多段算術推論タスクにおける思考の連鎖の忠実性

工藤 慧音 [1,2]　青木 洋一 [1,2]　栗林 樹生 [3,1]　曾根 周作 [1]　谷口 雅弥 [2,1]　Ana Brassard[2]
坂口 慶祐 [1,2]　乾 健太郎 [3,1,2]
[1] 東北大学　[2] 理化学研究所　[3]MBZUAI
keito.kudo.q4@dc.tohoku.ac.jp

概要

本研究では，言語モデルが思考の連鎖の結果として出力する最終解答が，その思考の連鎖に対して忠実であるかを評価する．特に，算術推論タスクを用いた実験を通して，(i) 言語モデル内部ではいつ途中結果/最終解答が導出されるのか，(ii) 思考の連鎖からの情報が最終解答に対してどの程度強い因果的影響があるのか，に焦点を当て調査する．実験の結果，言語モデルは入力が与えられた時点では内部的にも解答を導出しておらず，思考の連鎖の過程で(内部的に) 解答を導出していることが明らかとなった．この結果は，思考の連鎖はモデルの内部計算を忠実に反映したものであることを示唆している．

1　はじめに

言語モデルは利用者からの入力を与えられると，典型的には最終解答だけでなくその解答に至るまでの推論過程も生成する．このような推論過程は思考の連鎖 / Chain of Thought (以下 CoT) [1] と呼ばれ，しばしばモデル自身の推論プロセスの説明とみなされる [2, 3]．このような説明に対する懸念は，CoT による説明が，最終解答に対してどの程度忠実であるか，すなわち CoT と最終解答の間の因果関係である．例えば，CoT 生成前の問題文の読み込み中にすでに最終解答に到達しており，その後，その解答に対する説明を後付け的に生成する場合が考えられる．この場合，最終解答は厳密には CoT の説明に忠実であるとは言えない．本研究では言語モデルの内部分析の観点から，この CoT による説明とモデルの最終解答との間の忠実性を分析する．この問いを，モデルは CoT による推論中にいつ解答を思いつくのか，そしてそれらが内部的にどのように参照されているのか，という 2 つの問いに分解し調査する．

本研究では，初めに線形プローブ [4] を用いて，記号的な算術推論問題データセット上でモデルが内部的にいつ（どの層，どの位置で）解答を導出するのかを分析する（図 1）．各位置でのプローブの正解率を比較することで，どの時点でモデルが内部的に(途中の) 計算結果の情報を持ち始めるか（変数の値が線形分離可能な表現として隠れ状態に表現されるか）を確認できる．実験の結果，モデルは問題を最初に読んだ時点では内部的に解答を導出しておらず，CoT を生成しながら途中結果/最終解答に到達していることが明らかとなった．これは最終解答に対する CoT の忠実性を示唆している．さらに，モデルの内部表現と解答との間の因果関係を明らかにするために，因果介入分析を実施した（§ 4）．その結果，モデルの最終解答は，CoT 開始後の内部表現への因果介入によって変化させることができるが，問題部分への介入では変化しないことがわかった．この結果はプロービングによる結果と整合していた．

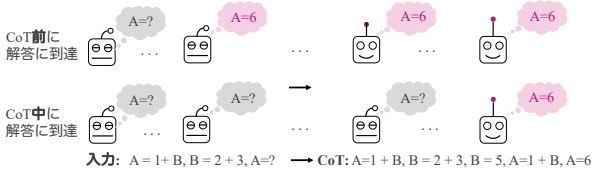

図 1　本研究の概要図．線形プローブ/因果介入により，言語モデルは CoT の過程のどの時点で各変数の値を内部的に予測しているかを調査する．

2　実験設定

2.1　算術推論タスク

既存研究 [5, 6] と同様に，記号的な算術推論問題データセットを作成する．このデータセットの各事例は，代入（例：A=1）及び，演算（例：B=1+3 または B=1+A）の 2 つの操作と，ある変数の値を問う質問文（例：B=?）から構成される．我々は 5 つの異なる複雑度を持つタスクを定義し分析を行う．

表 1　算術推論タスクの例．#Step は最終解答に到達するために必要な演算の回数，また，各式の右下に示された値は，式番号を表している．

複雑度	入力	出力	#Step
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
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
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
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
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

