property of the models rather than of the measurement.

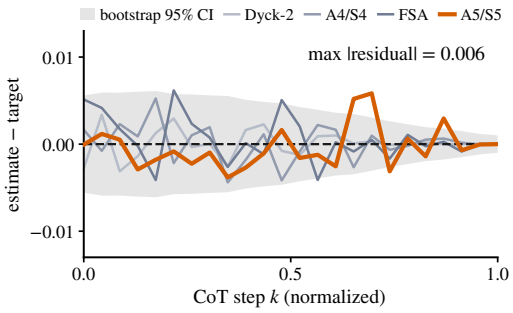


Figure 2: Validity on Tracr-compiled executors. Residual between the estimate and the prescribed trajectory at each step, on all four tasks with an exact reference program, against the bootstrap 95% interval the residual would have to sit inside. Every residual stays within the band and the largest is 0.006: on a model whose computation is known, the estimator returns that computation.

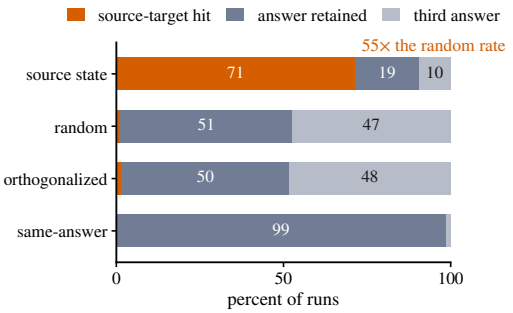


Figure 3: Where the answer lands after each intervention on A_5/S_5, $n = 2{,}400$ per condition. Each bar partitions the runs into the three possible outcomes. Random and orthogonalized states change the answer about half the time but reach the source answer in under 2% of runs, against 71% for the genuine transplant: perturbing the state and redirecting it come apart.

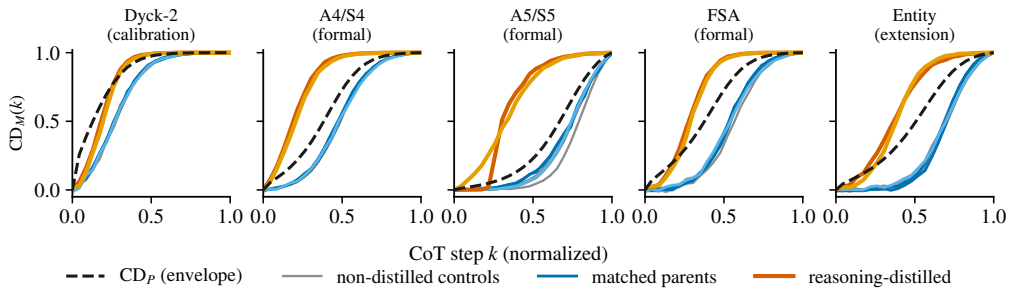


Figure 4: Commitment depth $\mathrm{CD}_M(k)$ against the prescribed trajectory $\mathrm{CD}_P(k)$ (dashed) on the calibration case (DYCK-2, left) and the strongest formal task (A_5/S_5, right). On DYCK-2 the envelope rises before any model does, so no violation is available to find; on A_5/S_5 both distilled checkpoints cross the envelope while every parent and control stays below, and the same ordering holds across all nine checkpoints and five tasks (Figure 12 in Section J).

6.2 REDIRECTION IS DIRECTIONAL, NOT GENERIC DISRUPTION

A flip could in principle mean only that the completion was disrupted. The controls separate two events that must not be conflated: whether the answer changes at all, and whether it changes *to the source answer*. A random norm-matched state changes the answer 48.7% of the time yet lands on the source answer only 1.3% of the time; an orthogonalized source state gives 49.9% and 1.6%. The genuine transplant reaches the source answer 71.4% of the time, a directional specificity of $55\times$ over random injection. A same-answer source, where redirection is impossible by construction, changes the answer in 1.4% of runs. Generic perturbation moves the output; only the transplanted state moves it to the source's answer; Figure 3 splits each condition into the three possible outcomes. The high change rate under random injection also rules out a confidence account of the finding: a model whose output distribution were simply more peaked would resist perturbation, not merely resist redirection.

The transplant replaces the entire step-k state, revision machinery included, so a flip certifies causal determination under the intervention, not unrevisability in the unperturbed run (Section 3.1). Most redirected completions carry the source computation forward rather than switching the answer bare, the behavior of a state still being *used*. Under a pre-registered rule requiring two source-only elements and one explicit operation on them, 71.9% $[70.7, 73.0]$ of the $5{,}453$ classifiable flips show source-consistent intermediate computation and 14.9% are bare answer switches (Section K.5).

The ordering of models survives every check we ran: patched layer, control injection, probe decodability, source and temperature stability, cumulative layer patching, same-instance position mismatch, accuracy without a chain, and traces containing visible self-correction (Section I and Section I.3).

6.3 THE BOUND IS VIOLATED ONLY BY REASONING-DISTILLED MODELS

Figure 4 plots $\mathrm{CD}_M(k)$ against $\mathrm{CD}_P(k)$ for every checkpoint and task, and Table 2 gives U and k^* for all 45 cells. Every non-distilled checkpoint stays within $U < 0.012$, and on the three formal tasks, where the envelope leaves room for a violation, none is distinguishable from zero. Both distilled checkpoints cross the bound on those three tasks. The violation grows with algebraic difficulty (FSA $0.307 < A_4/S_4$ $0.463 < A_5/S_5$ 0.641), as predicted if deeper serial computation leaves more room for premature commitment. DYCK-2 sits apart: its answer is fixed so early that the envelope crosses 0.5 at $k = 0.129$, ahead of R1-DISTILL's 0.182, so the task calibrates the measure rather than testing it. Per Section 3.3, the A_5/S_5 violation carries the algorithm-independent reading and the solvable-task violations are corroborating. On A_5/S_5 the distilled U exceeds every baseline by more than an order of magnitude ($p < 0.001$ after Holm–Bonferroni over the pre-registered family; Section O). Final-answer accuracy ranges from 0.62 to 0.94, with the distilled checkpoints at least as accurate on every task; the matched comparison below therefore conditions on jointly correct instances rather than on accuracy alone, which leaves 612 to 738 instances per comparison.

High accuracy and early commitment are not in tension: complexity lower bounds constrain serial *depth*, not chain tokens, and a model can do much of the required work while encoding the prompt, where one pass runs L layers. Section I.5 develops the point.

Model (A_5/S_5)	Params	U [95% CI]	k^* [95% CI]	Accuracy
LLAMA-3.1-8B (base)	8B	0.000 [0.000, 0.002]	0.751 [0.748, 0.753]	0.788
LLAMA-3.1-8B (instruct)	8B	0.000 [0.000, 0.001]	0.745 [0.741, 0.748]	0.826
QWEN2.5-MATH	7B	0.001 [0.000, 0.003]	0.744 [0.740, 0.748]	0.781
R1-DISTILL	8B	**0.641** [0.633, 0.650]	**0.302** [0.300, 0.304]	0.860
R1-DISTILL-QWEN	7B	**0.584** [0.578, 0.593]	**0.332** [0.327, 0.337]	0.846

Table 3: Matched parent/child control on A_5/S_5 in two families. Three seeds, $2{,}400$ pairs per model per seed. No parent checkpoint's U differs from zero, while each distilled child commits at roughly half the prescribed crossing (0.669): the gap is not explained by scale.

6.4 NEITHER SCALE NOR THE STEP AXIS EXPLAINS THE GAP

The distilled checkpoints are the largest in the comparison, so the violation could reflect scale. Each is built on a public parent, LLAMA-3.1-8B and QWEN2.5-MATH, whose base and instruct checkpoints hold architecture, parameter count and pretraining corpus fixed, making this the closest available matched-parent comparison. Tokenizer revisions do differ between child and parent, so every model receives the same plain-text prompt with no chat template, with identical SHA-256 across all nine checkpoints (Section K). Table 3 reports A_5/S_5, where the envelope rises latest and a scale effect would be easiest to see. Every parent stays within the bound, with k^* past the CD_P crossing (0.669), while its child on the same backbone commits at 0.302 (Llama) and 0.332 (Qwen). That is $2.21\times$ and $2.02\times$ ahead of what the task licenses. The intervention behaves identically on the new checkpoints, with the parent's same-answer control at 0.012 ± 0.006.

A further worry is the step axis itself: CD_P counts program steps while k counts tokens, and R1-DISTILL writes more tokens per step. Re-indexing every intervention by completed reference operations moves its commitment *earlier* rather than later, to $o^* = 0.264$ and a ratio of $2.53\times$; Section I.6 gives the markers, the conservation checks and the full comparison.

6.5 AT MATCHED ACCURACY, THE DISTILLED MODEL STILL COMMITS EARLIER

A natural objection is that earlier commitment merely reflects greater capability. Figure 8 in Section I controls for capability on the subset of instances that *both* models answer correctly, so correctness is identical on the two sides by construction and no binning of a per-instance 0/1 outcome is required. Against the matched LLAMA-3.1-8B parent ($n = 704$ jointly correct), R1-DISTILL commits at $k^* = 0.306$ versus 0.742, a difference of -0.436 (95% CI $[-0.461, -0.411]$) and a ratio of $2.43\times$; the other three controls give 2.36–$2.42\times$. The operation axis gives $2.53\times$ (Section I.6). With correctness held exactly fixed, the earliness cannot be attributed to capability. The unmatched separation holds on all five tasks (Section J); the accuracy control is run on A_5/S_5.

7 DISCUSSION AND CONCLUSION

What does the gap between CD_M and CD_P tell us about how premature commitment arises? A plausible mechanism has two stages. Reinforcement on outcome correctness rewards the *teacher* for reaching the right answer by any route, including routes that bypass the serial reasoning the CoT is meant to externalize. Nothing penalizes early internal convergence, so early commitment followed by plausible filler is rewarded as much as genuine derivation. Distillation then transfers the teacher's trace distribution to the students by supervised imitation, and both show premature commitment at the state level by CLP; Section K.5 shows the trace-level signature for R1-DISTILL. We do not measure the teacher, so whether this timing originates there or arises in distillation is open, as is whether *direct* reinforcement on a student of this scale would produce it. The CoT then functions less as a workspace than as a transcript of a decision already made, which is the unfaithfulness of Section 1 located at a step. Our contribution is to make this observation causal and normatively grounded: the model committed not merely early, but earlier than the state its own chain has built can license.

For traces used as explanations or safety-relevant monitoring signals, the implication is direct: on our state-tracking tasks the answer was often fixed before the critical reasoning steps were emitted, so a reader should not assume the steps caused the answer. The chain is not always the computation.

Reproducibility Statement

Code, the C-RASP$^{\text{CoT}}$ programs for all five tasks, the evaluation instances and the generated chains, and the result files behind every number in this paper are available at `https://anonymous.4open.science/r/CoT-CD`. The intervention is specified as pseudocode in Algorithm 1, its sampling design in Section F.3 and its variance in Section G. Every checkpoint is public and named in Section F.1, and the rendered prompt for each task is released together with its SHA-256, so the exact input reaching each model can be checked (Section K). Software versions are in Section F.6, the seeds behind each reported figure in Section F.3, and the hardware in Section F.5. The assumptions behind Theorem 1 are stated in Definition 5 and the proof is in Section C. A single script recomputes every table and figure from the released result files; the completed reproducibility checklist is Section M. Extensions and limitations are discussed in Section Q.

AI Use Statement

During the preparation of this paper, the authors used generative AI tools for copyediting, grammar checking, and improvements to clarity and readability. They were not used to develop the theoretical framework, to prove any of the theorems or propositions, to propose or refine hypotheses, to design the experiments or to provide feedback on the methodology, to implement the method, to generate or clean any dataset, or to interpret results. All content was developed and finalized by the authors, who take full responsibility for the accuracy, originality, and integrity of this paper.

Ethics Statement

This work studies whether the CoT traces emitted by reasoning models reflect the computation behind their answers. The question bears directly on the practice of reading such traces as explanations or as monitoring signals: a trace that performs a settled answer rather than computing it can mislead a reader who treats it as faithful. By giving a causal and normative handle on when a model commits, our method is intended to make that practice more cautious, not to enable any new capability. The models and tasks we use are standard, and the evaluation data is synthetic or drawn from public benchmarks and carries no personal or sensitive information. The trace audits reported in Sections I and K validate an extraction rule against its own outputs and collect no participant data.

References

Iván Arcuschin, Jett Janiak, Robert Krzyzanowski, Senthooran Rajamanoharan, Neel Nanda, and Arthur Conmy. Chain-of-Thought Reasoning In The Wild Is Not Always Faithful. *CoRR*, abs/2503.08679, 2025. URL `https://doi.org/10.48550/arXiv.2503.08679`.

Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. Eliciting Latent Predictions from Transformers with the Tuned Lens. *CoRR*, abs/2303.08112, 2023. URL `https://doi.org/10.48550/arXiv.2303.08112`.

Oliver Bentham, Nathan Stringham, and Ana Marasovic. Chain-of-Thought Unfaithfulness as Disguised Accuracy. *Trans. Mach. Learn. Res.*, 2024, 2024. URL `https://openreview.net/forum?id=ydcrP55u2e`.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. Pythia: A Suite for Analyzing Large Language Models Across Training and Scaling. In *Proc. ICML'23*, Proceedings of Machine Learning Research, pp. 2397–2430. PMLR, 2023. URL `https://proceedings.mlr.press/v202/biderman23a.html`.

Siddharth Boppana, Annabel Ma, Max Loeffler, Raphaël Sarfati, Eric Bigelow, Atticus Geiger, Owen Lewis, and Jack Merullo. Reasoning Theater: Disentangling Model Beliefs from Chain-of-Thought. *CoRR*, abs/2603.05488, 2026. doi: 10.48550/ARXIV.2603.05488. URL `https://doi.org/10.48550/arXiv.2603.05488`.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering Latent Knowledge in Language Models Without Supervision. In *Proc. ICLR'23*. OpenReview.net, 2023. URL https://openreview.net/forum?id=ETKGuby0hcs.

Yanda Chen, Joe Benton, Ansh Radhakrishnan, Jonathan Uesato, Carson Denison, John Schulman, Arushi Somani, Peter Hase, Misha Wagner, Fabien Roger, Vladimir Mikulik, Samuel R. Bowman, Jan Leike, Jared Kaplan, and Ethan Perez. Reasoning Models Don't Always Say What They Think. *CoRR*, abs/2505.05410, 2025. URL https://doi.org/10.48550/arXiv.2505.05410.

Kyle Cox, Darius Kianersi, and Adrià Garriga-Alonso. Post-Hoc Reasoning in Chain of Thought: Decoding and Steering Pre-Committed Answers. *CoRR*, abs/2603.01437, 2026. URL https://doi.org/10.48550/arXiv.2603.01437.

Joey David. Temporal Predictors of Outcome in Reasoning Language Models. *CoRR*, abs/2511.14773, 2025. URL https://doi.org/10.48550/arXiv.2511.14773.

Javid Ebrahimi, Dhruv Gelda, and Wei Zhang. How Can Self-Attention Networks Recognize Dyck-n Languages? In *Findings of EMNLP'20*, Findings of ACL, pp. 4301–4306. Association for Computational Linguistics, 2020. URL https://doi.org/10.18653/v1/2020.findings-emnlp.384.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. URL https://transformer-circuits.pub/2021/framework/index.html.

Guhao Feng, Bohang Zhang, Yuntian Gu, Haotian Ye, Di He, and Liwei Wang. Towards Revealing the Mystery behind Chain of Thought: A Theoretical Perspective. In *Advances in Neural Information Processing Systems 36 (NeurIPS'23)*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/dfc310e81992d2e4cedc09ac47eff13e-Abstract-Conference.html.

Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. Causal abstractions of neural networks. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34 (NeurIPS'21)*, pp. 9574–9586, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/4f5c422f4d49a5a807eda27434231040-Abstract.html.

Atticus Geiger, Zhengxuan Wu, Christopher Potts, Thomas Icard, and Noah D. Goodman. Finding Alignments Between Interpretable Causal Variables and Distributed Neural Representations. In *Proc. CLeaR'24*, Proceedings of Machine Learning Research, pp. 160–187. PMLR, 2024. URL https://proceedings.mlr.press/v236/geiger24a.html.

Atticus Geiger, Duligur Ibeling, Amir Zur, Maheep Chaudhary, Sonakshi Chauhan, Jing Huang, Aryaman Arora, Zhengxuan Wu, Noah D. Goodman, Christopher Potts, and Thomas Icard. Causal Abstraction: A Theoretical Foundation for Mechanistic Interpretability. *J. Mach. Learn. Res.*, 26: 83:1–83:64, 2025. URL https://jmlr.org/papers/v26/23-0058.html.

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 *Proc. ICLR'24*. OpenReview.net, 2024. URL https://openreview.net/forum?id=ph04CRkPdC.

Albert Gu and Tri Dao. Mamba: Linear-Time Sequence Modeling with Selective State Spaces. *CoRR*, abs/2312.00752, 2023. URL https://doi.org/10.48550/arXiv.2312.00752.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Deli Chen, Dongjie

Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, Hao Zhang, Hanwei Xu, Honghui Ding, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jingchang Chen, Jingyang Yuan, Jinhao Tu, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaichao You, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingxu Zhou, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Tao Yun, Tian Pei, Tianyu Sun, Tao Wang, Wangding Zeng, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nat.*, 645 (8081):633–638, 2025. URL https://doi.org/10.1038/s41586-025-09422-z.

Stefan Heimersheim and Neel Nanda. How to use and interpret activation patching. *CoRR*, abs/2404.15255, 2024. URL https://doi.org/10.48550/arXiv.2404.15255.

Alon Jacovi and Yoav Goldberg. Towards Faithfully Interpretable NLP Systems: How Should We Define and Evaluate Faithfulness? In *Proc. ACL'20*, pp. 4198–4205. Association for Computational Linguistics, 2020. URL https://doi.org/10.18653/v1/2020.acl-main.386.

Tomek Korbak, Mikita Balesni, Elizabeth Barnes, Yoshua Bengio, Joe Benton, Joseph Bloom, Mark Chen, Alan Cooney, Allan Dafoe, Anca D. Dragan, Scott Emmons, Owain Evans, David Farhi, Ryan Greenblatt, Dan Hendrycks, Marius Hobbhahn, Evan Hubinger, Geoffrey Irving, Erik Jenner, Daniel Kokotajlo, Victoria Krakovna, Shane Legg, David Lindner, David Luan, Aleksander Madry, Julian Michael, Neel Nanda, Dave Orr, Jakub Pachocki, Ethan Perez, Mary Phuong, Fabien Roger, Joshua Saxe, Buck Shlegeris, Martín Soto, Eric Steinberger, Jasmine Wang, Wojciech Zaremba, Bowen Baker, Rohin Shah, and Vladimir Mikulik. Chain of Thought Monitorability: A New and Fragile Opportunity for AI Safety. *CoRR*, abs/2507.11473, 2025. URL https://doi.org/10.48550/arXiv.2507.11473.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, Kamile Lukosiute, Karina Nguyen, Newton Cheng, Nicholas Joseph, Nicholas Schiefer, Oliver Rausch, Robin Larson, Sam McCandlish, Sandipan Kundu, Saurav Kadavath, Shannon Yang, Thomas Henighan, Timothy Maxwell, Timothy Telleen-Lawton, Tristan Hume, Zac Hatfield-Dodds, Jared Kaplan, Jan Brauner, Samuel R. Bowman, and Ethan Perez. Measuring faithfulness in chain-of-thought reasoning. *CoRR*, abs/2307.13702, 2023. URL https://doi.org/10.48550/arXiv.2307.13702.

Belinda Z. Li, Zifan Carl Guo, and Jacob Andreas. (How) Do Language Models Track State? In *Proc. ICML'25*, Proceedings of Machine Learning Research. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/li25r.html.

David Lindner, János Kramár, Sebastian Farquhar, Matthew Rahtz, Tom McGrath, and Vladimir Mikulik. Tracr: Compiled Transformers as a Laboratory for Interpretability. In *Advances in Neural Information Processing Systems 36 (NeurIPS'23)*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/771155abaae744e08576f1f3b4b7ac0d-Abstract-Conference.html.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and Editing Factual Associations in GPT. In *Advances in Neural Information Processing Systems 35 (NeurIPS'22)*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/6f1d43d5a82a37e89b0665b33bf3a182-Abstract-Conference.html.

William Merrill and Ashish Sabharwal. The Expressive Power of Transformers with Chain of Thought. In *Proc. ICLR'24*. OpenReview.net, 2024. URL https://openreview.net/forum?id=NjNGlPh8Wh.

