LLMs Faithfully and Iteratively Compute Answers During CoT: A Systematic Analysis With Multi-step Arithmetics

Keito Kudo[1,2], **Yoichi Aoki**[1,2], **Tatsuki Kuribayashi**[3,1], **Shusaku Sone**[1]
Masaya Taniguchi[2,1], **Ana Brassard**[2], **Keisuke Sakaguchi**[1,2], **Kentaro Inui**[3,1,2]
[1]Tohoku University, [2]RIKEN, [3]MBZUAI
{keito.kudo.q4, youichi.aoki.p2, sone.shusaku.r8}@dc.tohoku.ac.jp,
{tatsuki.kuribayashi, kentaro.inui}@mbzuai.ac.ae,
keisuke.sakaguchi@tohoku.ac.jp, {masaya.taniguchi, ana.brassard}@riken.jp

Abstract

This study investigates the internal information flow of large language models (LLMs) while performing chain-of-thought (CoT) style reasoning. Specifically, with a particular interest in the faithfulness of the CoT explanation to LLMs' final answer, we explore (i) when the LLMs' answer is (pre)determined, especially before the CoT begins or after, and (ii) how strongly the information from CoT specifically has a causal effect on the final answer. Our experiments with controlled arithmetic tasks reveal a systematic internal reasoning mechanism of LLMs. They have not derived an answer at the moment when input was fed into the model. Instead, they compute (sub-)answers while generating the reasoning chain on the fly. Therefore, the generated reasoning chains can be regarded as faithful reflections of the model's internal computation.

1 Introduction

Modern large language models (LLMs), given a query from users, typically produce intermediate reasoning steps as well as a final answer. Such reasoning steps to achieve the answer, i.e., explanation, are helpful to grasp the model's underlying reasoning process and the plausible justification for the produced answer. One critical concern is how *faithful* the explanation, typically with a chain-of-thought (CoT) (Wei et al., 2022) style, is to their final answer, i.e., the causal relationship between the CoT process and the final answer. In particular, how models *internally* associate the CoT part and its following answer is an intriguing question in the interpretability field. In this study, we analyze the faithfulness between the CoT explanation and the model's final answer through the lens of (mechanistic) interpretability. This question involves two key subquestions—when, in the first place, models come up with (sub-)answers during CoT-style reasoning, and how these are referred to internally. One possible strategy, for example, would be to reach the final answer even during reading problem statements before CoT generation (the first pass), and then the model produces a fluent explanation just to follow an expected format, and ultimately, the model simply refers to their predetermined answers. In this case, the final answer is no longer strictly faithful to the CoT explanation, and users should be aware of this limitation in interpreting the model's outputs.

In our first experiments, we exploratorily analyze *when* (at each layer at each timestep) models represent answers internally, using linear probes. To prevent situations where CoT is not required for deriving the answer, we set up controlled testbeds of symbolic arithmetic reasoning and observed when/where probes could accurately extract and control the answer (Figure 1). By comparing accuracies across each timestep, one can observe at which point models internally start being informative (the values of the variables are encoded in the hidden states as linearly separable representations) to the probes, illustrating the model's internal reasoning pattern. We found that models have not derived an answer at the moment they first read the problem; instead, they obtain (sub-)answers while generating the reasoning chain, suggesting the faithfulness of CoT to the final answer.

Based on the probing results, we further conducted causal intervention analyses to clarify the causal relationship between the model's internal representations and answers (§ 4). We found that, after CoT generation, the model's final answer can be changed by causal intervention to internal representations for the CoT explanation part, while not for the first pass part. The discovered causal graph follows recency bias, i.e., each reasoning step in CoT causally depends on its previous step, and the final answer heavily relies on the latter part of CoT. Thus, we tentatively conclude that models derive an answer during CoT on the fly, and the gener-

Findings of the Association for Computational Linguistics: EACL 2026, pages 1114–1153
March 24-29, 2026

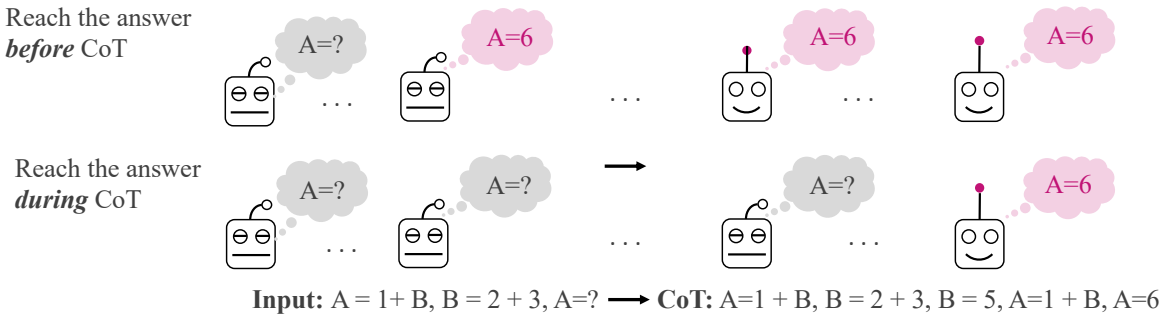

Figure 1: Using linear probes, we investigated at which time during the LLM's problem-solving process it is possible to determine the values of each variable, illustrating the model's problem-solving process. Our analysis indicates that LLMs come up with (sub-)answers during CoT. This conclusion is also consistent with the findings from the causal experiments in § 4.

ated reasoning chains can be regarded as faithful reflections of the model's final answer. [1]

2 General settings

2.1 Arithmetic problems

We prepared a synthetic dataset of multi-hop arithmetic problems similarly to Kudo et al. (2023) and Yu (2025). Each problem instance consists of strings of assignments (e.g., `A=1`) and operations (e.g., `B=1+3` or `B=1+A`) ending with a query for a variable's value (e.g., `B=?`). By using this synthetic dataset, we can conduct controlled experiments across different levels of problem complexity and confirm the generality of the obtained results. Moreover, by training and evaluating probes for each common problem format (level), we can conduct fine-grained, token-level analyses. Such controlled analyses are difficult with datasets written in natural language.

We specifically defined five complexity levels (Table 1), depending on (i) how many equations need to be resolved to reach the answer (#Step in Table 1), (ii) how many variables' values cannot be immediately resolved (and thus pended to a stack) in their first appearance when incrementally reading the problem from left to right (#Stack), (iii) and the number of unnecessary distractor equations (#Dist.). For example, in the level 5 example in Table 1, where #Step is three and #Stack is two, calculating `A` requires at least three steps of reasoning: `C(=1+2)=3`, `B(=2+3)=5`, and then `A(=1+5)=6`, and two variables need to be resolved before reaching `A`: `B` and `C`.

[1]Code and data are available at https://github.com/keitokudo/faithful-cot-multistep-arithmetic

Notation. Formally, let v denote a variable name sampled from the 26 letters of the English alphabet $\Sigma = \{\mathrm{a}, \mathrm{b}, \mathrm{c}, \ldots, \mathrm{z}\}$, and d denote a number sampled from the set of decimal digits $D = \{0, 1, 2, \ldots, 9\}$. Each instance consists of multiple equations $[e_1, e_2, \cdots, e_n]$ followed by a final query q. Each equation follows the format $v = d$, $v = d \pm d$, or $v = d \pm v$. We denote i-th variable in an instance from the left as v_i; e.g., in the Level 5 example in Table 1, $v_1 =$ `A`, $v_2 =$ `B`, and $v_3 =$ `C`.[2] The value assigned to a variable v_i is denoted as $\$\{v_i\} \in D$.

Data constraints. We ensure that $\$\{v_i\}$ for any v_i is also a single-digit number, and $\$\{v_i\}$ is constant within the same instance (i.e., we exclude cases such as `A=1+2,A=B+2,B=6`). All samples of the same complexity level follow the exact same format except for the actual numbers, variable names, and operators. In other words, an answer $\$\{v_i\}$ in each level can be obtained with exactly the same procedure. For example, first calculate $\$\{v_3\}$, and then calculate $\$\{v_2\}$ with $\$\{v_3\}$, and then finally calculate $\$\{v_1\}$ with $\$\{v_2\}$. Non-duplicated instances are created for each level by varying the variable names, numbers, and operators appearing in the equations. We use 10,000 instances to train probing classifiers and 2,000 to test their accuracy. To prevent the probe from simply memorizing (sub-)answers, the training and test sets were constructed with no overlap in arithmetic expressions.[3]

[2]For brevity, all examples in this paper use the uppercased variables `A`, `B`, `C`, and only the operator `+`, but the actual instances have a variety in the variable names and operator types.

[3]The overlap is computed at the equation-level. For example, if `1+2` appears in the training set, then `1+2` does not appear in any test instances.

Level	INPUT	OUTPUT	#Step	#Stack	#Dist.
1	A = 1 + B $_{-3}$, B = 2 $_{-2}$; A =? $_{-1}$	A = 1 + B $_{0}$, B = 2 $_{1}$, A = 1 + B $_{2}$, A = 1 + 2 $_{3}$, A = 3 $_{4}$	1	1	0
2	A = 2 + 3 $_{-3}$, B = 1 + A $_{-2}$; B =? $_{-1}$	B = 1 + A $_{0}$, A = 2 + 3 $_{1}$, A = 5 $_{2}$, B = 1 + A $_{3}$, B = 1 + 5 $_{4}$, B = 6 $_{5}$	2	0	0
3	A = 1 + B $_{-3}$, B = 2 + 3 $_{-2}$; A =? $_{-1}$	A = 1 + B $_{0}$, B = 2 + 3 $_{1}$, B = 5 $_{2}$, A = 1 + B $_{3}$, A = 1 + 5 $_{4}$, A = 6 $_{5}$	2	1	0
4	A = 1 + B $_{-4}$, B = 2 + 3 $_{-3}$, C = 4 + 5 $_{-2}$; A =? $_{-1}$	A = 1 + B $_{0}$, B = 2 + 3 $_{1}$, B = 5 $_{2}$, A = 1 + B $_{3}$, A = 1 + 5 $_{4}$, A = 6 $_{5}$	2	1	1
5	A = 1 + B $_{-4}$, B = 2 + C $_{-3}$, C = 1 + 2 $_{-2}$; A =? $_{-1}$	A = 1 + B $_{0}$, B = 2 + C $_{1}$, C = 1 + 2 $_{2}$, C = 3 $_{3}$, B = 2 + C $_{4}$, B = 2 + 3 $_{5}$, B = 5 $_{6}$, A = 1 + B $_{7}$, A = 1 + 5 $_{8}$, A = 6 $_{9}$	3	2	0

Table 1: Examples of arithmetic reasoning tasks used in our experiments at each complexity level. #Step indicates the number of required operations to reach the final answer. #Stack indicates how many variables' values are not immediately determined in their first appearing equation. #Dist. is the number of unnecessary distractor equations. The number (e.g., $-\mathbf{3}$) indicated in the lower right corner of each equation represents the equation's position. This position is used as a reference point for calculating t^*_{eq} in § 3.2.

2.2 CoT-style inference

Given the input described in § 2.1, the model generates CoT z and a final answer y. Henceforth, we refer to the part before CoT as the INPUT x and the CoT-reasoning part (y, z) as the OUTPUT of an instance, as shown in the right part of Table 1. For example, for the Level 1 instance in Table 1 (topmost), $x =$ "`A=1+B, B=2`," $z =$ "`A=1+B, B=2, A=1+2, A=`", and $y =$ "3." We demonstrate three examples in the same problem level for the model to inform the problem setting and expected CoT-style reasoning style (Table 1). That is, the task is, given a demonstration of three examples, and an INPUT x, to generate intermediate steps z and derive a final answer y. As a sanity check, we confirmed that the target models could follow the expected OUTPUT format and solve the tasks with nearly 100% accuracy in this setting (§ B.1).

2.3 Research scope

We are particularly concerned about the faithfulness between the CoT part z and its following final answer y, since these two parts are typically shown to users, and thus are of primary concern regarding the model's reliability from the user's perspective. In contrast, some recent studies (Afzal et al., 2025; Cox, 2025; Ye et al., 2025) have investigated whether the CoT z depends on the internal state of the model x before CoT begins (whether the CoT z is *post hoc* or not). This causal relationship $x \to z$ is out of scope in this study, while we are still interested in whether the model x has a predetermined answer y in order to distinguish the possibilities of $x \to y$ and $z \to y$, i.e., which part is causally relied on to the final answer y.

3 Linear probing

To begin with the analysis, it will be good to know where the final answer, or the necessary sub-answer for it, can possibly come from. We first analyze where final-/sub- answers can be extracted from the model's internal representations, using linear probes.

Position t. We denote a token position within the entire concatenated sequence $x \oplus z \oplus y$ with $t \in \mathbb{Z}$. The position t is relative to CoT; that is, t is zero at the moment CoT begins, negative within the INPUT, and positive within the OUTPUT. Similarly, we assign an equation position $t_{\text{eq}} \in \mathbb{Z}$ to each equation in the INPUT and OUTPUT (subscripts on the underlines in Table 1).

3.1 Training linear probes

Linear probes (Alain and Bengio, 2017a) are trained to identify when the model derives the answer internally. We separately train a probe for each combination of token position $t \in \mathbb{Z}$, layer depth $l \in \mathbb{N}$, and the variable of interest $v_i \in \Sigma$ in each level of the problem. That is, each probe is solely responsible for the position (t, l, v_i) to identify the answer $\hat{\$\{v_i\}}$ represented. Formally, given a model's d-dimensional hidden state $\boldsymbol{h}_{t,l} \in \mathbb{R}^d$, the probing classifier $f_{t,l,v_i}(\cdot) : \mathbb{R}^d \to D$ is trained to predict the correct answer $\$\{v_i\}$. If a probe f_{t,l,v_i} achieves high accuracy, this suggests that the answer for a particular variable $\$\{v_i\}$ is already computed at the corresponding position (t, l). The probe f_{t,l,v_i} is a single linear transformation:

$$\begin{aligned} \hat{\$\{v_i\}}_{t,l} &= f_{t,l,v_i}(\boldsymbol{h}_{t,l}) \\ &= \arg\max_{D} \boldsymbol{W}_{t,l,v_i} \boldsymbol{h}_{t,l} + \boldsymbol{b}_{t,l,v_i} \quad , \end{aligned} \tag{1}$$

where, $\boldsymbol{W}_{t,l,v_i} \in \mathbb{R}^{|D| \times d}$ and $\boldsymbol{b}_{t,l,v_i} \in \mathbb{R}^{|D|}$ are the trainable weight and bias parameters of the probe, respectively. The symbol $\hat{\cdot}$ refers to the model's predicted answer. We train the probes using stochastic gradient descent (Robbins, 1951) to minimize the cross entropy loss. The hyperparameters are listed in Table 8 in the appendix.

Each probe classifier is trained with 10,000 of $\boldsymbol{h}_{t,l}$ from training instances and then computes the accuracy with 2,000 hidden states from test instances.

3.2 Evaluation metrics

The probing results from all the token positions t and layers l are aggregated as follows:

$$t^*(v_i) = \min\{t \mid \max_l \mathrm{acc}(t, l, v_i) > \tau\}, \quad (2)$$

where $\mathrm{acc}(t, l, v_i) \in [0, 1]$ denotes the probing accuracy of f_{t,l,v_i}. The position $t^*(v_i) \in \mathbb{Z}$ indicates when the first time the probing classifier achieved a reasonable accuracy above τ. We set $\tau = 0.9$ in our main experiment, and we show additional results with different thresholds in Appendix B.3. As a more coarse but comprehensive value, we also report $t^*_{\mathrm{eq}}(v_i)$, indicating which equation t_{eq} in the input x the position $t^*(v_i)$ falls into. Given that the t (and t_{eq}) is relative to the CoT-beginning position, if $t^*_{(\mathrm{eq})}(v_i)$ is negative, the value $\${v_i\}$ is computed before CoT begins. In Figure 2, for example, both $t^*(\texttt{A})$ and $t^*(\texttt{B})$ are positive.

As complementary information of the position, we also report two types of accuracy:

$$\mathrm{Acc}_{\prec \mathrm{CoT}}(v_i) = \max_{t<0,l} \mathrm{acc}(t, l, v_i), \quad (3)$$

$$\mathrm{Acc}_{\succ \mathrm{CoT}}(v_i) = \max_{t\geq 0,l} \mathrm{acc}(t, l, v_i). \quad (4)$$

If $\mathrm{Acc}_{\prec \mathrm{CoT}}(v_i)$ is sufficiently high, $\${v_i\}$ is resolved internally before CoT begins. Conversely, if $\mathrm{Acc}_{\prec \mathrm{CoT}}(v_i)$ is low and $\mathrm{Acc}_{\succ \mathrm{CoT}}(v_i)$ is high, the answer is derived while performing CoT reasoning.

3.3 Results

Across task levels. We first analyze Qwen2.5-7B (Qwen Team, 2024) across the five task levels. Table 2 reports the position t^*_{eq} for each variable and its lower bound $t^{\mathrm{inf}}_{\mathrm{eq}}$ where the answer can be first identifiable. In most cases, (sub)answers are represented in the model's hidden states corresponding to the OUTPUT part ($t^*_{\mathrm{eq}} > 0$) in a linearly separable form, even when deriving relatively simple sub-answers, e.g., $\${B\}$ in Level 2. The exceptions are: (i) v_2 in Level 1, and (ii) v_3 in Level 4. Here, v_2 in Level 1 requires no computation (#Steps$= 0$), and v_3 in Level 4 is a distractor that is not needed to derive the final answer. To sum up, all calculations (at least 1 step) required to obtain the final answer are resolved **during CoT**, suggesting that it's unlikely to refer to a predetermined answer as model's final answer.

	Variable		When (↓)		Acc. (↑)	
Level	variable	#Step	t^*_{eq}	$t^{\mathrm{inf}}_{\mathrm{eq}}$	$\prec$ CoT	$\succ$ CoT
1	v_1 (A)	1	4	-2	35.8	100
	v_2 (B)	0	-2	-2	100	100
2	v_1 (A)	1	2	-3	49.2	100
	v_2 (B)	2	5	-2	21.6	94.7
3	v_1 (A)	2	5	-2	17.9	97.4
	v_2 (B)	1	2	-2	50.5	100
4	v_1 (A)	2	5	-3	17.2	100
	v_2 (B)	1	2	-3	47.7	100
	v_3 (C)	1	N/A	-2	43.7	23.7
5	v_1 (A)	3	9	-2	18.1	100
	v_2 (B)	2	6	-2	22.6	100
	v_3 (C)	1	3	-2	50.6	100

Table 2: The results of Qwen2.5-7B on the five levels. The t^*_{eq} is the time when the model comes up with the correct answer (see § 3.2). The $t^{\mathrm{inf}}_{\mathrm{eq}}$ column indicates the lower bound of t^*_{eq} score. The $\prec$ CoT and $\succ$ CoT scores correspond to the accuracies introduced in § 3.2. N/A indicates that the threshold τ was not exceeded at any position t.

Across models. We analyzed nine models (Table 3) on the Level 3 task. Similarly to Table 2, we observed $t^*_{\mathrm{eq}} > 0$. This pattern holds for other tasks and for different thresholds τ (Tables 9–23 in § B.3). On the other hand, in the INPUT part—particularly the latter half of equation -2 (B $= 2 + 3$, $\underline{}_{-2}$)—the pre-CoT accuracy $\mathrm{Acc}_{\prec \mathrm{CoT}}(v_i)$ increases modestly, peaking around 60%.

3.4 Analysis

In this section, we present the results of a prompt-sensitivity analysis and an error analysis. We also analyze the effects of factors such as reasoning-chain format (including implicit reasoning). For details, see § A.

Equation order. The difference between these two tasks lies in the order of the equations. Level 3 tasks involve forward references, whereas Level 2 tasks do not. An autoregressive language model

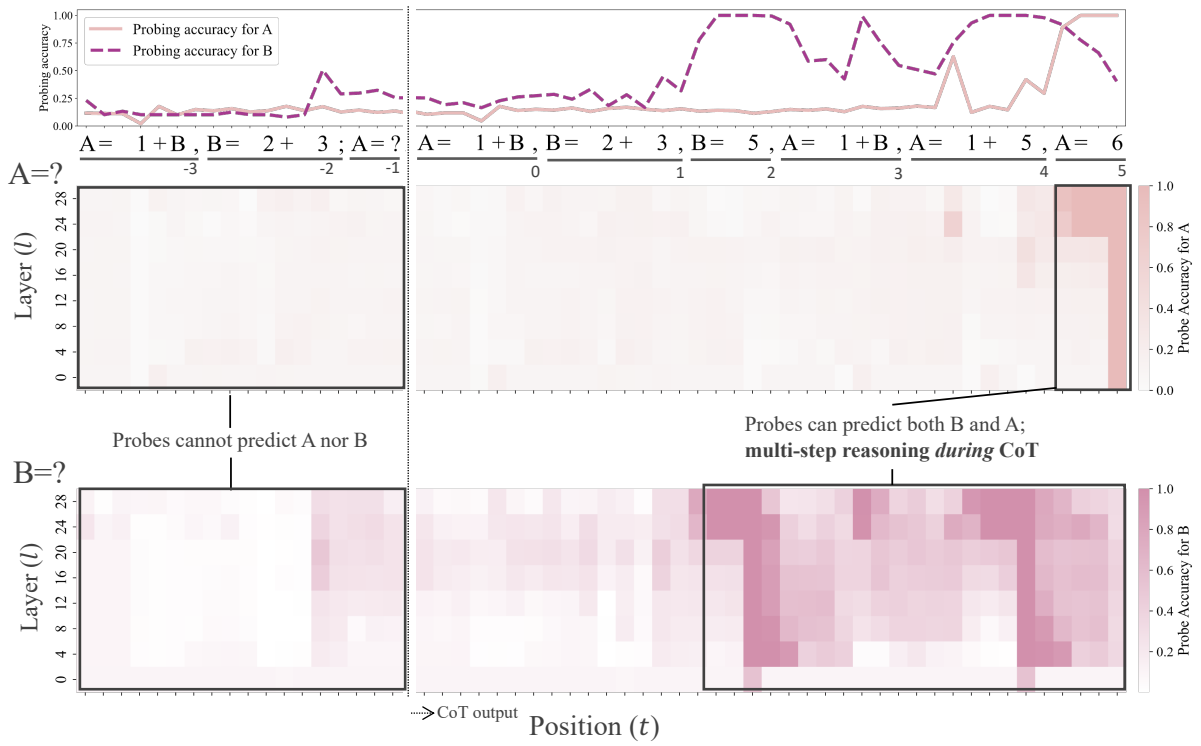

Figure 2: Probing results for Qwen2.5-7B at the Level 3 task. The heatmaps in the lower section represent the accuracy of probes computed on the evaluation set. Each cell shows the probing accuracies in each token t, layer l. The upper part indicates the maximum probing accuracy achieved at each token position t. The input sequence below the line graphs is just an example; in the actual evaluation set, each variable name, number, and operator are randomly sampled from $(D, \Sigma, \{+, -\})$.

can attend only to past inputs. Therefore, the model may be able to reach the final answer more quickly on Level 2 tasks than on Level 3 tasks. In practice, Table 2 indicates that the probe predicts the answer during CoT at both levels. Therefore, the impact of equation order (dependency structure) on the model's internal mechanism is thought to be limited.

Distractors. In the Level 4 task (see Table 1), the variable v_3 (`C`) is a distractor, that is, $\${v_3}$ is not necessary to derive the final answer. The models can infer this fact from the few-shot examples. According to Table 2, the $\mathrm{Acc}(v_3)$ in Level 4 was at most 44%, a relatively low accuracy. From this result, we can see that, unlike the variables required to derive the final answer, v_3 is not encoded in a simple form that can be extracted by a linear transformation alone. This suggests the possibility that the model employs an efficient internal mechanism that does not derive variables unnecessary for obtaining the final answer. It is also consistent with the finding that the CoT is faithful to the final answer.

Effect of depth of reasoning. The difference between Level 3 and Level 5 is whether two or three calculation steps are required to reach the final answer. Qualitatively, the probe's prediction accuracy for each variable increases a few tokens before the answer appears in the output text, indicating alignment between the internal state and the output. This finding supports the conclusion that the generated explanations are faithful to the internal state and that this conclusion generalizes across levels.

Prompt differences. We investigated whether the format of few-shot prompting influences the model's internal reasoning patterns. We analyzed the differences between two scenarios: *1. Same level prompting* and *2. General prompting*. In same level prompting, we provide three equations that match the level of the evaluation task, following the general setting (§ 2). In contrast, under general prompting the model receives 50 randomly generated demonstrations (three equations each).[4] Table 4 reports t^*_{eq} for both strategies for Level 4

[4]We increased the number of demonstrations in the general prompts because task performance was otherwise insufficient.

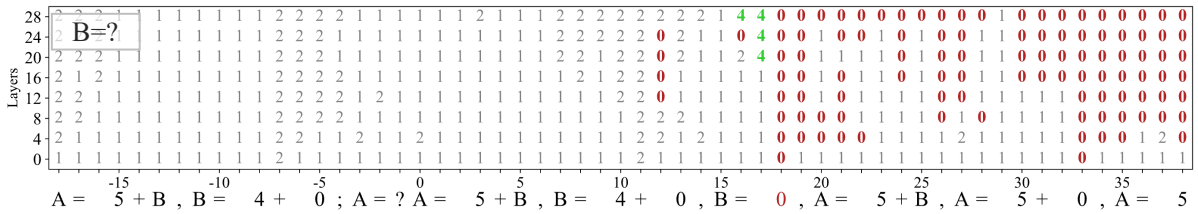

Figure 3: Error analysis of cases where Llama3.2-3B generated *incorrect* answers. The vertical axis represents the index of the transformer layer. The horizontal axis represents the tokens input to the model over time. The numbers highlighted in green represent the gold labels for the predictions, while those highlighted in red denote the values incorrectly generated by the model.