データセットの定義　v をアルファベット 26 文字 $\Sigma = \{\mathrm{a}, \mathrm{b}, \mathrm{c}, \ldots, \mathrm{z}\}$ からサンプリングされた変数名とする．また，d は $D = \{0, 1, 2, \ldots, 9\}$ からサンプリングされた数字とする．各事例は複数の式 $[e_1, e_2, \cdots, e_n]$ と，質問文 q から構成される．各式は $v = d$，$v = d \pm d$，または $v = d \pm v$ のいずれかの形式である．さらに，各事例の左から i 番目の変数を v_i と表記する．例えば，表 1 の複雑度 5 の例では，$v_1 = \mathtt{A}$，$v_2 = \mathtt{B}$，$v_3 = \mathtt{C}$ である[1)]．この変数 v_i に割り当てられた値を $\${v_i}$ とする．その他のデータセットの詳細については § A を参照されたい．

推論設定　§ 2.1 で述べた問題を言語モデルに与えられると，モデルは CoT z と最終解答 y を生成する．以降，CoT より前の部分を入力 x，CoT 推論部分 (y, z) を出力と呼ぶ．例えば，表 1 複雑度 1 の事例の場合，x = “A=1+B，B=2”，z = “A=1+B，B=2，A=1+2，A=”，y = “3” である．期待される CoT による推論形式を促すため，同じ複雑度の問題 3 つをモデルに例示する few-shot learning を行う．妥当性の確認として，実験対象モデルが期待される出力形式に従い，この設定でほぼ 100%の正解率でタスクを解くことができることを事前に確認した (§ B).

3　線形プロービング

はじめに，線形プローブにより途中結果/最終解答がモデル内部のどこから抽出できるかを分析する．分析は 9 種類の異なるサイズや種類の言語モデルに対して行う．本文では，代表例として Qwen2.5-7B [7] に対する分析結果を報告する．他のモデルの結果は § B を参照されたい．

以下では，入力系列全体 $x \oplus z \oplus y$ におけるトークン位置を $t \in \mathbb{Z}$ で表す．この位置 t は CoT の開始位置を基準とした相対位置で表現する．すなわち，t は CoT が始まる位置でゼロであり，入力部では負，出力部では正となる．同様に，各数式に対して，式番号 $t_{\mathrm{eq}} \in \mathbb{Z}$ を割り当てる（表 1 の下線部の添え字）．

3.1　線形プローブの学習

トークン位置 $t \in \mathbb{Z}$，層の深さ $l \in \mathbb{N}$，変数 $v_i \in \Sigma$ の組み合わせごとに，個別にプローブを訓練する．形式的には，言語モデルの d 次元隠れ状態 $\boldsymbol{h}_{t,l} \in \mathbb{R}^d$ が与えられたとき，プローブ $f_{t,l,v_i}(\cdot) : \mathbb{R}^d \to D$ は，解答 (変数の値) $\${v_i}$ を予測するように訓練される．もしプローブ f_{t,l,v_i} がテストセットにおいて高い正解率を達成すれば，変数 $\${v_i}$ に対する解答が，位置 t，層 l において導出されていることを意味する．

線形プローブ f_{t,l,v_i} は以下のように定義される：

$$\$\{\hat{v}_i\}_{t,l} = f_{t,l,v_i}(\boldsymbol{h}_{t,l}) = \arg\max \boldsymbol{W}_{t,l,v_i} \boldsymbol{h}_{t,l} + \boldsymbol{b}_{t,l,v_i} \quad , \quad (1)$$

ここで，$\boldsymbol{W}_{t,l,v_i} \in \mathbb{R}^{|D| \times d}$ および $\boldsymbol{b}_{t,l,v_i} \in \mathbb{R}^{|D|}$ は，それぞれ学習可能な重みおよびバイアス項である．また，$\hat{\cdot}$ はモデルの予測解答を指す．

3.2　評価指標

トークン位置 t，層 l ごとのプロービング結果を集約する指標として以下の $t^*(v_i)$ を定義する．

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

ここで $\mathrm{acc}(t, l, v_i) \in [0, 1]$ は f_{t,l,v_i} のプローブの正解率を表す．この $t^*(v_i) \in \mathbb{Z}$ は，プローブが τ を超える正解率を初めて達成した位置を意味する．本研究では $\tau = 0.9$ とする．また，位置 $t^*(v_i)$ が入力 x 内のどの式番号 t_{eq} に該当するかを示す $t^*_{\mathrm{eq}}(v_i)$ も報告する．もし $t^*_{\mathrm{eq}}(v_i)$ が負であれば，$\${v_i}$ は CoT の前に