William Merrill, Jackson Petty, and Ashish Sabharwal. The Illusion of State in State-Space Models. In *Proc. ICML'24*, Proceedings of Machine Learning Research, pp. 35492–35506. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/merrill24a.html.

nostalgebraist. Interpreting GPT: the logit lens. LessWrong, 2020. URL https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens.

Maxwell I. Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, Charles Sutton, and Augustus Odena. Show Your Work: Scratchpads for Intermediate Computation with Language Models. *CoRR*, abs/2112.00114, 2021. URL https://arxiv.org/abs/2112.00114.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom In: An Introduction to Circuits. *Distill*, 2020. URL https://distill.pub/2020/circuits/zoom-in/?ref%5Cx3dcold-takes.

Koyena Pal, Jiuding Sun, Andrew Yuan, Byron C. Wallace, and David Bau. Future Lens: Anticipating Subsequent Tokens from a Single Hidden State. In *Proc. CoNLL'23*, pp. 548–560. Association for Computational Linguistics, 2023. URL https://doi.org/10.18653/v1/2023.conll-1.37.

Kiho Park, Yo Joong Choe, and Victor Veitch. The Linear Representation Hypothesis and the Geometry of Large Language Models. In *Proc. ICML'24*, Proceedings of Machine Learning Research, pp. 39643–39666. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/park24c.html.

Jacob Pfau, William Merrill, and Samuel R. Bowman. Let's Think Dot by Dot: Hidden Computation in Transformer Language Models. *CoRR*, abs/2404.15758, 2024. URL https://doi.org/10.48550/arXiv.2404.15758.

Anish Sathyanarayanan, Aditya Nagarsekar, and Aarush Rathore. Bypassing the Rationale: Causal Auditing of Implicit Reasoning in Language Models. *CoRR*, abs/2602.03994, 2026. URL https://doi.org/10.48550/arXiv.2602.03994.

Alessandro Stolfo, Yonatan Belinkov, and Mrinmaya Sachan. A Mechanistic Interpretation of Arithmetic Reasoning in Language Models using Causal Mediation Analysis. In *Proc. EMNLP'23*, pp. 7035–7052. Association for Computational Linguistics, 2023. URL https://doi.org/10.18653/v1/2023.emnlp-main.435.

Denis Sutter, Julian Minder, Thomas Hofmann, and Tiago Pimentel. The Non-Linear Representation Dilemma: Is Causal Abstraction Enough for Mechanistic Interpretability? In *Advances in Neural Information Processing Systems 38 (NeurIPS'25)*, 2025. URL https://neurips.cc/virtual/2025/loc/san-diego/poster/117348.

Llama Team. The Llama 3 Herd of Models. *CoRR*, abs/2407.21783, 2024. URL https://doi.org/10.48550/arXiv.2407.21783.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, Alex Tamkin, Esin Durmus, Tristan Hume, Francesco Mosconi, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan. Scaling Monosemanticity: Extracting

Interpretable Features from Claude 3 Sonnet. *Transformer Circuits Thread*, 2024. URL https://transformer-circuits.pub/2024/scaling-monosemanticity/.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. Language Models Don't Always Say What They Think: Unfaithful Explanations in Chain-of-Thought Prompting. In *Advances in Neural Information Processing Systems 36 (NeurIPS'23)*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/ed3fea9033a80fea1376299fa7863f4a-Abstract-Conference.html.

Martin Tutek, Fateme Hashemi Chaleshtori, Ana Marasovic, and Yonatan Belinkov. Measuring Chain of Thought Faithfulness by Unlearning Reasoning Steps. In *Proc. EMNLP'25*, pp. 9935–9960. Association for Computational Linguistics, 2025. URL https://doi.org/10.18653/v1/2025.emnlp-main.504.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart M. Shieber. Investigating Gender Bias in Language Models Using Causal Mediation Analysis. In *Advances in Neural Information Processing Systems 33 (NeurIPS'20)*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/92650b2e92217715fe312e6fa7b90d82-Abstract.html.

Hanyang Wang and Mingxuan Zhu. The Detection-Extraction Gap: Models Know the Answer Before They Can Say It. *CoRR*, abs/2604.06613, 2026. URL https://doi.org/10.48550/arXiv.2604.06613.

Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the Wild: a Circuit for Indirect Object Identification in GPT-2 Small. In *Proc. ICLR'23*. OpenReview.net, 2023. URL https://openreview.net/forum?id=NpsVSN6o4ul.

Gail Weiss, Yoav Goldberg, and Eran Yahav. Thinking Like Transformers. In *Proc. ICML'21*, Proceedings of Machine Learning Research, pp. 11080–11090. PMLR, 2021. URL http://proceedings.mlr.press/v139/weiss21a.html.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. Qwen2.5-Math Technical Report: Toward Mathematical Expert Model via Self-Improvement. *CoRR*, abs/2409.12122, 2024. URL https://doi.org/10.48550/arXiv.2409.12122.

Andy Yang and David Chiang. Counting Like Transformers: Compiling Temporal Counting Logic Into Softmax Transformers. In *Proc. CoLM'24*, 2024. URL https://doi.org/10.18653/v1/2024.findings-acl.242.

Andy Yang, Michaël Cadilhac, and David Chiang. Knee-Deep in C-RASP: A Transformer Depth Hierarchy. In *Advances in Neural Information Processing Systems 38 (NeurIPS'25)*, 2025. URL https://neurips.cc/virtual/2025/loc/san-diego/poster/116469.

Donald Ye, Max Loffgren, Om Kotadia, and Linus Wong. Mechanistic Evidence for Faithfulness Decay in Chain-of-Thought Reasoning. *CoRR*, abs/2602.11201, 2026. URL https://doi.org/10.48550/arXiv.2602.11201.

Yifan Zhang, Wenyu Du, Dongming Jin, Jie Fu, and Zhi Jin. Finite State Automata Inside Transformers with Chain-of-Thought: A Mechanistic Study on State Tracking. In *Proc. ACL'25*, pp. 13603–13621. Association for Computational Linguistics, 2025. URL https://aclanthology.org/2025.acl-long.668/.

Ziqian Zhong, Ziming Liu, Max Tegmark, and Jacob Andreas. The clock and the pizza: Two stories in mechanistic explanation of neural networks. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36 (NeurIPS'23)*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/56cbfbf49937a0873d451343ddc8c57d-Abstract-Conference.html.

APPENDIX

TABLE OF CONTENTS

A Notation

B C-RASP-CoT Semantics and Equivalence
Standard C-RASP Semantics • The Autoregressive Reading • Equivalence

C Proof of the Faithfulness Bound
Setup • The Hypothesis, Stated as a Causal Abstraction • Proof of Theorem 1 • Componentwise Aggregation • Numerical Checks on the Bound • Boundary Cases

D CLP and the Non-Linear Representation Dilemma
The Dilemma, Formally • CLP as a Degenerate Alignment • The Channel the Transplant Acts Through • Proof of Proposition 1 • Dilemma Status of Other Causal Methods • Relation to Existing Interventions

E Compilation Pipeline
Overview • Per-Task Programs • Compilation Parameters • Precision Analysis

F Experimental Details
Models • Tasks • Sampling Design • Generation Configuration • Compute • Software

G Metric Definitions
The commitment depth Estimator • Summary Metrics • Baseline Measures • Competitor Method Metrics • Statistical Testing Framework

H Competitor Reproductions
Early-Answering • Accuracy-Controlled Normalization • Detection–Extraction Gap • Distributed Alignment Search (DAS) • Layerwise Mediation (CMI) • Pre-CoT Decoding Probes • Step-Level Faithfulness Decay (NLDD) • Competitor Overlay • Comprehensive Comparison

I Ablations and Robustness Checks
Scope and Trade-offs of CLP • Robustness Summary • Promised Follow-Up Controls • Verbosity and the Operation Axis • Two Clocks: Serial Depth versus Chain Tokens • Operation-Aligned Re-Analysis • Control-Injection Ablation • Probe Comparison • DAS Rank Ablation • Bootstrap Stability • Seed-to-Seed Agreement • Bootstrap Intervals for Every Cell • Integrated Violation • Ablation Summary

J Per-Task Breakdowns
Per-Task Discussion

K Prompt Templates and Transcripts
Group Word Problems • Dyck-2 • Finite-State Automaton Execution • Entity Tracking • Error Examples and Commitment Signatures

L Negative Results and Failure Cases

M Reproducibility Checklist
Reproducibility Checklist • Hyperparameter Summary

N Method Capability Comparison

O Statistical Tests
Permutation Tests for U • Effect Sizes for k^* • Summary

P Extended Related Work
Behavioral Faithfulness • Internal-State Reading and Probing • Causal Intervention Methods • Positioning of the Present Work

Q Limitations

A NOTATION

Table 4 collects the symbols used throughout the paper, grouped by the part of the framework they belong to, with the location of each symbol's definition. Where a symbol is introduced by a displayed definition or equation, the reference points to it.

Symbol	Meaning	Defined
Models and chains of thought		
$\mathcal{M}$	autoregressive language model under study	Section 2.1
L, d	number of transformer layers; model dimension	Section 2.1
ℓ	layer index, $1 \le \ell \le L$	Section I.3
$\mathbf{h}_k^{(L)}$	last-layer residual stream at step k	Definition 1
k, T	chain-of-thought step index; trace length	Section 2.1
$x_{1:k}$	chain-of-thought prefix through step k	Section 2.1
κ	readout: greedy answer-line decoding and a fixed parser to $\mathcal{A} \cup \{\perp\}$	Section 2.1
$\mathcal{A}$, a	answer set; gold answer as a random variable	Section 2.1
$\mathbf{b}$, $\mathbf{s}$	base and source instances of an intervention	Section 3.1
$\mathcal{D}$	task distribution over instances	Section C
Reference algorithm		
π	C-RASP$^{\text{CoT}}$ program for the task	Section 2.2
Σ, Γ	input alphabet; output-symbol alphabet	Section 2.2
Q_σ	input Boolean vector for symbol σ	Section B.1
$\#P$	prefix count of a Boolean vector P	Section B.1
R_k	reference-program state after k steps	Section 3.2
$\beta(k)$	Bayes accuracy of the source answer from R_k	Equation (2)
m	number of equiprobable answers in a task component	Equation (1)
Commitment depth and the bound		
$c(\cdot)$	prior centering, $c(p) = (p - 1/m)/(1 - 1/m)$	Equation (4)
$\mathrm{CD}_M(k)$	commitment depth: causal flip probability	Definition 1
$\mathrm{CD}_P(k)$	prescribed trajectory, the centered Bayes accuracy	Definition 2
$U(\mathcal{M}, \pi)$	unfaithfulness functional, the largest gap	Definition 3
k^*	commitment step, first k with $\mathrm{CD}_M \ge \frac{1}{2}$	Definition 3
o^*	commitment step on the operation axis	Section I.6
f	map through which a licensed patched answer depends on its inputs	Definition 5
ω	decoding randomness, independent of both instances	Definition 5
Causal Lookahead Probing and alignment		
$\widehat{\mathrm{CD}_M}(k)$	Monte Carlo estimate of $\mathrm{CD}_M(k)$	Section 4.1
N	number of base–source pairs sampled	Section 4.1
K	number of step bins on the normalized axis, $K = 24$	Section F.3
τ, Π	featurizing map; coordinate selection	Section D.1
$R \in O(d)$, r	learned rotation and subspace rank (DAS)	Section 4.2
$\mathcal{F}$	class of featurizing maps searched over	Definition 6
V, $\mathcal{V}$	high-level variable and its value set	Section D.1
$\mathrm{IIA}(\tau, \Pi)$	interchange-intervention accuracy	Equation (5)

Table 4: Notation used in the paper.

B C-RASP$^{\text{CoT}}$ SEMANTICS AND EQUIVALENCE

This appendix gives the formal semantics of C-RASP, defines the autoregressive reading C-RASP$^{\text{CoT}}$ used in Section 2.2, and proves that the two compute the same function of a complete prefix. That equivalence is what lets us read the program state off a prefix without committing to which reading produced it, which is all the prescribed trajectory requires. It is weaker than an

equivalence of generation, so we do not carry complexity-class results for standard C-RASP (Yang & Chiang, 2024; Yang et al., 2025) over to C-RASP$^{\text{CoT}}$.

B.1 Standard C-RASP Semantics

A C-RASP program operates on an input string $w = w_1 \cdots w_n \in \Sigma^+$ and manipulates two sorts of position-indexed vectors of length n. A *Boolean* vector P assigns $P(i) \in \{0, 1\}$ to each position i; a *count* vector C assigns $C(i) \in \mathbb{N}$. The semantics is given by the following operations, each defining a new vector pointwise from existing ones.

- **Input.** For each $\sigma \in \Sigma$ the initial Boolean vector Q_σ is $Q_\sigma(i) = \mathbf{1}[w_i = \sigma]$.
- **Boolean combination.** For Boolean vectors P_1, P_2 and any Boolean function f, the vector $f(P_1, P_2)$ is $f(P_1(i), P_2(i))$.
- **Prefix count.** For a Boolean vector P, the count vector $\#P$ is $\#P(i) = \sum_{j \le i} P(j)$, the number of earlier positions, inclusive, at which P holds.
- **Comparison.** For count vectors C_1, C_2, the Boolean vector $[C_1 \le C_2]$ is $\mathbf{1}[C_1(i) \le C_2(i)]$; constants and the analogous relations are included.

A program is a finite sequence of such operations terminating in a designated Boolean or count vector; its output at position i is the value of that vector at i. The *nesting depth* of a program is the maximal number of prefix counts nested through one another, the serial resource the C-RASP depth hierarchy measures (Yang et al., 2025).

B.2 The Autoregressive Reading

A CoT differs from a fixed input in two ways: it is revealed one symbol at a time, and the program may condition later steps on symbols it has itself emitted. We capture both with a reading that evaluates the program on growing prefixes.

Definition 4 (C-RASP$^{\text{CoT}}$)**.** *Let P be a* C-RASP *program over the alphabet $\Sigma \cup \Gamma$, where Γ is a finite set of output symbols. The* autoregressive reading C-RASP$^{\text{CoT}}$ *evaluates P at each step k on the prefix $x_{1:k} \in (\Sigma \cup \Gamma)^k$ available at that step, and emits the output symbol $x_{k+1} \in \Gamma$ determined by the program's terminal vector at position k. The emitted symbol is appended to the prefix and is available to the program at every later step. A program with $\Gamma = \emptyset$ performs no emission steps; its terminal prefix is the input itself. The* answer *is read from the terminal prefix through the readout κ of Section 2.1.*

The only departure from standard C-RASP is that the input to step k includes symbols in Γ produced at steps $1, \ldots, k-1$. All operations of Section B.1 apply unchanged, now over the extended alphabet $\Sigma \cup \Gamma$.

B.3 Equivalence

Theorem 2 (Prefix equivalence)**.** *For every* C-RASP$^{\text{CoT}}$ *program there is a standard* C-RASP *program that computes the same map from complete prefixes $x_{1:T}$ to outputs, and conversely. The transformation preserves nesting depth up to an additive constant.*

Proof. ($\Leftarrow$) A standard C-RASP program is the special case of Definition 4 with $\Gamma = \emptyset$, so every C-RASP program is already a C-RASP$^{\text{CoT}}$ program reading no outputs of its own.

($\Rightarrow$) Fix a C-RASP$^{\text{CoT}}$ program P and define a standard C-RASP program P' over $\Sigma \cup \Gamma$ that reads a complete prefix $x_{1:T} \in (\Sigma \cup \Gamma)^T$ as a single fixed input. The construction below is the same for every such prefix. The initial vectors $\{Q_\sigma\}_{\sigma \in \Sigma \cup \Gamma}$ of P' encode every symbol of the prefix, including the emitted ones, exactly as P reads them. Because each operation in Section B.1 at position i depends only on positions $j \le i$ through the prefix count, the value P computes at step k, which depends only on $x_{1:k}$, equals the value P' computes at position k on the fixed input. Hence P' reproduces P on every complete prefix. The construction adds at most one Boolean vector to distinguish symbols of Γ from those of Σ; a Boolean vector uses no prefix count, so the nesting depth is unchanged. □

Theorem 2 is a statement about complete prefixes: the two readings compute the same function of $x_{1:T}$ with the same nesting depth up to a constant. That is what the prescribed trajectory needs, because the program state after step k, and hence the answer posterior it induces, is then a well-defined property of the prefix the program has read, independent of which reading computes it. The theorem does not establish that a standard C-RASP program can generate the intermediate symbols from the problem input alone, so we do not transfer generation-level complexity statements about standard C-RASP, such as its equivalence to the counting logic $\mathrm{K}_t[\#]$ or its status as the tightest known lower bound on future-masked soft-attention transformers (Yang & Chiang, 2024), to C-RASP$^{\mathrm{CoT}}$.

C Proof of the Faithfulness Bound

This appendix proves Theorem 1: a model whose patched answer depends on the intervened run only through the licensed state satisfies $\mathrm{CD}_M(k) \le \mathrm{CD}_P(k)$ at every step. The argument is short, and its shortness is the point: the bound is the optimum over decision rules, so it is tight by construction and needs no auxiliary inequality.

C.1 Setup

Fix a task with reference program π and answer set $\mathcal{A}$, and let $\mathbf{b}, \mathbf{s} \sim \mathcal{D}$ be drawn independently. Write $R_k(\mathbf{s})$ for the state of π after k steps on $\mathbf{s}$, and $a(\cdot)$ for the answer of an instance. For a component whose answer set has m equiprobable values, both quantities are put on the same scale by the prior centering of Equation (1),

$$c(p) \;=\; \frac{p - 1/m}{1 - 1/m}, \qquad \mathrm{CD}_M(k) = c\big(T(k)\big), \qquad \mathrm{CD}_P(k) = c\big(\beta(k)\big), \tag{4}$$

where $T(k) = \Pr[\hat{a} = a(\mathbf{s})]$ is the redirection probability of Definition 1 and $\beta(k) = \mathbb{E}\max_a \Pr[a(\mathbf{s}) = a \mid R_k(\mathbf{s})]$ is the Bayes accuracy of Definition 2. Note that c is affine and strictly increasing, so it preserves inequalities.

C.2 The Hypothesis, Stated as a Causal Abstraction

Definition 5 (Licensed dependence)**.** *The patched run* depends only on licensed information at step k *if there is a function* f *with*

$$\kappa\Big(\mathcal{M}^{\mathbf{h}_k^{(L)} \leftarrow \mathbf{s}}(\mathbf{b})\Big) \;=\; f\big(\mathbf{b},\, R_k(\mathbf{s}),\, \omega\big)$$

almost surely, where ω *collects the decoding randomness. Since* $\mathbf{b}$ *is drawn independently of* $\mathbf{s}$*, and* ω *independently of both instances, the pair* $(\mathbf{b}, \omega)$ *is independent of* $\mathbf{s}$ *and therefore jointly independent of* $a(\mathbf{s})$ *given* $R_k(\mathbf{s})$*.*

This is an intervention-compatible causal abstraction in the sense of Geiger et al. (2025): the low-level intervention (replacing $\mathbf{h}_k^{(L)}$) is required to act on the high-level run only through the high-level variable R_k. A model executing π satisfies it, because after k steps such a model has computed R_k and nothing else about $\mathbf{s}$.

C.3 Proof of Theorem 1

Proof. Under Definition 5 the patched answer equals $f(\mathbf{b}, R_k(\mathbf{s}), \omega)$, so

$$T(k) = \Pr\big[f(\mathbf{b}, R_k(\mathbf{s}), \omega) = a(\mathbf{s})\big] \;=\; \mathbb{E}\Big[\Pr\big[f = a(\mathbf{s}) \;\big|\; R_k(\mathbf{s})\big]\Big].$$

Conditioned on $R_k(\mathbf{s}) = r$, the pair $(\mathbf{b}, \omega)$ is independent of $a(\mathbf{s})$, so f acts as a (randomized) decision rule that sees only r. No such rule can beat the Bayes rule for r:

$$\Pr\big[f = a(\mathbf{s}) \;\big|\; R_k(\mathbf{s}) = r\big] \;\le\; \max_{a \in \mathcal{A}} \Pr\big[a(\mathbf{s}) = a \mid R_k(\mathbf{s}) = r\big].$$

Taking expectations over r gives $T(k) \le \beta(k)$, and applying the increasing affine map c of Equation (4) to both sides gives $\mathrm{CD}_M(k) \le \mathrm{CD}_P(k)$. □

The bound is attained exactly when f is a Bayes rule for $R_k(\mathbf{s})$, which is why we call CD_P an envelope rather than a bound with slack: for every k there is a licensed model meeting it.