		When (↓)		Acc (↑)	
	Var.	t^*_{eq}	t^*	≺ CoT	≻ CoT
Qwen2.5 (7B)	v_1 (A)	5	36	17.9	100
(Qwen Team, 2024)	v_2 (B)	2	16	50.5	100
Qwen2.5 (14B)	v_1 (A)	5	35	17.8	100
(Qwen Team, 2024)	v_2 (B)	2	16	50.5	100
Qwen2.5 (32B)	v_1 (A)	5	36	17.8	100
(Qwen Team, 2024)	v_2 (B)	2	15	67.4	100
Qwen2.5-Math (7B)	v_1 (A)	5	35	18.6	100
(Yang et al., 2024a)	v_2 (B)	2	15	56.1	100
Yi1.5 (9B)	v_1 (A)	5	41	17.8	100
(Young et al., 2024)	v_2 (B)	2	18	36.9	100
Yi1.5 (34B)	v_1 (A)	5	41	22.4	100
(Young et al., 2024)	v_2 (B)	2	18	37.4	100
Llama3.1 (8B)	v_1 (A)	5	35	26.0	100
(Dubey et al., 2024)	v_2 (B)	2	16	29.6	100
Llama3.2 (3B)	v_1 (A)	5	36	17.8	93.2
(Dubey et al., 2024)	v_2 (B)	2	17	33.2	95.4
Mistral-Nemo (12B)	v_1 (A)	5	36	17.8	100
(Mistral AI Team, 2024)	v_2 (B)	2	16	28.9	100

Table 3: Results for various models on the Level 3 task. The t^* column shows the token-wise time (described in § 3.2), and the other columns are the same as Table 2. The t^* and t^*_{eq} scores that are the same as their lower bounds are bolded.

	Variable		When (↓)		Acc (↑)	
Setting	Variable	#Step	t^*_{eq}	$t^\dagger_{\text{eq}}$	≺ CoT	≻ CoT
Same level	v_1	2	5	−3	17.2	100
	v_2	1	2	−3	47.7	100
	v_3	1	N/A	−2	43.7	23.7
General	v_1	2	5	−3	18.1	94.7
	v_2	1	2	−3	49.4	100
	v_3	1	N/A	−2	43.3	28.1

Table 4: Evaluation results of same level prompting and general prompting. Each column is the same as Table 2.

task. We observe no difference in t^*_{eq}, suggesting that the few-shot prompt format has minimal effect on the model's internal reasoning patterns.

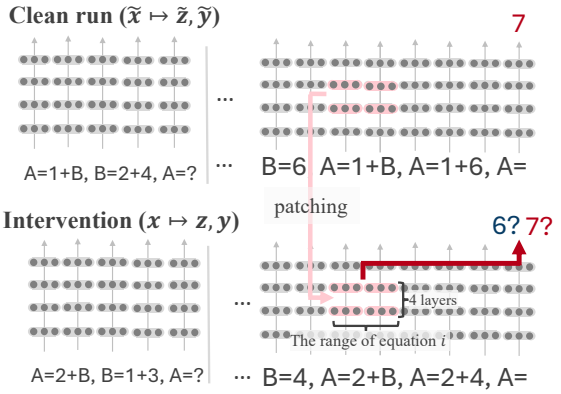

Figure 4: Overview of the causal intervention experiment. First, we perform normal inference (Clean run) and cache its hidden states. Subsequently, we evaluate whether the output changes by replacing some of the hidden states of a model solving a different problem with the cached hidden states.

Behavior in incorrect instances. Figure 3 shows the top-1 predictions of the probe for B in instances where Llama3.2-3B produced *incorrect* answers for level 3 (where B happened to be 4). When the model produced an incorrect answer, we often found that the correct answer had appeared at an earlier decoding step (see § B.2 for the probe's predictions on additional samples). The underlying reason for these mistakes remains unclear, but through our probing framework, one can track how such an incorrect answer is propagated to the subsequent generation, which might be practically helpful in debugging the model internals.

4 Causal interventions

From our probing experiments in § 3, we tentatively concluded that the final answer is determined after CoT-style generation begins. That is, CoT is faithful to the final answer. In this section, we cor-

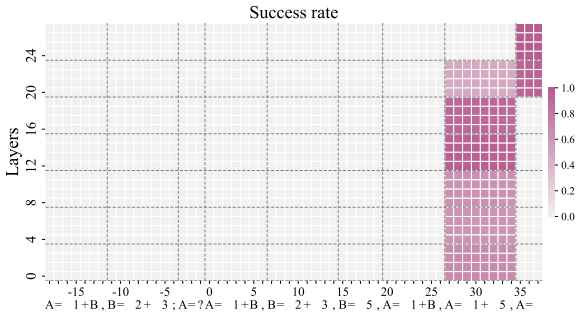

Figure 5: Results of the causal intervention on Qwen2.5-7B. Each grid cell shows the success rate when the final answer y ($\underline{\text{A} = \mathbf{6}_{\mathbf{5}}}$) is the target token.

roborate this conclusion using causal intervention analysis and further clarify how input information flows internally while generating CoT reasoning and the final answer.

4.1 Settings

Activation patching. We employ activation patching (Vig et al., 2020; Meng et al., 2022; Zhang and Nanda, 2024), which is a widely adopted technique for causal intervention analysis. To inspect the causal relationship between specific hidden states $\boldsymbol{h}_{t,l}$ and a final answer y, we compare two generation scenarios: (i) the ordinary inference and (ii) the intervened inference. In the latter scenario, we replace the specific hidden states $\boldsymbol{h}_{t,l}$ with other variants $\tilde{\boldsymbol{h}}_{t,l}$ obtained from the same model but with a different input $\tilde{x}$ (Intervention in Figure 4).

The input x and $\tilde{x}$ have different correct answer y and $\tilde{y}$ as well as different chains z and $\tilde{z}$, respectively. For example, for the triple ($\tilde{x}$ = "A=1+B,B=2+4;A=?," $\tilde{z}$ = "A=1+B,B=2+4,B=6,A=1+B,A=1+6,A=7," $\tilde{y}$ =7), one may use (x = "A=2+B,B=1+3;A=?," z = "A=2+B,B=1+3,B=4,A=2+B,A=2+4,A=6," y =6). If the model's output turns from y into $\tilde{y}$ or z_t into $\tilde{z}_t$ due to the intervention to $\boldsymbol{h}_{t,l}$ with $\tilde{\boldsymbol{h}}_{t,l}$, we can confirm the causal relationship between $\boldsymbol{h}_{t,l}$ and the original answer y. Henceforth, to precisely explain the setting, we denote the model's final output with intervention as $\hat{y}^{\text{patch}}$ and that without intervention as $\hat{y}$, respectively (y and $\tilde{y}$ denote respective gold answers). In the same way, we denote the generated reasoning chain with intervention as $\hat{z}_t{}^{\text{patch}}$ and that without intervention as $\hat{z}_t$, respectively.

Evaluation metrics. We report *Success Rate* as a metric for this experiment. The Success rate indicates how frequently (%) the intervened output $\hat{y}^{\text{patch}}$ aligns with the correct answer $\tilde{y}$. For reasoning chains, we report the Success rate for $\hat{z}_t{}^{\text{patch}}$ as well.

Patching targets. We specifically focus on Level 3 tasks and Qwen2.5-7B. The experimental results for the other models are reported in § B.4. Inspired by sliding window patching (Zhang and Nanda, 2024), we partition the hidden states into coarse grids, corresponding to each equation and every four layers, and perform activation patching on each grid separately (illustrated in Figure 6).[5] For every grid, we compute the *Success rate* by applying activation patching. We also examine multiple target tokens, specifically, at (i) the end of the equation 2 (z_{17} in $\underline{\text{B} = \mathbf{5}_{\mathbf{2}}}$ in Figure 2), (ii) the end of the equation 4 (z_{32} in $\underline{\text{A} = 1{+}\mathbf{5}_{\mathbf{4}}}$ in Figure 2), and (iii) the final answer (y). When we apply activation patching, we generate only the target token with greedy decoding while forced-decoding the context. Note that the above equations are examples. We create a test set of 2,000 instances for evaluation.

4.2 Results.

CoT is faithful to the final answer. Figure 5 presents the success rate at each grid position when the final answer y ($\underline{\text{A} = \mathbf{6}_{\mathbf{2}}}$) is the target token. The results show a strong causal dependence of the final answer on the OUTPUT part, whereas the INPUT part has only a limited effect. We observe the same pattern for other target tokens (e.g., z_{17} and z_{32}). This trend is broadly consistent across models (§ B.4). Furthermore, we focus on the generation of $\underline{\text{B} = \mathbf{5}_{\mathbf{2}}}$, which is a necessary sub-answer to reach the final answer (see Figure 6). Intervention to the immediately preceding equation $\underline{\text{B} = 2 + 3}_{\mathbf{1}}$ affects the generation of $\underline{\text{B} = \mathbf{5}_{\mathbf{2}}}$, but patching the equation $\underline{\text{B} = 2 + 3}_{\mathbf{2}}$ in the problem statement x (the region where a slight improvement in probing accuracy before the CoT was observed; $t = -11$ to $t = -4$) had almost no effect on the output of $\underline{\text{B} = \mathbf{5}_{\mathbf{2}}}$. Similarly, when we focus on the case where the target token is $\underline{\text{A} = 1{+}\mathbf{5}_{\mathbf{4}}}$, patching the equation $\underline{\text{B} = 2 + 3}_{\mathbf{2}}$ again resulted in almost no change in the output. Therefore, the influence of the information processed in the INPUT part on the sub-answers ($\text{B} = \mathbf{5}$) is limited. This tendency is especially strong in the Qwen family (except for Qwen2.5-Math-7B). By contrast, other model families show a weak causal relationship between $\underline{\text{B} = \mathbf{5}_{\mathbf{2}}}$ and the INPUT part (§ B.4). This

[5] All the hidden states in each grid are intervened at once.

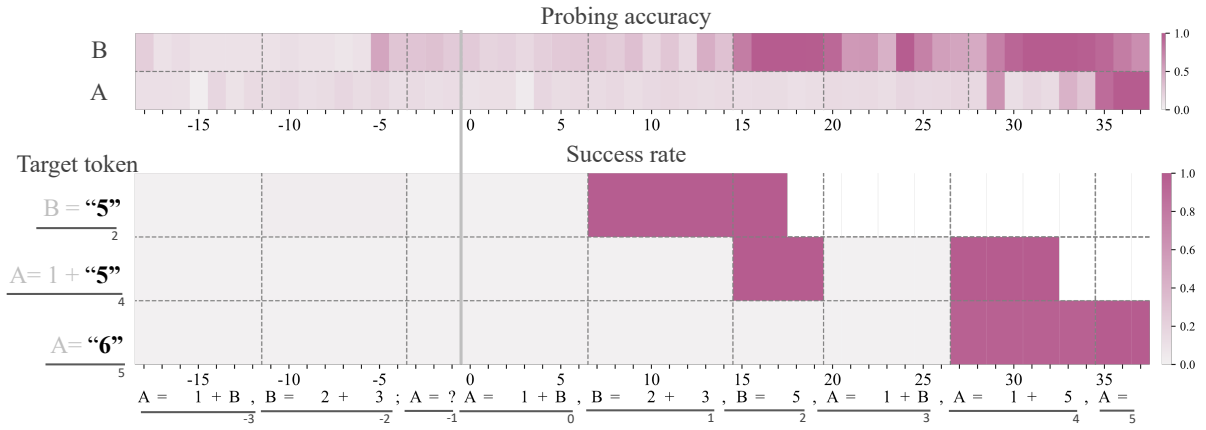

Figure 6: The upper part is the accuracy of probes, as shown in Figure 2. The lower part is the result of max pooling the Success rates from Figure 5 in the layer direction.

aligns with the results observed in probing experiments § 3.3.

Recency bias. We further investigate which parts of CoT output causally depend on which input. The bottom part of Figure 6 suggests strong *recency bias* in the causal relationship between hidden states and output tokens for Qwen2.5-7B. That is, intervention succeeded only when the target hidden state is (i) in the same equation as the target token, (ii) in the last equation where necessary information is stated (e.g., `B=2+3`→`B=5`), or (iii) in the last equation where a value of a relevant variable is explicitly stated (e.g., `B=5`→`A=1+5`). This finding suggests strong recency bias in the internal process of LLMs' multi-hop reasoning. This aligns with our finding that models derive an answer during CoT on the fly, and the generated reasoning chains can be regarded as faithful reflections of the model's final answer.

5 Related Work

Post hoc nature of CoT. Several prior studies investigate similar questions (Afzal et al., 2025; Ye et al., 2025; Cox, 2025; Lewis-Lim et al., 2025). For instance, Cox (2025) examined whether CoT explanations are post-hoc or not, specifically, confirmed that, in a binary classification task, the correct label can be predicted from the hidden state at the end of the input (prompt). In their setting, the CoT z is (freely) produced from the intervened hidden state x, while we force-decode the CoT chain. In this sense, their scope is rather on the causality of $x \rightarrow z$, and thus, as stated in § 2.3, the scope is slightly different. They also showed that, in some tasks (e.g., sports understanding), the final answer can be predicted with the model's final state in the prompt, apparently contradicting our probing experiments, but their tasks would be answerable with factual or commonsense knowledge without CoT, and in their specific logical deduction task—where on-the-fly reasoning is required, similarly to our arithmetic tasks—the probe indeed struggles to predict the final answer, similarly to our results. Afzal et al. (2025) also suggests the model reaches the answer internally before CoT begins, but their experimental design is somewhat different from ours. Specifically, they try to predict whether the model will succeed/fail the task before CoT begins as a binary problem, rather than tracing how models reach the correct answer during CoT or identifying where the concrete answer comes up with during CoT reasoning. Ye et al. (2025) also find that the model, in advance, anticipates which components (variables) will be required to reach the final answer by tracking whether a variable has been computed, and how models retain computed values during reasoning. Although the experiments are similar, they do not investigate causal relationships and therefore do not fully answer the question of whether the reasoning chain is faithful.

Faithfulness of CoT. Paul et al. (2024); Bentham et al. (2024) similarly evaluate the faithfulness of CoT to the final answer. They test the consistency between the final answer and the CoT by intervening on the CoT text. Specifically, they replace it with counterfactual content and then check whether the final answer changes. They report that CoT is not sufficiently faithful to the final answer. In addition, Turpin et al. (2023); Chen et al. (2025) define CoT as faithful when the following two con-

ditions hold: (i) The CoT explicitly refers to the hint provided in the prompt. (ii) The model relies on the hint to reach the answer (that is, without the hint, the model would not have produced the hinted answer). Using these conditions, they evaluate faithfulness indirectly by testing whether the model's internal computation depends on the hinted information.

By contrast, in our work, we trace the model's internal reasoning more directly by probing the values of intermediate answers, and thus track the faithfulness of CoT to the final answer using a more direct method and a controlled testbed than these prior studies. Through token-level probing, we analyze when and at which token position the reasoning is actually carried out. In this respect, our work differs in that it also allows us to observe the faithfulness of the internal states to the CoT text along the temporal axis.

Interpreting multi-hop reasoning in language models. Interpreting internal mechanisms of LLMs has been actively investigated (Conneau et al., 2018; Tenney et al., 2019; Niven and Kao, 2019; nostalgebraist, 2020; Geva et al., 2023; Lieberum et al., 2024; Ghandeharioun et al., 2024; Ferrando and Voita, 2024). They revealed, for example, specialized attention heads responsible for specific operations (Cabannes et al., 2024) or decision-making, copying, and induction (Dutta et al., 2024). With a more concrete example, Yang et al. (2024c) showed that, even during the first pass of the problem statements such as *The mother of the singer of Thriller is* ___, language models first resolve a *bridge entity*, Stevie Wonder in this case, then identify the final answer. This study is more focused on the difference between the first pass of the problem statements (before CoT generation) and their second pass involving explicit problem solving (while CoT generation).

Arithmetic representations in LLMs. How models handle numerical information has also been closely studied. For instance, Heinzerling and Inui (2024) used partial least squares regression (Wold et al., 2001) to demonstrate that numeric attributes, such as birth years and population numbers, are encoded as monotonic representations in the activation space of LLMs and can be manipulated with interventions. In turn, Stolfo et al. (2023) showed that, in autoregressive LLMs, the operations and numerical information necessary for solving quantitative reasoning are processed in the lower layers of the model, and these results are used by the attention layers to predict the final calculation outcomes. Zhu et al. (2025) studied the representation of numbers in language models' hidden states during single-hop arithmetic tasks (e.g., `What is the sum of 12 and 34?`). Their analysis revealed that numerical information is encoded linearly within the hidden states and demonstrated that individual digits could be manipulated independently. In this study, we add to this literature by introducing incremental arithmetic problem solving, i.e., what numerical information is contained in the model's hidden states at each time step of multi-hop arithmetic reasoning.

Model interpretability methods. Linear probing (Alain and Bengio, 2017b) is one of the representative methods for analyzing the internal representations of neural models—a small model predicts a specific feature from them, thereby determining whether the input contains information about that feature. In this study, we use them to derive the models' intermediate answers. The causality with the model's output can be further verified by examining if a model's predictions change when the hidden states are intervened (Li et al., 2023; Wu et al., 2023). One representative intervention method is activation patching (Vig et al., 2020; Meng et al., 2022; Zhang and Nanda, 2024), where hidden states obtained from one model instance are transplanted onto another during inference to change its predictions. Such techniques can be applied as a way to control model behavior in practical scenarios such as mitigating inherent biases (Zhao et al., 2019; Ganguli et al., 2023; Yang et al., 2024b). Here, we employ activation patching to validate the plausibility of the probing results.

6 Conclusions

We conducted probing experiments to determine when (sub)answers are derived during the CoT-style reasoning, using synthetic arithmetic problems as a controlled testbed. Across multiple models and task difficulties, we found that models tend to resolve the necessary (sub)answer after the CoT begins, i.e., computing the answer on the fly during CoT generation. Moreover, causal experiments support that the intervention to the CoT part specifically and causally impacts the final answer; that is, we conclude that the CoT is faithful to the final answer at least in our controlled experimental settings.

Limitations

Variety of tasks We analyzed the internal reasoning patterns of language models using synthetic arithmetic reasoning tasks. The use of synthetic data allows for more detailed control compared to experiments on natural language tasks. However, vocabulary and expression diversity, for example, are limited compared to natural language tasks. Therefore, conducting similar analyses on more realistic reasoning tasks in natural language will verify whether the results of this study apply to other broader, realistic contexts as well. In addition, we focused on the tasks that presumably require step-by-step reasoning as the CoT process, but as suggested in existing studies (§ 5), the situation may be different if the task is, in the first place, so simple that CoT is no longer needed.

Probing methods Interpreting internal mechanisms of LLMs using probing has been actively conducted in our field (Conneau et al., 2018; Tenney et al., 2019; Campbell et al., 2023; Li et al., 2023); however, there are criticisms regarding the validity of some probing approaches (Liu et al., 2023; Burns et al., 2023). One way to overcome such concerns will be to analyze the generality of obtained results through more diverse methodologies (Gurnee et al., 2023; Bricken et al., 2023).

Causal intervention purity Hidden states carry mixed information (e.g., features of the input text itself). Thus, even though activation patching is standard in prior work, we cannot fully rule out noise introduced by this intervention.

Ethics statement

This paper will not raise particular ethical concerns, considering that (i) no human experiments were conducted, and (ii) our tasks do not involve ethically sensitive topics.

Acknowledgments

This work was supported by JST CREST Grant Number JPMJCR20D2, JSPS KAKENHI Grant Numbers JP25KJ0615 and JP25K03175, JST SPRING Grant Number JPMJSP2114, Google Research grant, and the Nakajima Foundation. Ana Brassard's contribution was supported by a RIKEN Incentive Research Project (FY2024). Part of this work was carried out using the computer resource offered under the category of "General Projects" by Research Institute for Information Technology, Kyushu University. We thank the member of the Tohoku NLP Group for their cooperation in this research.

References

Anum Afzal, Florian Matthes, Gal Chechik, and Yftah Ziser. 2025. Knowing before saying: LLM representations encode information about chain-of-thought success before completion. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 12791–12806, Vienna, Austria. Association for Computational Linguistics.

Guillaume Alain and Yoshua Bengio. 2017a. Discovering latent knowledge in language models without supervision. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net.

Guillaume Alain and Yoshua Bengio. 2017b. Understanding intermediate layers using linear classifier probes. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net.

Oliver Bentham, Nathan Stringham, and Ana Marasovic. 2024. Chain-of-thought unfaithfulness as disguised accuracy. *Transactions on Machine Learning Research*. Reproducibility Certification.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nicholas L. Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Alex Tamkin, Karina Nguyen, Brayden McLean, and 5 others. 2023. Towards monosemanticity: Decomposing language models with dictionary learning. In *Anthropic*.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. 2023. Discovering latent knowledge in language models without supervision. In *The Eleventh International Conference on Learning Representations*.

Vivien Cabannes, Charles Arnal, Wassim Bouaziz, Xingyu Alice Yang, Francois Charton, and Julia Kempe. 2024. Iteration head: A mechanistic study of chain-of-thought. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

James Campbell, Phillip Guo, and Richard Ren. 2023. Localizing lying in Llama: Understanding instructed dishonesty on true-false questions through prompting, probing, and patching. In *Socially Responsible Language Modelling Research*.

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. 2025. Reasoning models don't always say what they think. *CoRR*, abs/2505.05410.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia. Association for Computational Linguistics.

Kyle Cox. 2025. Post-hoc reasoning in chain of thought. https://www.lesswrong.com/posts/ScyXz74hughga2ncZ/post-hoc-reasoning-in-chain-of-thought. LessWrong. Accessed: 2025-10-03.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, and 82 others. 2024. The Llama 3 herd of models. *CoRR*, abs/2407.21783.

Subhabrata Dutta, Joykirat Singh, Soumen Chakrabarti, and Tanmoy Chakraborty. 2024. How to think step-by-step: A mechanistic understanding of chain-of-thought reasoning. *Transactions on Machine Learning Research*.

Javier Ferrando and Elena Voita. 2024. Information flow routes: Automatically interpreting language models at scale. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17432–17445, Miami, Florida, USA. Association for Computational Linguistics.

Deep Ganguli, Amanda Askell, Nicholas Schiefer, Thomas I Liao, Kamilė Lukošiūtė, Anna Chen, Anna Goldie, Azalia Mirhoseini, Catherine Olsson, Danny Hernandez, and 1 others. 2023. The capacity for moral self-correction in large language models. *arXiv preprint arXiv:2302.07459*.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. 2023. Dissecting recall of factual associations in auto-regressive language models. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Asma Ghandeharioun, Avi Caciularu, Adam Pearce, Lucas Dixon, and Mor Geva. 2024. Patchscopes: A unifying framework for inspecting hidden representations of language models. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Wes Gurnee, Neel Nanda, Matthew Pauly, Katherine Harvey, Dmitrii Troitskii, and Dimitris Bertsimas. 2023. Finding neurons in a haystack: Case studies with sparse probing. *Transactions on Machine Learning Research*.

Benjamin Heinzerling and Kentaro Inui. 2024. Monotonic representation of numeric attributes in language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 175–195, Bangkok, Thailand. Association for Computational Linguistics.

Keito Kudo, Yoichi Aoki, Tatsuki Kuribayashi, Ana Brassard, Masashi Yoshikawa, Keisuke Sakaguchi, and Kentaro Inui. 2023. Do Deep Neural Networks Capture Compositionality in Arithmetic Reasoning? In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 1351–1362.

Samuel Lewis-Lim, Xingwei Tan, Zhixue Zhao, and Nikolaos Aletras. 2025. Analysing chain of thought dynamics: Active guidance or unfaithful post-hoc rationalisation? *CoRR*, abs/2508.19827.

Kenneth Li, Aspen K Hopkins, David Bau, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023. Emergent World Representations: Exploring a Sequence Model Trained on a Synthetic Task. In *The Eleventh International Conference on Learning Representations (ICLR)*.

Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, János Kramár, Anca D. Dragan, Rohin Shah, and Neel Nanda. 2024. Gemma scope: Open sparse autoencoders everywhere all at once on Gemma 2. *CoRR*, abs/2408.05147.

Kevin Liu, Stephen Casper, Dylan Hadfield-Menell, and Jacob Andreas. 2023. Cognitive dissonance: Why do language model outputs disagree with internal representations of truthfulness? In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4791–4797, Singapore. Association for Computational Linguistics.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, volume 35, pages 17359–17372. Curran Associates, Inc.

Mistral AI Team. 2024. Mistral NeMo.

Timothy Niven and Hung-Yu Kao. 2019. Probing neural network comprehension of natural language arguments. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4658–4664, Florence, Italy. Association for Computational Linguistics.

nostalgebraist. 2020. interpreting GPT: the logit lens. *LessWrong*.

Debjit Paul, Robert West, Antoine Bosselut, and Boi Faltings. 2024. Making reasoning matter: Measuring and improving faithfulness of chain-of-thought reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 15012–15032, Miami, Florida, USA. Association for Computational Linguistics.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Herbert E. Robbins. 1951. A stochastic approximation method. *Annals of Mathematical Statistics*, 22:400–407.

Alessandro Stolfo, Yonatan Belinkov, and Mrinmaya Sachan. 2023. A mechanistic interpretation of arithmetic reasoning in language models using causal mediation analysis. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7035–7052, Singapore. Association for Computational Linguistics.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Sam Bowman, Dipanjan Das, and Ellie Pavlick. 2019. What do you learn from context? probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations*.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel Bowman. 2023. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. In *Advances in Neural Information Processing Systems*, volume 36, pages 74952–74965. Curran Associates, Inc.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. 2020. Investigating gender bias in language models using causal mediation analysis. In *Advances in Neural Information Processing Systems*, volume 33, pages 12388–12401. Curran Associates, Inc.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Svante Wold, Michael Sjöström, and Lennart Eriksson. 2001. PLS-regression: a basic tool of chemometrics. *Chemometrics and Intelligent Laboratory Systems*, 58(2):109–130. PLS Methods.