1)　簡単のため，本論文中のすべての例では大文字の変数 A, B, C と演算子 + のみを使用している．実際の事例では変数名や演算子の種類は多様である点に注意されたい．

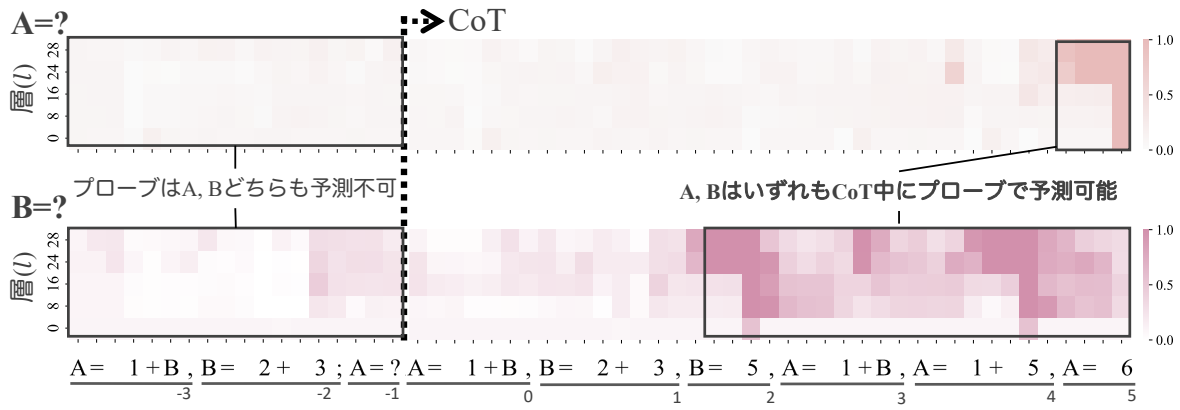

図 2　Qwen2.5-7B の複雑度 3 のタスクにおけるプロービング結果．各セルは位置 t，層 l のプローブの正解率を示す．

表 2　5 つの複雑度における Qwen2.5-7B の結果．N/A は，どの位置 t においても閾値 τ を超えなかったことを示す．

複雑度	変数	#Step	t^*_{eq}	$t^{\text{inf}}_{\text{eq}}$	$\text{Acc}_{\prec\text{CoT}}$	$\text{Acc}_{\succ\text{CoT}}$
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

モデル内部で導出された可能性が示唆される．

さらに，以下の 2 種類の正解率も報告する：

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

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

$\text{Acc}_{\prec\text{CoT}}(v_i)$ が高い場合には，$\{v_i\}$ は CoT が始まるより前に解答が導出されていることが示唆される．一方で，$\text{Acc}_{\prec\text{CoT}}(v_i)$ が低く $\text{Acc}_{\succ\text{CoT}}(v_i)$ が高い場合には，CoT 中に導出されていることが示唆される．

3.3　結果

表 2 に，5 つの複雑度の異なるタスクにおいて，各変数での t^*_{eq} と，解答が最初に導出可能になる位置の下限 $t^{\text{inf}}_{\text{eq}}$ を示す．ほとんどの場合，途中結果は出力部分に対応するモデルの隠れ状態において，線形分離可能な形で表現されていた（$t^*_{\text{eq}} > 0$）．例外ケースは (i) 複雑度 1 の v_2，および (ii) 複雑度 4 の v_3 の場合であったが，いずれも，複雑度 1 の v_2 は計算を必要としない変数であり（#Steps= 0），複雑度 4 の v_3 は最終解答を導くのに不要な計算結果である．以上から，最終解答を得るために必要なすべての計算は **CoT の間に**実行されており，事前に決定された解答をモデルの最終解答として参照している可能性は低いことが示唆された．この結果は，他のモデルでも共通した傾向であった (§ B)．

4　因果介入

§ 3 の結果から，最終解答は CoT が始まった後に導出されると暫定的に結論付けた．つまり，CoT は最終解答に対して忠実である．本節では，因果介入実験によりこの結論を裏付けるとともに，CoT 中のモデル内部での情報の流れを明らかにする．