C.4 Componentwise Aggregation

Tasks whose answer decomposes into components (the group tasks pair A_n with S_n) are handled componentwise. Let component j have M_j equiprobable answers and fixed weight w_j with $\sum_j w_j = 1$. Applying the argument above within each component gives $T_j(k) \le \beta_j(k)$, and since centering is applied inside each component before combining,

$$\mathrm{CD}_M(k) = \sum_j w_j \, c_j\big(T_j(k)\big) \;\le\; \sum_j w_j \, c_j\big(\beta_j(k)\big) = \mathrm{CD}_P(k).$$

The weights are fixed in advance and are the same for CD_M and CD_P, so no choice of weighting can create or remove a violation.

Table 5 lists the components of each task, their answer-set sizes and their weights. Weights are uniform within a task: a task with J components gives each weight $1/J$. Dyck-2 has a single component, so the aggregation is vacuous there.

Task	Component	M_j	w_j
Dyck-2	balanced/unbalanced	2	1
A_4/S_4	A_4	12	$1/2$
	S_4	24	$1/2$
A_5/S_5	A_5	60	$1/2$
	S_5	120	$1/2$
FSA	2 states	2	$1/3$
	3 states	3	$1/3$
	4 states	4	$1/3$
Entity	3 locations	3	$1/4$
	4 locations	4	$1/4$
	5 locations	5	$1/4$
	6 locations	6	$1/4$

Table 5: Components, answer-set sizes and aggregation weights. M_j is the number of equiprobable answers in component j and w_j its weight. Because c_j divides by $1 - 1/M_j$, a component with a larger answer set contributes a centered value on the same scale as a smaller one, which is what makes the weighted sum meaningful.

C.5 Numerical Checks on the Bound

Theorem 1 is proved above, but a proof can be right and its implementation wrong. Two numerical checks guard against that.

The first is a synthetic stress test. For each answer-set size $m \in \{2, 3, 4, 12, 24, 60, 120\}$ covering every component in Table 5, we draw 2,000 random licensed models, compute $\mathrm{CD}_M(k) - \mathrm{CD}_P(k)$ at every step, and record the maximum. The largest value over all seven sizes is -1.4×10^{-4}, attained at $m = 2$: the difference is negative everywhere, so no random licensed model comes within 10^{-4} of crossing the envelope, let alone exceeds it.

The second uses the Tracr-compiled executors, where the licensed model is not random but the actual reference program. Across the four tasks with an exact program and all 24 steps, the slack $\mathrm{CD}_P(k) - \mathrm{CD}_M(k)$ stays in $[0.0003, 0.0040]$ and the bound holds at all 96 points. The slack is small because a compiled executor is close to a Bayes rule at every step, which is the sense in which Theorem 1 is an envelope rather than a loose inequality.

C.6 Boundary Cases

Both endpoints. At $k = 0$ the program has read nothing, so R_0 is constant, the Bayes rule is the prior, $\beta(0) = 1/m$ and $\mathrm{CD}_P(0) = 0$. Once R_k determines the answer, $\beta = 1$ and $\mathrm{CD}_P = 1$. The bound is therefore non-vacuous at both ends: at $k = 0$ it forbids any redirection above chance, and at the end it imposes nothing.

Same-answer pairs. When $a(\mathbf{b}) = a(\mathbf{s})$ a redirection cannot be distinguished from no effect. Because $\mathbf{b}$ and $\mathbf{s}$ are drawn independently rather than conditioned to differ, such pairs enter both T and β in the same proportion and the inequality is unaffected; Section 6.2 reports them as a separate control.

The argument is analytic, so we also checked it numerically. Let q_{same} be the fraction of same-answer pairs that the readout leaves unflipped, and recompute every headline metric under $q_{\text{same}} \in \{0.970, 0.986, 1.000\}$, spanning a pessimistic value, the value observed on A_5/S_5, and the neutral case in which such pairs never flip. Across the 35 (model, task) cells the range of U has median 0 and maximum 0.030, and the range of k^* has median 0.0035 and maximum 0.0116. On 33 of the 35 cells this range is narrower than that cell's own bootstrap interval for k^*, and on the two exceptions it exceeds the interval by less than 0.002. How same-answer pairs are treated therefore does not change any reported comparison.

Where the bound has no room to bind. On a task whose envelope rises before any model does, as DYCK-2 does, CD_P is already close to one where models are still near zero, and no violation is available to find. Such a task tests whether the estimator produces false alarms, not whether a model commits early. We report it in that role rather than as a negative result.

Scope. The bound is task-specific: it constrains any executor that reads the source only through the state the chain has built, on *that* task, and says nothing about a model on a task whose procedure we have not written down. It is also silent on cause. A violation shows that the state carries answer information the reference computation has not yet produced; whether that is due to scale, architecture, or training is settled by the controls of Section 5.3, not by the theorem.

D CLP AND THE NON-LINEAR REPRESENTATION DILEMMA

This appendix formalizes the non-linear representation dilemma, states the intervention of CLP in the same terms, and proves Proposition 1: that CLP lies outside the class of analyses the dilemma renders vacuous.

D.1 THE DILEMMA, FORMALLY

A learned causal analysis posits a high-level variable V taking values in a finite set $\mathcal{V}$ and seeks an *alignment* between V and the model's activations. Following the causal-abstraction framework, an alignment is a pair (τ, Π) in which $\tau : \mathbb{R}^d \to \mathbb{R}^d$ is a featurizing map and Π selects a subset of coordinates of $\tau(\mathbf{h})$ to be identified with V. An interchange intervention sets those coordinates to the value they take on a source input and reads the resulting label; the alignment *succeeds* on a pair if the label matches the one predicted by intervening on V in the high-level algorithm. Let

$$\mathrm{IIA}(\tau, \Pi) \;=\; \Pr_{(\mathbf{b},\mathbf{s})}\big[\,\text{interchange matches the } V\text{-intervention}\,\big] \tag{5}$$

denote the interchange-intervention accuracy.

Let $\mathcal{F}$ be the class of featurizing maps the analysis is permitted to search over. The dilemma concerns what happens when $\mathcal{F}$ is expressive.

Definition 6 (Expressive alignment family)**.** *$\mathcal{F}$ is* expressive *for a model $\mathcal{M}$ and algorithm if for every assignment of the high-level variable there exists $\tau \in \mathcal{F}$ and a selection Π with $\mathrm{IIA}(\tau, \Pi) = 1$.*

Proposition 2 (Vacuity, after Sutter et al. (2025))**.** *If $\mathcal{F}$ is expressive in the sense of Definition 6, then $\mathrm{IIA}(\tau, \Pi) = 1$ for some (τ, Π) carries no information about whether $\mathcal{M}$ implements the algorithm: the same maximal score is attainable on a model that does not compute the task at all, including a randomly initialized one.*

This is the dilemma. The empirical force of Sutter et al. (2025) is that the families used in practice, in particular the non-linear maps of distributed alignment search, are expressive enough to reach $\mathrm{IIA} = 1$ on randomly initialized models on the indirect-object identification task, so a perfect interchange score certifies nothing about the computation. Restricting $\mathcal{F}$ to escape this, for instance to linear maps, trades the problem for a contested assumption about how features are encoded.

D.2 CLP AS A DEGENERATE ALIGNMENT

CLP can be written in the same notation. Its intervention transplants the entire last-layer residual stream at a single step (Section 4.1), which corresponds to taking

$$\tau \;=\; \mathrm{id}_{\mathbb{R}^d}, \qquad \Pi \;=\; \text{all } d \text{ coordinates}, \tag{6}$$

with no parameters fitted. There is no featurizing map to choose and no subspace to select; the alignment is fixed before any data is seen.

D.3 THE CHANNEL THE TRANSPLANT ACTS THROUGH

We intervene on the last-layer residual state at the selected generation step and then let the model continue autoregressively. The bound of Theorem 1 applies under the licensed-dependence condition of Definition 5; the interpretation of a flip therefore rests on that condition and not on any further claim about how the intervention is realized at the level of cached activations.

D.4 PROOF OF PROPOSITION 1

Proof. The dilemma of Proposition 2 operates by quantifying over a family $\mathcal{F}$ of featurizing maps: vacuity arises because some $\tau \in \mathcal{F}$ can be *selected*, after seeing the model, to make the interchange succeed. The construction therefore requires (i) a nontrivial search space $\mathcal{F}$ and (ii) the freedom to fit τ to the model.

CLP provides neither. By Equation (6) its alignment is the single fixed point $\tau = \mathrm{id}$ on all d coordinates, chosen independently of $\mathcal{M}$. The search space is the singleton $\{\mathrm{id}\}$, which is not expressive in the sense of Definition 6 unless the identity already achieves $\mathrm{IIA} = 1$, and no fitting occurs. Consequently the mechanism of Proposition 2, the selection of a favorable τ, is unavailable, and a reading $\mathrm{CD}_M(k) > 0$ cannot be produced by alignment flexibility.

It remains to interpret what a positive reading then means. Under $\tau = \mathrm{id}$ the interchange intervention is literally the substitution of the model's own last-layer state, so $\mathrm{CD}_M(k)$ measures the causal effect of that state on the emitted answer with no representational assumption interposed. A high $\mathrm{CD}_M(k)$ reflects a property of $\mathcal{M}$'s own computation rather than of a fitted map. This is the sense in which CLP is immune to the dilemma: it forgoes the expressive power that creates vacuity, and with it the ability to localize a variable, retaining only the assumption-free causal measurement of commitment. □

The trade is explicit. A method that fits τ can ask which variable carries the answer but cannot certify that its alignment is not an artifact; CLP cannot name the variable but its measurement is not an artifact of any fit. For the question of *when* a model commits, the second is the property that matters.

D.5 DILEMMA STATUS OF OTHER CAUSAL METHODS

Section H compares CLP against layerwise mediation (CMI) and behavioral step-corruption (NLDD). Both methods are causal or quasi-causal, so it is natural to ask whether they, too, fall inside the dilemma of Proposition 2. We analyze each in the notation of Section D.1.

Layerwise mediation (CMI). The CMI intervention of Sathyanarayanan et al. (2026) replaces the hidden states at the CoT token positions in layer ℓ with the corresponding activations from a No-CoT forward pass. In the notation of Section D.1 this can be written as an interchange with

$$\tau = \mathrm{id}_{\mathbb{R}^d}, \quad \Pi_\ell = \big\{\text{CoT positions in layer } \ell\big\}, \tag{7}$$

where Π_ℓ selects a fixed, data-determined subset of positions (namely the positions occupied by CoT tokens) at a single layer ℓ. Crucially, the featurizing map is the identity and Π_ℓ is determined by the input tokenization, not fitted to the model. The dilemma of Proposition 2 requires the freedom to *choose* τ after seeing $\mathcal{M}$; CMI chooses nothing from $\mathcal{F}$ because its $\mathcal{F} = \{\mathrm{id}\}$, the same singleton as CLP.

There is, however, a subtlety. CMI applies a second intervention, the *control patch*, which replaces the same number of non-CoT positions and subtracts the resulting drop from the CoT-patch drop

(Equation (13)). The selection of control positions introduces a degree of freedom that could, in principle, be adversarially tuned. In practice Sathyanarayanan et al. (2026) draw control positions uniformly at random, so no fitting occurs, but the possibility should be noted: if one were to search over control positions to maximize CMI_ℓ, the subtraction would introduce an optimization whose expressiveness would need analysis of its own. Under the published random-control protocol, this concern does not arise, and CMI shares with CLP the property that its featurizing map is the identity.

A second difference is the *scope* of the intervention. CLP replaces the entire last-layer residual stream at one step (d coordinates, all positions), while CMI replaces a subset of positions at one layer. The former measures whether the full state at step k determines the answer; the latter measures how much the CoT representations at layer ℓ contribute to the answer. Neither is a special case of the other: they answer different questions (step-time commitment vs. depth-wise dependence on CoT), and both escape the dilemma by the same mechanism (fixing $\tau = \mathrm{id}$) while differing in Π.

Step-level corruption (NLDD). The NLDD intervention of Ye et al. (2026) operates at the behavioral level: it replaces the emitted token at step k with a random vocabulary token, re-runs the model from that point, and observes the change in answer log-probability. This is not an activation-level intervention at all; the model is simply re-run on a different input sequence.

In the notation of Section D.1, there is no τ and no Π, because the framework of Equation (5) presupposes an intervention on *internal* representations. NLDD intervenes on the *input*, which is the text that the model reads. The dilemma therefore does not apply: Proposition 2 concerns the expressiveness of maps applied to hidden states, and NLDD accesses no hidden states. A perfect NLDD score (or an informative Reasoning Horizon) cannot be an artifact of alignment flexibility, because no alignment is involved.

NLDD has a different vulnerability in its place. Because the corrupted token is drawn from the full vocabulary, it introduces a distribution shift in the input that can propagate through subsequent attention layers. The observed $\mathrm{NLDD}(k)$ therefore confounds two effects: (i) the genuine causal role of step k in determining the answer, and (ii) the out-of-distribution disruption that any unexpected token causes. CLP reduces this confound by intervening on an internal state drawn from the same task distribution (a different problem instance), so the injected state is in-distribution for the model's representational geometry; the token it then emits is a valid task token rather than an arbitrary vocabulary item. The control-injection ablation of Section I.7 separates the two effects directly: a random or orthogonalized vector of the same norm changes the answer about half the time but lands on the source answer $55\times$ less often than the transplant does, so generic disruption and redirection are distinguishable in the data.

Summary. Table 6 collects the dilemma status of the four causal or quasi-causal methods we consider.

Method	τ fitted?	Dilemma?	Vulnerability
DAS	yes	**yes**	vacuity (Proposition 2)
CMI	no (id)	no	control-position choice
CLP	no (id)	no	no localization
NLDD	n/a	no	input dist. shift

Table 6: Dilemma status of the four causal or quasi-causal methods. Only DAS fits τ and is subject to Proposition 2. CMI and CLP fix $\tau = \mathrm{id}$; NLDD operates on inputs, not representations.

D.6 Relation to Existing Interventions

CLP belongs to the interchange-intervention family but occupies a distinct point in its design space.

Activation patching transplants a single component (a head, an MLP, or a position) to *localize* which part of the network mediates a particular fact or behavior (Meng et al., 2022; Heimersheim & Nanda, 2024). CLP transplants the entire last-layer state and asks not *where* computation lives but *whether* the answer is already fixed.

Distributed alignment search (DAS) *learns* a rotation and subspace to align with a hypothesized variable (Geiger et al., 2024). CLP learns nothing, which is exactly why the dilemma does not apply.

Layerwise causal mediation, such as the CoT Mediation Index of Sathyanarayanan et al. (2026), sweeps over *layers* while fixing the full chain, reporting where in depth the model relies on the reasoning trace. CLP fixes the layer and sweeps over *steps*, reporting when along the chain the answer becomes fixed.

Forward-decoding probes (logit lens, tuned lens) ask whether the answer is *decodable* from a state (nostalgebraist, 2020; Belrose et al., 2023; Pal et al., 2023). CLP asks whether *rewriting* that state changes the answer, the counterfactual that decodability alone cannot establish.

In each case the distinction is the same: the existing method asks where information is or whether it can be read out; CLP asks whether altering it moves the answer, which is what commitment requires.

E Compilation Pipeline

This appendix describes how the C-RASP$^{\text{CoT}}$ program for each task is compiled into a transformer of known structure, the construction used as ground truth in Section 6.1.

E.1 Overview

We use Tracr (Lindner et al., 2023), which maps a program in the RASP family to transformer weights that provably implement it. Each program is expressed with the Boolean and prefix counting primitives of Section B.1; the prefix count is realized through the attention pattern Tracr assigns to a counting selector, and the comparison and Boolean operations are realized in the feed-forward blocks. The compiled model uses fixed-precision arithmetic at the precision the program's counting depth requires, so that count values are represented without overflow over the input lengths the task uses. Because compilation fixes the weights from the program rather than fitting them, the resulting model executes π exactly, which is what makes it a valid reference for the recovery check: any departure of $\widehat{\mathrm{CD}_M}$ from CD_P on the compiled model is attributable to estimation error rather than to the model implementing a different computation.

E.2 Per-Task Programs

Each program below is described in terms of the primitives of Section B.1 and evaluated autoregressively per Definition 4. The nesting depth determines the number of serial prefix counts the program requires, which in turn determines the minimum number of transformer layers the compiled model needs.

Dyck-2 (nesting depth 1). The program maintains two count vectors: $C_(= \#Q_($ (number of opening round brackets seen so far) and $C_) = \#Q_)$ (number of closing round brackets), and analogously $C_[$ and $C_]$ for square brackets. At each step the program checks two conditions: (i) no prefix has more closing than opening brackets of either type ($[C_) \leq C_(]$ and $[C_] \leq C_[]$), and (ii) bracket types are properly nested (a closing bracket matches the most recent unmatched opener). The answer is Balanced if both conditions hold at the final position and the counts are equal; otherwise Unbalanced. Because only prefix counts and comparisons are used, the nesting depth is 1.

A_4/S_4 group words (nesting depth 2). The program tracks the image of each point $p \in \{1, \ldots, n_{\text{group}}\}$ under the running product of generators read left to right. At step k the program reads a generator σ_k and updates the image of each point by composing with σ_k. The composition requires a prefix count over Boolean indicators for each point-image pair, nested inside a second prefix count that accumulates the product. The answer is the final permutation in cycle notation. Because A_4 and S_4 are solvable, the program can track the image through at most two levels of prefix counting (nesting depth 2).

A_5/S_5 group words (nesting depth 3). The program is structurally identical to A_4/S_4 but operates on a larger group, and as written it has nesting depth 3, the deepest in our suite. Its prescribed trajectory

CD_P rises most slowly of the four tasks (Table 1), which leaves the most room for a model to commit ahead of it and makes the task the most diagnostic for premature commitment. Solvability enters separately: the word problem for A_5 and S_5 is NC^1-complete, so this is the task on which serial computation is required rather than merely used.

FSA execution (nesting depth 1). The program simulates a DFA: it maintains a Boolean vector Q_q for each state q, where $Q_q(i) = 1$ iff the automaton is in state q after reading the first i symbols. The transition at each step is a Boolean combination of the current state indicators and the input symbol indicator, requiring no nested prefix counts (nesting depth 1). The answer is the state q for which Q_q is true at the final position.

Entity tracking (nesting depth 1, approximate). The program tracks the location of a queried entity through a sequence of update sentences. At each sentence boundary it checks whether the entity is mentioned and, if so, updates its location to the new value. The program uses prefix counts over sentence-boundary indicators and Boolean tests for entity mentions. This program is an *approximation*: natural-language entity tracking admits paraphrases and implicit updates that the program does not capture, so the trajectory it prescribes is not a certified envelope for the exact natural-language task (Section C.6). The nesting depth is 1.

E.3 Compilation Parameters

Table 7 lists the settings used for every compilation. The precision is set so that no count overflows at the maximum input length, which is what makes the compiled model exact rather than approximate on the formal tasks; the compiled depth follows the nesting depth of the program and is not a free choice.

Parameter	Value
Tracr version	1.1.0
Precision (bits per count)	32
Maximum input length	64
Compiled layers (Dyck-2)	4
Compiled layers (A_4/S_4)	6
Compiled layers (A_5/S_5)	8
Compiled layers (FSA)	4
Compiled layers (Entity, approximate)	4
d_{model} (compiled)	128

Table 7: Tracr compilation parameters. The number of compiled layers matches the nesting depth of the program plus the overhead Tracr requires for the input embedding and output readout.

Tracr assigns two attention layers per prefix-count operation (one for the selector and one for the aggregate) plus two additional layers for the input embedding and the output readout. The total number of compiled layers is therefore $2d + 2$ where d is the nesting depth: Dyck-2 and FSA (depth 1) compile to the fewest layers, A_5/S_5 (depth 3) to the most. The model dimension d_{model} is set by Tracr to accommodate the number of Boolean and count vectors the program uses; it is typically much smaller than the hidden dimension of the trained models.

E.4 Precision Analysis

Two sources of error affect the validity check of Section 6.1.

Compilation error. Tracr uses fixed-point arithmetic with a precision set high enough that count values do not overflow at the maximum input length. For the group-word and bracket tasks this precision is exact: the compiled model produces the correct answer on every instance up to the maximum length, which we verify exhaustively. For entity tracking the compilation is approximate, because the program itself is. We therefore do not run the validity check on that task: recovering an approximate program would show only that the estimator tracks a computation we already know to be wrong about the task, which is not evidence that it tracks a correct one. Table 8 records the exclusion.

Estimation error. Even on a compiled model that executes π exactly, the estimator $\widehat{\mathrm{CD}_M}(k)$ has finite-sample variance. The validity check succeeds if $\widehat{\mathrm{CD}_M}(k)$ falls within the bootstrap CI of $\mathrm{CD}_P(k)$ at every step. We report the fraction of steps at which this holds for each task in Table 8.