Zhengxuan Wu, Atticus Geiger, Thomas Icard, Christopher Potts, and Noah Goodman. 2023. Interpretability at scale: Identifying causal mechanisms in alpaca. In *Thirty-seventh Conference on Neural Information Processing Systems*.

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. 2024a. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*.

Nakyeong Yang, Taegwan Kang, Stanley Jungkyu Choi, Honglak Lee, and Kyomin Jung. 2024b. Mitigating biases for instruction-following language models via bias neurons elimination. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9061–9073, Bangkok, Thailand. Association for Computational Linguistics.

Sohee Yang, Elena Gribovskaya, Nora Kassner, Mor Geva, and Sebastian Riedel. 2024c. Do large language models latently perform multi-hop reasoning? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10210–10229, Bangkok, Thailand. Association for Computational Linguistics.

Tian Ye, Zicheng Xu, Yuanzhi Li, and Zeyuan Allen-Zhu. 2025. Physics of language models: Part 2.1, grade-school math and the hidden reasoning process. In *The Thirteenth International Conference on Learning Representations*.

Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, Kaidong Yu, Peng Liu, Qiang Liu, Shawn Yue, Senbin Yang, Shiming Yang, Tao Yu, Wen Xie, Wenhao Huang, and 11 others. 2024. Yi: Open foundation models by 01.ai. *CoRR*, abs/2403.04652.

Yijiong Yu. 2025. Do LLMs really think step-by-step in implicit reasoning? *Preprint*, arXiv:2411.15862.

Fred Zhang and Neel Nanda. 2024. Towards best practices of activation patching in language models: Metrics and methods. In *The Twelfth International Conference on Learning Representations*.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Ryan Cotterell, Vicente Ordonez, and Kai-Wei Chang. 2019. Gender bias in contextualized word embeddings. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 629–634, Minneapolis, Minnesota. Association for Computational Linguistics.

Fangwei Zhu, Damai Dai, and Zhifang Sui. 2025. Language models encode the value of numbers linearly. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 693–709, Abu Dhabi, UAE. Association for Computational Linguistics.

	INPUT	OUTPUT
Simple	A = 1 + B _3, B = 2 + 3 _2; A =? _1	B = 5 0, A = 6 1
Implicit	A = 1 + B _3, B = 2 + 3 _2; A =? _1	A = 6 0

Table 5: Examples of arithmetic reasoning tasks used for reasoning chain format comparison. This position is used as a reference point for calculating t^*_{eq} in § 3.2.

	Variable		When (↓)		Acc (↑)	
Setting	Variable	#Step	t^*_{eq}	$t^\dagger_{\text{eq}}$	$\prec$ CoT	$\succ$ CoT
Simple CoT	v_1	2	1	5	34.8	99.7
	v_2	1	0	1	69.7	1.0
Implicit	v_1	2	N/A	N/A	40.8	80.7
	v_2	1	N/A	N/A	69.1	79.2

Table 6: Probing evaluation results with different reasoning chains. Each column is the same as Table 2.

A Alternative CoT formats.

For comparison, we also ran probing experiments on Qwen2.5-7B under two reasoning chain formats. Specifically, we defined two formats. The first is the Simple CoT setting, in which the reasoning chain outputs only the intermediate subresults. The second is the Implicit (reasoning) setting, which omits intermediate computation steps in the output text. Table 5 lists example equations.

The experimental results are shown in Table 6 and Figures 57 and 58. Under the Simple CoT setting, the model's task accuracy was 99.5%, whereas under the Implicit reasoning setting it was 77.8%[6]. As in the Simple CoT setting, intermediate results are linearly separable in the OUTPUT segment ($t^*_{\text{eq}} > 0$). This trend is consistent with the results obtained using the default CoT format in § 3.3. In the implicit reasoning setting, the accuracy did not reach the threshold at any position (both $\text{Acc}_{\prec\text{CoT}}$ and $\text{Acc}_{\succ\text{CoT}}$ are N/A). These results suggest that when the model fails to solve a problem, the corresponding (sub-)answers are also unlikely to be represented in its internal states. This, in turn, indicates that the answers decoded from internal states are faithful, in the sense that when the model cannot solve the problem, there may be no linearly decodable representation either.

	Level				
	1	2	3	4	5
Qwen2.5 (7B)	100.00	100.00	100.00	100.00	100.00
Qwen2.5 (14B)	100.00	100.00	100.00	100.00	98.25
Qwen2.5 (32B)	100.00	100.00	100.00	100.00	100.00
Qwen2.5-Math (7B)	100.00	100.00	100.00	100.00	100.00
Yi1.5 (9B)	100.00	100.00	100.00	100.00	100.00
Yi1.5 (34B)	100.00	100.00	100.00	100.00	100.00
Llama3.1 (8B)	100.00	100.00	100.00	100.00	99.40
Llama3.2 (3B)	99.95	99.75	93.15	90.90	38.45
Mistral-Nemo (12B)	100.00	100.00	99.95	100.00	99.85

Table 7: The performance of language models on the arithmetic reasoning tasks. The Task column shows the accuracy for the evaluation set (exact match).

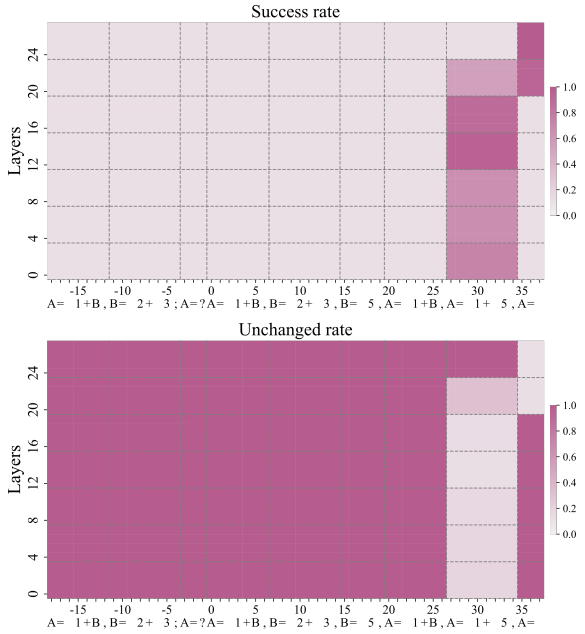

Figure 7: Success and Unchanged rates for each grid when the final answer y (A =**6** 5) is the target token. The Success rate heatmap at the top is the same as Figure 5.

B Supplemental results

B.1 Model performance on arithmetic tasks

Table 7 shows the accuracy of language models on arithmetic reasoning tasks for each experimental setting. We computed the accuracy based on exact matches between the output, including the chain ($\hat{z} \oplus \hat{y}$), and the gold labels ($z \oplus y$). The accuracy for all models is nearly 100%, indicating that they are capable of solving the arithmetic reasoning tasks used in this experiment.

[6]Please note that, unlike the other experimental configurations, under the implicit reasoning setting the model is unable to answer the problems with sufficiently high accuracy.

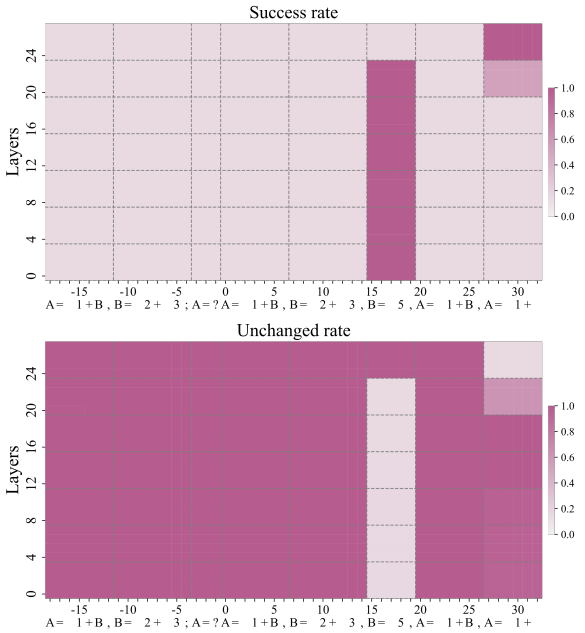

Figure 8: Success rate and Unchanged rate for each grid when intervention was performed with z_{17} ($\underline{\text{A = 1+5}}_{\mathbf{4}}$) as the target token.

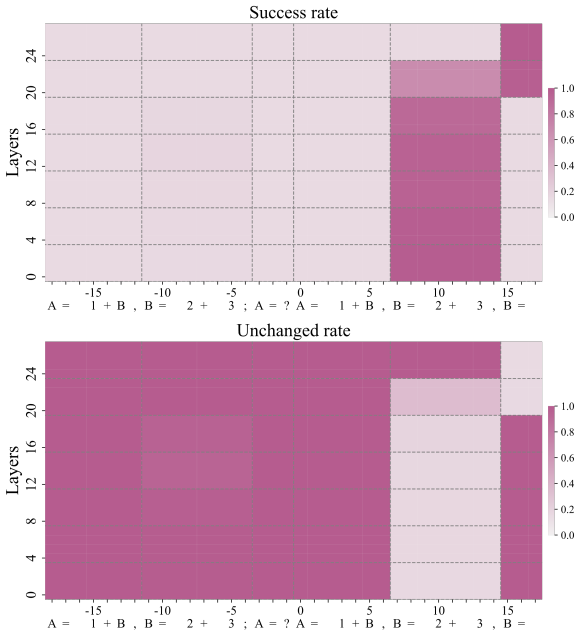

Figure 9: Success rate and Unchanged rate for each grid when intervention was performed with z_{32} ($\underline{\text{B =5}}_{\mathbf{2}}$) as the target token.

B.2 Additional probing predictions for correct/incorrect instances.

Figure 10 shows the top-1 predictions of the probe for B in instances where Llama3.2-3B produced *incorrect* answers for Task 3. In contrast, Figure 11 illustrates the probe predictions for instances where Llama3.2-3B generates correct responses. In Figure 3 the model should output (B=4) but instead produces 0. We observe the same pattern in other examples in Figure 10. One possible cause is that digits for computed results (e.g., 4) and digits from the input (e.g., 0) are represented in neighboring subspaces, which may cause confusion; we leave a detailed analysis for future work.

Train instances	10,000
Optimizer	SGD (Robbins, 1951)
Learning rate	1.0×10^{-3} (constant)
Batch size	10,000
Epochs	10,000

Table 8: Hyperparameters for training the probe

B.3 All probing results

Figures 12 through 56 present the probing results for all models and tasks discussed in this paper. For each figure, the input sequence below the graphs is one example (the results are averaged over the test set). The upper part indicates the maximum probing accuracy achieved at each token position t. The bottom part shows the probing accuracies in each token t, layer l, and variable v_i. Tables 9 through 23 summarize these results for thresholds (τ) ranging from 0.85 to 0.95. From these results, we observe trends similar to those described in § 3.3 across many settings. However, for the smaller model Llama3.2 (3B), increasing the threshold τ often leads to cases where the accuracy does not reach the threshold (N/A). Nonetheless, a consistent pattern remains: $\text{Acc}_{\prec \text{CoT}}(v_i)$ is low whereas $\text{Acc}_{\succ \text{CoT}}(v_i)$ is high.

B.4 Additional causal intervention results

Figures 59- 111 show the causal intervention results for the target tokens z_{17} and z_{32}, respectively.

Here, in addition to the Success rate, we also present the *Unchanged rate* as a metric. The Unchanged rate indicates how frequently (%) the intervened output $\hat{y}^{\text{patch}}$ remains the same as y. If this value is small, it indicates that the patched hidden states do not affect the output.

Additional discussion: Memory vs. recomputation. Focusing on the case where the target token is $\underline{\text{A = 1+}\mathbf{5}}_{\mathbf{4}}$ in Figure 6, if the model were recomputing when generating this **5**, it would be expected to show a causal relationship with the segment $\underline{\text{B = 2 + 3}}_{\mathbf{1}}$, where the pre-computation equation information is expected to be explicitly represented. However, in practice, a strong causal relationship

was observed only with the hidden state of the immediately preceding segment $\underline{\mathrm{B} = 5}_{\mathbf{2}}$. This pattern suggests that the model is not recomputing; instead, it likely relies on the immediately preceding intermediate result stored in the CoT text.

C Hyperparameters

Table 8 shows the hyperparameters used for training the probes.

D Computational resources

We used NVIDIA A100 GPUs (40GB and 80GB memory) and NVIDIA H100 GPUs to conduct this study.

E Usage of AI assistants

For writing this paper and the source code for the experiments, we use AI assistants (e.g., ChatGPT, GitHub Copilot). However, the use is limited to purposes such as code completion, translation, text editing, and table creation, and all content is solely based on the authors' ideas.

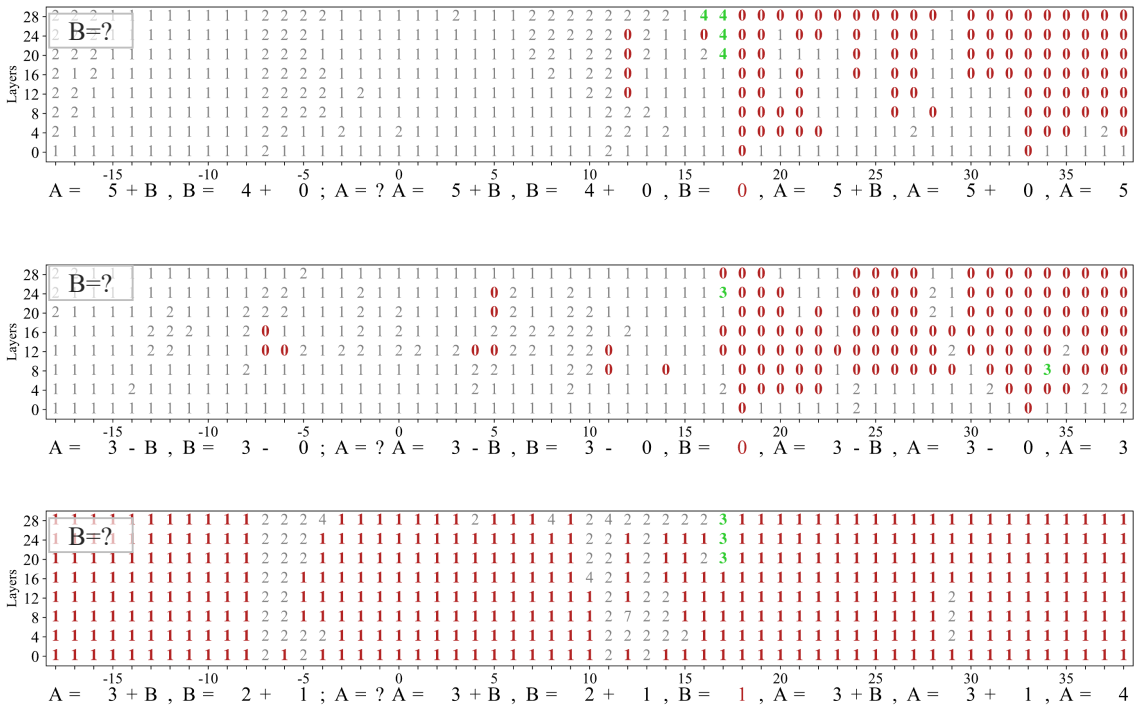

Figure 10: Error analysis of cases where Llama3.2-3B generated *incorrect* answers. The vertical axis represents the index of the transformer layer. The horizontal axis represents the tokens input to the model over time. The numbers in the figure indicate the labels predicted by each probe (top-1) at each layer and time step. The numbers highlighted in green represent the gold labels for the predictions, while those highlighted in red denote the values incorrectly generated by the model.

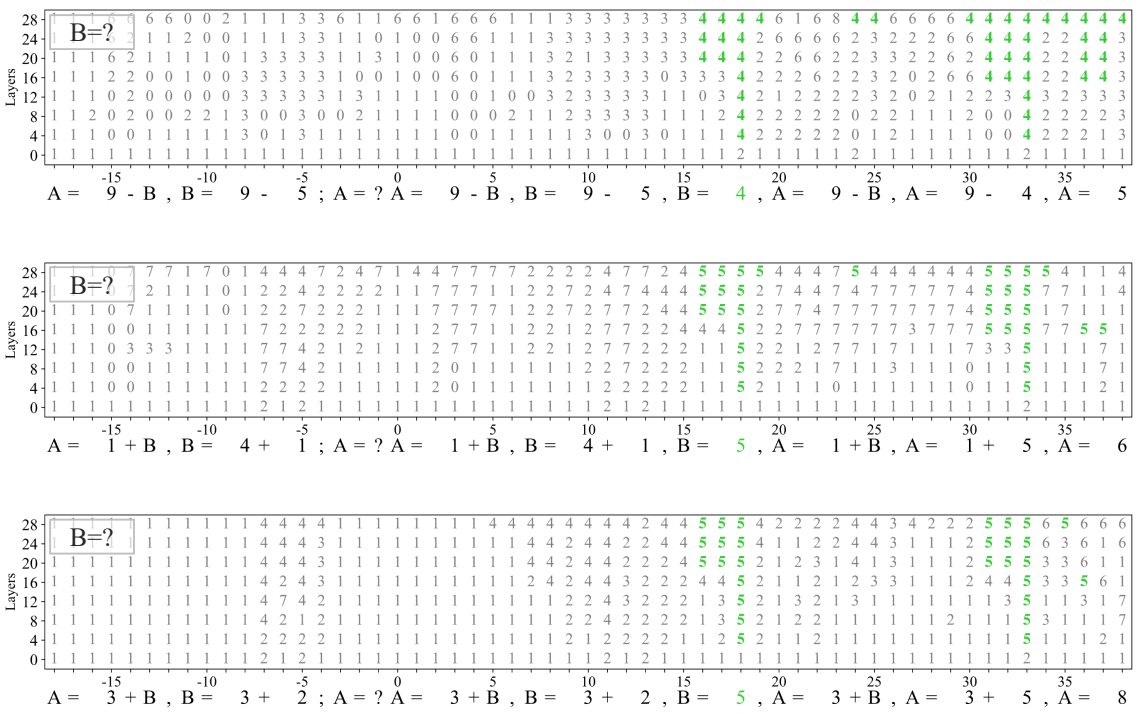

Figure 11: Top 1 predictions of probes where Llama3.2-3B generated *correct* answers. The vertical axis represents the index of the transformer layer. The horizontal axis represents the tokens input to the model over time. The numbers in the figure indicate the labels predicted by each probe (top-1) at each layer and time step. The numbers highlighted in green represent the gold labels for the predictions.

Model	Variable	When (↓) t^*_{eq}	When (↓) $t^* \prec$ CoT	Acc (↑) $\succ$ CoT	Acc (↑) CoT
Qwen2.5 (7B) (Qwen Team, 2024)	v_1 (A)	4	27	35.8	100
	v_2 (B)	**−2**	**−5**	100	100
Qwen2.5 (14B) (Qwen Team, 2024)	v_1 (A)	4	27	36.9	100
	v_2 (B)	**−2**	**−5**	100	100
Qwen2.5 (32B) (Qwen Team, 2024)	v_1 (A)	4	28	30.5	100
	v_2 (B)	**−2**	**−5**	100	100
Qwen2.5-Math (7B) (Yang et al., 2024a)	v_1 (A)	4	27	41.8	100
	v_2 (B)	**−2**	**−5**	100	100
Yi1.5 (9B) (Young et al., 2024)	v_1 (A)	4	32	28.1	100
	v_2 (B)	**−2**	**−5**	100	100
Yi1.5 (34B) (Young et al., 2024)	v_1 (A)	4	31	22.9	100
	v_2 (B)	**−2**	**−5**	100	100
Llama3.1 (8B) (Dubey et al., 2024)	v_1 (A)	4	27	20.6	100
	v_2 (B)	**−2**	**−5**	100	100
Llama3.2 (3B) (Dubey et al., 2024)	v_1 (A)	4	28	21.8	100.0
	v_2 (B)	**−2**	**−5**	100	100
Mistral-Nemo (12B) (Mistral AI Team, 2024)	v_1 (A)	4	27	18.0	100
	v_2 (B)	**−2**	**−5**	100	100

Table 9: Results for various models on the Level 1 task ($\tau = 0.85$).

Model	Variable	When (↓) t^*_{eq}	When (↓) $t^* \prec$ CoT	Acc (↑) $\succ$ CoT	Acc (↑) CoT
Qwen2.5 (7B) (Qwen Team, 2024)	v_1 (A)	2	16	49.2	100
	v_2 (B)	5	35	21.2	100
Qwen2.5 (14B) (Qwen Team, 2024)	v_1 (A)	2	15	48.8	100
	v_2 (B)	5	36	21.5	100
Qwen2.5 (32B) (Qwen Team, 2024)	v_1 (A)	2	15	66.4	100
	v_2 (B)	5	36	21.3	100
Qwen2.5-Math (7B) (Yang et al., 2024a)	v_1 (A)	2	15	53.7	100
	v_2 (B)	5	35	22.1	100
Yi1.5 (9B) (Young et al., 2024)	v_1 (A)	2	18	40.2	100
	v_2 (B)	5	41	17.8	100
Yi1.5 (34B) (Young et al., 2024)	v_1 (A)	2	18	35.6	100
	v_2 (B)	5	41	18.3	100
Llama3.1 (8B) (Dubey et al., 2024)	v_1 (A)	2	15	31.9	100
	v_2 (B)	5	35	17.8	100
Llama3.2 (3B) (Dubey et al., 2024)	v_1 (A)	2	16	36.2	99.9
	v_2 (B)	5	36	17.8	99.9
Mistral-Nemo (12B) (Mistral AI Team, 2024)	v_1 (A)	2	16	30.8	100
	v_2 (B)	5	36	17.8	100

Table 10: Results for various models on the Level 2 task ($\tau = 0.85$).

Model	Variable	When (↓) t^*_{eq}	When (↓) $t^* \prec$ CoT	Acc (↑) $\succ$ CoT	Acc (↑) CoT
Qwen2.5 (7B) (Qwen Team, 2024)	v_1 (A)	5	35	17.9	100
	v_2 (B)	2	16	50.5	100
Qwen2.5 (14B) (Qwen Team, 2024)	v_1 (A)	5	35	17.8	100
	v_2 (B)	2	15	50.5	100
Qwen2.5 (32B) (Qwen Team, 2024)	v_1 (A)	5	36	17.8	100
	v_2 (B)	2	15	67.4	100
Qwen2.5-Math (7B) (Yang et al., 2024a)	v_1 (A)	5	35	18.6	100
	v_2 (B)	2	15	56.1	100
Yi1.5 (9B) (Young et al., 2024)	v_1 (A)	5	41	17.8	100
	v_2 (B)	2	18	36.9	100
Yi1.5 (34B) (Young et al., 2024)	v_1 (A)	5	41	22.4	100
	v_2 (B)	2	18	37.4	100
Llama3.1 (8B) (Dubey et al., 2024)	v_1 (A)	5	35	26.0	100
	v_2 (B)	2	16	29.6	100
Llama3.2 (3B) (Dubey et al., 2024)	v_1 (A)	5	36	17.8	93.2
	v_2 (B)	2	16	33.2	95.4
Mistral-Nemo (12B) (Mistral AI Team, 2024)	v_1 (A)	5	36	17.8	100
	v_2 (B)	2	16	28.9	100

Table 11: Results for various models on the Level 3 task ($\tau = 0.85$).

Model	Variable	When (↓) t^*_{eq}	When (↓) $t^* \prec$ CoT	Acc (↑) $\succ$ CoT	Acc (↑) CoT
Qwen2.5 (7B) (Qwen Team, 2024)	v_1 (A)	4	29	30.4	100
	v_2 (B)	2	15	27.2	100
	v_3 (C)	N/A	N/A	18.5	17.6
Qwen2.5 (14B) (Qwen Team, 2024)	v_1 (A)	5	35	18.9	100
	v_2 (B)	2	15	44.3	100
	v_3 (C)	N/A	N/A	40.4	26.8
Qwen2.5 (32B) (Qwen Team, 2024)	v_1 (A)	5	36	17.4	100
	v_2 (B)	2	15	62.8	100
	v_3 (C)	N/A	N/A	64.4	32.6
Qwen2.5-Math (7B) (Yang et al., 2024a)	v_1 (A)	5	35	17.2	100
	v_2 (B)	2	15	55.6	100
	v_3 (C)	N/A	N/A	47.8	29.4
Yi1.5 (9B) (Young et al., 2024)	v_1 (A)	5	41	17.8	100
	v_2 (B)	2	18	43.5	100
	v_3 (C)	N/A	N/A	36.7	21.2
Yi1.5 (34B) (Young et al., 2024)	v_1 (A)	5	40	19.3	100
	v_2 (B)	2	18	40.8	100
	v_3 (C)	N/A	N/A	27.9	26.2
Llama3.1 (8B) (Dubey et al., 2024)	v_1 (A)	4	29	30.4	100
	v_2 (B)	2	15	27.2	100
	v_3 (C)	N/A	N/A	18.5	17.6
Llama3.2 (3B) (Dubey et al., 2024)	v_1 (A)	5	36	26.2	91.7
	v_2 (B)	2	16	29.1	98.7
	v_3 (C)	N/A	N/A	18.3	17.3
Mistral-Nemo (12B) (Mistral AI Team, 2024)	v_1 (A)	5	36	17.2	100
	v_2 (B)	2	16	29.9	100
	v_3 (C)	N/A	N/A	22.0	19.8

Table 12: Results for various models on the Level 4 task ($\tau = 0.85$).