4.1　実験設定

介入手法　因果介入手法として，既存研究で広く採用されている手法であるアクティベーションパッチング（activation patching）[8, 9, 10] を行う．特定の隠れ状態 $\boldsymbol{h}_{t,l}$ と最終解答 y との間の因果関係を明らかにするために，(i) 通常の推論と (ii) 介入ありの推論での比較を行う．介入ありの推論時には，特定の隠れ状態 $\boldsymbol{h}_{t,l}$ を，同じモデルで異なる入力 $\tilde{x}$ から得られた $\tilde{\boldsymbol{h}}_{t,l}$ に置き換えて推論を行う．入力 x と $\tilde{x}$ は，それぞれ異なる正解 y と $\tilde{y}$ および推論過程 z と $\tilde{z}$ を持つ (例: x = “A=1+B,B=2+4;A=?”, z = “A=1+B,...,A=1+6,A=7”, y =7 の時，$\tilde{x}$ = “A=2+B,B=1+3;A=?”, $\tilde{z}$ = “A=2+B,...,A=2+4,A=6”, $\tilde{y}$ =6)．$\boldsymbol{h}_{t,l}$ を $\tilde{\boldsymbol{h}}_{t,l}$ に置き換えたとき，モデルの出力が y から $\tilde{y}$ へ，あるいは z_t から $\tilde{z}_t$ へと変化した場合，$\boldsymbol{h}_{t,l}$ と元の解答 y (または z_t) との間に因果的な関係があると結論づけられる．以降，通常の推

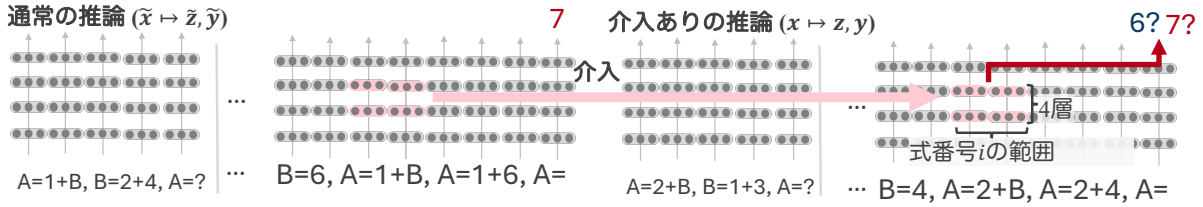

図 3　因果介入実験の概要．はじめに，通常の推論を行い，その時の隠れ状態を保存する．次に，異なる問題を解くモデルの隠れ状態の一部を保存されたものに置き換えて推論を行う．この結果，出力が変化するかを評価する．

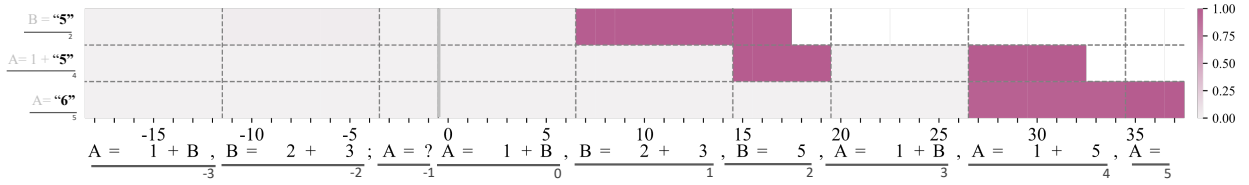

図 4　因果介入実験の結果．各グリッドはその位置に介入を行った際の介入成功率を示している．

論時のモデルの最終出力を $\hat{y}$，介入ありの出力を $\hat{y}$ とそれぞれ表記する（$\tilde{y}$ と y はそれぞれの正解を表す）．同様に，介入なしで生成された推論連鎖を $\hat{z}_t$，介入ありの場合を $\hat{z}_t$ と表記する．

評価指標　評価指標として，介入ありの推論時の出力 $\hat{y}$ が正解 $\tilde{y}$ と一致した割合である**介入成功率**を定義する．途中結果 $\hat{z}_t$ に対しても同様に計算する．その後，各式 (位置) ごとに最大の介入成功率を算出する．