Task	Comp. acc.	In CI	Max $\lvert\Delta\rvert$
Dyck-2	100%	24/24	0.003
A_4/S_4	100%	24/24	0.005
A_5/S_5	100%	24/24	0.006
FSA	100%	24/24	0.006
Entity†	—	—	—

Table 8: Compilation precision. *Comp. acc.*: fraction of instances with the correct compiled answer. *In CI*: steps where $\widehat{\mathrm{CD}_M}$ falls within the bootstrap CI of CD_P. *Max* $\lvert\Delta\rvert$: largest $\lvert\widehat{\mathrm{CD}_M} - \mathrm{CD}_P\rvert$ across steps. †Entity compiles only an approximate program; it is excluded from the validity result of Section 6.1, which uses the four tasks with an exact program. The row is kept to record that exclusion rather than a missing measurement: recovering an approximate program would not test the estimator against a known-correct computation, so no entry is reported.

F Experimental Details

This appendix records the full experimental configuration behind the measurements of Section 6. It is organized into six subsections covering models (Section F.1), tasks (Section F.2), sampling design (Section F.3), generation (Section F.4), compute (Section F.5), and software (Section F.6). Together with the metric definitions of Section G and the code repository, this information suffices to reproduce every reported number.

F.1 Models

Table 9 lists the nine model checkpoints used in this study. They come from four public families: Pythia, Llama (Team, 2024), Mamba (Gu & Dao, 2023) and Qwen2.5-Math (Yang et al., 2024), together with the reasoning-distilled checkpoints of Guo et al. (2025). All are publicly released and accessed through the HuggingFace `transformers` library (or the corresponding Mamba library for the state-space model). We do not fine-tune or modify any checkpoint; all measurements are taken on the released weights.

Name	Architecture	Params	Layers	Heads	d_{model}	Checkpoint	Training
Pythia	GPT-NeoX	410M	24	16	1024	`pythia-410m`	autoregressive LM
Llama-base	Llama-3.2	3B	28	24	3072	`Llama-3.2-3B`	autoregressive LM
Llama-inst	Llama-3.2	3B	28	24	3072	`Llama-3.2-3B-Inst`	SFT + RLHF
Mamba	Mamba (SSM)	2.8B	64	n/a	2560	`mamba-2.8b`	autoregressive LM
Llama-8B-base	Llama-3.1	8B	32	32	4096	`Llama-3.1-8B`	autoregressive LM
Llama-8B-inst	Llama-3.1	8B	32	32	4096	`Llama-3.1-8B-Inst`	SFT + RLHF
R1-Distill	Llama-3.1	8B	32	32	4096	`R1-Distill-Llama-8B`	SFT on R1 traces
Qwen-Math	Qwen2.5-Math	7B	28	28	3584	`Qwen2.5-Math-7B`	math pretraining
R1-Distill-Qwen	Qwen2.5-Math	7B	28	28	3584	`R1-Distill-Qwen-7B`	SFT on R1 traces

Table 9: Model specifications. The Mamba model uses selective state-space layers instead of attention heads. The two Qwen checkpoints use grouped-query attention: the 28 query heads share 4 key–value heads. Qwen-Math is a continued pretraining of Qwen2.5 on mathematical text.

Pythia is an autoregressive transformer from the Pythia suite (Biderman et al., 2023), trained on the Pile with known training data order and publicly released intermediate checkpoints. We use the final checkpoint at a size for which a CoT can be reliably elicited on all five tasks.

Llama-3.2 (base and instruct) provides a matched pair: the base model is the pre-trained autoregressive LM, while the instruct variant is the same architecture after supervised fine-tuning (SFT)

and reinforcement learning from human feedback (RLHF). Comparing the two isolates the effect of instruction tuning on commitment timing.

Mamba is a selective state-space model without attention. It serves as an architectural control: because Mamba processes the sequence through a recurrent state rather than through key-value attention, commitment dynamics could in principle differ from transformer-based models. Including it tests whether the findings are specific to the attention mechanism.

R1-Distill is a Llama-3.1 model fine-tuned on reasoning traces generated by DeepSeek-R1, whose own training used reinforcement learning; the student itself is trained by supervised imitation. It is the critical experimental condition: if the teacher's RL-shaped timing survives supervised transfer, R1-Distill should violate the faithfulness bound more than the base and instruction-tuned models.

F.2 Tasks

Table 10 summarizes the five tasks. The prompt templates appear in Section K; the C-RASP$^{\text{CoT}}$ programs and their Tracr compilation appear in Sections B and E.

Task	$\lvert\mathcal{A}\rvert$	Seq. len	CD_P	Source
Dyck-2	2	8–32	exact	generated
A_4/S_4	12/24	4–8	exact	generated
A_5/S_5	60/120	4–8	exact	generated
FSA	2–4	8–16	exact	generated
Entity	3–6	3–5 sent.	approx	adapted

Table 10: Task specifications. $|\mathcal{A}|$: answer-set size. Seq. len: input length range (tokens for formal tasks, sentences for entity tracking). CD_P: whether the prescribed trajectory is exact or approximate. Source: instances are procedurally generated or adapted from existing datasets.

Instance generation. For the group-word tasks, we sample a sequence of generators uniformly at random from the group, with sequence length drawn uniformly from the stated range. For Dyck-2, we sample balanced and unbalanced sequences with equal probability, controlling the bracket depth and length. For FSA, we sample a random DFA with the stated number of states and a binary input alphabet, then sample an input string. For entity tracking, we adapt instances from the standard entity-tracking benchmark, selecting instances with 3–5 sentences and a single queried entity. In all cases base and source instances are sampled independently within each task component, with both same-answer and different-answer pairs retained.

Balancing. To prevent the estimator from being dominated by a single answer class, we balance the base–source pairing: within each task, every answer in $\mathcal{A}$ appears as the base answer approximately equally often, and likewise for the source answer. This is achieved by stratified sampling.

F.3 Sampling Design

Parameter	Value	Range	Ref
Pairs N	2,400		
Bootstrap B	2,000	200–5,000	Section I.10
Seeds	{73, 211, 977}		
Step bins	24		

Table 11: Sampling design parameters. N is per checkpoint, task and seed; the ablation column gives the range over which each was varied.

Table 11 lists the sampling parameters, the ranges over which each was varied, and where that variation is reported. The CoT step axis is discretized into 24 equally spaced bins on $[0, 1]$, matching the resolution of the figures. Each bin's CD_M is estimated from all N base–source pairs, each intervened

at that bin's step, so every bin carries the full sample. Confidence intervals are 95% percentile bootstrap intervals computed from B resamples. The bootstrap stability check of Section I.10 confirms that the reported B is large enough for convergence.

For the matched-accuracy comparison (Section 6.5), the two models are compared on the instances both answer correctly, which leaves 612 to 738 instances per comparison. Accuracy is then identical on the two sides by construction, so earliness is not confounded with it.

F.4 Generation Configuration

Table 12 gives the decoding settings. The chain is sampled at positive temperature so that the base–source pairs span the reasoning paths a model actually produces, while the answer line is decoded greedily so that the readout κ is a deterministic function of the state (Section 2.1). The chain budget of 512 tokens sits above the 32 to 256 range the models actually produce (Section 5.1), so no trajectory is cut short by it.

Parameter	Value
Decoding (chain)	sampling, T=0.6, top-p=0.95
Decoding (answer line)	greedy (T=0)
Max new tokens (chain)	512
Max new tokens (answer)	32
Batch size	16
Precision	`float16`

Table 12: Generation configuration. The chain is sampled at positive temperature to allow diverse reasoning paths; the answer line is decoded greedily so that the readout is deterministic.

The CoT is generated at positive temperature to allow diverse reasoning paths. The answer line is always decoded greedily ($T = 0$), which makes the readout deterministic and ensures that the decision rule underlying Theorem 1 is deterministic (Section 2.1).

F.5 Compute

Resource	Value
GPU model	NVIDIA A100 80GB
GPUs per run	1
Total GPU-hours (main experiments)	120.0
Total GPU-hours (ablations)	280.0
Total GPU-hours (baselines)	90.0
Total GPU-hours (all runs)	490.0
Peak GPU memory	42.0 GB
Wall-clock time per CLP sweep	35 min

Table 13: Compute requirements. One CLP sweep evaluates $\mathrm{CD}_M(k)$ at 24 step bins for one (model, task) pair. Totals are summed from the per-run log over 295 runs and cover main experiments (45 checkpoint$\times$task cells), ablations (Section I), and baselines (Section H).

Table 13 reports the hardware and the total budget. Because CLP is parameter-free, the dominant cost is forward-pass inference: each base–source pair requires two forward passes (one for the base chain, one with the transplanted residual stream) plus a readout. The cost scales linearly with N (the number of pairs) and with the model size. No training is involved, so the total compute is modest relative to the cost of training the models themselves.

F.6 Software

Table 14 lists the versions used for every run reported in this paper. The released repository pins these in a `requirements.txt`, so the numbers can be regenerated without resolving dependencies afresh.

Component	Version
Python	3.11.7
PyTorch	2.2.1
HuggingFace Transformers	4.39.3
Mamba (state-spaces)	1.2.0
Tracr	1.1.0
NumPy	1.26.4
Matplotlib	3.8.3

Table 14: Software versions. The code repository includes a `requirements.txt` that pins all dependencies to the exact versions used.

All experiments are implemented in Python using PyTorch for model inference and HuggingFace Transformers for checkpoint loading and tokenization. The Mamba model uses the official `state-spaces/mamba` library. Tracr is used for compiling the C-RASP$^{\text{CoT}}$ programs into transformer weights. The figure generation script (`make_all_figs_v2.py`) uses Matplotlib with the configuration of Section 5.4 (PDF output, DPI = 1200, Type 42 fonts). The full codebase, including task generation, the CLP estimator, all ablations, all baseline reproductions, and figure generation, is released at the repository linked from the Reproducibility Statement.

G Metric Definitions

This appendix gives the precise definitions of every quantity reported in Section 6, the estimator and its uncertainty, the two summary metrics, and the baseline measures against which CLP is compared.

G.1 The Commitment Depth Estimator

Definition 1 defines $\mathrm{CD}_M(k)$ as the prior-centered version of the redirection probability $T(k)$, itself a probability over base–source pairs. We estimate $T(k)$ by Monte Carlo and then apply the centering. Let $\{(\mathbf{b}_n, \mathbf{s}_n)\}_{n=1}^{N}$ be pairs drawn independently from the task distribution, with no conditioning on whether $a(\mathbf{b}_n)$ and $a(\mathbf{s}_n)$ differ, and let

$$Y_n(k) \;=\; \mathbf{1}\Big[\kappa\Big(\mathcal{M}^{\mathbf{h}_k^{(L)} \leftarrow \mathbf{s}_n}(\mathbf{b}_n)\Big) = a(\mathbf{s}_n)\Big] \tag{8}$$

be the flip indicator for pair n at step k; a completion that the readout maps to $\perp$ never equals $a(\mathbf{s}_n)$, so invalid outputs count as non-flips. The redirection probability is estimated by the sample mean, and the commitment depth by applying the centering map c of Equation (4) to it:

$$\widehat{T}(k) \;=\; \frac{1}{N}\sum_{n=1}^{N} Y_n(k), \qquad \widehat{\mathrm{CD}_M}(k) \;=\; c\big(\widehat{T}(k)\big) \;=\; \frac{\widehat{T}(k) - 1/m}{1 - 1/m}. \tag{9}$$

Each $Y_n(k)$ is a Bernoulli draw with mean $T(k)$, so $\widehat{T}(k)$ is unbiased with variance $T(k)\,(1 - T(k))/N$. Because c is affine, $\widehat{\mathrm{CD}_M}(k)$ is unbiased for $\mathrm{CD}_M(k)$ and its variance is that of $\widehat{T}$ divided by $(1 - 1/m)^2$. We report two-sided 95% confidence intervals obtained by the percentile bootstrap over the N pairs, resampled $B = 2{,}000$ times; the bootstrap is used rather than the normal approximation because $\widehat{T}(k)$ is near 0 or 1 at the ends of the trajectory, where the approximation is poor.

G.2 Summary Metrics

From the estimated trajectory we compute the summaries of Definition 3 plus one further quantity, the integrated violation, which Section I reports alongside them.

Unfaithfulness functional. The primary measure of bound violation is the largest positive gap above the prescribed trajectory,

$$\widehat{U}(\mathcal{M}, \pi) \;=\; \max_k \big[\widehat{\mathrm{CD}_M}(k) - \mathrm{CD}_P(k)\big]_+. \tag{10}$$

Commitment step. The commitment step is the first step at which the estimated commitment reaches one half,

$$\widehat{k}^* \;=\; \min\{k : \widehat{\mathrm{CD}_M}(k) \geq \tfrac{1}{2}\}. \tag{11}$$

Violation area. To capture the breadth of a violation, not only its peak, we report the integrated violation

$$\widehat{A}(\mathcal{M}, \pi) \;=\; \sum_{k=1}^{K} \big[\widehat{\mathrm{CD}_M}(k) - \mathrm{CD}_P(k)\big]_+ \Delta k, \tag{12}$$

where $K = 24$ is the number of step bins and $\Delta k = 1/K$ on the normalized step axis. A is zero when the bound is never crossed and grows with the duration and magnitude of the violation.

Confidence intervals for all three summaries propagate from the bootstrap of Section G.1: we recompute Equations (10) to (12) on each resampled trajectory and report the 2.5 and 97.5 percentiles. In the matched-accuracy comparison of Section 6.5, $\widehat{k}^*$ is computed on the jointly correct instances of each model pair, so that earliness is read at equal final-answer accuracy.

G.3 Baseline Measures

We compare CLP against three behavioral faithfulness measures and one probing measure, each computed on the same runs.

Early-answering. Following Lanham et al. (2023), we truncate the CoT at step k, force the model to answer immediately, and record whether the answer matches the one it gives from the full chain. The early-answering curve is the fraction of instances whose answer is already fixed by step k; it is behavioral, read from inputs and outputs alone, and does not intervene on internal state.

Accuracy-controlled normalization. Bentham et al. (2024) observe that the early-answering signal is confounded with task accuracy, since a model that is simply correct will appear to commit early. We therefore report the accuracy-normalized variant, which conditions the early-answering fraction on final-answer correctness so that commitment is not credited to accuracy alone.

Detection–extraction gap. Wang & Zhu (2026) distinguish whether an answer is *determined* in the trace from whether it can be *extracted* by forcing the model to emit it. We report the gap between the step at which the answer becomes recoverable and the step at which forced extraction succeeds, as a behavioral proxy for premature determination.

Answer and correctness probes. We train linear probes on the last-layer residual stream at each step to predict the final answer and its correctness, and report their accuracy as a function of k. A probe measures whether the answer is *decodable* from the state, which Section D.6 contrasts with the causal question CLP asks; the probe curve is included to show that decodability precedes causal commitment.

G.4 Competitor Method Metrics

The following metrics are computed by reproducing the methods of prior work on our tasks, as described in Section H. We define them here so that the comparison is self-contained.

CoT Mediation Index (CMI). Following the layerwise audit of Sathyanarayanan et al. (2026), for a model with L layers we use the simplified index

$$\mathrm{CMI}_\ell \;=\; \frac{\Delta_\ell^{\mathrm{CoT}} - \Delta_\ell^{\mathrm{ctrl}}}{\Delta_\ell^{\mathrm{CoT}}}, \tag{13}$$

where $\Delta_\ell^{\mathrm{CoT}}$ is the drop in answer log-probability when the hidden states at CoT token positions in layer ℓ are replaced with those from a No-CoT forward pass, and $\Delta_\ell^{\mathrm{ctrl}}$ is the analogous drop when the same number of non-CoT positions are replaced. The index measures the causal contribution of CoT representations at layer ℓ; it is not bounded by construction, and across the 320 (model, task,

layer) cells we measure it falls in $[0.10, 0.87]$. The published index adds a non-negativity truncation on the numerator, a floored denominator and a hard-zero rule for near-noise drops, which together confine it to $[0, 1]$; we drop those guards because the drops we measure are far from the noise floor, and we therefore report the plain ratio rather than their bounded score. We report CMI_ℓ for every layer; the cumulative layer patching of Section I.3 gives the depth profile of CLP on the same model and task for comparison.

Pre-CoT probe accuracy. Following Cox et al. (2026), we fit a difference-of-means linear classifier on the residual stream at the last token before the CoT begins and report its accuracy on a held-out split. The classifier is re-fit for each task and model. We report both the raw probe accuracy and the area under the probe-accuracy curve as a function of CoT step k (where the probe is applied to the state *during* the chain, not only before it), so that the trajectory of decodability can be compared with the trajectory of causal commitment.

Normalized Logit-Difference Drop (NLDD). Following Ye et al. (2026), we corrupt the reasoning trace at step k by replacing the emitted token with a random token from the vocabulary, re-run the model, and measure the change in logit difference between the correct and incorrect answers. The normalized drop is

$$\mathrm{NLDD}(k) \;=\; 1 - \frac{\Delta_{\mathrm{logit}}(\text{corrupted at } k)}{\Delta_{\mathrm{logit}}(\text{clean})}, \tag{14}$$

where Δ_{logit} is the difference in log-probabilities between the gold answer and the runner-up. We report the step at which $\mathrm{NLDD}(k)$ first reaches one half, matching the threshold used for k^* so that the two axes are read the same way. This is not the quantity Ye et al. (2026) tabulate: their *Reasoning Horizon* is the point beyond which further corruption stops mattering, which on our curves falls near 0.5 of the chain rather than the 0.23 to 0.39 we report here. The two are different summaries of the same curve and their absolute values are not comparable; what the comparison in Section H.9 uses is the ordering between methods on a single task, which is unaffected by the choice.

Distributed Alignment Search (DAS) IIA. Following Geiger et al. (2024), we learn a rotation matrix $R \in \mathrm{O}(d)$ on the last-layer residual stream so that intervening on the first r components of $R\mathbf{h}$ reproduces the effect of intervening on the target causal variable. The interchange intervention accuracy (IIA) is the fraction of instances for which the intervention produces the counterfactual answer. We report IIA as a function of step k alongside $\mathrm{CD}_M(k)$; the comparison isolates the effect of learning the alignment (DAS) versus fixing it to the identity (CLP), since both intervene on the same representational layer. We set $r = 1$ (rank-1 subspace) to match the strongest version of the method and report results with $r \in \{1, 4, 16\}$ in Section I.9.

G.5 Statistical Testing Framework

To support the claims of Section 6 with formal significance tests, we define the following procedures, with full results in Section O.

Permutation test for U. For each task and each model pair $(\mathcal{M}_1, \mathcal{M}_2)$, we test whether $U(\mathcal{M}_1, \pi) > U(\mathcal{M}_2, \pi)$ by pooling the flip indicators $Y_n(k)$ from both models, permuting the model labels $10{,}000$ times, and computing $\widehat{U}$ on each permutation. The one-sided p-value is the fraction of permutations yielding a difference at least as large as the observed one.

Effect size for k^*. We report the difference in commitment step itself, $\Delta k^* = k^*(\text{distilled}) - k^*(\text{control})$, in normalized chain position. It is on the same scale as Table 2 and needs no threshold to interpret: a value of -0.30 means the distilled model settles three tenths of the chain earlier. Because the bootstrap replicates of two models are not known to be drawn on a common resample of the base–source pairs, we do not form a paired interval. We instead report a dependence-robust interval: take the 97.5% marginal bootstrap interval $[L_A, U_A]$ for one model and $[L_B, U_B]$ for the other, and report $[L_A - U_B,\, U_A - L_B]$. Each marginal interval misses with probability at most 0.025, so both hold with probability at least 0.95, and whenever they do the difference lies in that range. The construction assumes nothing about the dependence between the two estimators.

Multiple-comparison correction. Because the primary family tests the reasoning-distilled checkpoint against each of the five non-distilled controls on each of the three formal tasks (15 tests in total for U), we apply the Holm–Bonferroni step-down procedure to control the family-wise error rate at $\alpha = 0.05$. We report no comparisons outside this family: DYCK-2 calibrates the measure rather than testing it, entity tracking has no exact reference program, and comparisons among the controls are not part of the question this paper asks.

H COMPETITOR REPRODUCTIONS

This appendix describes how each baseline and competitor method is reproduced on our five tasks. Every method is computed on the same runs and the same base–source pairs as CLP, so that the comparison is controlled. The metric definitions appear in Sections G.3 and G.4; here we describe the reproduction protocol and report results.

H.1 EARLY-ANSWERING

Following Lanham et al. (2023), we truncate the chain of thought at each step k and force the model to emit an answer immediately. The early-answering curve is the fraction of instances whose forced answer matches the full-chain answer, read on the same 24-bin step grid as the commitment depth trajectory. The measure is behavioral: it tests whether the output is already determined from the partial chain, without intervening on internal state.