Model	Variable	When (↓) t^*_{eq}	When (↓) t^*	Acc (↑) $\prec$ CoT	Acc (↑) $\succ$ CoT
Qwen2.5 (7B) (Qwen Team, 2024)	v_1 (A)	9	62	18.1	100
	v_2 (B)	6	42	22.6	100
	v_3 (C)	3	23	50.6	100
Qwen2.5 (14B) (Qwen Team, 2024)	v_1 (A)	9	63	18.1	98.8
	v_2 (B)	6	42	18.7	98.9
	v_3 (C)	3	23	42.2	100
Qwen2.5 (32B) (Qwen Team, 2024)	v_1 (A)	9	63	18.7	100
	v_2 (B)	6	43	22.6	100
	v_3 (C)	3	23	62.4	100
Qwen2.5-Math (7B) (Yang et al., 2024a)	v_1 (A)	9	62	18.1	100
	v_2 (B)	6	42	22.6	100
	v_3 (C)	3	22	54.5	100
Yi1.5 (34B) (Young et al., 2024)	v_1 (A)	9	71	18.1	100
	v_2 (B)	6	49	22.6	100
	v_3 (C)	3	26	41.2	100
Llama3.1 (8B) (Dubey et al., 2024)	v_1 (A)	9	62	16.0	99.5
	v_2 (B)	6	43	20.0	99.5
	v_3 (C)	3	23	30.6	99.8
Llama3.2 (3B) (Dubey et al., 2024)	v_1 (A)	N/A	N/A	14.2	43.7
	v_2 (B)	N/A	N/A	26.3	47.4
	v_3 (C)	N/A	N/A	37.7	71.7
Mistral-Nemo (12B) (Mistral AI Team, 2024)	v_1 (A)	9	63	18.1	99.9
	v_2 (B)	6	43	16.3	99.9
	v_3 (C)	3	23	32.0	99.9

Table 13: Results for various models on the Level 5 task ($\tau = 0.85$).

Model	Variable	When (↓) t^*_{eq}	When (↓) t^*	Acc (↑) $\prec$ CoT	Acc (↑) $\succ$ CoT
Qwen2.5 (7B) (Qwen Team, 2024)	v_1 (A)	4	27	35.8	100
	v_2 (B)	**−2**	**−5**	100	100
Qwen2.5 (14B) (Qwen Team, 2024)	v_1 (A)	4	27	36.9	100
	v_2 (B)	**−2**	**−5**	100	100
Qwen2.5 (32B) (Qwen Team, 2024)	v_1 (A)	4	28	30.5	100
	v_2 (B)	**−2**	**−5**	100	100
Qwen2.5-Math (7B) (Yang et al., 2024a)	v_1 (A)	4	27	41.8	100
	v_2 (B)	**−2**	**−5**	100	100
Yi1.5 (9B) (Young et al., 2024)	v_1 (A)	4	32	28.1	100
	v_2 (B)	**−2**	**−5**	100	100
Yi1.5 (34B) (Young et al., 2024)	v_1 (A)	4	32	22.9	100
	v_2 (B)	**−2**	**−5**	100	100
Llama3.1 (8B) (Dubey et al., 2024)	v_1 (A)	4	27	20.6	100
	v_2 (B)	**−2**	**−5**	100	100
Llama3.2 (3B) (Dubey et al., 2024)	v_1 (A)	4	28	21.8	100.0
	v_2 (B)	**−2**	**−5**	100	100
Mistral-Nemo (12B) (Mistral AI Team, 2024)	v_1 (A)	4	28	18.0	100
	v_2 (B)	**−2**	**−5**	100	100

Table 14: Results for various models on the Level 1 task ($\tau = 0.90$).

Model	Variable	When (↓) t^*_{eq}	When (↓) t^*	Acc (↑) $\prec$ CoT	Acc (↑) $\succ$ CoT
Qwen2.5 (7B) (Qwen Team, 2024)	v_1 (A)	2	16	49.2	100
	v_2 (B)	5	35	21.2	100
Qwen2.5 (14B) (Qwen Team, 2024)	v_1 (A)	2	16	48.8	100
	v_2 (B)	5	36	21.5	100
Qwen2.5 (32B) (Qwen Team, 2024)	v_1 (A)	2	16	66.4	100
	v_2 (B)	5	36	21.3	100
Qwen2.5-Math (7B) (Yang et al., 2024a)	v_1 (A)	2	15	53.7	100
	v_2 (B)	5	35	22.1	100
Yi1.5 (9B) (Young et al., 2024)	v_1 (A)	2	18	40.2	100
	v_2 (B)	5	41	17.8	100
Yi1.5 (34B) (Young et al., 2024)	v_1 (A)	2	18	35.6	100
	v_2 (B)	5	41	18.3	100
Llama3.1 (8B) (Dubey et al., 2024)	v_1 (A)	2	15	31.9	100
	v_2 (B)	5	35	17.8	100
Llama3.2 (3B) (Dubey et al., 2024)	v_1 (A)	2	16	36.2	99.9
	v_2 (B)	5	36	17.8	99.9
Mistral-Nemo (12B) (Mistral AI Team, 2024)	v_1 (A)	2	16	30.8	100
	v_2 (B)	5	36	17.8	100

Table 15: Results for various models on the Level 2 task ($\tau = 0.90$).

Model	Variable	When (↓) t^*_{eq}	When (↓) t^*	Acc (↑) $\prec$ CoT	Acc (↑) $\succ$ CoT
Qwen2.5 (7B) (Qwen Team, 2024)	v_1 (A)	5	36	17.9	100
	v_2 (B)	2	16	50.5	100
Qwen2.5 (14B) (Qwen Team, 2024)	v_1 (A)	5	35	17.8	100
	v_2 (B)	2	16	50.5	100
Qwen2.5 (32B) (Qwen Team, 2024)	v_1 (A)	5	36	17.8	100
	v_2 (B)	2	15	67.4	100
Qwen2.5-Math (7B) (Yang et al., 2024a)	v_1 (A)	5	35	18.6	100
	v_2 (B)	2	15	56.1	100
Yi1.5 (9B) (Young et al., 2024)	v_1 (A)	5	41	17.8	100
	v_2 (B)	2	18	36.9	100
Yi1.5 (34B) (Young et al., 2024)	v_1 (A)	5	41	22.4	100
	v_2 (B)	2	18	37.4	100
Llama3.1 (8B) (Dubey et al., 2024)	v_1 (A)	5	35	26.0	100
	v_2 (B)	2	16	29.6	100
Llama3.2 (3B) (Dubey et al., 2024)	v_1 (A)	5	36	17.8	93.2
	v_2 (B)	2	17	33.2	95.4
Mistral-Nemo (12B) (Mistral AI Team, 2024)	v_1 (A)	5	36	17.8	100
	v_2 (B)	2	16	28.9	100

Table 16: Results for various models on the Level 3 task ($\tau = 0.90$).

Model	Variable	When (↓) t^*_{eq}	When (↓) t^*	Acc (↑) $\prec$ CoT	Acc (↑) $\succ$ CoT
Qwen2.5 (7B) (Qwen Team, 2024)	v_1 (A)	5	35	17.2	100
	v_2 (B)	2	16	47.7	100
	v_3 (C)	N/A	N/A	43.7	23.7
Qwen2.5 (14B) (Qwen Team, 2024)	v_1 (A)	5	36	18.9	100
	v_2 (B)	2	15	44.3	100
	v_3 (C)	N/A	N/A	40.4	26.8
Qwen2.5 (32B) (Qwen Team, 2024)	v_1 (A)	5	36	17.4	100
	v_2 (B)	2	15	62.8	100
	v_3 (C)	N/A	N/A	64.4	32.6
Qwen2.5-Math (7B) (Yang et al., 2024a)	v_1 (A)	5	35	17.2	100
	v_2 (B)	2	15	55.6	100
	v_3 (C)	N/A	N/A	47.8	29.4
Yi1.5 (9B) (Young et al., 2024)	v_1 (A)	5	41	17.8	100
	v_2 (B)	2	18	43.5	100
	v_3 (C)	N/A	N/A	36.7	21.2
Yi1.5 (34B) (Young et al., 2024)	v_1 (A)	5	40	19.3	100
	v_2 (B)	2	18	40.8	100
	v_3 (C)	N/A	N/A	27.9	26.2
Llama3.1 (8B) (Dubey et al., 2024)	v_1 (A)	5	35	30.4	100
	v_2 (B)	2	16	27.2	100
	v_3 (C)	N/A	N/A	18.5	17.6
Llama3.2 (3B) (Dubey et al., 2024)	v_1 (A)	5	37	26.2	91.7
	v_2 (B)	2	17	29.1	98.7
	v_3 (C)	N/A	N/A	18.3	17.3
Mistral-Nemo (12B) (Mistral AI Team, 2024)	v_1 (A)	5	36	17.2	100
	v_2 (B)	2	16	29.9	100
	v_3 (C)	N/A	N/A	22.0	19.8

Table 17: Results for various models on the Level 4 task ($\tau = 0.90$).

Model	Variable	When (↓) t^*_{eq}	When (↓) t^*	Acc (↑) $\prec$ CoT	Acc (↑) $\succ$ CoT
Qwen2.5 (7B) (Qwen Team, 2024)	v_1 (A)	9	63	18.1	100
	v_2 (B)	6	42	22.6	100
	v_3 (C)	3	23	50.6	100
Qwen2.5 (14B) (Qwen Team, 2024)	v_1 (A)	9	63	18.1	98.8
	v_2 (B)	6	42	18.7	98.9
	v_3 (C)	3	23	42.2	100
Qwen2.5 (32B) (Qwen Team, 2024)	v_1 (A)	9	63	18.7	100
	v_2 (B)	6	43	22.6	100
	v_3 (C)	3	23	62.4	100
Qwen2.5-Math (7B) (Yang et al., 2024a)	v_1 (A)	9	62	18.1	100
	v_2 (B)	6	42	22.6	100
	v_3 (C)	3	22	54.5	100
Yi1.5 (34B) (Young et al., 2024)	v_1 (A)	9	72	18.1	100
	v_2 (B)	6	49	22.6	100
	v_3 (C)	3	26	41.2	100
Llama3.1 (8B) (Dubey et al., 2024)	v_1 (A)	9	62	16.0	99.5
	v_2 (B)	6	43	20.0	99.5
	v_3 (C)	3	23	30.6	99.8
Llama3.2 (3B) (Dubey et al., 2024)	v_1 (A)	N/A	N/A	14.2	43.7
	v_2 (B)	N/A	N/A	26.3	47.4
	v_3 (C)	N/A	N/A	37.7	71.7
Mistral-Nemo (12B) (Mistral AI Team, 2024)	v_1 (A)	9	63	18.1	99.9
	v_2 (B)	6	43	16.3	99.9
	v_3 (C)	3	23	32.0	99.9

Table 18: Results for various models on the Level 5 task ($\tau = 0.90$).

Model	Variable	When (↓) t^*_{eq}	When (↓) t^*	Acc (↑) $\prec$ CoT	Acc (↑) $\succ$ CoT
Qwen2.5 (7B) (Qwen Team, 2024)	v_1 (A)	4	27	35.8	100
	v_2 (B)	**−2**	**−5**	100	100
Qwen2.5 (14B) (Qwen Team, 2024)	v_1 (A)	4	28	36.9	100
	v_2 (B)	**−2**	**−5**	100	100
Qwen2.5 (32B) (Qwen Team, 2024)	v_1 (A)	4	28	30.5	100
	v_2 (B)	**−2**	**−5**	100	100
Qwen2.5-Math (7B) (Yang et al., 2024a)	v_1 (A)	4	27	41.8	100
	v_2 (B)	**−2**	**−5**	100	100
Yi1.5 (9B) (Young et al., 2024)	v_1 (A)	4	32	28.1	100
	v_2 (B)	**−2**	**−5**	100	100
Yi1.5 (34B) (Young et al., 2024)	v_1 (A)	4	32	22.9	100
	v_2 (B)	**−2**	**−5**	100	100
Llama3.1 (8B) (Dubey et al., 2024)	v_1 (A)	4	27	20.6	100
	v_2 (B)	**−2**	**−5**	100	100
Llama3.2 (3B) (Dubey et al., 2024)	v_1 (A)	4	28	21.8	100.0
	v_2 (B)	**−2**	**−5**	100	100
Mistral-Nemo (12B) (Mistral AI Team, 2024)	v_1 (A)	4	28	17.9	100
	v_2 (B)	**−2**	**−5**	100	100

Table 19: Results for various models on the Level 1 task ($\tau = 0.95$).

Model	Variable	When (↓) t^*_{eq}	When (↓) t^*	Acc (↑) $\prec$ CoT	Acc (↑) $\succ$ CoT
Qwen2.5 (7B) (Qwen Team, 2024)	v_1 (A)	2	16	49.2	100
	v_2 (B)	5	36	21.2	100
Qwen2.5 (14B) (Qwen Team, 2024)	v_1 (A)	2	16	48.8	100
	v_2 (B)	5	36	21.5	100
Qwen2.5 (32B) (Qwen Team, 2024)	v_1 (A)	2	16	66.4	100
	v_2 (B)	5	36	21.3	100
Qwen2.5-Math (7B) (Yang et al., 2024a)	v_1 (A)	2	15	53.7	100
	v_2 (B)	5	36	22.1	100
Yi1.5 (9B) (Young et al., 2024)	v_1 (A)	2	18	40.2	100
	v_2 (B)	5	41	17.8	100
Yi1.5 (34B) (Young et al., 2024)	v_1 (A)	2	18	35.6	100
	v_2 (B)	5	41	18.3	100
Llama3.1 (8B) (Dubey et al., 2024)	v_1 (A)	2	15	31.9	100
	v_2 (B)	5	35	17.8	100
Llama3.2 (3B) (Dubey et al., 2024)	v_1 (A)	2	16	36.2	99.9
	v_2 (B)	5	36	17.8	99.9
Mistral-Nemo (12B) (Mistral AI Team, 2024)	v_1 (A)	2	16	30.8	100
	v_2 (B)	5	36	17.8	100

Table 20: Results for various models on the Level 2 task ($\tau = 0.95$).

Model	Variable	When (↓) t^*_{eq}	When (↓) t^*	Acc (↑) $\prec$ CoT	Acc (↑) $\succ$ CoT
Qwen2.5 (7B) (Qwen Team, 2024)	v_1 (A)	5	36	17.9	100
	v_2 (B)	2	16	50.5	100
Qwen2.5 (14B) (Qwen Team, 2024)	v_1 (A)	5	36	17.8	100
	v_2 (B)	2	16	50.5	100
Qwen2.5 (32B) (Qwen Team, 2024)	v_1 (A)	5	36	17.8	100
	v_2 (B)	2	15	67.4	100
Qwen2.5-Math (7B) (Yang et al., 2024a)	v_1 (A)	5	35	18.6	100
	v_2 (B)	2	15	56.1	100
Yi1.5 (9B) (Young et al., 2024)	v_1 (A)	5	41	17.8	100
	v_2 (B)	2	18	36.9	100
Yi1.5 (34B) (Young et al., 2024)	v_1 (A)	5	41	22.4	100
	v_2 (B)	2	18	37.4	100
Llama3.1 (8B) (Dubey et al., 2024)	v_1 (A)	5	35	26.0	100
	v_2 (B)	2	16	29.6	100
Llama3.2 (3B) (Dubey et al., 2024)	v_1 (A)	N/A	N/A	17.8	93.2
	v_2 (B)	2	17	33.2	95.4
Mistral-Nemo (12B) (Mistral AI Team, 2024)	v_1 (A)	5	36	17.8	100
	v_2 (B)	2	16	28.9	100

Table 21: Results for various models on the Level 3 task ($\tau = 0.95$).

Model	Variable	When (↓) t^*_{eq}	When (↓) t^*	Acc (↑) $\prec$ CoT	Acc (↑) $\succ$ CoT
Qwen2.5 (7B) (Qwen Team, 2024)	v_1 (A)	5	36	17.2	100
	v_2 (B)	2	16	47.7	100
	v_3 (C)	N/A	N/A	43.7	23.7
Qwen2.5 (14B) (Qwen Team, 2024)	v_1 (A)	5	36	18.9	100
	v_2 (B)	2	15	44.3	100
	v_3 (C)	N/A	N/A	40.4	26.8
Qwen2.5 (32B) (Qwen Team, 2024)	v_1 (A)	5	36	17.4	100
	v_2 (B)	2	16	62.8	100
	v_3 (C)	N/A	N/A	64.4	32.6
Qwen2.5-Math (7B) (Yang et al., 2024a)	v_1 (A)	5	35	17.2	100
	v_2 (B)	2	15	55.6	100
	v_3 (C)	N/A	N/A	47.8	29.4
Yi1.5 (9B) (Young et al., 2024)	v_1 (A)	5	41	17.8	100
	v_2 (B)	2	18	43.5	100
	v_3 (C)	N/A	N/A	36.7	21.2
Yi1.5 (34B) (Young et al., 2024)	v_1 (A)	5	41	19.3	100
	v_2 (B)	2	18	40.8	100
	v_3 (C)	N/A	N/A	27.9	26.2
Llama3.1 (8B) (Dubey et al., 2024)	v_1 (A)	5	35	30.4	100
	v_2 (B)	2	16	27.2	100
	v_3 (C)	N/A	N/A	18.5	17.6
Llama3.2 (3B) (Dubey et al., 2024)	v_1 (A)	N/A	N/A	26.2	91.7
	v_2 (B)	2	17	29.1	98.7
	v_3 (C)	N/A	N/A	18.3	17.3
Mistral-Nemo (12B) (Mistral AI Team, 2024)	v_1 (A)	5	36	17.2	100
	v_2 (B)	2	16	29.9	100
	v_3 (C)	N/A	N/A	22.0	19.8

Table 22: Results for various models on the Level 4 task ($\tau = 0.95$).

Model	Variable	When (↓) t^*_{eq}	When (↓) t^*	Acc (↑) $\prec$ CoT	Acc (↑) $\succ$ CoT
Qwen2.5 (7B)	v_1 (A)	9	63	18.1	100
(Qwen Team, 2024)	v_2 (B)	6	43	22.6	100
	v_3 (C)	3	23	50.6	100
Qwen2.5 (14B)	v_1 (A)	9	63	18.1	98.8
(Qwen Team, 2024)	v_2 (B)	6	43	18.7	98.9
	v_3 (C)	3	23	42.2	100
Qwen2.5 (32B)	v_1 (A)	9	63	18.7	100
(Qwen Team, 2024)	v_2 (B)	6	43	22.6	100
	v_3 (C)	3	23	62.4	100
Qwen2.5-Math (7B)	v_1 (A)	9	63	18.1	100
(Yang et al., 2024a)	v_2 (B)	6	42	22.6	100
	v_3 (C)	3	22	54.5	100
Yi1.5 (34B)	v_1 (A)	9	72	18.1	100
(Young et al., 2024)	v_2 (B)	6	49	22.6	100
	v_3 (C)	3	26	41.2	100
Llama3.1 (8B)	v_1 (A)	9	62	16.0	99.5
(Dubey et al., 2024)	v_2 (B)	6	43	20.0	99.5
	v_3 (C)	3	23	30.6	99.8
Llama3.2 (3B)	v_1 (A)	N/A	N/A	14.1	43.7
(Dubey et al., 2024)	v_2 (B)	N/A	N/A	26.3	47.4
	v_3 (C)	N/A	N/A	37.7	71.7
Mistral-Nemo (12B)	v_1 (A)	9	63	18.1	99.9
(Mistral AI Team, 2024)	v_2 (B)	6	43	16.3	99.9
	v_3 (C)	3	23	32.0	99.9

Table 23: Results for various models on the Level 5 task ($\tau = 0.95$).

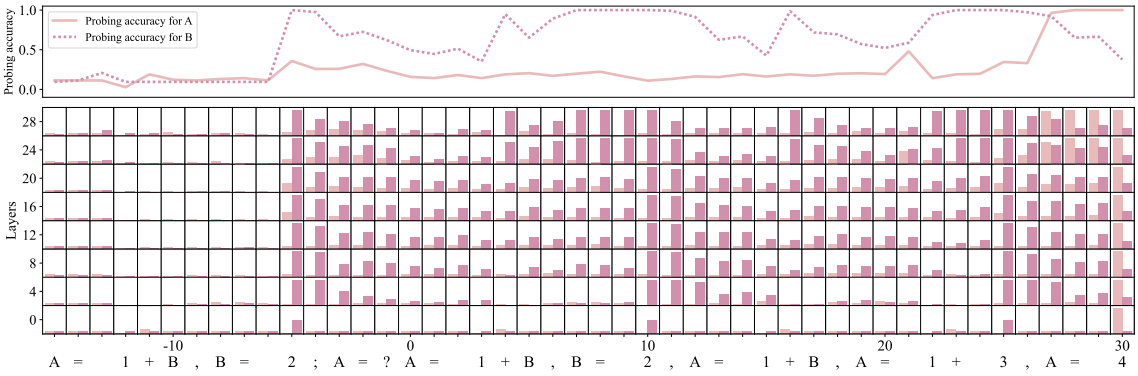

Figure 12: Probing results when Qwen2.5-7B solves Level 1.

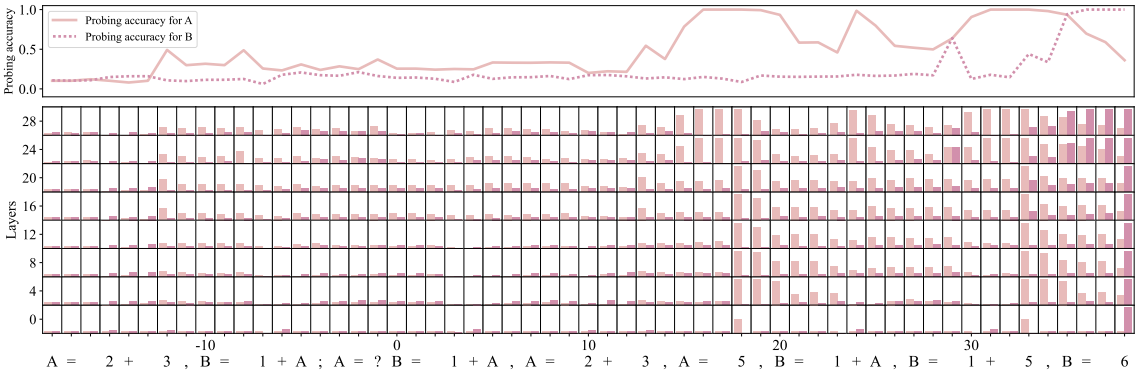

Figure 13: Probing results when Qwen2.5-7B solves Level 2.

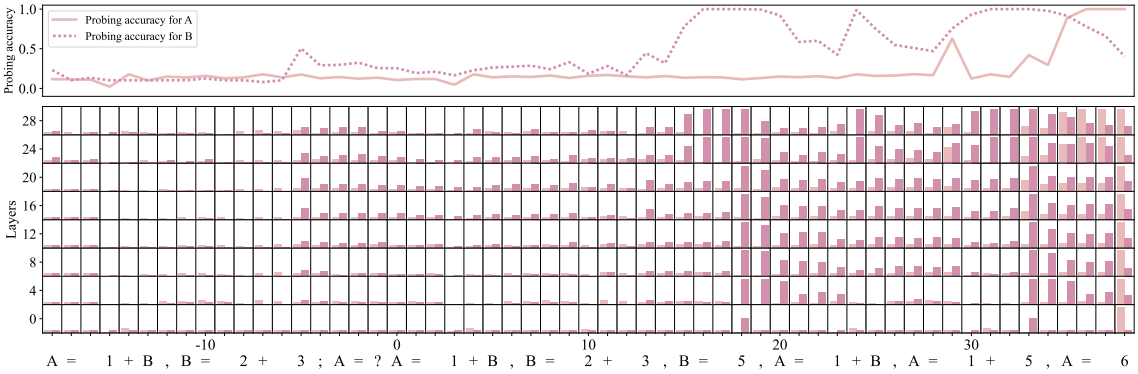

Figure 14: Probing results when Qwen2.5-7B solves Level 3.

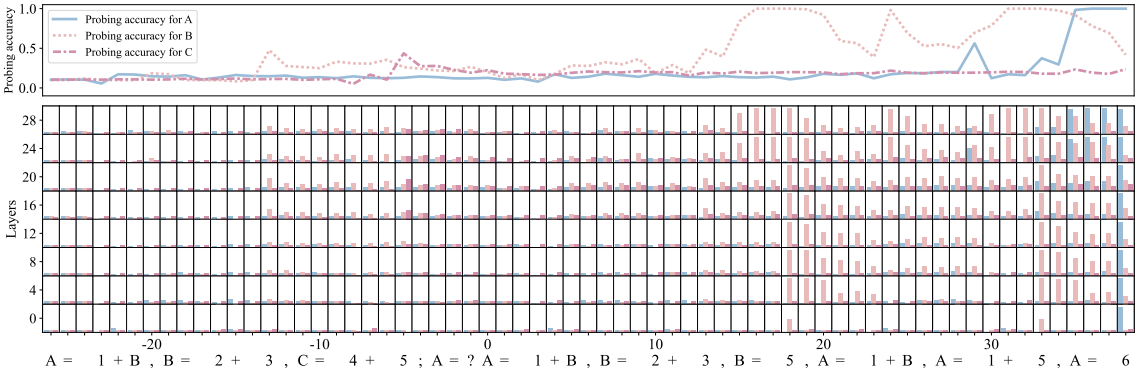

Figure 15: Probing results when Qwen2.5-7B solves Level 4.

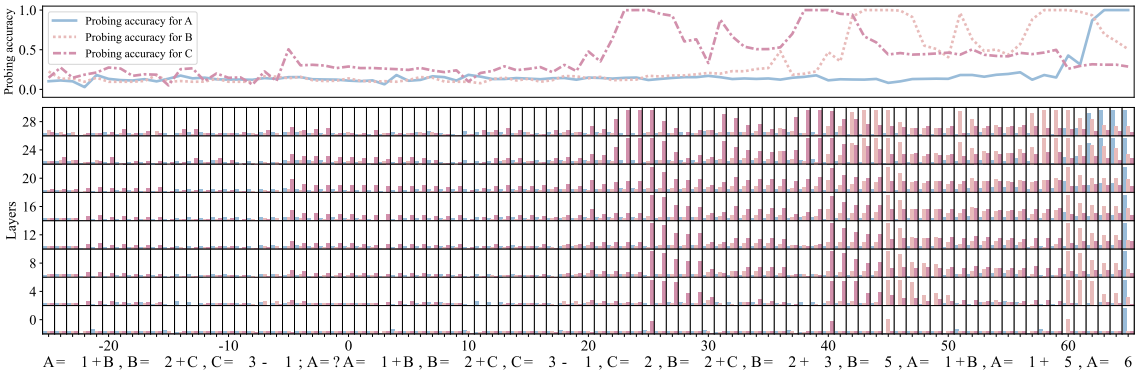

Figure 16: Probing results when Qwen2.5-7B solves Level 5.