介入対象　特に Qwen2.5-7B が複雑度 3 のタスクを解く設定に着目する[2)]．既存研究 [10] に着想を得て，図 4 のように，隠れ状態を各式および 4 層ごとにグリッドに分割し，個別に介入を実施した時の介入成功率を計算する．また，推論の過程で重要と考えられるトークンを分析対象トークンと定め，これらを生成する際の因果分析を実施する．具体的には (i) 式番号 2 の数式の最後の数字（例: 図 2 における $\underline{\text{B =}\mathbf{5}_2}$ の z_{17}），(ii) 式番号 4 の数式の最後の数字（例: 図 2 $\underline{\text{A = 1+}\mathbf{5}_4}$ の z_{32}），および (iii) 最終解答（y）を分析対象トークンとする．介入ありの推論時には，分析対象トークンのみを貪欲法により生成し，それ以前の CoT はコンテキストとしてモデルに入力する．

4.2　実験結果

図 4 は，3 種類の分析対象トークンにおける各式ごとの最大介入成功率を示している．全ての分析対象トークンが出力部分と強い因果的影響がある一方で，入力部分の影響は限定的であった．この結果は入力部分で処理された情報が (途中の) 計算結果に与える影響が限定的であることを示唆する．この傾向は Qwen2.5-Math-7B を除く，Qwen 系列のモデルにおいて顕著であった．対照的に，他の種類のモデルでは $\underline{\text{B =}\mathbf{5}_2}$ と 入力部分との間に弱い因果関係を示していた（§ B）．これは，モデルは CoT 中にその場で解答を導き出しており，生成された推論過程はモデルの最終解答に対して忠実であるとみなせるというプロービングによる結果 § 3.3 と整合している．

また介入は，介入した隠れ状態が (i) 分析対象トークンと同じ式の位置にある，(ii) 必要な情報が記述されている最後の式の位置にある（例：B=2+3→B=5），または (iii) 関連する変数の値が明示的に記述されている最後の式の位置にある（例：B=5→A=1+5）場合にのみ成功していた．この知見は，言語モデルによる多段推論の内部処理において強い直近バイアスが存在することを示唆している．

5　おわりに

我々は算術推論問題を用いて，CoT による推論中に最終解答/途中結果がモデル内部でいつ導出されるかを特定するために，線形プローブを用いた分析を実施した．複数のモデル/タスクの難易度にわたる実験の結果，言語モデルは CoT 開始後に必要な最終解答/途中結果を導出していることが示唆された．また，因果介入による実験でも，CoT 開始後の内部状態への介入が最終解答に対して因果的に影響を与えることが確認された．少なくとも我々の制御された実験設定においては，CoT は最終解答に対して忠実であることが示唆された．

2)　他のモデルに対する実験結果は § B を参照されたい．

謝辞

本研究は JST CREST JPMJCR20D2，JSPS 科研費 JP25KJ0615/JP25K03175，JST SPRING JPMJSP2114 の助成を受けたものです．また，Ana Brassard の貢献は，理化学研究所の奨励課題 24 の助成を受けたものです．本研究の一部は九州大学情報基盤研究開発センター研究用計算機システムの「一般利用」を利用した．終わりに，本研究を進めるにあたり多くの協力を賜りました Tohoku NLP グループの皆様に感謝申し上げます．

参考文献

[1] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, **Advances in Neural Information Processing Systems**, 2022.

[2] Martin Tutek, Fateme Hashemi Chaleshtori, Ana Marasovic, and Yonatan Belinkov. Measuring chain of thought faithfulness by unlearning reasoning steps. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, **Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing**, pp. 9935–9960, Suzhou, China, November 2025. Association for Computational Linguistics.

[3] Yanda Chen, Joe Benton, Ansh Radhakrishnan, Jonathan Uesato, Carson Denison, John Schulman, Arushi Somani, Peter Hase, Misha Wagner, Fabien Roger, Vladimir Mikulik, Samuel R. Bowman, Jan Leike, Jared Kaplan, and Ethan Perez. Reasoning models don't always say what they think. **CoRR**, Vol. abs/2505.05410, , 2025.

[4] Guillaume Alain and Yoshua Bengio. 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, 2017.

[5] Keito Kudo, Yoichi Aoki, Tatsuki Kuribayashi, Ana Brassard, Masashi Yoshikawa, Keisuke Sakaguchi, and Kentaro Inui. 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)**, pp. 1351–1362, 2023.