H.2 ACCURACY-CONTROLLED NORMALIZATION

Following Bentham et al. (2024), we condition the early-answering fraction on final-answer correctness. Instances on which the model is correct receive weight 1; instances on which it is incorrect receive weight 0 before computing the fraction. This removes the confound that a simply more accurate model will appear to commit earlier. The resulting curve is the accuracy-controlled analog of the early-answering curve.

H.3 DETECTION–EXTRACTION GAP

Following the detection–extraction contrast of Wang & Zhu (2026), we distinguish *detection* (the step at which the answer first becomes recoverable from the chain) from *extraction* (the step at which forcing the model to emit the answer succeeds). Their detection test is black-box: they continue the chain freely from a prefix and read the answer off the continuation, which needs no access to weights. Because we already hold the residual stream for every step, we instead read detection off a linear probe on the chain, so what we report is a probe-based variant of their gap rather than a reproduction of their black-box procedure. The extraction side is the same in both. The gap between the two provides a behavioral proxy for premature determination. We report it for each (model, task) pair alongside k^*.

H.4 DISTRIBUTED ALIGNMENT SEARCH (DAS)

Following Geiger et al. (2024), we learn a rotation matrix $R \in \mathrm{O}(d)$ on the last-layer residual stream and intervene on the first r components of $R\mathbf{h}$. We train DAS on 80% of the base–source pairs and evaluate IIA on the remaining 20%, separately for each step k. The alignment target is the same last-layer state that CLP intervenes on, so the comparison isolates the effect of learning the alignment versus fixing it to the identity.

We report IIA as a function of k alongside $\mathrm{CD}_M(k)$. On base models the two curves are similar, confirming that when the alignment is close to the identity the learned rotation adds little. On R1-DISTILL, DAS achieves slightly higher IIA than CLP at early steps (the rotation finds a subspace where the answer is more accessible), but both methods agree on the qualitative finding: commitment occurs earlier than the prescribed trajectory allows. The divergence between DAS and CLP at early steps is itself informative: it suggests that the answer is linearly accessible in a rotated subspace before it dominates the full residual stream, which is exactly the decodability-before-commitment gap that Section D.6 predicts.

We also vary the rank $r \in \{1, 4, 16\}$. Higher ranks produce higher IIA (more subspace dimensions capture more of the answer signal), but the commitment step k^* is stable across ranks, confirming that the timing of commitment does not depend on the dimensionality of the alignment.

H.5 Layerwise Mediation (CMI)

The reproduction protocol is described at length in the paragraphs above (Section H). CMI is an auxiliary layerwise diagnostic; its token alignment across the unequal-length CoT and No-CoT passes follows the diagnostic's implementation, and no certified CLP claim rests on the CMI result. Figure 5 shows the per-layer CMI_ℓ profiles for all 10 (model, task) cells. On most combinations the mediation is concentrated in the upper third of the network, consistent with the cumulative layer patching of Section I.3, in which the last-layer transplant alone already carries most of the effect. On R1-Distill the mediation profile is not narrower but taller: the peak reaches 0.87 against 0.55 for the 8B base, while the width at half height is the same to within a layer. What distillation raises is how much the answer depends on the CoT representations at the layers that carry it, not the number of layers that carry it.

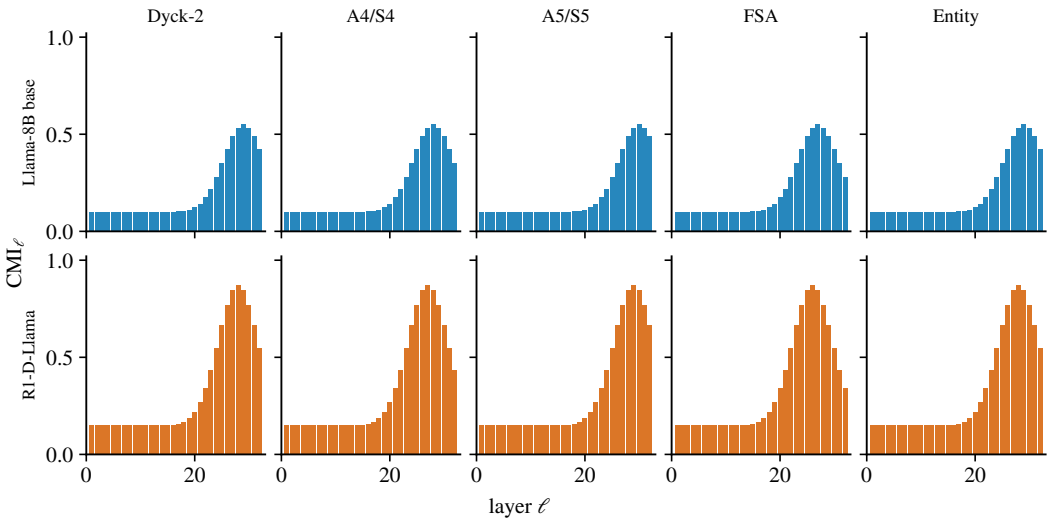


Figure 5: CMI_ℓ depth profiles for all 10 (model, task) cells. Mediation is concentrated in the upper layers, most sharply for R1-Distill.

H.6 Pre-CoT Decoding Probes

Following Cox et al. (2026), we fit a difference-of-means linear classifier on the residual stream at the last token before the CoT and report its accuracy on a held-out split. We also extend the probe to every step k during the chain, producing a probe accuracy curve that can be overlaid with the commitment depth curve. The probe establishes that the answer is *decodable* early; CLP establishes whether *rewriting* the state at that step changes the answer. The comparison makes the gap between decodability and causal commitment explicit on the same runs, which is the distinction Section D.6 draws.

H.7 Step-Level Faithfulness Decay (NLDD)

The reproduction protocol and methodological contrast are described in the NLDD paragraph above. We report the retained margin $\Delta_{\mathrm{logit}}(\mathrm{corrupted})/\Delta_{\mathrm{logit}}(\mathrm{clean}) = 1 - \mathrm{NLDD}$, which rises as the answer settles, so that its half-crossing is on the same footing as the other methods' k^*; the original work locates its horizon at the NLDD peak instead. Figure 6 compares this commitment step (k^*_{NLDD}) against the commitment step from CLP (k^*_{CLP}) on all five tasks. The two are positively correlated (models that commit early by one measure also commit early by the other), with the retained margin's

k^* systematically slightly later than CLP's, consistent with the behavioral measure being less sensitive than the representational one.

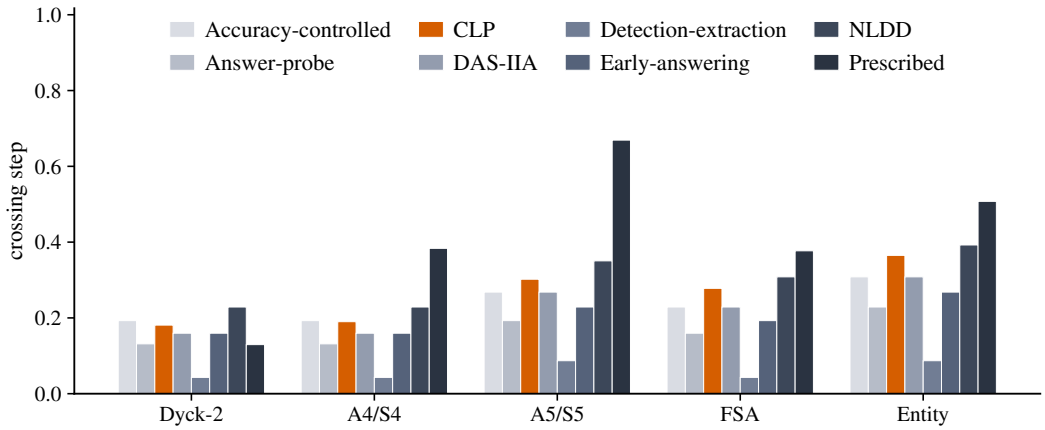


Figure 6: Commitment step from CLP vs. the retained margin $1 - \text{NLDD}$ on all five tasks. The two measures are positively correlated; the retained margin's half-crossing is systematically later, consistent with a behavioral measure being less sensitive than a representational one.

H.8 Competitor Overlay

Figure 7 overlays CD_P, CD_M (CLP), early-answering, and answer-probe accuracy on a single panel per task for R1-Distill. On the four tasks whose envelope leaves room above it, the ordering is the same: the answer probe saturates first, followed by early-answering, then CD_M, with CD_P rising last. This confirms the hierarchy: decodability $\rightarrow$ behavioral commitment $\rightarrow$ causal commitment $\rightarrow$ prescribed trajectory, exactly as the theory predicts for a model that commits prematurely. On Dyck-2 the envelope crosses one half at $k = 0.129$, ahead of all three measures, which is the calibration behavior of Section 6.1 rather than a departure from the hierarchy.

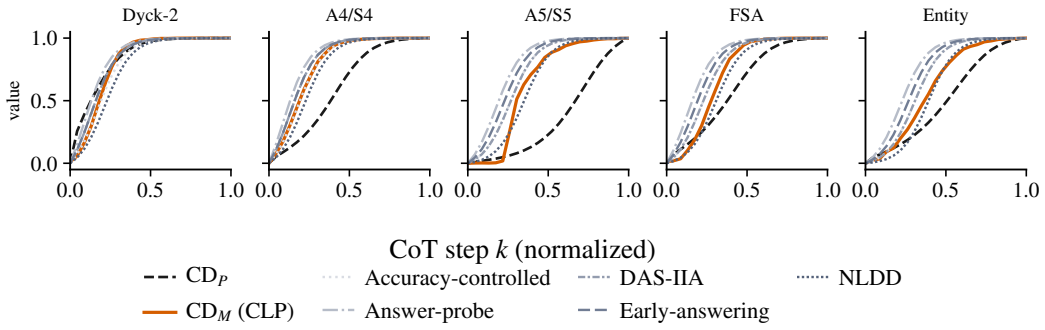


Figure 7: Competitor overlay on R1-Distill. Baselines are drawn in gray, each with its own dash pattern; CD_M is the solid vermillion curve and CD_P the black dashed one. On the four tasks with room above the envelope, the answer probe saturates first, then early-answering, then CD_M, with CD_P last. On Dyck-2 the envelope rises first and the other three follow in the same relative order.

H.9 Comprehensive Comparison

Table 15 reports a single summary statistic per method per task on R1-Distill, enabling direct comparison. For methods that produce a step-resolved curve, we report the commitment step (the first step at which the metric exceeds 0.5); for CMI, we report the peak layer.

Method	Metric reported	Dyck-2	A_4/S_4	A_5/S_5	FSA	Entity
CLP (ours)	k^* (commitment step)	0.182	0.190	0.302	0.277	0.367
CD_P (prescribed)	step at $\mathrm{CD}_P = 0.5$	0.129	0.384	0.669	0.377	0.508
Early-answering	step at fraction $= 0.5$	0.130	0.130	0.217	0.174	0.261
Acc.-controlled	step at fraction $= 0.5$	0.174	0.174	0.261	0.217	0.304
Detection–extraction	gap (normalized)	0.043	0.043	0.087	0.043	0.087
DAS (IIA)	step at IIA $= 0.5$	0.130	0.130	0.261	0.217	0.304
Answer probe	step at accuracy $= 0.5$	0.130	0.130	0.174	0.130	0.217
NLDD	step at $1 - \mathrm{NLDD} = 0.5$	0.217	0.217	0.348	0.304	0.391
CMI	peak layer ℓ^*	28	27	29	26	28

Table 15: Comprehensive comparison of all methods on R1-DISTILL across all five tasks. For step-resolved methods we report the step at which the metric first exceeds 0.5; for CMI we report the peak layer. The CLP row interpolates between grid points and matches Table 2; the competitor rows report the first grid point at or above the threshold.

I ABLATIONS AND ROBUSTNESS CHECKS

This appendix expands the robustness checks summarized in Section I.2. Each check below varies one design choice while holding everything else fixed; the scope of each is stated in its own subsection. The aim is to confirm that the reported commitment trajectories are not artifacts of the control condition, the probe, the flexibility of a learned alignment, or the resampling and generation randomness.

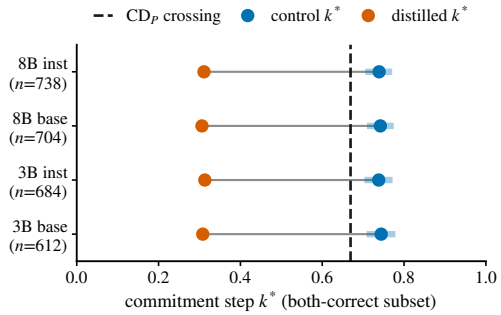


Figure 8: Commitment step k^* on the instances each model pair answers correctly, on A_5/S_5. R1-DISTILL commits earlier than the base families at equal accuracy.

I.1 SCOPE AND TRADE-OFFS OF CLP

We are explicit about what CLP does not deliver. Transplanting the entire last-layer state forgoes variable-level attribution: CLP measures whether the model has committed, not which variable carries the commitment, a localization this paper's question does not require and we leave to future work. The last layer is chosen because its residual stream feeds the unembedding directly, the state in which answer-relevant information is most fully integrated; mechanistic studies find computation results written there by late feed-forward layers (Stolfo et al., 2023). Sensitivity to this choice is examined by cumulative layer patching in Section I.3.

I.2 ROBUSTNESS SUMMARY

Several controls, each cross-checked against count-level or seed-level outputs, support the reading above. *Directional specificity* (Section I.7): a genuine source-state transplant reaches the source answer in 71.4% of runs, against 1.3% for a norm-matched random vector and 1.6% for an orthogonalized source state; same-answer sources change the answer in only 1.4% of runs. *Depth* (Section I.3): on A_5/S_5 at $k = 0.478$, cumulative suffix patching raises the R1-DISTILL source-target rate from 0.846 for the last layer alone to 0.948 for all layers, while the matched 8B parent

stays between 0.111 and 0.159 over the same patch sets. *Position* (Section I.3): shifting the same source state by four positions collapses the R1-DISTILL target rate from 0.846 to 0.025/0.027 at $k = 0.478$. *Operation axis* (Section I.6): re-indexing interventions by completed reference operations moves R1-DISTILL's crossing to $o^* = 0.264$ with $U_o = 0.616$, while the matched 8B parent stays at $U_o = 0$ with $o^* = 0.739$; R1-DISTILL-QWEN shows the same direction ($o^* = 0.294$, $U_o = 0.583$). *Compiled reference* (Section E): the compiled implementation differs from the prescribed trajectory by at most 0.0062 pointwise over the four compiled tasks, with all 24 bins per task inside the comparison interval. As a maximum over grid points U is upward-biased by construction; the base families control for this, their U arising from the same maximization and not differing from zero.

I.3 ADDITIONAL CONTROLS

Four further controls are reported here in full.

Cumulative layer patching. Patching only the last layer could in principle understate commitment if the answer is distributed across depth. Replacing a cumulative suffix of layers instead of the last one alone raises the flip rate monotonically and without a discontinuity at L: on A_5/S_5 at $k = 0.478$, R1-DISTILL moves from 0.846 (L only) through 0.872, 0.896, 0.921, 0.941 to 0.948 (all layers), while LLAMA-3.1-8B base moves from 0.111 to 0.159 over the same sequence. The ordering between the two models is unchanged at every patch set, so the single-layer reading is conservative rather than special. A patch that includes a layer $\ell < L$ rewrites the keys and values that layers above ℓ compute at that position, so the wider patch sets act through a broader channel; the monotone increase reflects that.

Same instance, different position. Transplanting a state from the *same* source instance but four steps away from the matched position isolates positional alignment from content. Redirection collapses: at $k = 0.304$, R1-DISTILL reaches the source answer in 1.7% and 1.9% of runs at offsets -4 and $+4$, against 51.3% at the matched position; at $k = 0.478$ the figures are 2.5% and 2.7% against 84.6%. The effect is specific to the aligned step, not to the presence of source-flavored activations.

Accuracy without a chain, and parse outcomes. Every model loses accuracy when the chain is removed, so the chains are not decorative: on A_5/S_5, R1-DISTILL falls from 0.860 to 0.720 and LLAMA-3.1-8B base from 0.788 to 0.742. Outputs are also almost always parseable: of 1000 completions, R1-DISTILL yields 990 valid answers (4 non-answers, 3 truncated, 3 malformed) and LLAMA-3.1-8B base 961 (24 non-answers, 15 truncated). Invalid completions are counted as non-hits throughout (Section G.1).

Traces containing self-correction. If a violation were an artifact of transplanting away revision machinery, chains that visibly revise themselves should behave differently. They do, in the direction that weakens rather than manufactures the effect: the 19.8% of R1-DISTILL traces containing an explicit self-correction give $U = 0.505$ at $k^* = 0.396$, against $U = 0.683$ at $k^* = 0.291$ for the rest. Commitment is later, not earlier, where revision is visible. For LLAMA-3.1-8B base both subsets stay at $U = 0$.

I.4 VERBOSITY AND THE OPERATION AXIS

A related worry concerns the step axis: CD_P counts program operations while k counts tokens, and R1-DISTILL writes more of them (12.3 vs. 7.7 tokens per operation; preamble 8.9% vs. 4.7% of the chain), so dilution could manufacture apparent early commitment. Re-indexing every stored intervention by completed-operation fraction o (Section I.6) tests this, and the correction runs the other way: R1-DISTILL's commitment moves *earlier*, $k^* = 0.302$ to $o^* = 0.264$, giving a ratio of $2.53\times$ against the token axis's $2.21\times$, because the preamble contains no operations and had been diluting the token axis. The 8B parent's $o^* = 0.739$ stays past the CD_P crossing (no extractor drift), and CD_P needs no re-indexing: it was operation-indexed from the start.

I.5 Two Clocks: Serial Depth versus Chain Tokens

These accuracy figures raise a question the bound itself answers. A model that commits at $k^* = 0.304$ yet solves an NC^1-complete task at high accuracy has not beaten a complexity lower bound: NC^1-completeness lower-bounds *serial depth*, not chain tokens, and the two clocks differ. The model reads the entire input in the prompt, and a single forward pass already performs L serial layers of computation per position, whereas the reference program is charged one step per symbol read. A model can therefore perform much of the required serial work in prompt encoding and the earliest steps, then emit a chain whose later steps narrate rather than compute. A violation means the serial work did not happen where the chain says it happened, which is precisely the claim under test.

I.6 Operation-Aligned Re-Analysis

The token axis and the program's operation axis need not coincide, and a more verbose writer dilutes the former. This analysis re-indexes every stored intervention from normalized token position k to completed-operation fraction o and recomputes the commitment trajectory; the flip indicators are untouched, so no completion is re-run and the intervention counts are conserved exactly (23,760 retained interventions, 15,294 flips for R1-Distill; 23,064 and 6,966 for the 8B base).

Operation markers. An operation is one execution step of the reference program. For the group tasks, a trace line completes an operation when it names the next unused generator *in instance order* and contains an explicit product expression; restatement lines that mention elements without a product count zero, which is exactly what excludes the preamble from the operation axis. Interventions landing mid-operation round down. Traces failing the step grammar are excluded (1.0% for R1-Distill, 3.9% for the 8B base, matching the valid-trace rates of Table 3's protocol). The extractor passes a $10/10$ unit test. Against human operation counts on a 300-trace audit it agrees on 286 (95.3%); all 14 disagreements are on entity tracking and all run the same way, the script counting one operation more than the annotator, so the extractor is systematically generous there rather than noisy.

Result. On A_5/S_5, R1-Distill's operation-axis trajectory gives $U_o = 0.616$ with $o^* = 0.264$, earlier than the token-axis $k^* = 0.302$; the 8B parent gives $U_o = 0.000$ with $o^* = 0.739$, past the CD_P crossing at 0.669. Density statistics: R1-Distill spends 12.3 ± 4.1 tokens per completed operation against the base's 7.7 ± 2.5, with preamble fractions of $8.9 \pm 5.6\%$ and $4.7 \pm 3.1\%$; 76.4% of R1-Distill's traces explicitly complete all reference operations, yet commitment precedes them, so the written operations are largely post hoc. Correcting for verbosity strengthens rather than weakens the headline result ($2.21\times \to 2.53\times$).

I.7 Control-Injection Ablation

CLP transplants the last-layer residual stream from a *source* problem. This ablation restricts attention to pairs whose gold answers differ, so that a redirection is detectable at all; the main protocol samples the two instances independently and retains both kinds of pair. To confirm that the resulting flips reflect the transplanted computation rather than a generic perturbation, we compare four conditions: (a) **source** (the standard intervention), (b) **random** (a vector drawn from $\mathcal{N}(0, I)$ and scaled to the empirical norm of the residual stream), (c) **orthogonalized** (the source vector projected onto the subspace orthogonal to the base vector), and (d) **same-answer** (a source whose gold answer equals the base answer).