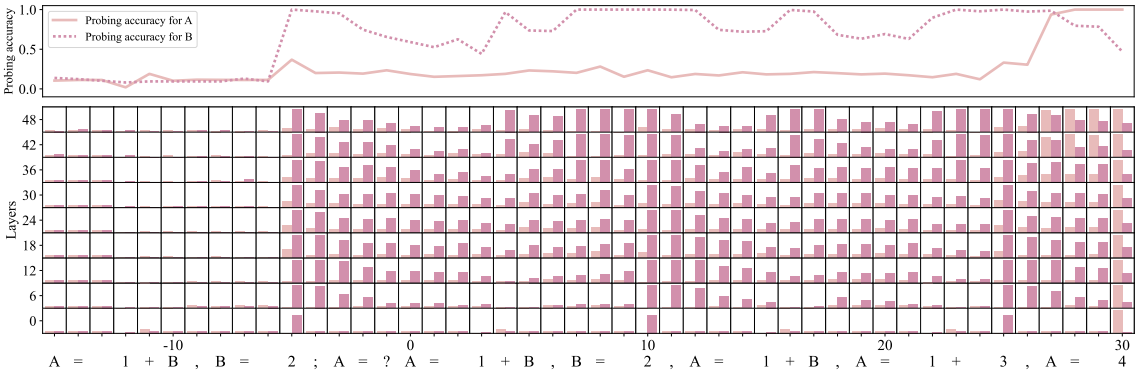

Figure 17: Probing results when Qwen2.5-14B solves Level 1.

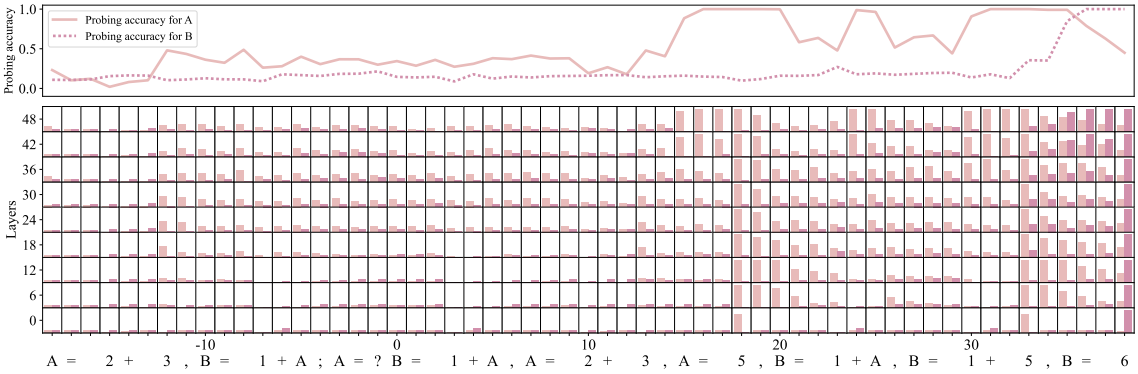

Figure 18: Probing results when Qwen2.5-14B solves Level 2.

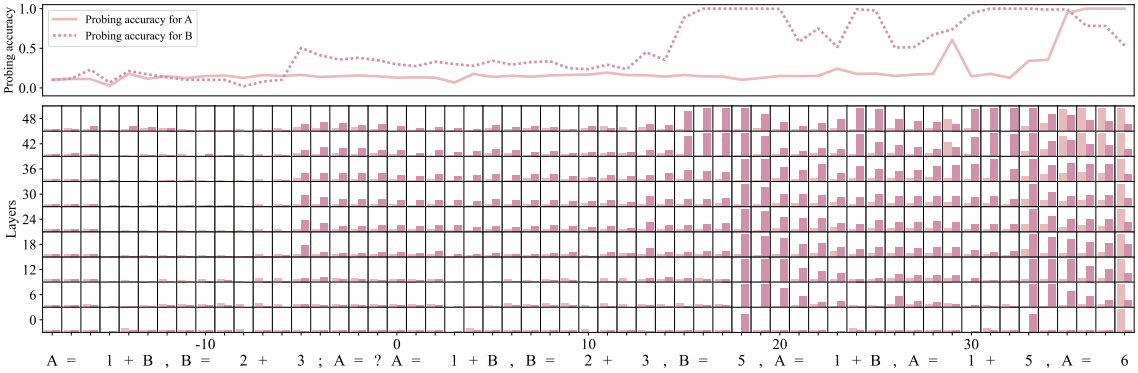

Figure 19: Probing results when Qwen2.5-14B solves Level 3.

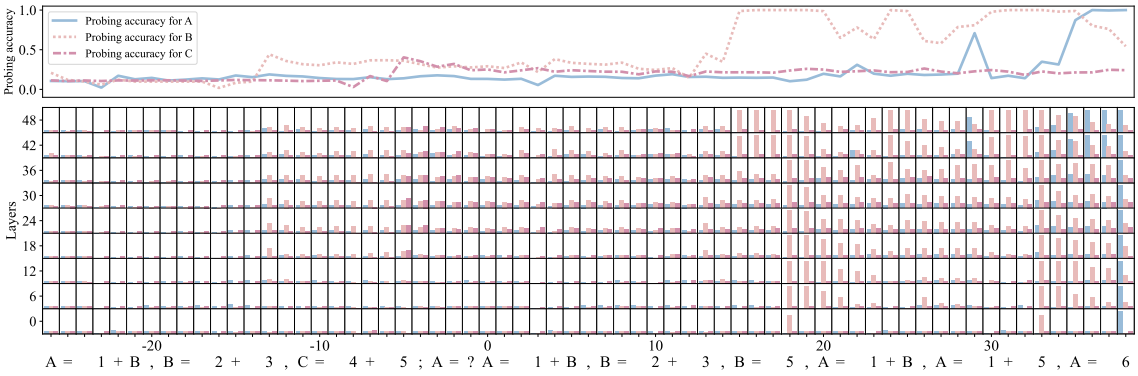

Figure 20: Probing results when Qwen2.5-14B solves Level 4.

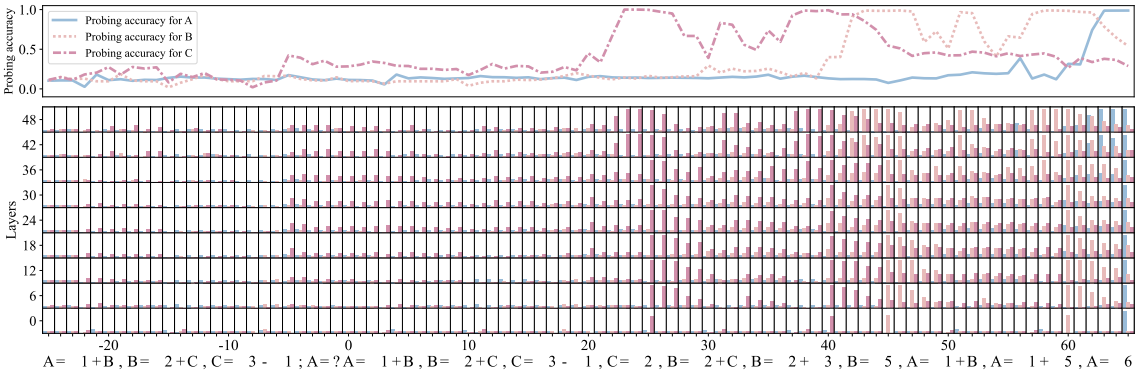

Figure 21: Probing results when Qwen2.5-14B solves Level 5.

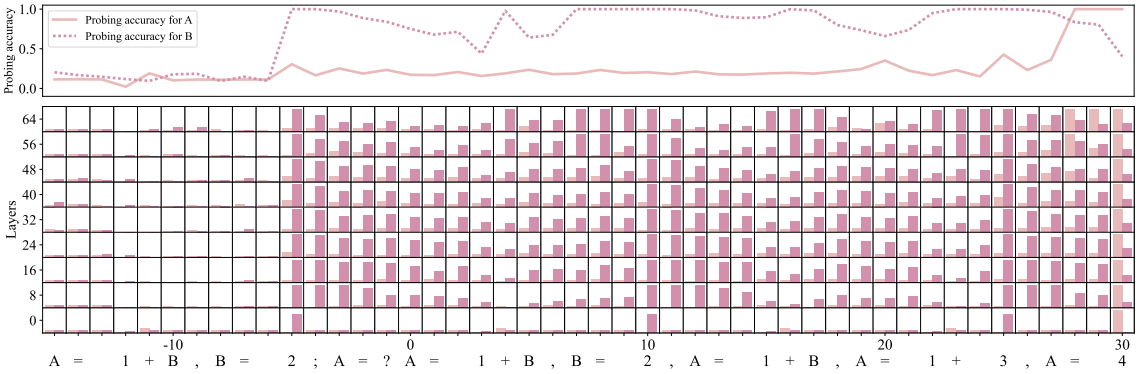

Figure 22: Probing results when Qwen2.5-32B solves Level 1.

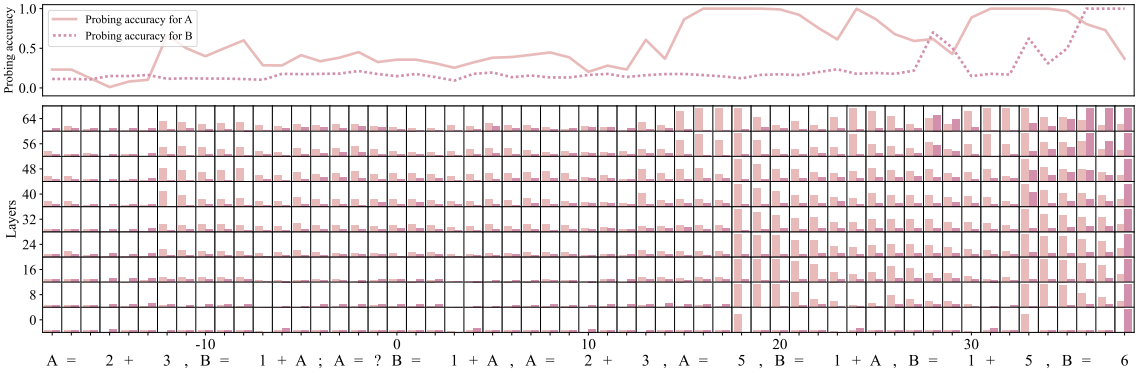

Figure 23: Probing results when Qwen2.5-32B solves Level 2.

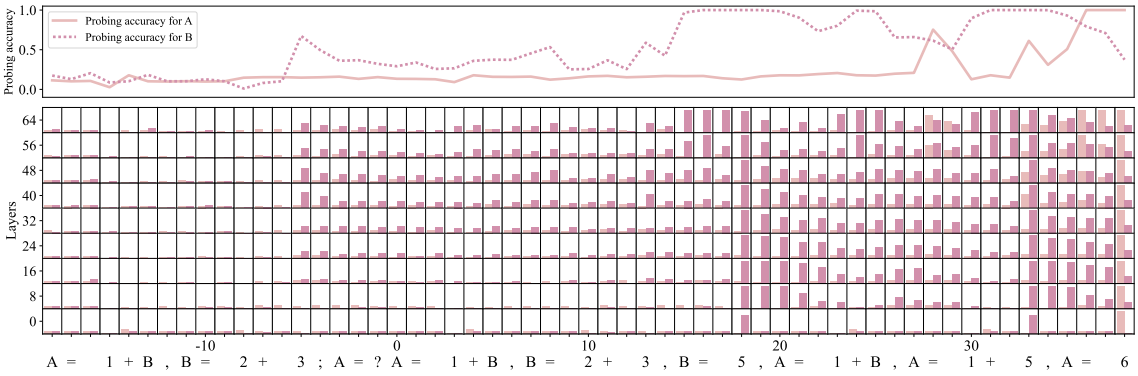

Figure 24: Probing results when Qwen2.5-32B solves Level 3.

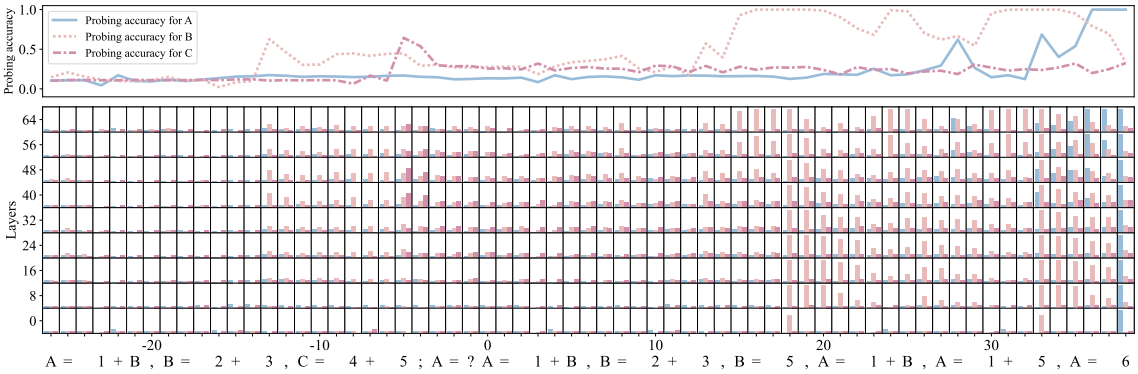

Figure 25: Probing results when Qwen2.5-32B solves Level 4.

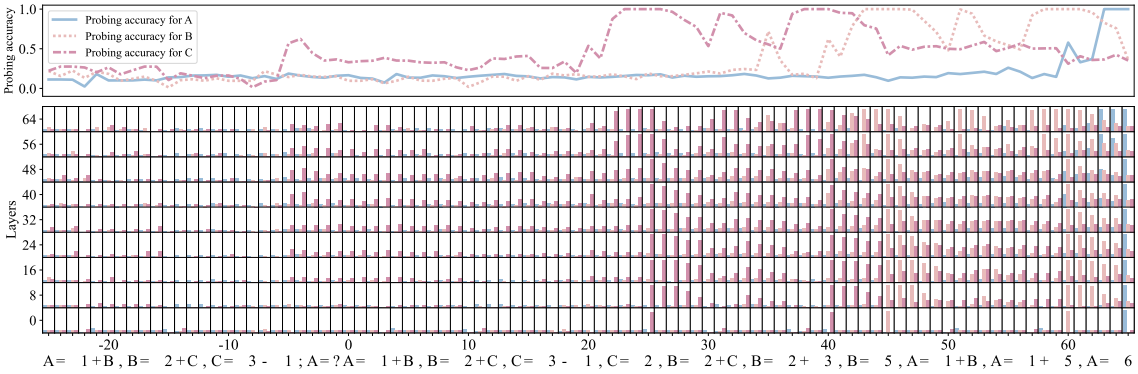

Figure 26: Probing results when Qwen2.5-32B solves Level 5.

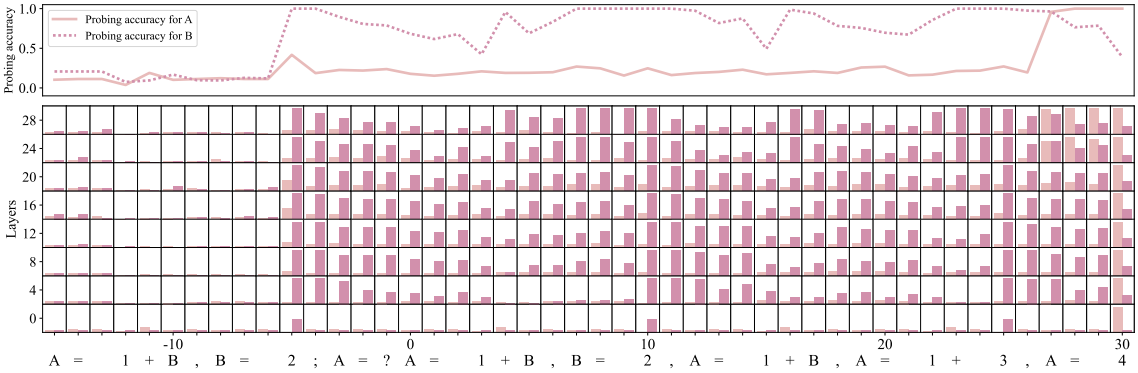

Figure 27: Probing results when Qwen2.5-Math-7B solves Level 1.

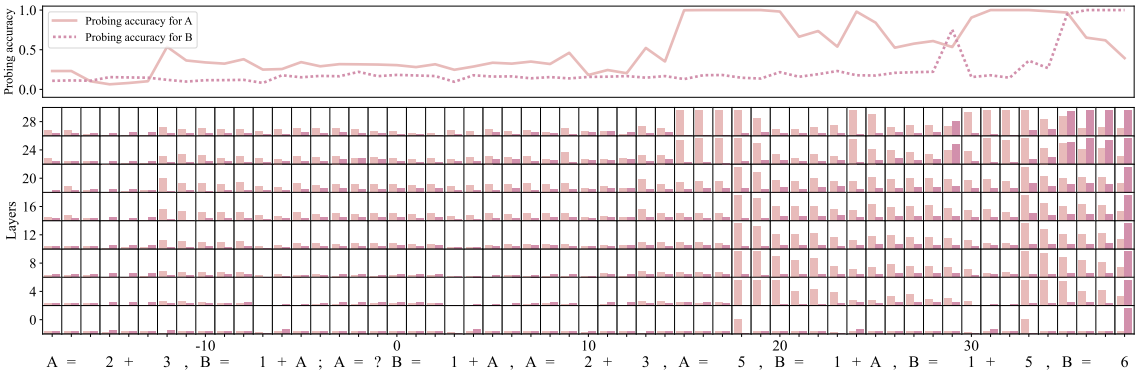

Figure 28: Probing results when Qwen2.5-Math-7B solves Level 2.

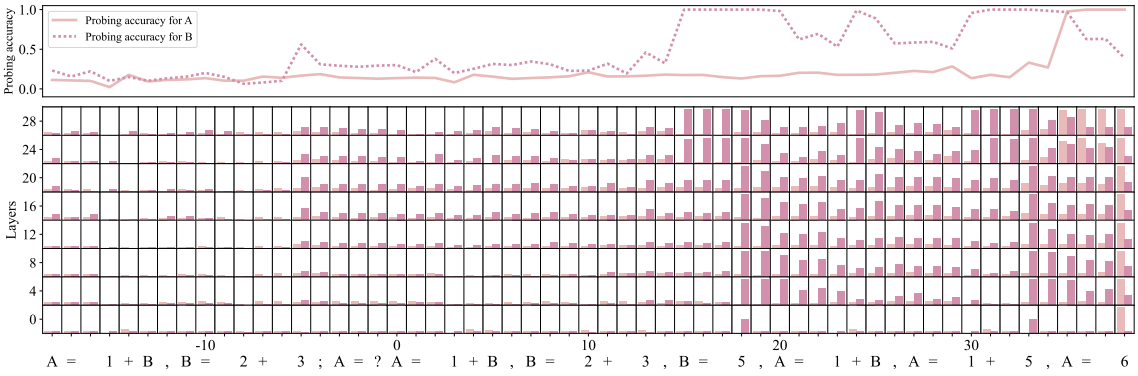

Figure 29: Probing results when Qwen2.5-Math-7B solves Level 3.

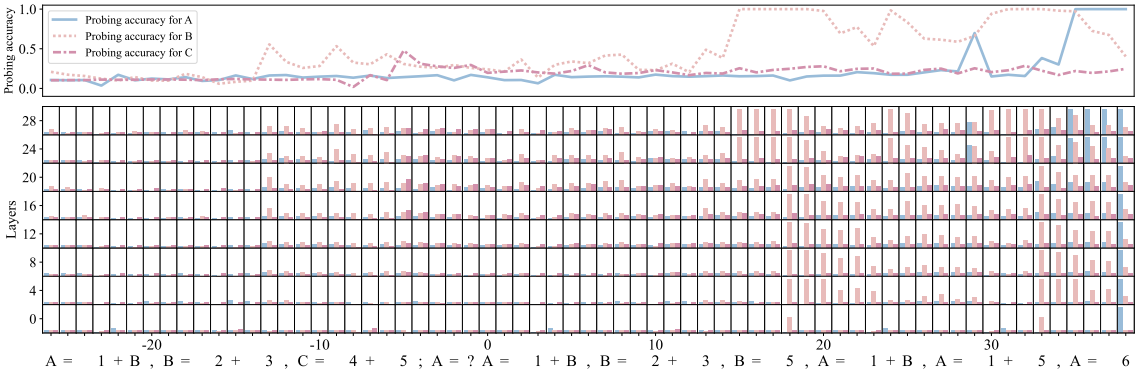

Figure 30: Probing results when Qwen2.5-Math-7B solves Level 4.

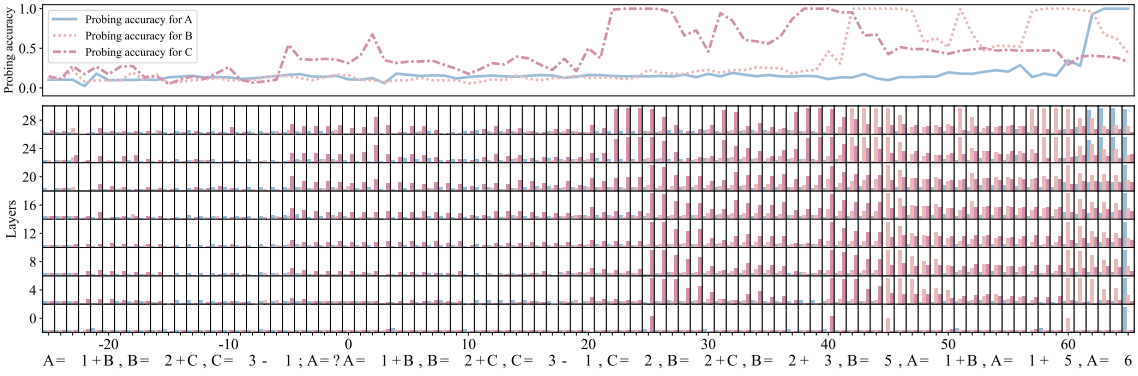

Figure 31: Probing results when Qwen2.5-Math-7B solves Level 5.

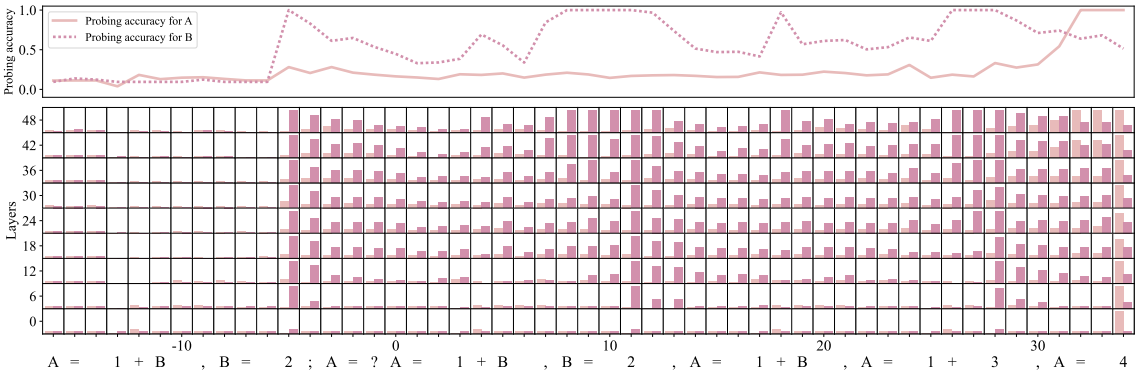

Figure 32: Probing results when Yi-1.5-9B solves Level 1.

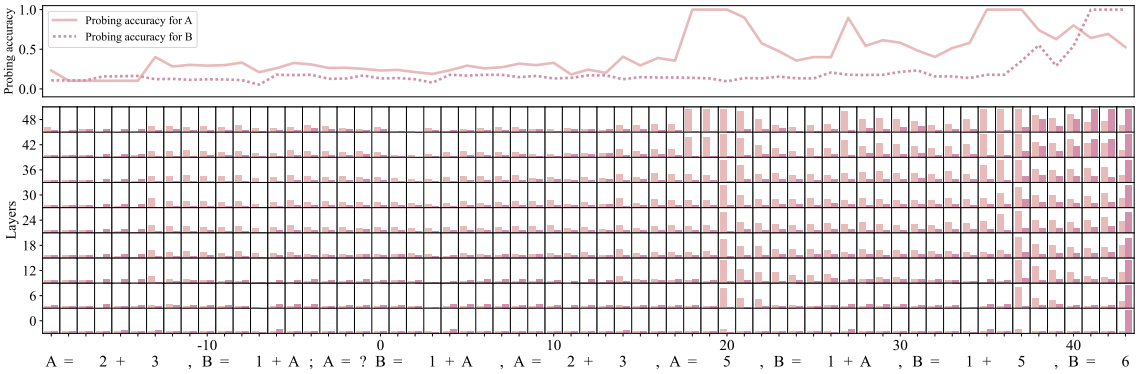

Figure 33: Probing results when Yi-1.5-9B solves Level 2.