[6] Yijiong Yu. Do LLMs really think step-by-step in implicit reasoning?, 2025.

[7] Qwen Team. Qwen2.5: A party of foundation models, September 2024.

[8] Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. Investigating gender bias in language models using causal mediation analysis. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, **Advances in Neural Information Processing Systems**, Vol. 33, pp. 12388–12401. Curran Associates, Inc., 2020.

[9] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, **Advances in Neural Information Processing Systems**, Vol. 35, pp. 17359–17372. Curran Associates, Inc., 2022.

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

[11] An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. **arXiv preprint arXiv:2409.12122**, 2024.

[12] 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, Xiaohui Hu, Xiaoyi Ren, Xinyao Niu, Pengcheng Nie, Yuchi Xu, Yudong Liu, Yue Wang, Yuxuan Cai, Zhenyu Gu, Zhiyuan Liu, and Zonghong Dai. Yi: Open foundation models by 01.ai. **CoRR**, Vol. abs/2403.04652, , 2024.

[13] 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, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. The Llama 3 herd of models. **CoRR**, Vol. abs/2407.21783, , 2024.

[14] Mistral AI Team. Mistral NeMo, July 2024.

[15] Anum Afzal, Florian Matthes, Gal Chechik, and Yftah Ziser. Knowing before saying: LLM representations encode information about chain-of-thought success before completion. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, **Findings of the Association for Computational Linguistics: ACL 2025**, pp. 12791–12806, Vienna, Austria, July 2025. Association for Computational Linguistics.

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

[17] Anonymous. Post-hoc reasoning in chain-of-thought: Evidence from pre-cot probes and activation steering. In **Submitted to The Fourteenth International Conference on Learning Representations**, 2025. under review.

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

表 3　各言語モデルの算術推論タスクにおける性能．“正解率”の行は，テストセットにおける CoT$_z$ と最終解答 y を結合した文字列 $z \oplus y$ との完全一致率を示す．

	複雑度	正解率
Qwen2.5 (7B)	1	100
	2	100
	3	100
	4	100
	5	100

	複雑度	正解率
Qwen2.5 (14B)	3	100
Qwen2.5 (32B)	3	100
Qwen2.5-Math (7B)	3	100
Yi1.5 (9B)	3	100
Yi1.5 (34B)	3	100
Llama3.1 (8B)	3	100
Llama3.2 (3B)	3	97.6
Mistral-Nemo (12B)	3	99.6

表 4　複雑度 3 のタスクにおけるプロービング結果．

	変数	t^*_{eq}	t^*	$\text{Acc}_{\prec \text{CoT}}$	$\text{Acc}_{\succ \text{CoT}}$
Qwen2.5 (14B)[7]	v_1 (A)	5	35	17.8	100
	v_2 (B)	2	16	50.5	100
Qwen2.5 (32B)[7]	v_1 (A)	5	36	17.8	100
	v_2 (B)	2	15	67.4	100
Qwen2.5-Math (7B)[11]	v_1 (A)	5	35	18.6	100
	v_2 (B)	2	15	56.1	100
Yi1.5 (9B)[12]	v_1 (A)	5	41	17.8	100
	v_2 (B)	2	18	36.9	100
Yi1.5 (34B)[12]	v_1 (A)	5	41	22.4	100
	v_2 (B)	2	18	37.4	100
Llama3.1 (8B)[13]	v_1 (A)	5	35	26.0	100
	v_2 (B)	2	16	29.6	100
Llama3.2 (3B)[13]	v_1 (A)	5	36	17.8	93.2
	v_2 (B)	2	17	33.2	95.4
Mistral-Nemo (12B)[14]	v_1 (A)	5	36	17.8	100
	v_2 (B)	2	16	28.9	100

A　データセットの詳細

データセットの各事例には，任意の v_i に対して $\{v_i\}$ が 1 桁の数字であること，および同一事例内では $\{v_i\}$ が定数である（すなわち，`A=1+2,A=B+2,B=6` のようなケースは除外する）という制約を設ける．また，同じ複雑度のすべての事例は，数値，変数名，演算子を除き，同じ形式に従う．すなわち，各複雑度における解答 $\{v_i\}$ は，全く同じ手順で導出できる．この制約の元，式中に現れる変数名，数値，演算子を無作為に変化させ，各複雑度ごとに重複のない事例からなるデータセットを作成する．複雑度ごとに，合計 12,000 件の事例を作成し，そのうち 10,000 件をプローブの学習データ，2,000 件をテストセットとして使用する．また，§ 4 の実験で利用するテストセットについても 2,000 件である．加えて，プローブが解答を単に丸暗記することを防ぐため，学習データセットとテストセットに含まれる事例には算術式に重複がないようにする．[3)]