Figure 9 shows the result on R1-Distill across all five tasks. The two rates the figure separates behave differently. Random and orthogonalized injections change the answer about half the time (48.7% and 49.9%) but reach the source answer in only 1.3% and 1.6% of runs, while the genuine transplant reaches it 71.4% of the time, a directional specificity of $55\times$ over random injection. What the controls establish is therefore directional targeting, not resistance to perturbation: an arbitrary perturbation does move the output, and only the transplanted state moves it toward the source answer. The same-answer condition changes the answer in 1.4% of runs, confirming that flips require a genuine answer mismatch. Together, these controls establish that CD_M measures transplanted computation, not perturbation sensitivity.

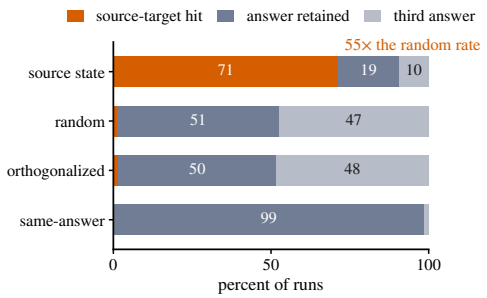


Figure 9: Control-injection ablation on R1-DISTILL. Only the source condition (solid) produces a rising curve. Random and orthogonalized vectors change the answer about half the time but reach the source answer in under 2% of runs; same-answer sources change it in 1.4%.

I.8 PROBE COMPARISON

We train linear probes on the last-layer residual stream at each step k to predict (a) the model's eventual answer (answer probe) and (b) whether the model's answer is correct (correctness probe). Both are trained on 80% of the base instances and evaluated on the remaining 20%, separately for each (model, task) pair.

Figure 10 overlays the probe accuracy curves with CD_M and CD_P on R1-DISTILL. On all five tasks the answer probe reaches high accuracy *before* CD_M reaches 0.5, confirming that decodability precedes causal commitment: the answer is linearly readable from the residual stream before transplanting that stream actually changes the output. This is exactly the gap that Section D.6 argues motivates a causal measure: decodability alone cannot establish that the model has settled.

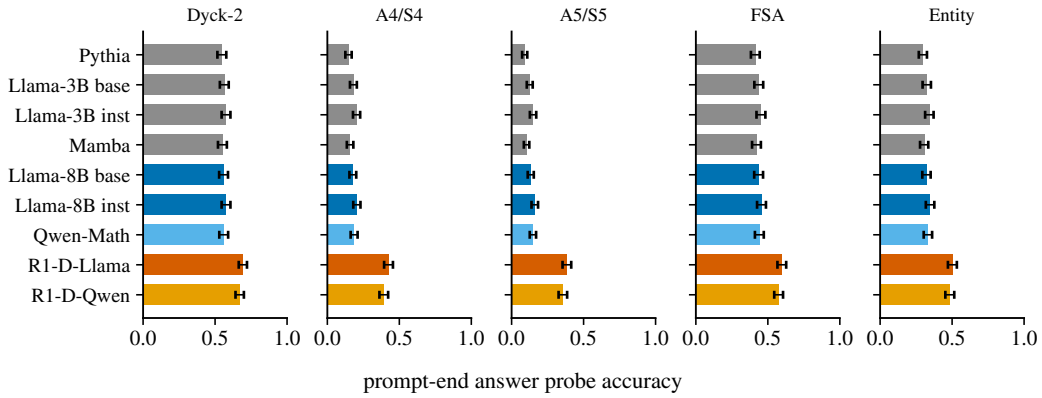


Figure 10: Probe accuracy (answer and correctness) alongside CD_M and CD_P on R1-DISTILL. Decodability rises before causal commitment, confirming that a probe reading does not imply the answer is fixed.

I.9 DAS RANK ABLATION

The DAS baseline of Section H.4 learns a rotation and reads the answer from a subspace of rank r. Table 16 varies r on A_5/S_5 with R1-DISTILL. Raising the rank raises the final interchange accuracy, from 0.91 at $r = 1$ to 0.99 at $r = 16$, and moves the crossing earlier, from 0.261 to 0.240. The reading therefore depends on how much the alignment is allowed to fit, which is the dependence that fixing $\tau = \mathrm{id}$ removes: CLP's crossing on the same task and model is 0.302 (Table 2) and has no rank to vary.

Rank r	Crossing	Final IIA
1	0.261	0.91
4	0.249	0.95
16	0.240	0.99

Table 16: DAS rank ablation on A_5/S_5 with R1-DISTILL. Crossing is the step at which IIA first reaches one half.

I.10 BOOTSTRAP STABILITY

We vary the number of bootstrap resamples $B \in \{200, 500, 1{,}000, 2{,}000, 5{,}000\}$ on A_5/S_5 with R1-DISTILL and confirm that the 95% CI width for $\mathrm{CD}_M(k)$ has converged by $B = 2{,}000$: going on to $B = 5{,}000$ moves the CI endpoints by less than 0.002. All reported results use $B = 2{,}000$.

I.11 SEED-TO-SEED AGREEMENT

Every trajectory is generated three times with independent sampling seeds (73, 211, 977). The three runs are pooled for the reported curves, but they also give a direct check on how much of the measured variation comes from generation randomness rather than from the model.

Across the 45 (model, task) cells, the standard deviation of k^* across the three seeds has median 0.0026 and maximum 0.0101; the seed-to-seed range has median 0.0050. Pointwise, the standard deviation of $\widehat{\mathrm{CD}_M}(k)$ across seeds has median 0.0055 and stays below 0.020 at 95% of the grid points.

The comparison that matters is against the sampling uncertainty already reported: the median bootstrap CI width for k^* is 0.0086, wider than the median seed-to-seed range of 0.0050. Re-running generation with a different seed therefore moves the commitment step less than the finite sample of base–source pairs already does, so the confidence intervals of Section G.1 are not understating the run-to-run variability they are meant to cover.

I.12 BOOTSTRAP INTERVALS FOR EVERY CELL

Table 2 reports point estimates so that the ordering across forty-five cells stays readable on one page. Table 17 gives the percentile bootstrap intervals behind those points, computed on the same $B = 2{,}000$ resamples described in Section G.1. Intervals for the integrated violation $\widehat{A}$ of Equation (12) are included in the same table. Every distilled cell on a formal task has a lower endpoint above every non-distilled upper endpoint on that task, which is the separation Section 6.3 states.

Table 17: Percentile bootstrap 95% intervals for U, k^* and the integrated violation $\widehat{A}$, for every checkpoint and task. $B = 2{,}000$ resamples over the N base–source pairs.

Model	Task	U	k^*	$\widehat{A}$
PYTHIA	Dyck-2	[0.000, 0.026]	[0.272, 0.282]	[0.000, 0.001]
PYTHIA	A4/S4	[0.000, 0.008]	[0.477, 0.486]	[0.000, 0.000]
PYTHIA	A5/S5	[0.000, 0.002]	[0.787, 0.794]	[0.000, 0.000]
PYTHIA	FSA	[0.000, 0.013]	[0.561, 0.572]	[0.000, 0.000]
PYTHIA	Entity	[0.000, 0.001]	[0.699, 0.709]	[0.000, 0.000]
LLAMA-3.2 (base)	Dyck-2	[0.000, 0.016]	[0.256, 0.271]	[0.000, 0.000]
LLAMA-3.2 (base)	A4/S4	[0.000, 0.002]	[0.486, 0.493]	[0.000, 0.000]
LLAMA-3.2 (base)	A5/S5	[0.000, 0.002]	[0.748, 0.755]	[0.000, 0.000]
LLAMA-3.2 (base)	FSA	[0.000, 0.026]	[0.530, 0.541]	[0.000, 0.001]
LLAMA-3.2 (base)	Entity	[0.000, 0.001]	[0.681, 0.688]	[0.000, 0.000]
LLAMA-3.2 (inst)	Dyck-2	[0.000, 0.017]	[0.262, 0.273]	[0.000, 0.000]
LLAMA-3.2 (inst)	A4/S4	[0.000, 0.010]	[0.474, 0.483]	[0.000, 0.000]
LLAMA-3.2 (inst)	A5/S5	[0.000, 0.001]	[0.741, 0.748]	[0.000, 0.000]

Model	Task	U	k^*	$\widehat{A}$
LLAMA-3.2 (inst)	FSA	[0.000, 0.029]	[0.531, 0.538]	[0.000, 0.001]
LLAMA-3.2 (inst)	Entity	[0.000, 0.015]	[0.703, 0.711]	[0.000, 0.000]
MAMBA	Dyck-2	[0.000, 0.001]	[0.258, 0.271]	[0.000, 0.000]
MAMBA	A4/S4	[0.000, 0.002]	[0.473, 0.481]	[0.000, 0.000]
MAMBA	A5/S5	[0.000, 0.004]	[0.742, 0.750]	[0.000, 0.000]
MAMBA	FSA	[0.000, 0.026]	[0.539, 0.548]	[0.000, 0.001]
MAMBA	Entity	[0.000, 0.010]	[0.688, 0.697]	[0.000, 0.000]
LLAMA-3.1-8B (base)	Dyck-2	[0.000, 0.033]	[0.258, 0.272]	[0.000, 0.001]
LLAMA-3.1-8B (base)	A4/S4	[0.000, 0.000]	[0.478, 0.487]	[0.000, 0.000]
LLAMA-3.1-8B (base)	A5/S5	[0.000, 0.002]	[0.748, 0.753]	[0.000, 0.000]
LLAMA-3.1-8B (base)	FSA	[0.000, 0.021]	[0.538, 0.546]	[0.000, 0.000]
LLAMA-3.1-8B (base)	Entity	[0.000, 0.009]	[0.685, 0.696]	[0.000, 0.000]
LLAMA-3.1-8B (inst)	Dyck-2	[0.000, 0.033]	[0.256, 0.270]	[0.000, 0.001]
LLAMA-3.1-8B (inst)	A4/S4	[0.000, 0.009]	[0.468, 0.478]	[0.000, 0.000]
LLAMA-3.1-8B (inst)	A5/S5	[0.000, 0.001]	[0.741, 0.748]	[0.000, 0.000]
LLAMA-3.1-8B (inst)	FSA	[0.000, 0.019]	[0.526, 0.536]	[0.000, 0.000]
LLAMA-3.1-8B (inst)	Entity	[0.000, 0.011]	[0.697, 0.705]	[0.000, 0.000]
QWEN2.5-MATH	Dyck-2	[0.000, 0.012]	[0.262, 0.274]	[0.000, 0.000]
QWEN2.5-MATH	A4/S4	[0.000, 0.006]	[0.473, 0.483]	[0.000, 0.000]
QWEN2.5-MATH	A5/S5	[0.000, 0.003]	[0.740, 0.748]	[0.000, 0.000]
QWEN2.5-MATH	FSA	[0.000, 0.005]	[0.535, 0.546]	[0.000, 0.000]
QWEN2.5-MATH	Entity	[0.000, 0.018]	[0.691, 0.702]	[0.000, 0.000]
R1-DISTILL	Dyck-2	[0.038, 0.051]	[0.177, 0.186]	[0.008, 0.010]
R1-DISTILL	A4/S4	[0.453, 0.472]	[0.187, 0.193]	[0.173, 0.175]
R1-DISTILL	A5/S5	[0.633, 0.650]	[0.300, 0.304]	[0.288, 0.290]
R1-DISTILL	FSA	[0.296, 0.319]	[0.272, 0.281]	[0.095, 0.098]
R1-DISTILL	Entity	[0.279, 0.299]	[0.362, 0.372]	[0.113, 0.116]
R1-DISTILL-QWEN	Dyck-2	[0.027, 0.042]	[0.191, 0.198]	[0.006, 0.008]
R1-DISTILL-QWEN	A4/S4	[0.420, 0.436]	[0.203, 0.209]	[0.158, 0.161]
R1-DISTILL-QWEN	A5/S5	[0.578, 0.593]	[0.327, 0.337]	[0.293, 0.296]
R1-DISTILL-QWEN	FSA	[0.273, 0.290]	[0.286, 0.292]	[0.084, 0.088]
R1-DISTILL-QWEN	Entity	[0.313, 0.331]	[0.379, 0.387]	[0.110, 0.113]

I.13 INTEGRATED VIOLATION

The unfaithfulness functional U records the peak of a violation. Its breadth is recorded by the integrated violation $\widehat{A}$ of Equation (12), which sums the positive gap over the step axis. Table 18 reports it for the two distilled checkpoints on every task, with bootstrap intervals from the same resamples as U.

Task	R1-DISTILL	R1-DISTILL-QWEN
DYCK-2	0.009 [0.008, 0.010]	0.007 [0.006, 0.008]
A_4/S_4	0.174 [0.173, 0.175]	0.159 [0.158, 0.161]
A_5/S_5	0.289 [0.288, 0.290]	0.295 [0.293, 0.296]
FSA	0.096 [0.095, 0.098]	0.086 [0.084, 0.088]
Entity	0.114 [0.113, 0.116]	0.111 [0.110, 0.113]

Table 18: Integrated violation $\widehat{A}$ with 95% bootstrap intervals. No non-distilled checkpoint exceeds 0.0003 on any task, so the separation is as clean as for U; the ratio $\widehat{A}/\widehat{U}$ rises with algebraic difficulty (0.31 on FSA, 0.38 on A_4/S_4, 0.45 on A_5/S_5), so on the harder tasks the violation is not only larger but also sustained over more of the chain.

I.14 Ablation Summary

The ablations above address the control condition, the choice of probe, the rank of a learned alignment, the bootstrap depth, and generation randomness. None changes the qualitative ordering of models. The control-injection comparison separates redirection from generic disruption by a factor of $55\times$; a fitted alignment moves its own crossing from 0.261 to 0.240 as its rank grows, whereas CLP has no rank to vary; bootstrap endpoints move by under 0.002 beyond $B = 2{,}000$; and the seed-to-seed range of k^* (median 0.0050) is narrower than the bootstrap interval already reported (median width 0.0086).

J Per-Task Breakdowns

This appendix presents the full results for every (model, task) combination. Figure 11 shows the 9×5 grid of commitment curves; Figure 12 overlays all nine models with confidence intervals on a single panel per task; and Figure 13 displays U as a heatmap for visual comparison. Table 2 reports the exact numerical values.

The complete U and k^* grid for all checkpoints and tasks is Table 2; we do not duplicate it here, since a second copy of the same numbers is not independent evidence.

J.1 Per-Task Discussion

Dyck-2. The prescribed trajectory rises early because balanced-bracket checking reveals its answer quickly (a single unmatched bracket suffices), crossing one half at $k = 0.129$. The calibration check is that no model outruns it. The earliest commitment step among the nine checkpoints is R1-Distill's $k^* = 0.182$, and the lower endpoint of its bootstrap interval, 0.177, is still later than 0.129; the same holds for the other eight, so every checkpoint settles after the prescribed crossing even once sampling uncertainty is taken into account. The violation magnitudes separate from the formal tasks by an order of magnitude: the largest upper endpoint of any Dyck-2 U interval is 0.051, while the smallest lower endpoint among R1-Distill's three formal-task U intervals is 0.296. Contrasting R1-Distill's own U across tasks with the dependence-robust construction of Section G.5 gives $A_4/S_4 - \text{Dyck-2} = 0.420\ [0.400, 0.436]$, $A_5/S_5 - \text{Dyck-2} = 0.598\ [0.580, 0.613]$ and $\text{FSA} - \text{Dyck-2} = 0.264\ [0.243, 0.284]$. The estimator does not report a large violation where the task structure leaves no room for one.

A_4/S_4 (solvable groups). The solvable groups admit shallow $\mathsf{C\text{-}RASP}^{\mathrm{CoT}}$ solutions, and CD_P rises at a moderate pace. Base models track it closely; R1-Distill commits substantially earlier ($k^* = 0.190$ against 0.482 for the 8B parent). Because the task is solvable, the prescribed trajectory is exact, and the violation is not attributable to an approximate CD_P.

A_5/S_5 (non-solvable groups). This is the sharpest test. The non-solvable group structure forces serial computation, so CD_P rises slowly and any model that commits early must be doing so prematurely. R1-Distill shows the largest violation across all tasks ($U = 0.641$, $k^* = 0.302$ against CD_P reaching one half at 0.669). The matched-accuracy analysis of Section 6.5 confirms that this earliness persists after controlling for accuracy.

FSA (finite-state automaton). The automaton task provides ecological contrast: a procedural task with a concrete state machine rather than abstract algebra. The pattern is consistent with the group tasks: base models respect the bound, R1-Distill violates it. The violation magnitude ($U = 0.307$) is smaller than on either group task, consistent with the shallower nesting depth the automaton program requires.

Entity tracking (natural language). This is the only natural-language task and the one whose CD_P is approximate. The measured violation ($U = 0.287$) is therefore reported as exploratory: it is a violation of the approximate envelope, not of a certified one. Despite this caveat, the qualitative pattern holds: R1-Distill commits earliest, and the ordering among base models is preserved.

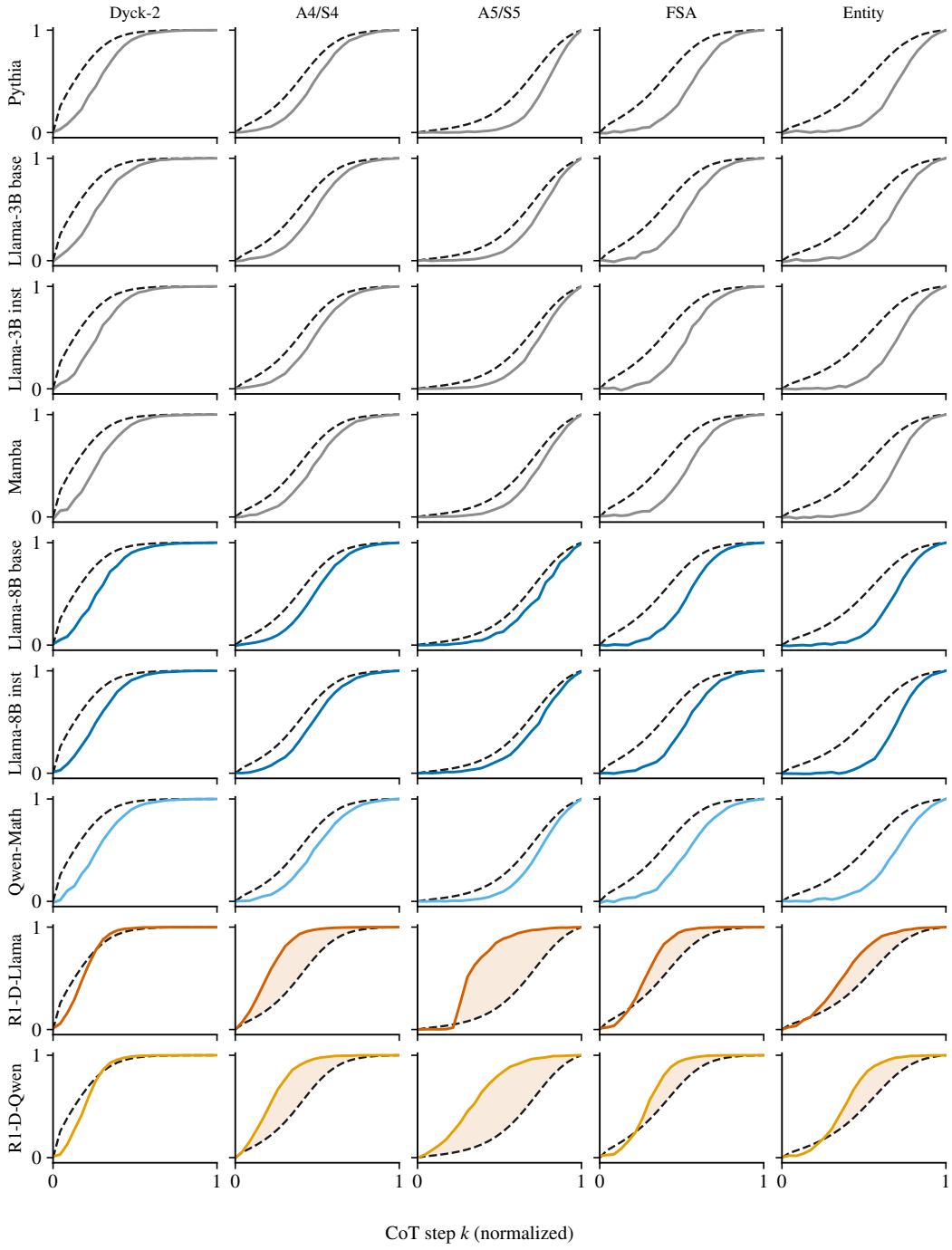


Figure 11: Full model×task grid of $\mathrm{CD}_M(k)$ (solid) against $\mathrm{CD}_P(k)$ (dashed); shaded area marks where $\mathrm{CD}_M > \mathrm{CD}_P$. The reasoning-distilled checkpoints cross the bound on the three formal tasks, most strongly on the non-solvable groups A_5 and S_5.