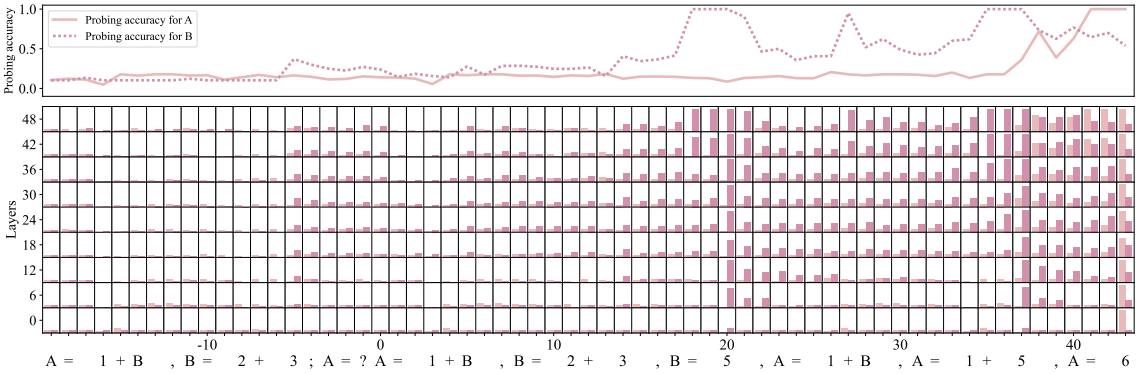

Figure 34: Probing results when Yi-1.5-9B solves Level 3.

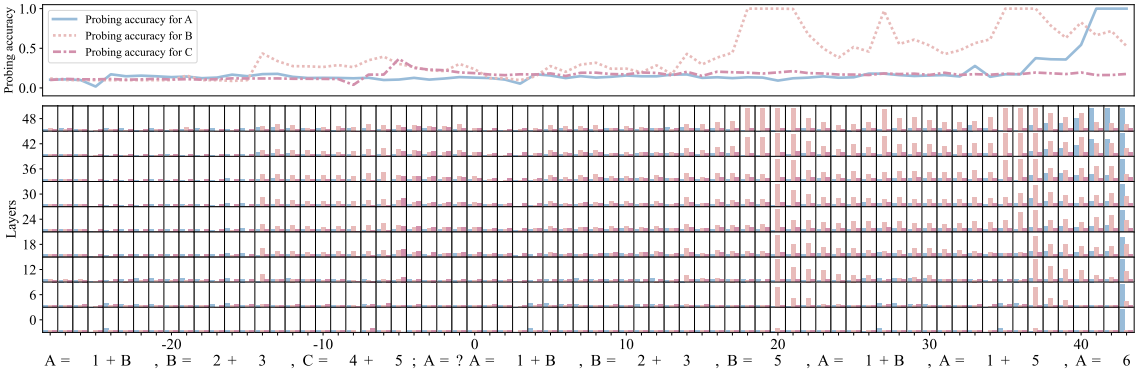

Figure 35: Probing results when Yi-1.5-9B solves Level 4.

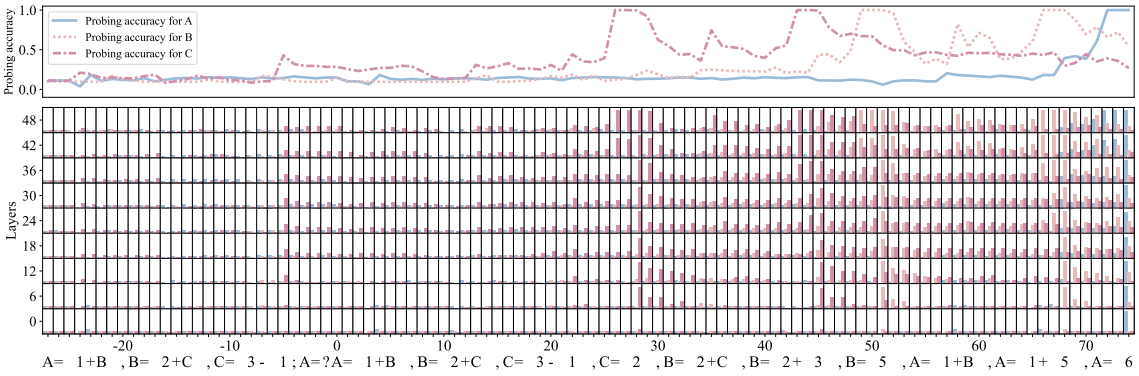

Figure 36: Probing results when Yi-1.5-9B solves Level 5.

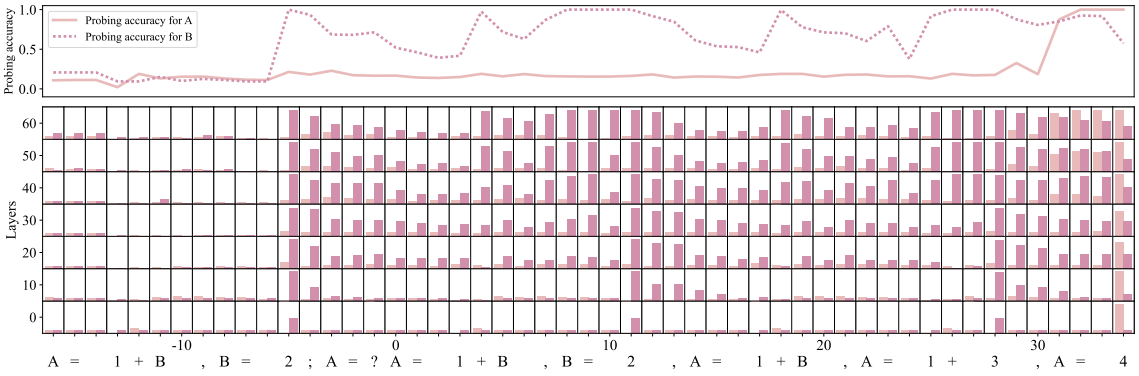

Figure 37: Probing results when Yi-1.5-34B solves Level 1.

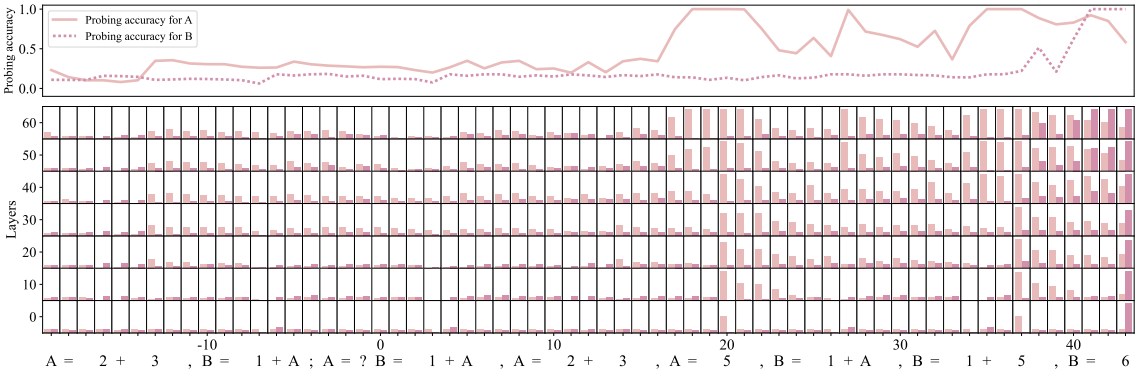

Figure 38: Probing results when Yi-1.5-34B solves Level 2.

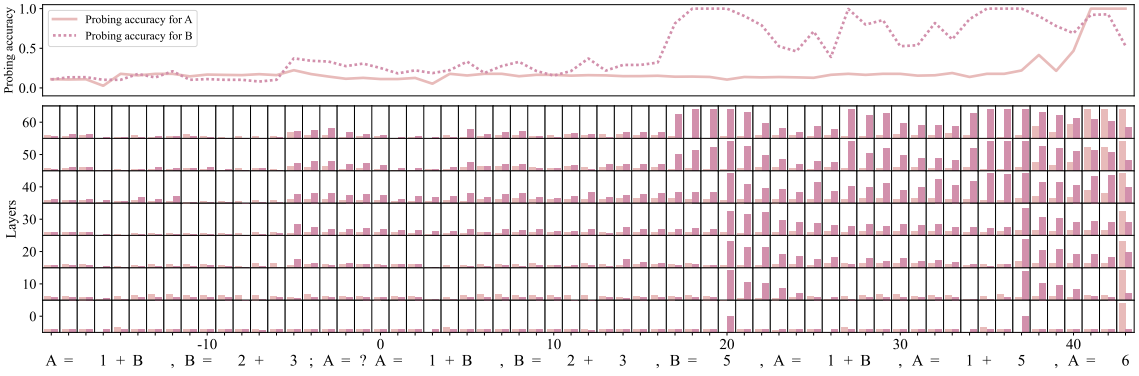

Figure 39: Probing results when Yi-1.5-34B solves Level 3.

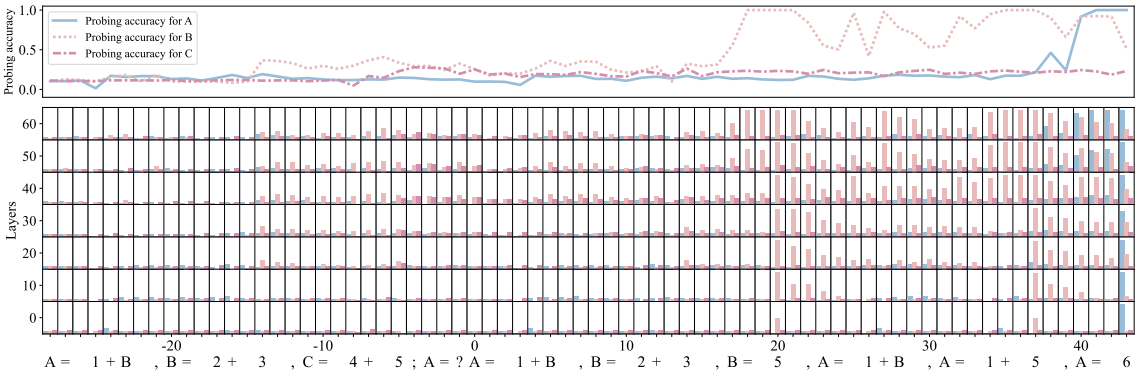

Figure 40: Probing results when Yi-1.5-34B solves Level 4.

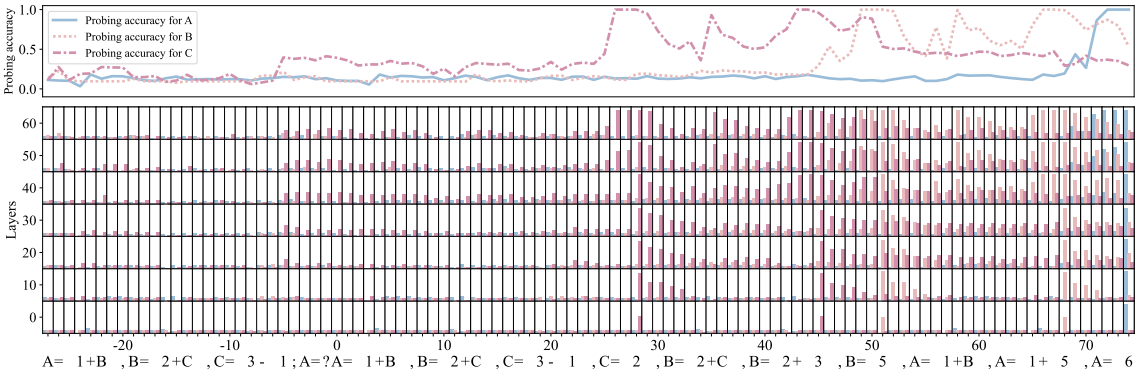

Figure 41: Probing results when Yi-1.5-34B solves Level 5.

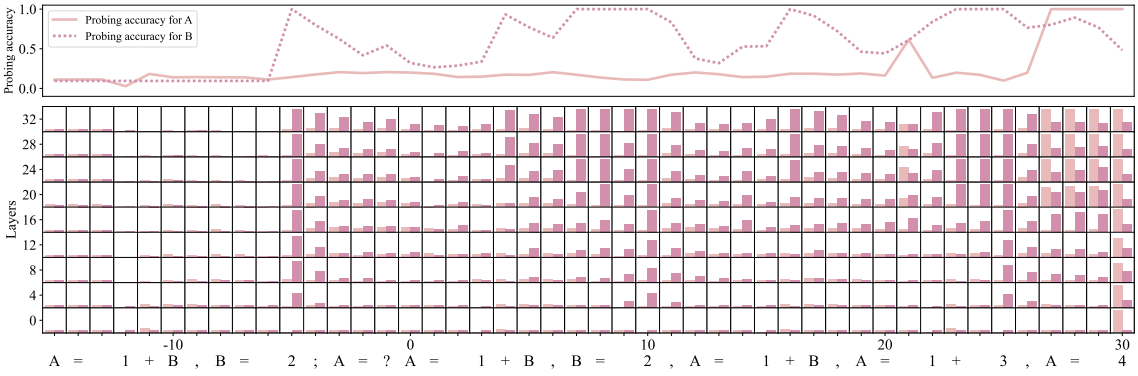

Figure 42: Probing results when Llama-3.1-8B solves Level 1.

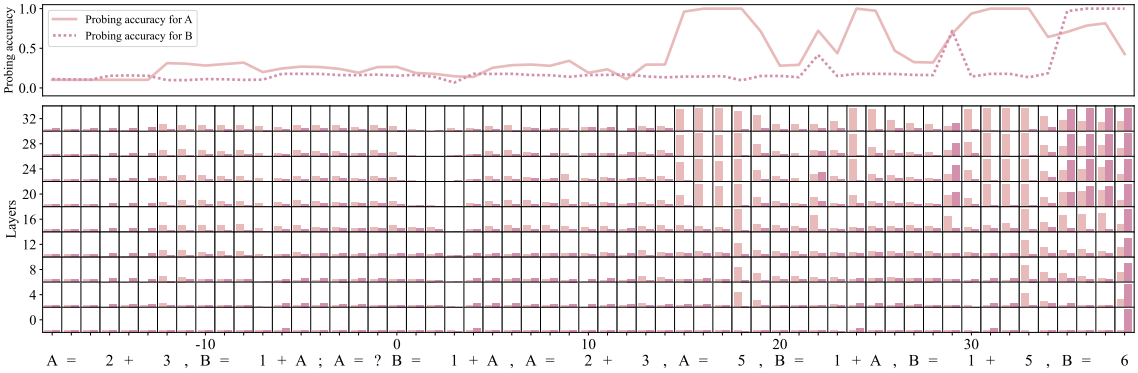

Figure 43: Probing results when Llama-3.1-8B solves Level 2.

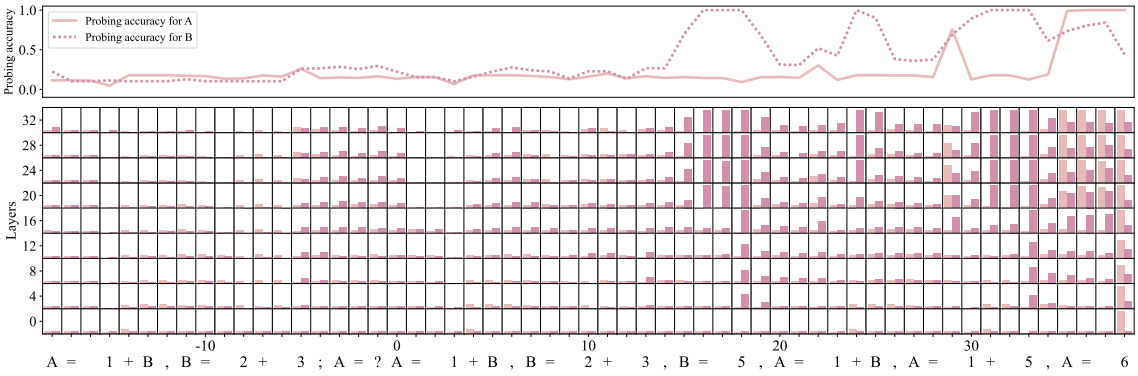

Figure 44: Probing results when Llama-3.1-8B solves Level 3.

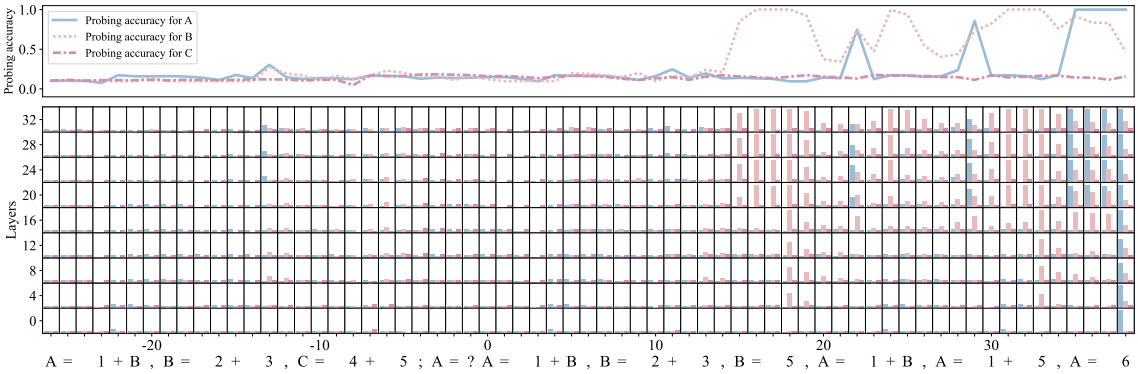

Figure 45: Probing results when Llama-3.1-8B solves Level 4.

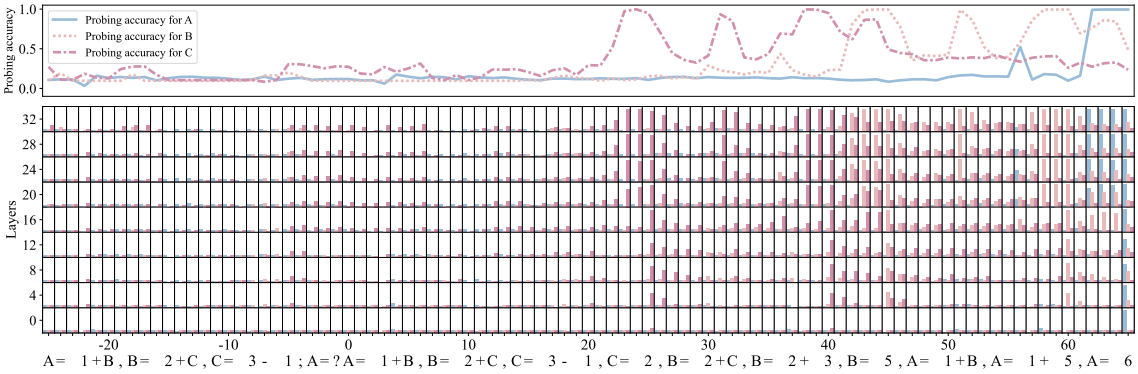

Figure 46: Probing results when Llama-3.1-8B solves Level 5.

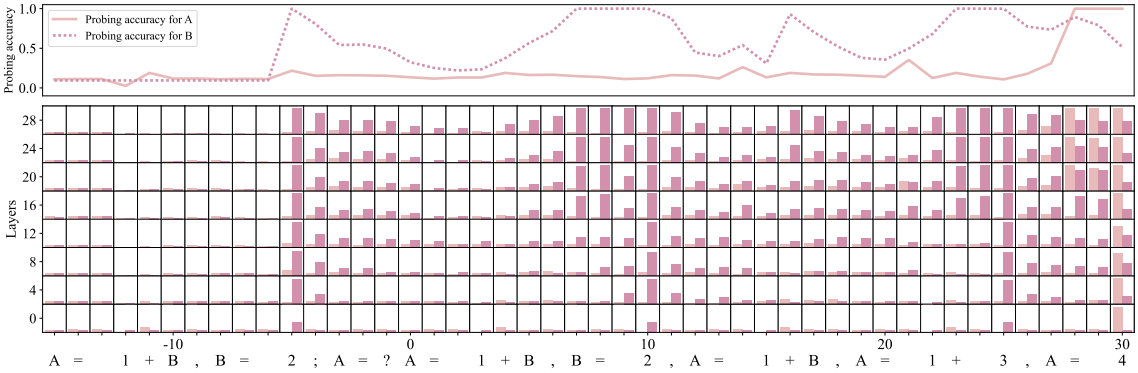

Figure 47: Probing results when Llama-3.2-3B solves Level 1.

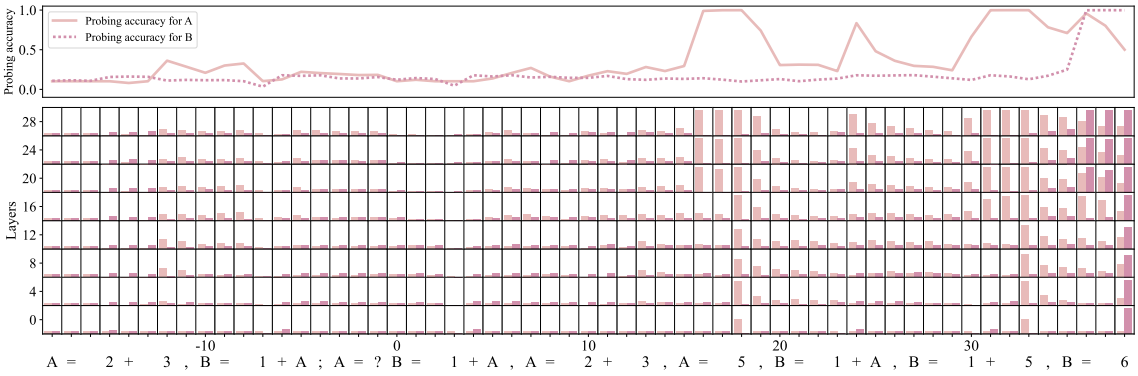

Figure 48: Probing results when Llama-3.2-3B solves Level 2.

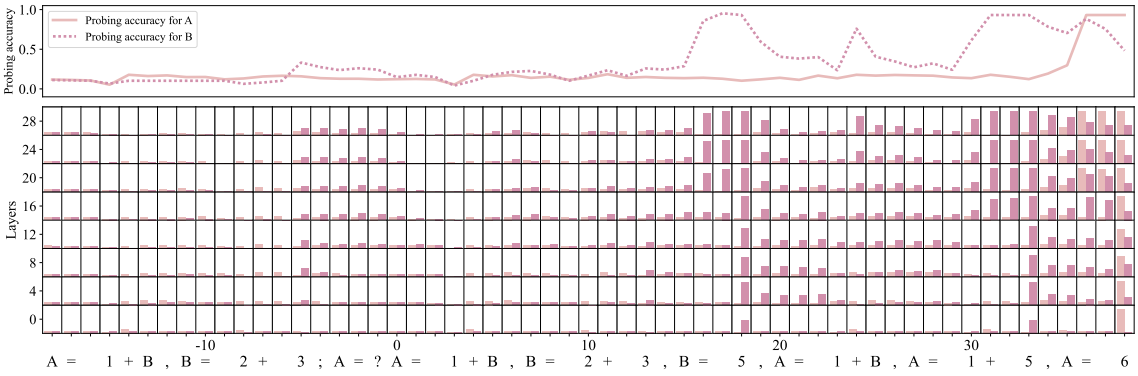

Figure 49: Probing results when Llama-3.2-3B solves Level 3.

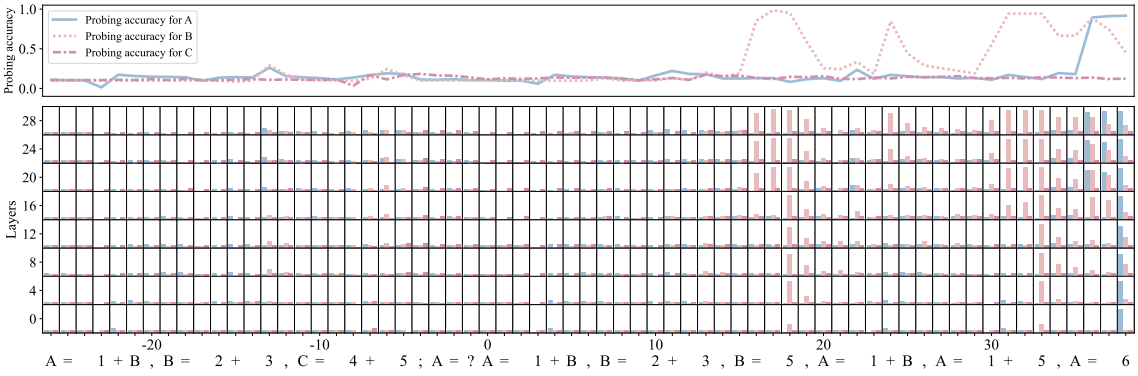

Figure 50: Probing results when Llama-3.2-3B solves Level 4.

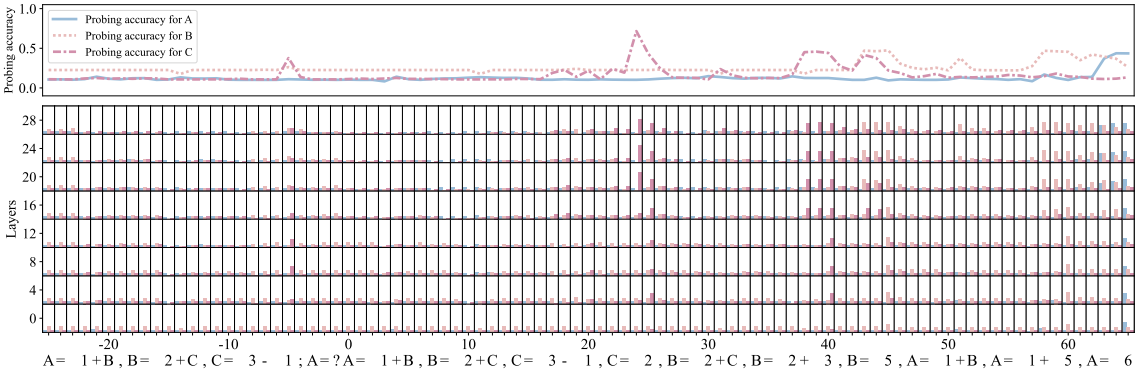

Figure 51: Probing results when Llama-3.2-3B solves Level 5.