B　追加の実験結果

タスク正解率　表 3 に使用した算術推論タスクにおける各モデルのタスクそのものの正解率を示す．いずれの複雑度/モデルも 97%以上の高い正解率であった．

3)　重複は式ごとに判定する．例えば，学習データセットに `1+2` が出現する場合，テストセットの中には `1+2` は出現しない．

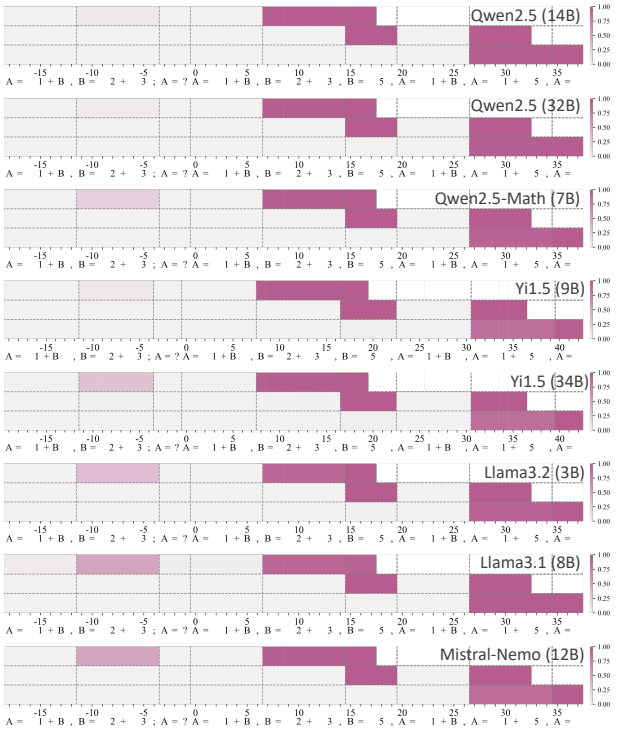

図 5　因果介入実験の結果．各グリッドはその位置に介入を行った際の介入成功率を示している．図 4 と同様に，各ヒートマップは上から，分析対象トークンを B =**5**$_2$, A = 1+**5**$_4$, A =**6**$_5$ とした時の最大介入成功率である．

プロービング　また，表 4 に 8 種類の言語モデルに対するプロービング結果を示す．いずれのモデルでも $t^*_{\text{eq}} > 0$ であり，CoT の間に途中の最終解答/途中結果がモデル内部で導出されている可能性が示された．よって，異なる言語モデルであっても (少なくとも今回の実験範囲では) CoT に対して最終解答は忠実であると考えられる．

因果介入　図 5 に Qwen2.5-7B 以外の 8 つの言語モデルに対する実験結果を示す．§ 4 でも述べたように，Qwen 系列のモデルにおいて特に入力部分に対する介入の成功率が低く，CoT に対して最終解答/途中の計算結果が忠実であることが示唆された，一方で，特に Llama, Mistral-Nemo などの他の種類のモデルでは B =**5**$_2$ と入力部分との間に弱い因果関係が観察された．

C　関連研究

いくつかの既存研究では類似の問いを調査している [15, 16, 17, 18]．たとえば，既存研究 [17] では，二値分類タスクにおいて CoT による説明が後付け的なものであるかを調査している．因果介入実験を実施し，CoT$_z$ は入力 x の位置に対応する隠れ状態への介入によって制御可能であることを示している．その意味で，当該研究は $x \to z$ の因果関係に着目した研究でありスコープが異なる．加えて，我々の利用したタスクに類似した論理推論タスクでは，プローブは最終解答の予測に苦戦している結果も報告されており，我々の結果とも整合する．その他についても (実験設定は類似しているものの)，CoT が最終解答に忠実であるかという問いに着目したものではなく，我々の問いには完全には答えられていない．