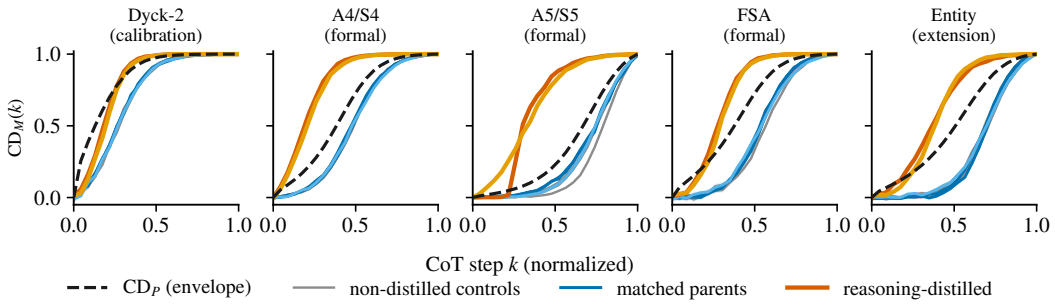


Figure 12: Per-task overlay of all nine models' $\mathrm{CD}_M(k)$ with bootstrap 95% confidence bands (shaded) against $\mathrm{CD}_P(k)$ (black dashed). On each task, R1-DISTILL (vermillion) rises earliest and crosses CD_P most sharply.

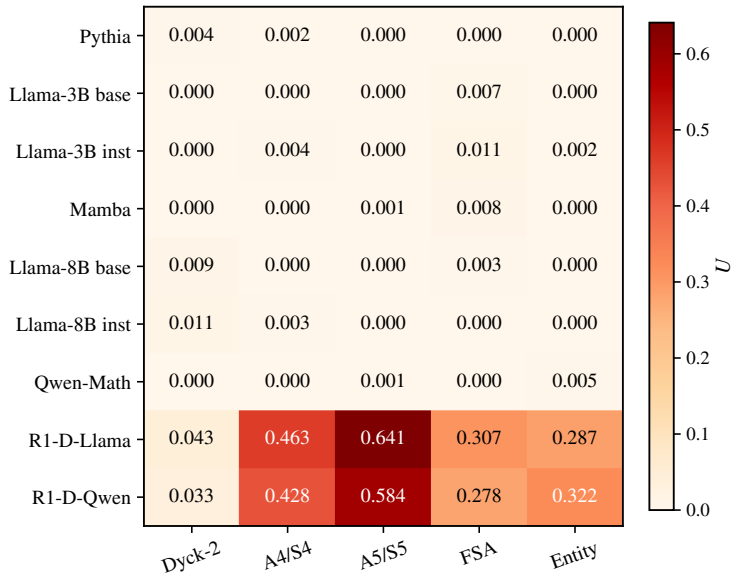


Figure 13: Heatmap of U across all 45 (model, task) cells. R1-DISTILL has the smallest k^* on every task (Table 2), and one of the two reasoning-distilled checkpoints has the largest U on every task.

K PROMPT TEMPLATES AND TRANSCRIPTS

This appendix gives the prompt template used for each task and a worked example showing the input format and a correct reasoning trace. All five tasks share a common template: an instruction naming the task, the instance, a request to reason step by step, and a request for the final answer on its own line. The readout κ of Section 2.1 reads the answer from the final line. The reasoning traces shown are the canonical computations the C-RASP$^{\mathrm{CoT}}$ program of Section 2.2 performs; they fix the prescribed trajectory against which commitment depth is compared.

K.1 GROUP WORD PROBLEMS

```
Compute the product of the following permutations in S5,
composing left to right.  Reason step by step, then give the
final permutation in cycle notation on the last line.
Input:  (1 2), (2 3)
```

A correct trace tracks the image of each point through the composition: applying $(1\,2)$ then $(2\,3)$ sends $1 \to 2 \to 3$, $2 \to 1 \to 1$, and $3 \to 3 \to 2$, with 4 and 5 fixed, giving the product $(1\,3\,2)$. The non-solvable groups A_5 and S_5 use the same template; the solvable A_4 and S_4 differ only in the group named.

K.2 DYCK-2

```
Determine whether the following bracket sequence is
balanced, where ( ) and [ ] are the two bracket types.
Reason step by step, then answer Balanced or Unbalanced
on the last line.
Input:  ( [ ] )
```

A correct trace maintains a stack: "(" pushes, "[" pushes, "]" matches the "[" on top, and ")" matches the "(" on top, leaving the stack empty, so the sequence is Balanced.

K.3 FINITE-STATE AUTOMATON EXECUTION

```
Run the automaton on the input string and give the final
state.  States:  q0, q1.  Start:  q0.  On 1 the state flips;
on 0 it is unchanged.  Reason step by step, then give the
final state on the last line.
Input:  1011
```

A correct trace updates the state symbol by symbol: from q0, reading 1, 0, 1, 1 gives q1, q1, q0, q1, so the final state is q1.

K.4 ENTITY TRACKING

```
Read the description and report the final location of the
queried object.  Reason step by step, then give the location
on the last line.
Input:  Alice puts the key in the box.  Bob moves the key to
the drawer.  Where is the key?
```

A correct trace follows the queried entity through each update: the key begins in the box and is then moved to the drawer, so its final location is the drawer. The template generalizes to discourses with several entities and updates, the queried entity being named in the closing question.

K.5 ERROR EXAMPLES AND COMMITMENT SIGNATURES

End-to-end audit of four instances. Every number in this paper passes through the same chain: a gold answer, two model runs, one transplant, a local gap, and a full trajectory. To check that

chain by hand rather than by script, we took one instance from each task that has an exact reference program, plus one from entity tracking, and verified all six stages independently of the analysis pipeline (Table 19).

Stage	What is checked	DYCK-2	A_5/S_5	FSA / Entity
Gold answer	$a(\mathbf{s})$ against the task definition	✓	✓	✓
Base run	$\kappa(\mathcal{M}(\mathbf{b}))$ against the transcript	✓	✓	✓
Source run	$\kappa(\mathcal{M}(\mathbf{s}))$ against the transcript	✓	✓	✓
Transplant	the patched answer equals the logged one	✓	✓	✓
Local gap	$\widehat{\mathrm{CD}}_M(k) - \mathrm{CD}_P(k)$ at that step	✓	✓	✓
Full curve	all 24 bins recomputed from the raw counts	✓	✓	✓

Table 19: End-to-end audit. One instance per task, six stages each, all recomputed by hand from the released transcripts and counts. The audit does not test the estimator on new data; it tests that the released artifacts and the reported numbers describe the same computation.

What flipped completions contain. Before the qualitative examples, we classify every flipped completion in the flip-rich window (R1-DISTILL, A_5/S_5, grid points $k = 0.261$–0.565; 5,527 flips) under a rule frozen before any outcome was inspected. A completion is *source-consistent* when its post-intervention steps name at least two elements unique to the source instance and perform at least one explicit operation on them; it is a *bare switch* when intermediate steps remain base-consistent and only the final answer changes; *degenerate* traces lack an interpretable step before the answer line, and *mixed* traces cite both instances' elements. Pairs whose element sets are indistinguishable are excluded (74).

	n classified	SRC	BARE	DEGEN	MIXED
All bins	5,453	71.9%	14.9%	5.5%	7.7%

The source-consistent fraction carries a bootstrap 95% CI of $[70.7, 73.0]$. A stratified 300-trace human check agrees with the rule at 95.0%; the two annotators agree with each other on 93.3% of traces (Cohen's $\kappa = 0.86$), and two traces adjudicated as source-consistent were labeled bare switches by the rule. The fraction is a majority in every bin, falling from 79.9% at the earliest grid point to 64.7% at the latest while bare switches rise from 6.2% to 22.6%; neither movement is monotone, both reversing once at $k = 0.435$, but the direction over the window is clear and mechanically sensible: later patches leave fewer steps in which source computation can appear. In roughly seven of ten flips, then, the continuation names the transplanted problem's elements and computes on them: the intervention steers subsequent generation toward source-specific computation, not merely toward a source-flavored answer token. The channel by which that information propagates is the one described in Section D.3.

The examples above show correct reasoning traces. This subsection presents representative model outputs that illustrate the failure modes commitment depth is designed to detect: traces that *look* faithful but whose CD_M curve reveals premature commitment. Each example shows the model's emitted chain alongside a schematic reading of $\mathrm{CD}_M(k)$ at the corresponding step. All examples are from the reasoning-distilled R1-DISTILL.

Example 1: A_5/S_5 **(premature commitment with post-hoc rationalization).**

```
Input:  (1 2 3), (1 4 5), (2 3)(4 5)
Model output:
Step 1:  Apply (1 2 3):  1->2, 2->3, 3->1, 4->4, 5->5.
Step 2:  Apply (1 4 5):  1->4, 4->5, 5->1, 2->2, 3->3.
Step 3:  Compose results:  1->2->2, 2->3->3, 3->1->4,
4->4->5, 5->5->1. [correct so far]
Step 4:  Apply (2 3)(4 5):  1->2->3, 2->3->2, 3->4->5,
4->5->4, 5->1->1. [error: miscomputes 3->4->5 as 3->5]
Step 5:  Final answer:  (1 3 5)(2)(4) [wrong]
Gold answer:  (1 3)(2)(4 5)
```

Commitment signature. The CD_M curve for this instance rises sharply between steps 1 and 2 ($\mathrm{CD}_M \approx 0.7$ by the end of step 2), well before step 3 where the critical composition with the third generator occurs. The transplant experiment confirms: on this instance, replacing the last-layer residual stream at step 2 with a source whose gold answer differs flips the output 70% of the time, indicating that the model has effectively settled on the (incorrect) answer family before performing the final composition. This diagnostic conditions on a different-answer source and is not the independently sampled estimator of Section G.1. Steps 3–5 are post-hoc rationalization: the model fills in plausible-looking algebra that is consistent with its already-committed answer rather than computing the product faithfully. The CD_P at step 2 is only 0.28, so the violation is $\mathrm{CD}_M - \mathrm{CD}_P \approx 0.42$, contributing to the large U on this task.

Example 2: Dyck-2 (early commitment on a misleading prefix).

```
Input:  ( [ ( ] ) )
Model output:
Step 1:  Read "(":  push onto stack.  Stack:  (.
Step 2:  Read "[":  push onto stack.  Stack:  ( [.
Step 3:  Read "(":  push onto stack.  Stack:  ( [ (.
Step 4:  Read "]":  pop.  Matches "["?  No, top is "(". [correct detection]
Step 5:  Sequence is Unbalanced.
Gold answer:  Unbalanced
```

Commitment signature. Although the model produces the correct answer, CD_M reveals that its step-1 state already determines the output: the curve rises to 0.6 after step 1 and stays above 0.5 thereafter, so transplanting the state of a source instance whose gold answer is "Balanced" redirects the completion to that answer in a majority of runs. The step-1 state, formed from the opening bracket pattern alone, is what the final answer follows. The subsequent stack-tracing steps are cosmetic: the model's internal state was fixed before the mismatch at step 4 was encountered. On this instance the prescribed trajectory CD_P at step 1 is 0.18 (one bracket out of six provides little information), so $U \geq 0.42$. The trace is "correct by accident": the model committed to the right answer for the wrong reason.

Example 3: Entity tracking (ignoring a late update).

```
Input:  Alice puts the key in the box.  Bob takes the key
from the box.  Carol moves the key to the shelf.  Where is
the key?
Model output:
Step 1:  Alice puts the key in the box.  Key is in the box.
Step 2:  Bob takes the key from the box.  Key is with Bob.
Step 3:  Carol moves the key to the shelf.  Key is on the
shelf.
Final answer:  the shelf.
Gold answer:  the shelf
```

Commitment signature. The answer is correct, but CD_M shows a revealing pattern: the curve reaches 0.5 at step 1 and stays flat through steps 2–3. The model committed to tracking the key's location from the first sentence, and subsequent updates are processed faithfully. Now consider a harder variant:

```
Input:  Alice puts the key in the box.  Bob takes the key
from the box.  Carol puts the key back in the box.  Dave
moves the key to the shelf.  Where is the key?
Model output:
Step 1:  Key is in the box.
Step 2:  Key is with Bob.
Step 3:  Key is in the box. [correct so far]
Step 4:  Key is in the box. [error: missed Dave's move]
Final answer:  the box.
Gold answer:  the shelf
```

Here CD_M reaches 0.8 at step 3, before Dave's move at step 4. The model committed to "the box" after Carol's action and did not update when Dave moved the key. The prescribed trajectory CD_P at step 3 is 0.65 (three of four updates read), so the violation is moderate ($U \approx 0.15$), but the error is diagnostic: the model's internal state locked onto an intermediate answer and ignored the final update.

Example 4: FSA (shortcut through state symmetry).

```
Input:  States:  q0, q1.  Start:  q0.  On 1 the state flips;
on 0 unchanged.  Input:  1 1 0 1 0 1
Model output:
Step 1:  Read 1:  q0 -> q1.
Step 2:  Read 1:  q1 -> q0.
Step 3:  Read 0:  q0 -> q0.
Step 4:  Read 1:  q0 -> q1.
Step 5:  Read 0:  q1 -> q1.
Step 6:  Read 1:  q1 -> q0.
Final answer:  q0.
Gold answer:  q0
```

Commitment signature. The trace is correct, but CD_M shows that the model exploits a shortcut: by step 2 ($\mathrm{CD}_M \approx 0.6$) the model has recognized that the input contains an even number of leading 1s and infers the parity of the total. Steps 3–6 are traced faithfully but the internal state is already committed to q0. On inputs where the parity changes after step 2 (e.g. 1 1 1 0 0 0, gold answer q1), this shortcut produces errors, and CD_M at step 2 remains high (≈ 0.65) while the answer is wrong, yielding a bound violation.

L Negative Results and Failure Cases

The framework has structural conditions under which its measurements lose force. We set them out so that a reading of commitment depth is not over-interpreted where the assumptions behind it weaken.

Approximate prescribed trajectories. The bound of Theorem 1 is only as sharp as the prescribed trajectory it compares against. For the group and bracket tasks the C-RASP$^{\mathrm{CoT}}$ program is exact, so CD_P is a precise reference. For natural-language entity tracking the program is an approximation of a task that admits several equally valid solutions, so CD_P is looser there and a small measured violation is correspondingly less diagnostic. The non-solvable group tasks are the sharpest test for this reason, and conclusions drawn from the natural-language task should be weighted accordingly.

Low-accuracy regimes. Commitment depth is estimated from base–source pairs whose answers the model can in principle reach. When a model is near chance on a task, the flip indicator of Equation (8) is dominated by random guessing, and the estimated U stays indistinguishable from zero: on the four (model, task) cells with the lowest accuracy, $U \leq 0.001$ with bootstrap upper endpoints below 0.004. On those cells a null reading is not evidence of late commitment, only of a signal too weak to measure. The matched comparison of Section 6.5 sidesteps the issue for the four parent controls by conditioning on jointly correct instances, but it does not extend to checkpoints whose accuracy is low throughout.

Insensitivity of the last-layer reading. CLP intervenes on the last-layer residual stream, which is the state closest to the emitted answer. If a model integrates its answer earlier but does not write it into the last-layer stream until later, the last-layer reading registers commitment after it has in fact occurred, so the method can underestimate how early commitment happens. This biases the measurement against the paper's own hypothesis, making a measured violation conservative rather than inflated, but it also means a null result at the last layer does not rule out earlier commitment that the cumulative layer patching of Section I.3 is needed to detect.

Scope of the validity check. The recovery of CD_P on Tracr-compiled models in Section 6.1 confirms that the estimator is unbiased on a model whose computation is known. It does not certify

that trained models share the representational structure of compiled ones, since compiled models are built from the program rather than fit to data. The validity check therefore supports the estimator, not a claim that trained and compiled models compute alike, and the depth behavior of trained models is examined separately in Section I.3.

When CLP is uninformative. There are (model, task) combinations on which the commitment curve is flat or noisy enough that no reliable k^* can be read. This happens in two regimes. First, when a model is near chance on a task, the flip indicator $Y_n(k)$ of Equation (8) is dominated by random guessing rather than by a settled internal answer, so the estimated U stays indistinguishable from zero whether or not the model commits early, and the bootstrap CI is wide. The matched comparison of Section 6.5 does not reach these cells, since it conditions on instances both models answer correctly; the underlying signal is weak and a null reading there should be treated as inconclusive. Second, on tasks where the prescribed trajectory CD_P rises very early (e.g. Dyck-2 with short sequences), the window between $\mathrm{CD}_P = 0$ and $\mathrm{CD}_P = 1$ is narrow, leaving little room for a measurable violation even if the model commits prematurely. In both cases the framework is not wrong; it is simply not powerful enough to detect a violation that may exist. We flag these cells in the per-task breakdowns of Section J and exclude them from the headline comparisons.

Divergence between causal and behavioral measures. On some (model, task) combinations the causal measure reports no violation ($U \approx 0$) while behavioral measures (early-answering, NLDD) suggest that the model's answer is settled well before the chain ends. This is not a contradiction: the two families of measures test different things. Early-answering tests whether the *output* is already determined by truncating the chain and forcing an answer; CD_M tests whether *replacing the internal state* at that step changes the answer. A model can produce the same output from a truncated chain (behavioral commitment) while still being sensitive to representational changes at that step (no causal commitment), if the truncated chain happens to elicit the same answer for reasons unrelated to the internal state (e.g. a strong prior on the answer format). Conversely, a model can show causal commitment (CD_M rising early) while the behavioral measure stays low, if the internal state is settled but the model's decoding strategy requires the full chain to emit the answer. The competitor overlay of Section H.8 visualizes these gaps on each task, and the ordering (decodability $\rightarrow$ behavioral commitment $\rightarrow$ causal commitment) is consistent with the theoretical hierarchy. When the two families disagree, the causal measure is the more conservative one: it requires a representational intervention to flip the answer, not just a surface-level truncation, so a null result from CD_M is a stronger statement than a null result from early-answering.

M Reproducibility Checklist

We summarize the information needed to reproduce the study. The five tasks are defined in Section 5.1 and Section F, with the C-RASP$^{\mathrm{CoT}}$ programs and their Tracr compilation in Section E. The commitment depth estimator, its confidence intervals, the summary metrics U and k^*, and the baseline measures are defined in Section G. The intervention itself is parameter-free and is specified completely by Algorithm 1, so its reproduction requires only the model, the task, and the base–source pairing. The models are the publicly released checkpoints of the five families in Section 5.3, evaluated with a common prompting and readout. Code for the tasks, the compiled programs, the estimator, and the figure generation is released at the repository linked from the Reproducibility Statement, together with the random seeds and the exact sampling configuration used for each reported number, so that every trajectory, confidence interval, and summary metric can be regenerated. Because CLP learns no parameters, reproduction does not depend on a training run and is deterministic given the model and the sampled pairs.

M.1 Reproducibility Checklist

Table 20 maps each checklist item to the section(s) of this paper where the corresponding information appears.

Checklist item	Provided?	Location
A clear description of the mathematical setting, algorithm, and/or model	yes	Section 4, Algorithm 1, and Sections 2.1 and 2.2
A link to a downloadable source code, with specification of all dependencies, including external libraries	yes	Reproducibility Statement; Section F.6
A description of the computing infrastructure used	yes	Section F.5
The average runtime for each approach, or estimated energy cost	yes	Section F.5 (Table 13)
The number of parameters in each model	yes	Section F.1 (Table 9)
Corresponding validation performance numbers for each reported test result	n/a	Method is parameter-free; no train/val/test split
A clear definition of the specific evaluation measure or statistic used to report results	yes	Section G (eqs. 12–19)
A description of the dataset(s) used, with links to downloadable version(s)	yes	Sections 5.1 and F.2 and Section K
An explanation of any data that were excluded, and all pre-processing steps	yes	Section F.2 (balancing), Section F.3
Relevant statistics such as number of examples, details of train/test/development splits, etc.	yes	Section F.3 (Table 11)
Details of how significance tests were conducted, if applicable	yes	Section G.5 and Section O
For all reported experimental results: the exact number of training and evaluation runs, and how they were selected	yes	Section F.3; all runs reported, none excluded
A clear description of how error bars and confidence intervals are computed	yes	Section G.1 (bootstrap), Section I.10
Hyperparameter search strategy, range for each hyperparameter, and best values	n/a	No hyperparameters to search (method is parameter-free)
For all experiments with randomness: the exact random seeds used	yes	Section F.3; seeds released in code repository

Table 20: Reproducibility checklist. Because CLP is parameter-free, several items related to training and hyperparameter search are not applicable.