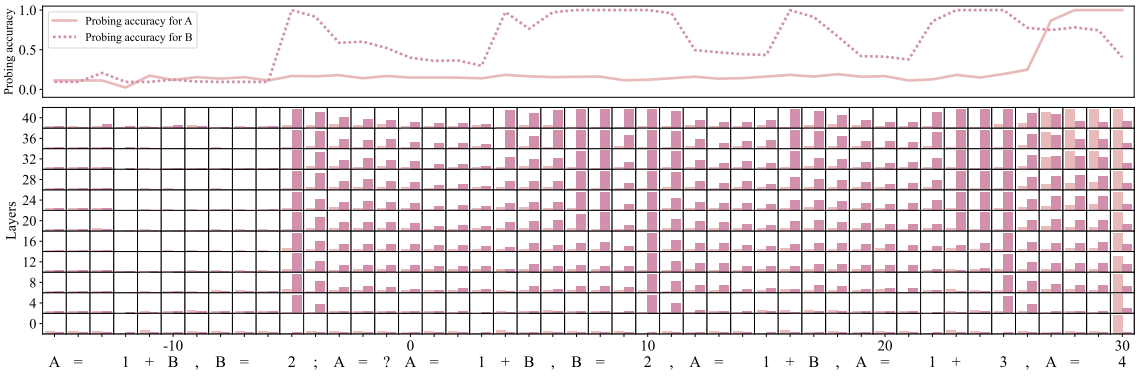

Figure 52: Probing results when Mistral-Nemo-Base-2407 solves Level 1.

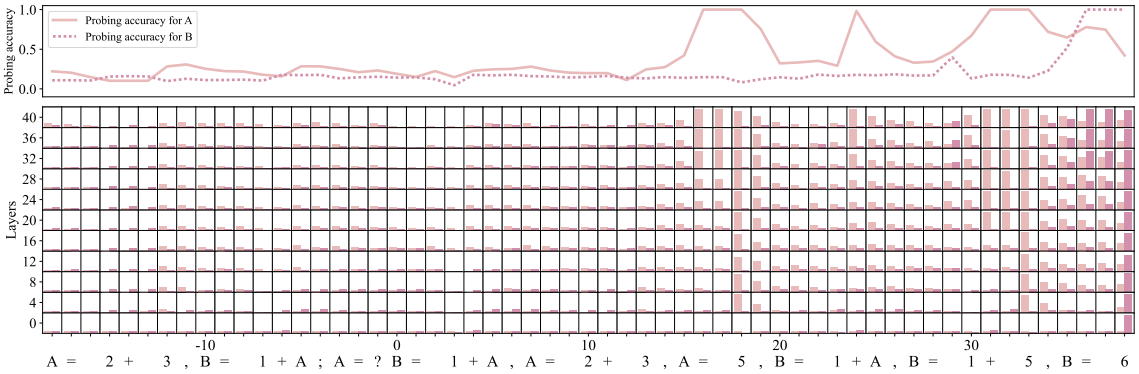

Figure 53: Probing results when Mistral-Nemo-Base-2407 solves Level 2.

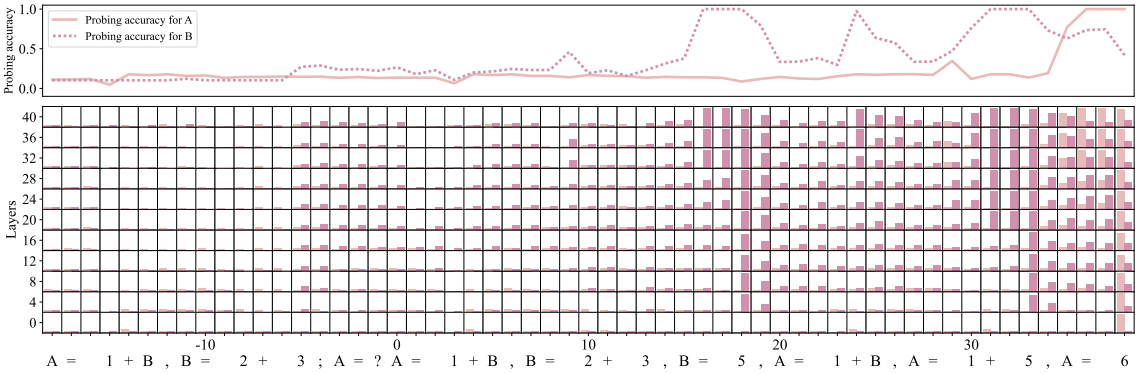

Figure 54: Probing results when Mistral-Nemo-Base-2407 solves Level 3.

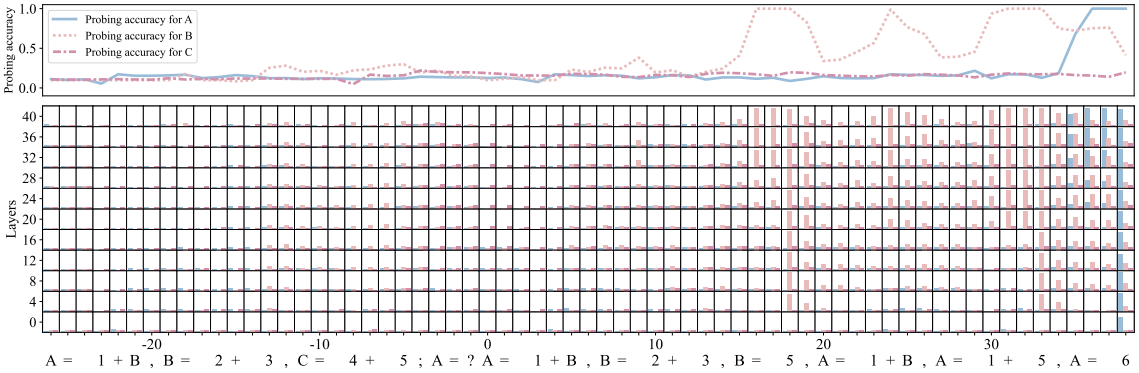

Figure 55: Probing results when Mistral-Nemo-Base-2407 solves Level 4.

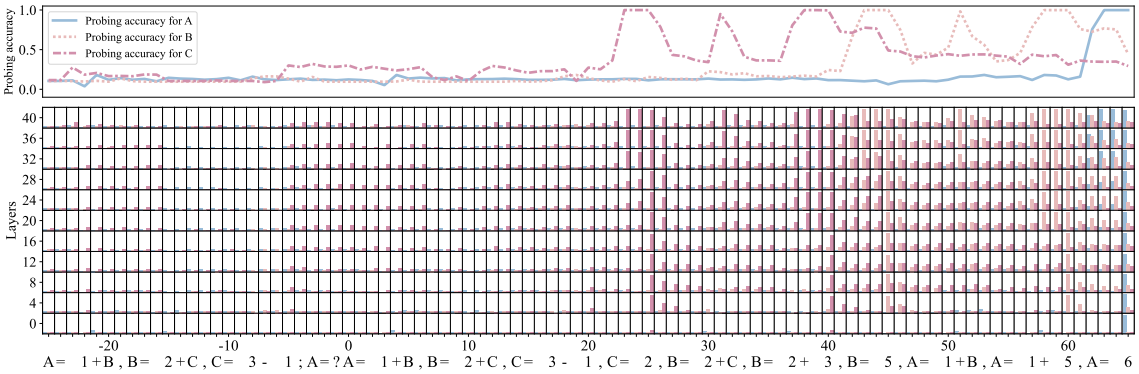

Figure 56: Probing results when Mistral-Nemo-Base-2407 solves Level 5.

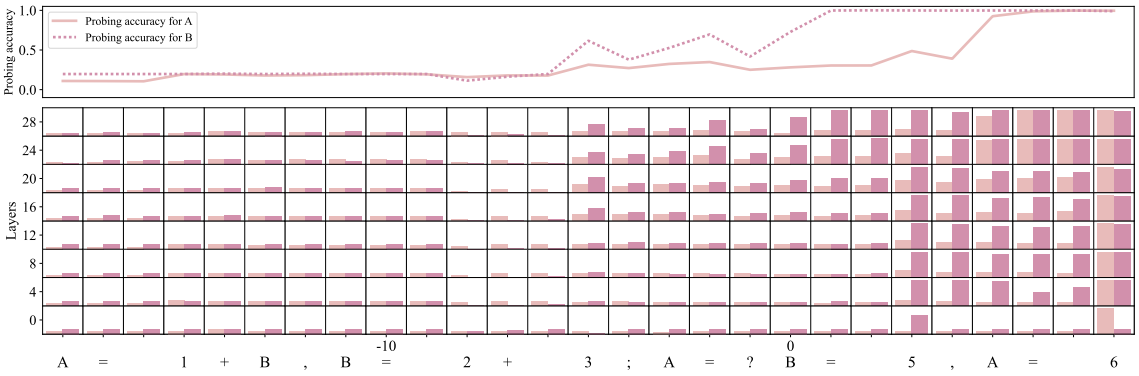

Figure 57: Probing results when Qwen2.5-7B solves Level 2 with **Simple CoT**.

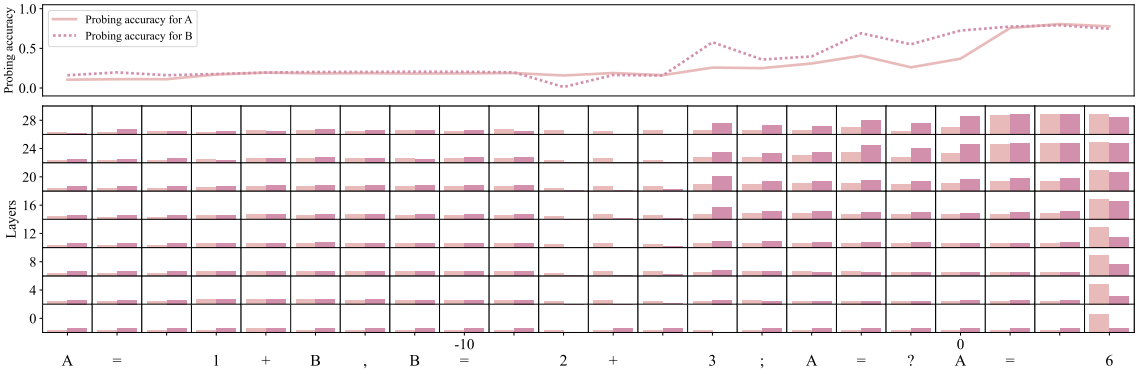

Figure 58: Probing results when Qwen2.5-7B solves Level 2 with **Implicit reasoning**.

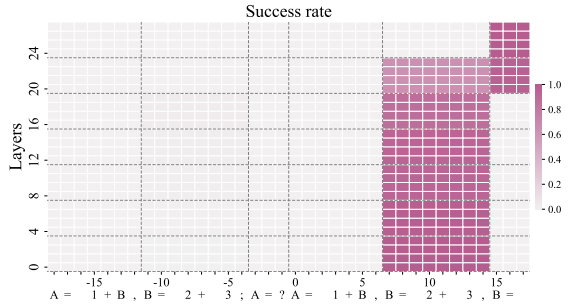

Figure 59: Results of the causal intervention on Qwen2.5-7B. Each grid cell shows the success rate when the intermediate token z_{17} ($\underline{\text{B} = 5}_{\mathbf{2}}$) is the target token.

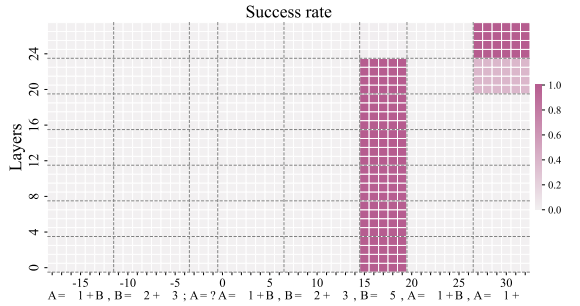

Figure 60: Results of the causal intervention on Qwen2.5-7B. Each grid cell shows the success rate when z_{32} ($\underline{\text{A} = 1{+}5}_{\mathbf{4}}$) is the target token.

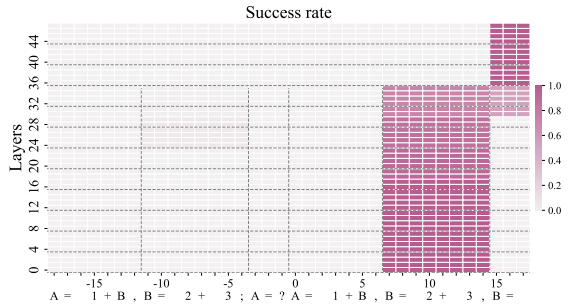

Figure 61: Results of the causal intervention on Qwen2.5-14B. Each grid cell shows the success rate when the intermediate token z_{17} ($\underline{\text{B} = 5}_{\mathbf{2}}$) is the target token.

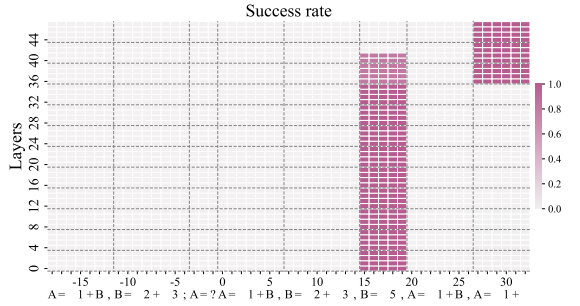

Figure 62: Results of the causal intervention on Qwen2.5-14B. Each grid cell shows the success rate when z_{32} ($\underline{\text{A} = 1{+}5}_{\mathbf{4}}$) is the target token.

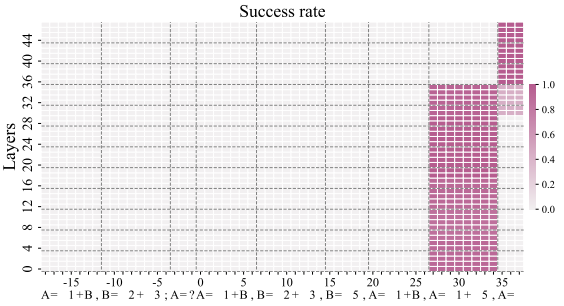

Figure 63: Results of the causal intervention on Qwen2.5-14B. Each grid cell shows the success rate when the final answer y ($\underline{\text{A} = \mathbf{6}}_{\mathbf{5}}$) is the target token.

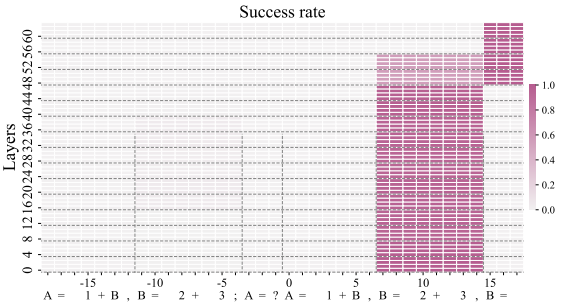

Figure 64: Results of the causal intervention on Qwen2.5-32B. Each grid cell shows the success rate when the intermediate token z_{17} ($\underline{\text{B} = 5}_{\mathbf{2}}$) is the target token.

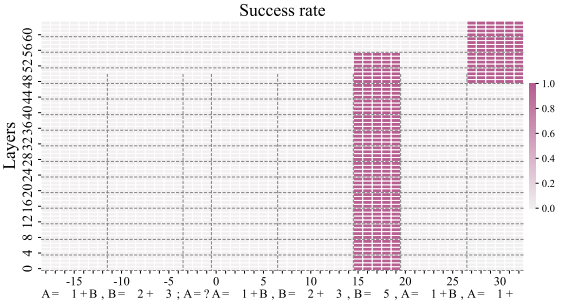

Figure 65: Results of the causal intervention on Qwen2.5-32B. Each grid cell shows the success rate when z_{32} ($\underline{\text{A} = 1{+}5}_{\mathbf{4}}$) is the target token.

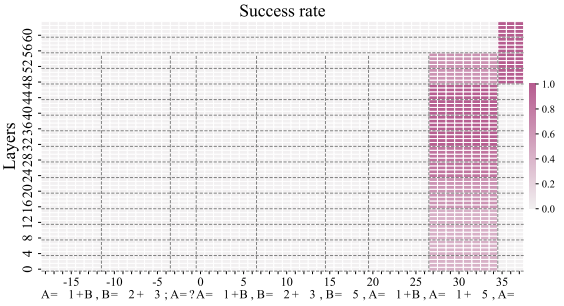

Figure 66: Results of the causal intervention on Qwen2.5-32B. Each grid cell shows the success rate when the final answer y ($\underline{\text{A} = \mathbf{6}}_{\mathbf{5}}$) is the target token.

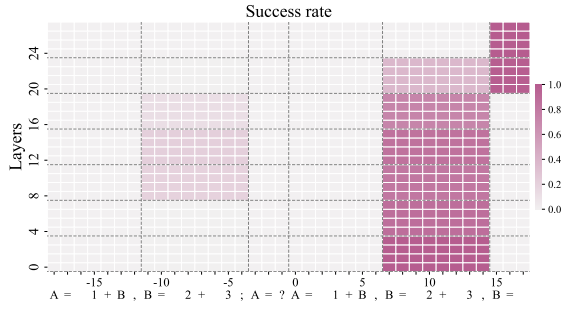

Figure 67: Results of the causal intervention on Qwen2.5-Math-7B. Each grid cell shows the success rate when the intermediate token z_{17} ($\underline{\text{B =5}}_{\mathbf{2}}$) is the target token.

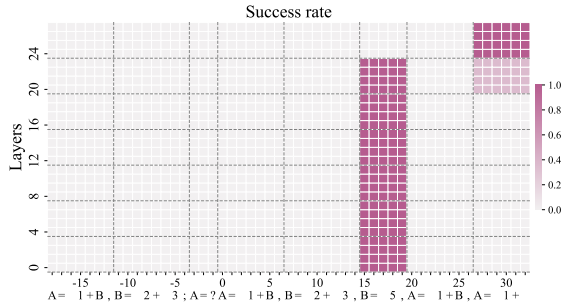

Figure 68: Results of the causal intervention on Qwen2.5-Math-7B. Each grid cell shows the success rate when z_{32} ($\underline{\text{A = 1+5}}_{\mathbf{4}}$) is the target token.

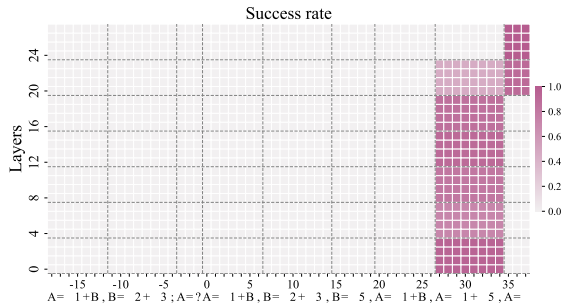

Figure 69: Results of the causal intervention on Qwen2.5-Math-7B. Each grid cell shows the success rate when the final answer y ($\underline{\text{A =}\mathbf{6}}_{\mathbf{5}}$) is the target token.

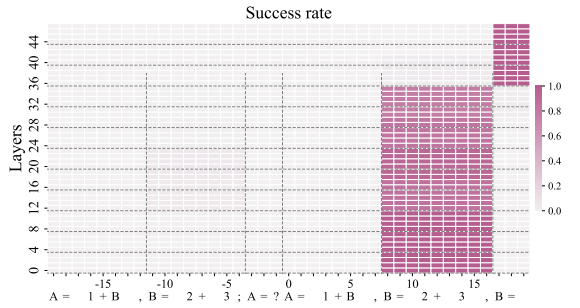

Figure 70: Results of the causal intervention on Yi-1.5-9B. Each grid cell shows the success rate when the intermediate token z_{17} ($\underline{\text{B =5}}_{\mathbf{2}}$) is the target token.

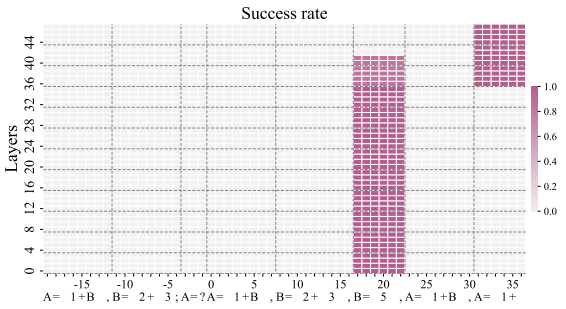

Figure 71: Results of the causal intervention on Yi-1.5-9B. Each grid cell shows the success rate when z_{32} ($\underline{\text{A = 1+5}}_{\mathbf{4}}$) is the target token.

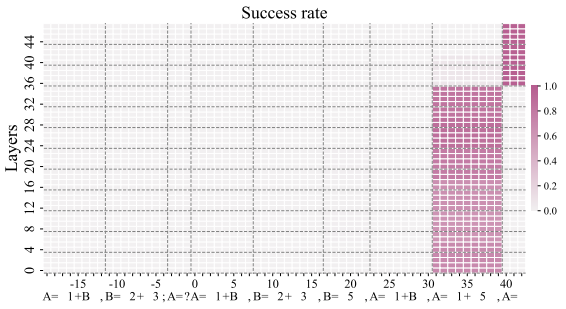

Figure 72: Results of the causal intervention on Yi-1.5-9B. Each grid cell shows the success rate when the final answer y ($\underline{\text{A =}\mathbf{6}}_{\mathbf{5}}$) is the target token.

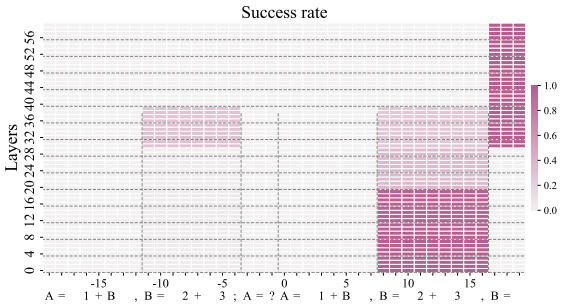

Figure 73: Results of the causal intervention on Yi-1.5-34B. Each grid cell shows the success rate when the intermediate token z_{17} ($\underline{\text{B =5}}_{\mathbf{2}}$) is the target token.

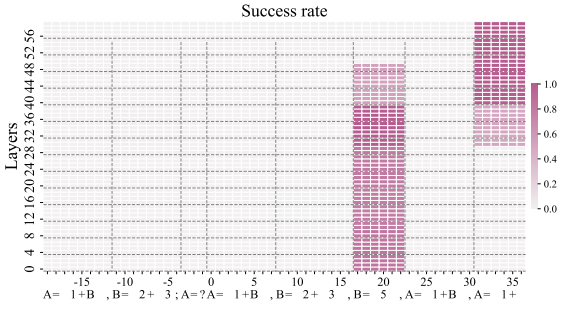

Figure 74: Results of the causal intervention on Yi-1.5-34B. Each grid cell shows the success rate when z_{32} ($\underline{\text{A = 1+5}}_{\mathbf{4}}$) is the target token.

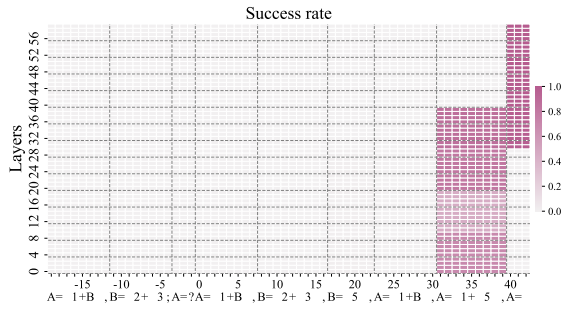

Figure 75: Results of the causal intervention on Yi-1.5-34B. Each grid cell shows the success rate when the final answer y ($\underline{\text{A} = \mathbf{6}_{\mathbf{5}}}$) is the target token.

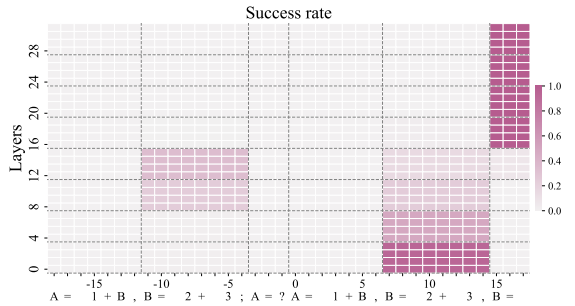

Figure 76: Results of the causal intervention on Llama-3.1-8B. Each grid cell shows the success rate when the intermediate token z_{17} ($\underline{\text{B} = 5_{\mathbf{2}}}$) is the target token.

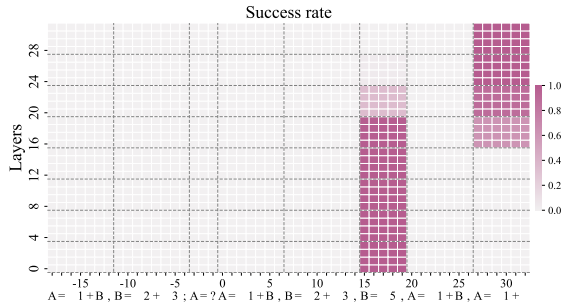

Figure 77: Results of the causal intervention on Llama-3.1-8B. Each grid cell shows the success rate when z_{32} ($\underline{\text{A} = 1{+}5_{\mathbf{4}}}$) is the target token.

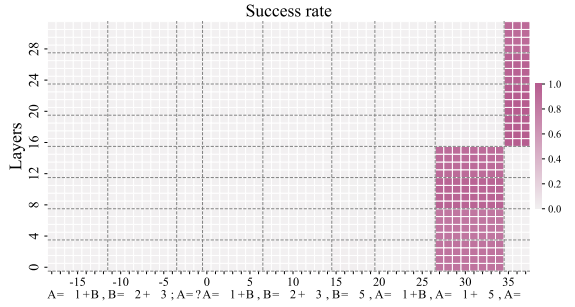

Figure 78: Results of the causal intervention on Llama-3.1-8B. Each grid cell shows the success rate when the final answer y ($\underline{\text{A} = \mathbf{6}_{\mathbf{5}}}$) is the target token.

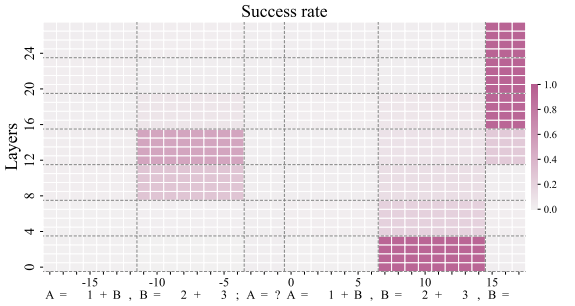

Figure 79: Results of the causal intervention on Llama-3.2-3B. Each grid cell shows the success rate when the intermediate token z_{17} ($\underline{\text{B} = 5_{\mathbf{2}}}$) is the target token.

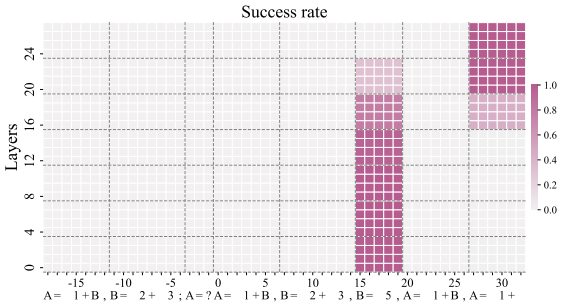

Figure 80: Results of the causal intervention on Llama-3.2-3B. Each grid cell shows the success rate when z_{32} ($\underline{\text{A} = 1{+}5_{\mathbf{4}}}$) is the target token.

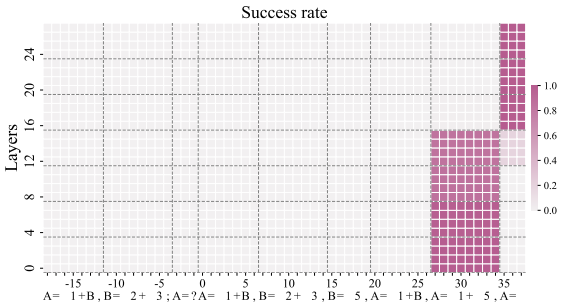

Figure 81: Results of the causal intervention on Llama-3.2-3B. Each grid cell shows the success rate when the final answer y ($\underline{\text{A} = \mathbf{6}_{\mathbf{5}}}$) is the target token.

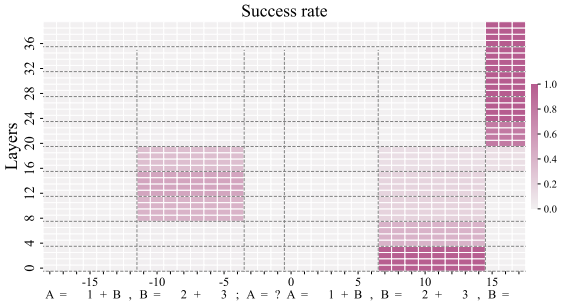

Figure 82: Results of the causal intervention on Mistral-Nemo-Base-2407. Each grid cell shows the success rate when the intermediate token z_{17} ($\underline{\text{B} = 5_{\mathbf{2}}}$) is the target token.

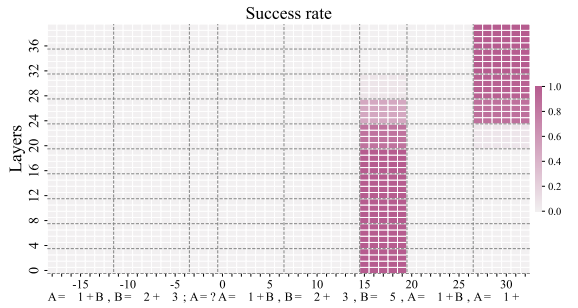

Figure 83: Results of the causal intervention on Mistral-Nemo-Base-2407. Each grid cell shows the success rate when z_{32} (A = 1+5 **4**) is the target token.

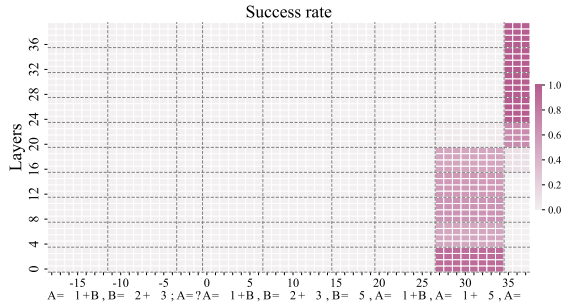

Figure 84: Results of the causal intervention on Mistral-Nemo-Base-2407. Each grid cell shows the success rate when the final answer y (A = **6** **5**) is the target token.

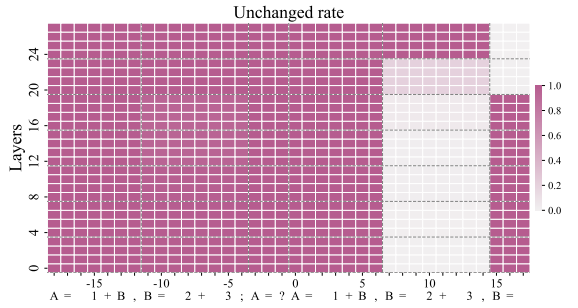

Figure 85: Results of the causal intervention on Qwen2.5-7B. Each grid cell shows the unchanged rate when the intermediate token z_{17} (B = 5 **2**) is the target token.

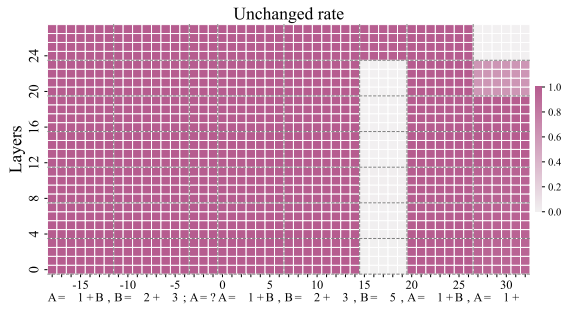

Figure 86: Results of the causal intervention on Qwen2.5-7B. Each grid cell shows the unchanged rate when z_{32} (A = 1+5 **4**) is the target token.

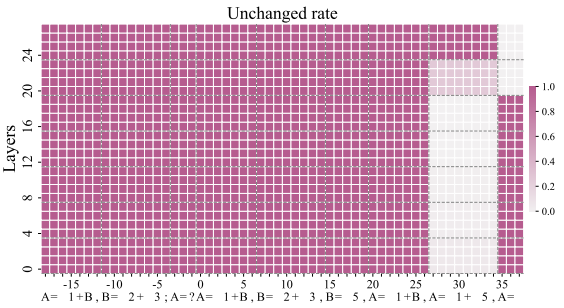

Figure 87: Results of the causal intervention on Qwen2.5-7B. Each grid cell shows the unchanged rate when the final answer y (A = **6** **5**) is the target token.

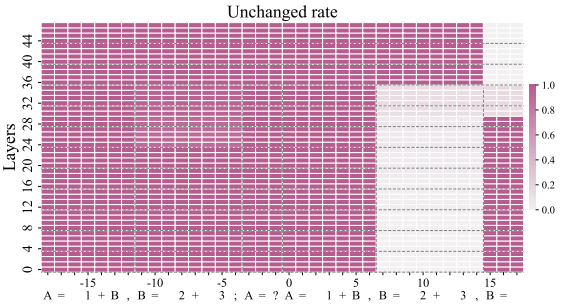

Figure 88: Results of the causal intervention on Qwen2.5-14B. Each grid cell shows the unchanged rate when the intermediate token z_{17} (B = 5 **2**) is the target token.

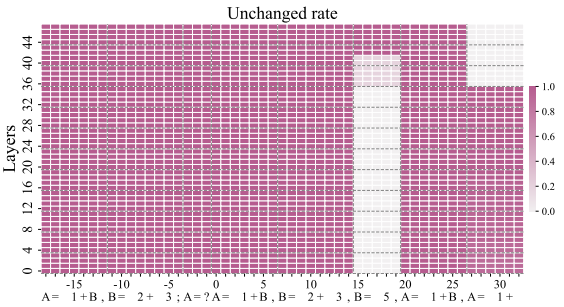

Figure 89: Results of the causal intervention on Qwen2.5-14B. Each grid cell shows the unchanged rate when z_{32} (A = 1+5 **4**) is the target token.

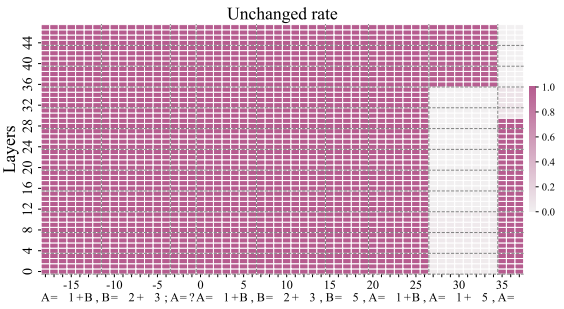

Figure 90: Results of the causal intervention on Qwen2.5-14B. Each grid cell shows the unchanged rate when the final answer y (A = **6** **5**) is the target token.

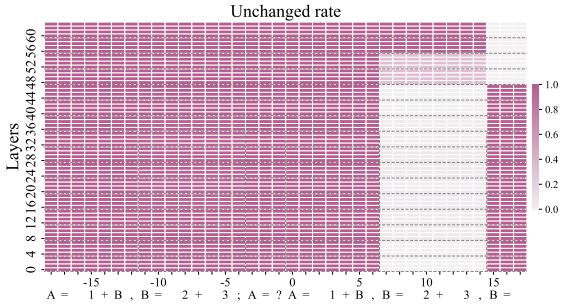

Figure 91: Results of the causal intervention on Qwen2.5-32B. Each grid cell shows the unchanged rate when the intermediate token z_{17} ($\underline{\text{B} = 5}_{\mathbf{2}}$) is the target token.

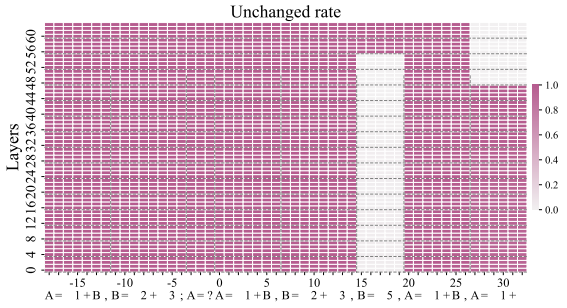

Figure 92: Results of the causal intervention on Qwen2.5-32B. Each grid cell shows the unchanged rate when z_{32} ($\underline{\text{A} = 1{+}5}_{\mathbf{4}}$) is the target token.

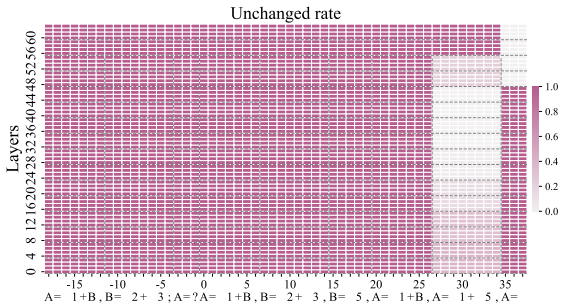

Figure 93: Results of the causal intervention on Qwen2.5-32B. Each grid cell shows the unchanged rate when the final answer y ($\underline{\text{A} = \mathbf{6}}_{\mathbf{5}}$) is the target token.

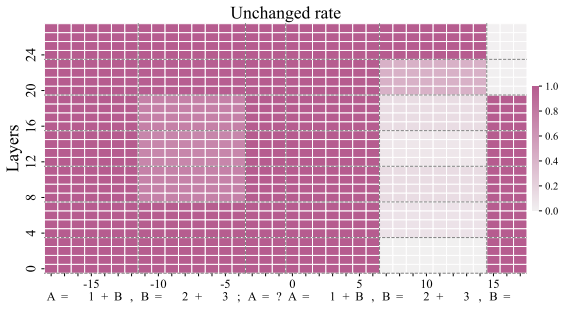

Figure 94: Results of the causal intervention on Qwen2.5-Math-7B. Each grid cell shows the unchanged rate when the intermediate token z_{17} ($\underline{\text{B} = 5}_{\mathbf{2}}$) is the target token.

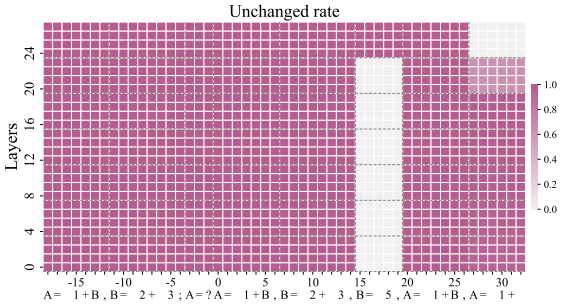

Figure 95: Results of the causal intervention on Qwen2.5-Math-7B. Each grid cell shows the unchanged rate when z_{32} ($\underline{\text{A} = 1{+}5}_{\mathbf{4}}$) is the target token.

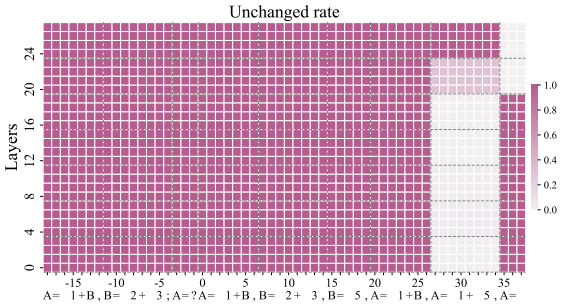

Figure 96: Results of the causal intervention on Qwen2.5-Math-7B. Each grid cell shows the unchanged rate when the final answer y ($\underline{\text{A} = \mathbf{6}}_{\mathbf{5}}$) is the target token.

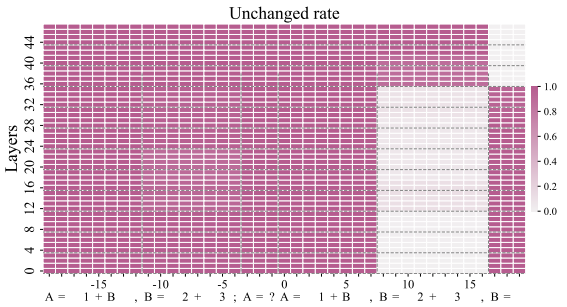

Figure 97: Results of the causal intervention on Yi-1.5-9B. Each grid cell shows the unchanged rate when the intermediate token z_{17} ($\underline{\text{B} = 5}_{\mathbf{2}}$) is the target token.

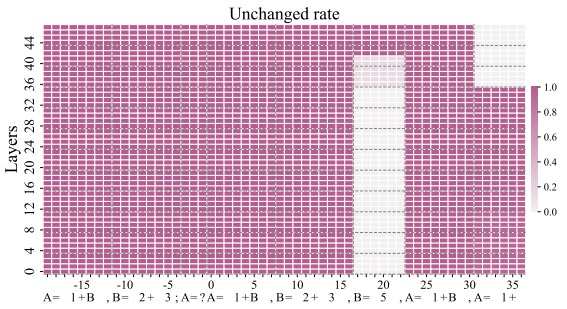

Figure 98: Results of the causal intervention on Yi-1.5-9B. Each grid cell shows the unchanged rate when z_{32} ($\underline{\text{A} = 1{+}5}_{\mathbf{4}}$) is the target token.

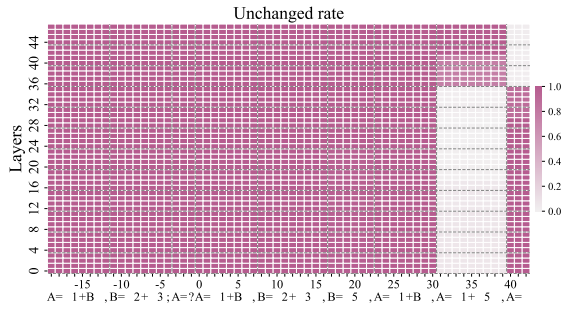

Figure 99: Results of the causal intervention on Yi-1.5-9B. Each grid cell shows the unchanged rate when the final answer y ($\underline{\text{A} = \mathbf{6}_{\mathbf{5}}}$) is the target token.

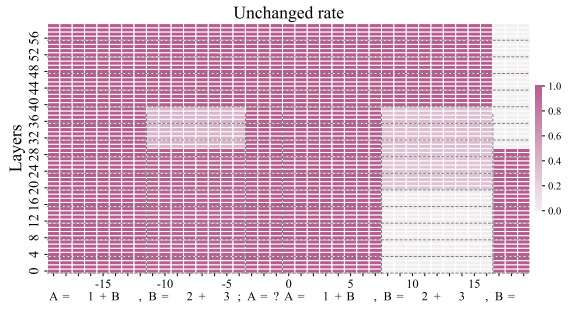

Figure 100: Results of the causal intervention on Yi-1.5-34B. Each grid cell shows the unchanged rate when the intermediate token z_{17} ($\underline{\text{B} = 5_{\mathbf{2}}}$) is the target token.

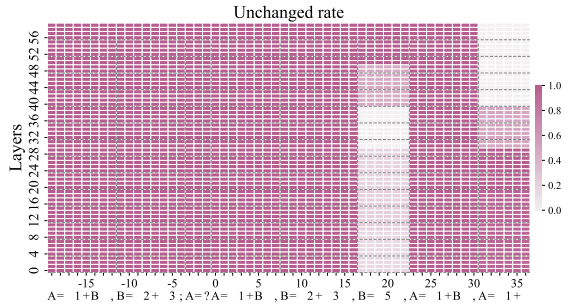

Figure 101: Results of the causal intervention on Yi-1.5-34B. Each grid cell shows the unchanged rate when z_{32} ($\underline{\text{A} = 1{+}5_{\mathbf{4}}}$) is the target token.

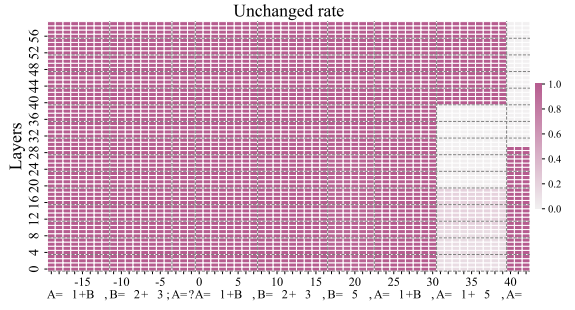

Figure 102: Results of the causal intervention on Yi-1.5-34B. Each grid cell shows the unchanged rate when the final answer y ($\underline{\text{A} = \mathbf{6}_{\mathbf{5}}}$) is the target token.

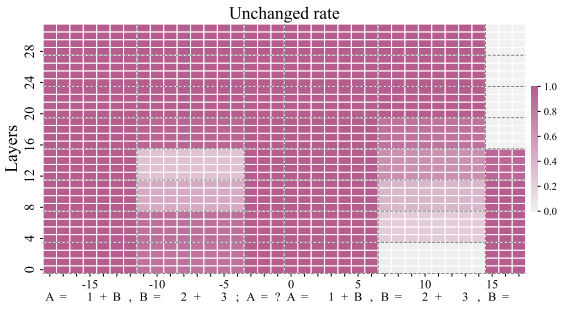

Figure 103: Results of the causal intervention on Llama-3.1-8B. Each grid cell shows the unchanged rate when the intermediate token z_{17} ($\underline{\text{B} = 5_{\mathbf{2}}}$) is the target token.

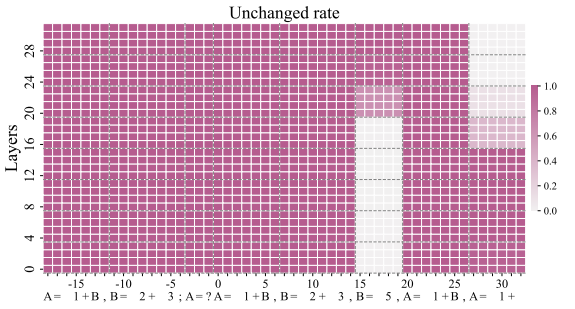

Figure 104: Results of the causal intervention on Llama-3.1-8B. Each grid cell shows the unchanged rate when z_{32} ($\underline{\text{A} = 1{+}5_{\mathbf{4}}}$) is the target token.

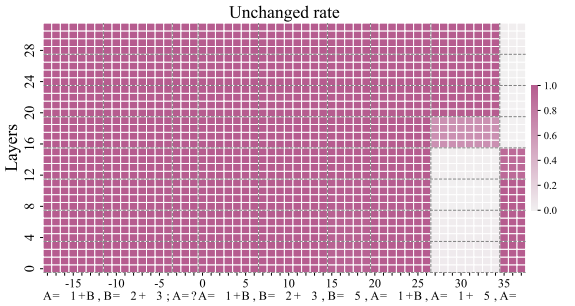

Figure 105: Results of the causal intervention on Llama-3.1-8B. Each grid cell shows the unchanged rate when the final answer y ($\underline{\text{A} = \mathbf{6}_{\mathbf{5}}}$) is the target token.

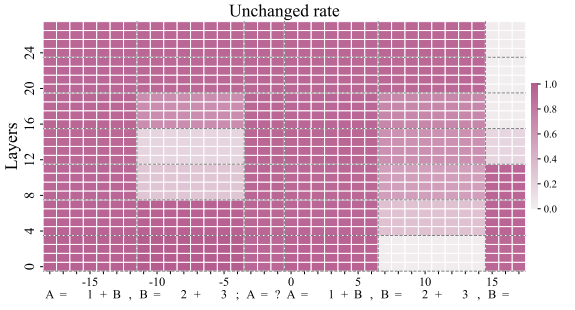

Figure 106: Results of the causal intervention on Llama-3.2-3B. Each grid cell shows the unchanged rate when the intermediate token z_{17} ($\underline{\text{B} = 5_{\mathbf{2}}}$) is the target token.

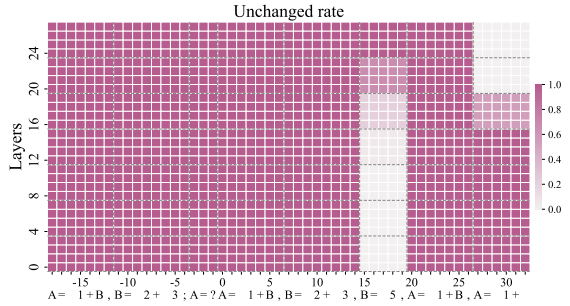

Figure 107: Results of the causal intervention on Llama-3.2-3B. Each grid cell shows the unchanged rate when z_{32} ($\underline{\text{A = 1+5}}_{\mathbf{4}}$) is the target token.

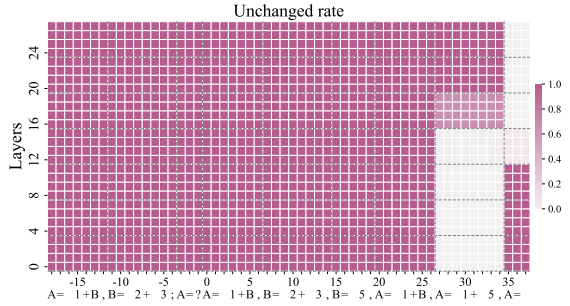

Figure 108: Results of the causal intervention on Llama-3.2-3B. Each grid cell shows the unchanged rate when the final answer y ($\underline{\text{A =}\mathbf{6}_{\mathbf{5}}}$) is the target token.

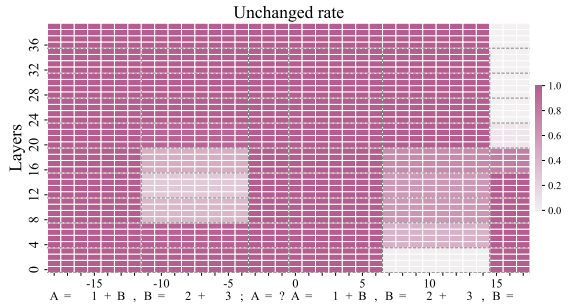

Figure 109: Results of the causal intervention on Mistral-Nemo-Base-2407. Each grid cell shows the unchanged rate when the intermediate token z_{17} ($\underline{\text{B =5}}_{\mathbf{2}}$) is the target token.

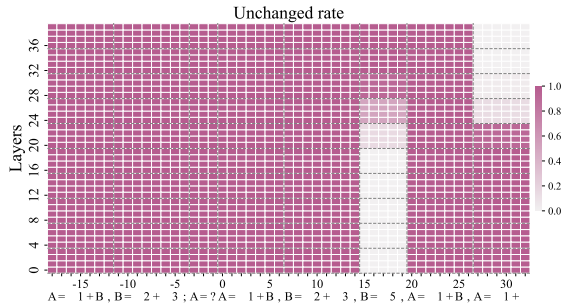

Figure 110: Results of the causal intervention on Mistral-Nemo-Base-2407. Each grid cell shows the unchanged rate when z_{32} ($\underline{\text{A = 1+5}}_{\mathbf{4}}$) is the target token.

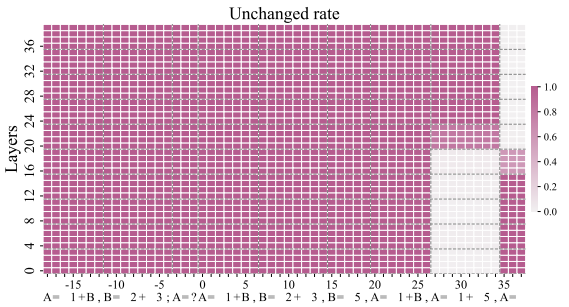

Figure 111: Results of the causal intervention on Mistral-Nemo-Base-2407. Each grid cell shows the unchanged rate when the final answer y ($\underline{\text{A =}\mathbf{6}_{\mathbf{5}}}$) is the target token.