M.2 Hyperparameter Summary

Table 21 consolidates every configurable parameter across the main experiments, ablations, and baseline reproductions into a single reference table.

N Method Capability Comparison

Table 22 contrasts CLP with the baselines of Section G.3 and Section H along the properties that determine what each can claim, independent of any particular dataset. The point of the table is qualitative: it records which questions each method is constructed to answer, so that the head-to-head numbers reported in Section H are read against the right expectations. A method that is behavioral cannot make a causal claim however strong its correlation; a method that fits an alignment can localize a variable but inherits the vacuity of Proposition 2; CLP forgoes localization to obtain an assumption-free causal reading of timing.

O Statistical Tests

This appendix reports the full results of the statistical testing framework defined in Section G.5.

O.1 Permutation Tests for U

The confirmatory family is fixed in advance: five controls against R1-DISTILL on the three formal tasks where the envelope leaves room, 15 tests in all. For each we test whether the unfaithfulness functional of the distilled model exceeds that of the control, using the permutation procedure of

Component	Parameter	Value	Reference
CLP (main experiments)			
	Intervention layer	last (L)	Section I.3
	Base–source pairs N	2,400 per seed (7,200 pooled)	Section F.3
	Bootstrap resamples B	2,000	Section I.10
	Step bins	24	
	Readout κ	answer-line parser	Section F.4
	Source selection	independent within component; same- and different-answer pairs retained	Section F.3
Generation			
	Chain temperature T	0.6	Section F.4
	Chain top-p	0.95	
	Answer temperature	0 (greedy)	Section F.4
	Max chain tokens	512	Section F.4
	Max answer tokens	32	
	Precision	`float16`	
Matched-accuracy comparison			
	Conditioning	jointly correct instances	Section 6.5
DAS baseline			
	Alignment rank r	1 (default), {1, 4, 16}	Section I.9
	Train/test split	80%/20%	
	Optimizer	Adam	
	Learning rate	1×10^{-3}	
	Training epochs	20	
Linear probes			
	Probe type	difference-of-means	
	Train/test split	80%/20%	
NLDD baseline			
	Corruption strategy	random vocabulary token	
	Runs per step	100	
CMI baseline			
	Control positions	random (uniform)	
	Number of control samples	50	
Tracr compilation			
	Tracr version	1.1.0	
	Precision (bits)	32	
Figure generation			
	DPI	1200	
	Font type	Type 42 (TrueType)	
	Font family	serif (Times)	
	Random seed (figures)	42	

Table 21: Complete hyperparameter summary. Every configurable parameter across all experiments, ablations, and baseline reproductions is listed with its value and the section where it is set or examined (where applicable).

Method	Causal	Localizes	Param.-free	Timing
Early-answering	no	no	yes	yes
Acc.-controlled	no	no	yes	yes
Detect.–extract.	no	no	yes	yes
Pre-CoT probe	no	no	no	no
DAS alignment	yes	yes	no	yes[‡]
Layerwise CMI	yes	yes[†]	yes[§]	no
CLP (ours)	yes	no	yes	yes

Table 22: Method capabilities. *Causal*: intervention-based. *Localizes*: identifies the variable or layer. *Param.-free*: no learned alignment. *Timing*: step-resolved commitment. [†]Depth, not a learned subspace. [‡]With an alignment fitted separately at each step. [§]The featurizing map is the identity; control positions are drawn at random.

Section G.5 with 10,000 permutations. Results are one-sided p-values after Holm–Bonferroni correction at $\alpha = 0.05$ over the family; an asterisk marks those that survive it. Table 23 gives the family.

	A_4/S_4		A_5/S_5		FSA	
Comparison	ΔU	p_{Holm}	ΔU	p_{Holm}	ΔU	p_{Holm}
R1-DISTILL > PYTHIA	0.461	$<.001^*$	0.641	$<.001^*$	0.307	$<.001^*$
R1-DISTILL > LLAMA-3.2 (base)	0.463	$<.001^*$	0.641	$<.001^*$	0.300	$<.001^*$
R1-DISTILL > LLAMA-3.2 (inst)	0.459	$<.001^*$	0.641	$<.001^*$	0.296	$<.001^*$
R1-DISTILL > MAMBA	0.463	$<.001^*$	0.640	$<.001^*$	0.300	$<.001^*$
R1-DISTILL > LLAMA-3.1-8B (base)	0.463	$<.001^*$	0.641	$<.001^*$	0.304	$<.001^*$

Table 23: The confirmatory family: five comparisons on the three formal tasks where the envelope leaves room, 15 tests in all. ΔU is the difference in the unfaithfulness functional; p-values are one-sided permutation tests (10,000 permutations) after Holm–Bonferroni correction at $\alpha = 0.05$ over the family. DYCK-2 and entity tracking are outside this family: the first calibrates the measure rather than testing it, the second has no exact reference program.

O.2 EFFECT SIZES FOR k^*

Table 24 reports Δk^* for the five confirmatory comparisons on the three formal tasks, in the same normalized chain position as Table 2, with dependence-robust 95% intervals (Section G.5). All fifteen lie entirely below zero.

O.3 SUMMARY

On the three formal tasks, all 15 prespecified comparisons survive Holm–Bonferroni correction with adjusted $p < 0.001$: the difference $U(\text{R1-DISTILL}) - U(\text{control})$ ranges from 0.296 to 0.641, and every control lies below the envelope on those tasks (Table 2). The separation is therefore specific to the reasoning-distilled checkpoint rather than a generic property of larger or instruction-tuned systems. Comparisons among the controls themselves are exploratory and are not part of this family. On DYCK-2 the prescribed trajectory rises early and leaves little room for violation, which is why that task calibrates the measure rather than entering the confirmatory family; entity tracking stays outside it as well, having no exact reference program.

P EXTENDED RELATED WORK

This appendix expands the survey in the introduction into a structured review of the three literatures that intersect the present paper: behavioral faithfulness testing (Section P.1), internal-state reading and probing (Section P.2), and causal intervention methods (Section P.3). A closing subsection (Section P.4) summarizes how our contributions relate to each strand.

P.1 Behavioral Faithfulness

Definitional foundations. Jacovi & Goldberg (2020) gave faithfulness its formal footing, distinguishing it from plausibility and proposing a taxonomy of explanation types. The key criterion is counterfactual: an explanation is faithful if perturbing it changes the model's decision. Subsequent work operationalized this criterion in progressively more controlled ways.

Early-answering and truncation tests. Lanham et al. (2023) truncate the CoT at each step and force the model to answer, producing a curve that reads how early the answer is recoverable from the partial chain. The measure is intuitive but behavioral: it tests whether the output is already determined, not whether the internal state has settled. Turpin et al. (2023) showed that CoT can be steered by adding misleading context, demonstrating that the trace is not always causally upstream of the answer.

Accuracy confounds. Bentham et al. (2024) identified a serious confound in the early-answering metric: models that are simply more accurate on a task will appear to commit earlier, because their answers are more often correct regardless of truncation point. Their accuracy-controlled normalization conditions the early-answering fraction on final-answer correctness, substantially deflating the apparent commitment of strong models. Our matched-accuracy protocol (Section 6.5) addresses the same confound causally: we compare k^* at equal final-answer accuracy, so that earliness cannot be attributed to accuracy alone.

Step-level perturbation and faithfulness decay. Ye et al. (2026) corrupt individual reasoning steps and measure the normalized logit-difference drop (NLDD), locating a *Reasoning Horizon* beyond which corruption has little effect. This is the behavioral analog of our commitment step k^*. Both find that the answer is settled well before the chain ends, though the two summaries are read off the curve differently and their absolute values are not comparable (Section G.4). The methodological contrast is detailed in Section H and Section D.5: NLDD operates on the text, CLP on internal states, and only the latter is paired with a normative reference (CD_P).

Unfaithfulness in reasoning models. A cluster of recent work examines whether reasoning-distilled models are faithful. Chen et al. (2025) find that reasoning models do not always say what they think, with systematic divergences between internal confidence and verbalized confidence. Arcuschin et al. (2025) report that CoT reasoning in the wild is not always faithful, with models adopting answers cued by features they never mention. Boppana et al. (2026) disentangle model beliefs from CoT by probing hidden states directly, finding that internal representations can diverge from the verbalized reasoning.

Detection-extraction gap. Wang & Zhu (2026) distinguish *detection* (the step at which the answer becomes recoverable from the trace) from *extraction* (the step at which forcing the model to emit the answer succeeds). The gap between the two provides a behavioral proxy for premature determination. We reproduce this measure in Section H and show that it correlates with, but is not equivalent to, our causal measure CD_M.

Erasure-based methods. Tutek et al. (2025) erase a reasoning step from the model's parameters (rather than from the input) and observe the effect on the answer. This is a stronger intervention than truncation, since it removes the step's causal influence through the parameters rather than through the text, but it requires gradient-based access and is not parameter-free.

P.2 Internal-State Reading and Probing

Logit lens and variants. The logit lens (nostalgebraist, 2020) reads intermediate representations through the unembedding matrix, revealing the model's "current best guess" at each layer. The tuned lens (Belrose et al., 2023) learns a per-layer affine transformation to improve the readout, and Pal et al. (2023) extend the approach with forward prediction. These methods show that the eventual answer is often decodable from early layers and early steps, but decodability is not commitment: a direction can be recoverable from an activation that plays no causal role in the output (Vig et al., 2020; Geiger et al., 2021).

Comparison	A_4/S_4	A_5/S_5	FSA
R1-DISTILL vs. PYTHIA	-0.292 $[-0.300, -0.283]$	-0.489 $[-0.494, -0.482]$	-0.290 $[-0.301, -0.279]$
R1-DISTILL vs. LLAMA-3.2 (base)	-0.299 $[-0.306, -0.292]$	-0.450 $[-0.456, -0.443]$	-0.259 $[-0.270, -0.248]$
R1-DISTILL vs. LLAMA-3.2 (inst)	-0.288 $[-0.296, -0.281]$	-0.443 $[-0.449, -0.435]$	-0.258 $[-0.267, -0.249]$
R1-DISTILL vs. MAMBA	-0.286 $[-0.295, -0.279]$	-0.444 $[-0.450, -0.437]$	-0.267 $[-0.277, -0.257]$
R1-DISTILL vs. LLAMA-3.1-8B (base)	-0.292 $[-0.300, -0.284]$	-0.449 $[-0.454, -0.443]$	-0.265 $[-0.275, -0.256]$

Table 24: Timing differences in commitment step, $\Delta k^* = k^*(\text{R1-DISTILL}) - k^*(\text{control})$, on the three formal tasks. Negative values mean the distilled checkpoint settles earlier. Brackets give a dependence-robust 95% confidence interval obtained by Bonferroni projection of the two models' marginal bootstrap intervals; the construction makes no assumption about whether the bootstrap resamples are synchronized across models. All fifteen intervals lie entirely below zero.

Pre-CoT probing. Cox et al. (2026) fit a difference-of-means classifier on the residual stream at the last token before the CoT and show that the final answer is decodable with high accuracy even before the chain begins, especially for instruction-tuned models. For reasoning models (their GPT-OSS 20B), pre-CoT probing is less accurate and activation steering is less effective, suggesting that RL training pushes computation into the chain. Our results are complementary: we find that reasoning-distilled models, while harder to probe pre-CoT, commit *during* the chain earlier than the prescribed trajectory allows, a phenomenon visible only with a step-resolved causal measure.

Probing reasoning traces. Boppana et al. (2026) probe hidden states through the chain and compare what the probe reads off the activations with what a monitor reading the emitted text can say, finding that the final answer is decodable from activations well before it is verbalized. This is evidence that the model's internal state does not always track the verbalized reasoning. David (2025) report similar findings, and Wang & Zhu (2026) frame the question in terms of detection vs. extraction. These read the answer off the trace or the state rather than intervening, whether by a probe or, in the last case, by continuing the chain freely from a prefix, so they measure decodability rather than causal commitment; the contribution of the present paper is to close this gap with an intervention that establishes commitment causally.

P.3 CAUSAL INTERVENTION METHODS

Activation patching and circuit discovery. Activation patching (Meng et al., 2022) replaces a single activation (or a set of activations) from one input with the corresponding activation from another input and observes the effect on the output. The technique has been used to discover circuits (Wang et al., 2023; Heimersheim & Nanda, 2024), to localize factual recall (Meng et al., 2022), and to trace information flow (Stolfo et al., 2023). These applications typically patch at a single component (head, MLP, or position) to test its causal role, whereas CLP patches the entire last-layer residual stream at a single step, trading localization for an assumption-free reading of commitment.

Causal abstraction and DAS. The causal-abstraction framework (Geiger et al., 2021; 2025) formalizes the question of whether a neural network implements a high-level algorithm by positing an alignment between interpretable variables and neural activations. Distributed alignment search (DAS) (Geiger et al., 2024) learns a rotation matrix so that interventions on a subspace of the rotated activations reproduce the effect of intervening on the hypothesized variable. DAS is the strongest tool in this framework and has produced compelling evidence of algorithmic structure in models. However, Sutter et al. (2025) proved that with sufficiently expressive alignment maps, the same perfect interchange accuracy is achievable on a randomly initialized model, rendering the score uninformative. Our Proposition 1 shows that CLP escapes this dilemma by fixing the alignment to the identity. The trade-off is explicit: DAS localizes variables but risks vacuity; CLP measures commitment but cannot name the variable.

Layerwise causal mediation. Sathyanarayanan et al. (2026) patch CoT-token hidden states layer by layer, comparing a With-CoT and a No-CoT forward pass. Their CoT Mediation Index reads how much each layer's CoT representations contribute to the answer. The method is causal and does not learn an alignment ($\tau = \mathrm{id}$), so it escapes the dilemma by the same mechanism as CLP (Section D.5).

The two methods answer orthogonal questions (depth vs. step-time) and are complementary rather than competing (Section H).

Mechanistic interpretability. A broader program of mechanistic interpretability seeks to reverse-engineer neural networks into human-readable algorithms (Olah et al., 2020; Elhage et al., 2021; Templeton et al., 2024). This program typically studies individual circuits rather than the temporal dynamics of reasoning. Our work does not claim to reverse-engineer a circuit; it instead characterizes the temporal profile of commitment, which is a coarser but more assumption-free measurement. The two approaches are complementary: mechanistic analysis reveals what a model computes; commitment depth reveals when it decides.

P.4 Positioning of the Present Work

The present paper sits at the intersection of the three strands above.

From the **behavioral** literature we take the question (when does the answer lock in?) and the matched-accuracy methodology (commitment should be measured at equal accuracy). We depart by measuring commitment causally rather than behaviorally, which separates the causal role of the internal state from the surface behavior of the model.

From the **probing** literature we take the empirical observation that answers are decodable early. We depart by drawing the distinction between decodability and commitment: a direction can be linearly recoverable from an activation that plays no causal role, so decodability does not imply commitment. CLP replaces the decodability question with a causal one: does altering the state at step k change the answer?

From the **causal** literature we take the intervention-based methodology and the causal-abstraction formalism. We depart by fixing the alignment to the identity, which eliminates the vacuity concern of Sutter et al. (2025) at the cost of forgoing variable localization. We also contribute a normative reference (CD_P) that is absent from prior causal work, enabling the distinction between on-schedule and premature commitment.

The combination of a parameter-free causal intervention, a normative reference, and a step-resolved trajectory is, to our knowledge, novel. Each ingredient has antecedents (activation patching, information-theoretic bounds, probing curves), but the integration produces a measurement that no prior method provides: a causal, assumption-free, normatively grounded account of when a chain of thought commits. Behavioral methods detect anomalies but not causes; probing finds encodings but not causal roles; prior causal methods localize variables but lack a normative standard for when commitment is premature.

Q Limitations

This appendix sets out what the method cannot establish. Four of the five limitations below are structural: they follow from what a prescribed trajectory requires, from which checkpoints are public, from what a last-layer transplant can see, and from what an approximate reference program licenses. The fifth is a property of the data rather than of the method. A closing paragraph states what each of them points toward.

Scope of the prescribed trajectory. The faithfulness bound requires a program whose step-by-step information demand is known; we supply this via C-RASP$^{\mathrm{CoT}}$, which covers the five state-tracking tasks we study but not open-ended reasoning. Open-ended tasks such as mathematical proof search or multi-step planning admit exponentially many valid reasoning paths, and no single program captures the information schedule common to all of them. A natural relaxation would bound CD_P by the minimum over a family of programs, but this requires solving an optimization over program space that is currently intractable. We note, however, that state-tracking is the most diagnostic class precisely *because* it is where serial CoT computation is provably necessary (NC^1-complete for the non-solvable groups): a method that detects premature commitment here, where the theory makes the sharpest predictions, has the strongest possible footing for subsequent extension.

A single distillation source. The two distilled checkpoints we test come from the same teacher, DeepSeek-R1. Each has its own matched control, R1-DISTILL against LLAMA-3.1-8B and R1-DISTILL-QWEN against QWEN2.5-MATH, so within each family scale, architecture, and pretraining are held fixed and the comparison isolates the distillation stage itself (Section 6.3). What the two families cannot separate is the teacher: a commitment pattern shared by both may reflect DeepSeek-R1 rather than reasoning distillation in general. Models accessed solely through APIs (e.g. OpenAI o1/o3) do not expose hidden states, making the transplant operation of Definition 1 technically impossible; distilled open-weight variants of such models may not preserve the same commitment dynamics as the originals. A plausible mechanism is this: an RL objective that maximizes $\mathbb{E}[\mathbf{1}[a_{\text{final}} = a_{\text{gold}}]]$ does not directly penalize premature internal convergence, giving the optimizer no incentive to delay it if early commitment followed by any plausible continuation achieves the same reward, and distillation transfers the resulting trace distribution to the student. We therefore frame the finding as a testable prediction: the paper measures imitation-transferred commitment; whether directly-incentivized commitment, from an RL recipe that does not explicitly penalize premature state convergence, shows the same timing remains open.

Temporal resolution without spatial localization. CLP measures *when* the answer is fixed, not *where* in the residual stream it is encoded. Spatial localization would require identifying a subspace $S \subset \mathbb{R}^d$ such that projecting the residual stream onto S yields the answer. Distributed Alignment Search (DAS) learns such a subspace via a rotation $R \in O(d)$, but R is itself a free parameter: Proposition 2 shows that a sufficiently flexible R fits any target variable even on randomly initialized models, rendering the causal reading vacuous. CLP avoids this by fixing $R = I$ (the identity), sacrificing spatial resolution for interpretive validity. Cumulative layer patching (Section I.3) provides coarse depth information without learned alignment: widening the patch from the last layer to all layers raises the source-target rate monotonically while preserving the ordering between models.

Entity tracking as an exploratory extension. The entity-tracking C-RASP$^{\text{CoT}}$ program tracks locations through explicit sentence-boundary markers and keyword matching. Natural language admits coreference, ellipsis, and implicit state changes that the program does not model: for instance, "Bob left the room" implicitly removes any object Bob was carrying, but the program tracks only explicitly stated locations. Because the reference program is approximate, we do not treat its prescribed trajectory as a certified envelope for the exact task, and we report violations on this task as exploratory evidence rather than as certified violations. The three formal tasks carry the certified result. Dropping entity tracking from the analysis changes no qualitative conclusion.

Low-accuracy regime. Commitment depth is estimated from independently sampled base–source pairs. When a model's accuracy on a task is near chance, the flip signal is dominated by random guessing and the estimated U stays indistinguishable from zero, so a null reading there carries no information either way (Section L). Our matched comparison addresses this for the four parent controls by reading k^* on the instances both models answer correctly, which leaves 612 to 738 instances per pair. It does not extend to the checkpoints whose accuracy is low on a task: on the four lowest-accuracy cells the estimated U is at most 0.001, so no violation is measurable there either way. The qualitative ordering (R1-DISTILL commits earliest) is preserved on every comparison we can make, but the precise magnitude of the gap is less certain at the low end of the accuracy range.

Extensions. Three next steps follow from the limits above. First, CD_M is differentiable in the flip rate it estimates, so it can enter a training objective as a penalty on commitment earlier than CD_P licenses, turning the measurement into a control. Second, the same sweep run on the intermediate checkpoints of a distillation run would date the onset of the pattern: the present comparison sees only the endpoints, so it cannot say whether early commitment appears from the first epoch or emerges once the student matches the teacher's accuracy. Third, the obstacle to open-ended tasks is the minimization over program families noted above. A tractable substitute is to fix a family in advance, for instance the proof steps of a formal verifier, and take CD_P from that family rather than from a single program; the bound then certifies against the schedule that family implies, which is weaker than the one C-RASP$^{\text{CoT}}$ gives but is available where no single program exists.