網羅的統制実験による Transformer の加算能力獲得機序の解明

熊懐 祐太
東京大学
ykumadak@tkl.iis.u-tokyo.ac.jp

吉永 直樹
東京大学 生産技術研究所
ynaga@iis.u-tokyo.ac.jp

概要

LLM の社会実装が進む一方で，算術演算のような厳密な論理を要するタスクへの不完全性が課題となっている．内部機序の観点による分析が進められているが，記号処理能力の獲得機序は明らかとなっていない．本研究は，整数加算問題に関する統制された合成データを用いた実験により，Transformer の加算能力獲得機序を幾何学的に検証した．結果，モデルの深さと特定の桁数構成が，埋め込み空間における螺旋構造の形成と Grokking の発生を決定づけることを解明した．これに基づき，埋め込みの螺旋構造を固定する新たな学習戦略を提案し，従来の全データ一括学習よりも効率的に高精度なモデルを獲得できることを確認した．

1 はじめに

大規模言語モデル（LLM）の適用範囲は拡大しているが，規則的な記号処理能力には本質的な不完全性が残る．Dziri ら [1] が指摘するように，Transformer は規則に基づく記号操作を統計的パターンで近似するため，多段階の構成的推論において破綻しやすい．例えば算術演算については，演算過程における入出力の桁数増大に伴う精度低下 [2] や，小学校レベルの算数文章題ベンチマーク GSM8K [3] での誤答が報告されており，モデルが演算アルゴリズムを真に習得できていない証左となっている．

こうした課題に対し，学習済みの大規模言語モデルを対象に，算術演算を担う推論回路の特定等の内部機序の分析が行われている．一方で，言語モデルが算術能力をどのように獲得するかについては，過学習後に汎化性能が突如発現する Grokking 現象が算術能力にも認められるとの報告がある [4]．しかし，学習データが不明な学習済みモデルを対象とした分析では，学習データの特性とモデル特性との区別が困難である．また言語データが混在したデータからの学習においては純粋な算術能力獲得過程の解析が困難であり，制御性の欠如が課題である．

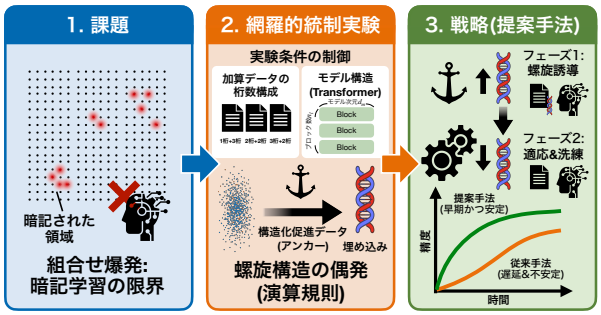

図 1: Transformer の加算能力獲得機序の分析と強化．

そこで本研究では，数値に対する適切な埋め込み表現の獲得が汎用的な記号操作の習得に不可欠であるとの仮説を立て，整数加算演算に焦点を当て明示的に制御されたベンチマークを用いて一から Transformer デコーダを学習し，その内部表現を追跡する．このような方法論を用いて，Transformer の加算能力に関する以下の問い（RQ）に答える．

RQ1: 加算演算能力の獲得過程において，Grokking が生じた学習では埋め込み空間にはどのような幾何学的構造が形成されるか。また，その形成度は学習データの構成にどう依存するか？

RQ2: 幾何学的構造の形成を人為的に誘導し，その構造を基礎とする学習戦略は学習の効率化と最終的な精度向上に寄与するか？

実験では，2 項の複数桁加算データセットを用い，4 層から 16 層の Transformer で層数や次元を変更しながら演算能力獲得過程における精度推移と内部構造の洗練度を詳細に追跡した．実験の結果，演算能力の獲得にはモデルの深さが不可欠であり，同一パラメタ数では多層構成が有利であることを確認した．また，3 桁+2 桁等の不均衡データが周期 $T = 10$ の鮮明な螺旋構造を誘発する一方，2 桁+2 桁等は不規則な構造へ収束することを解明した．本知見に基づき，螺旋構造を固定する新たな学習戦略を提案し，その有効性を実証する．

表 1: 加算タスクのデータセット構成（Op1 + Op2）.

Op1 \ Op2	1 桁	2 桁	3 桁	合計
1 桁	100	900	9,000	10,000
2 桁	900	8,100	81,000	90,000
3 桁	9,000	81,000	810,000	900,000
合計	10,000	90,000	900,000	1,000,000

2 加算演算能力の獲得機序分析手法

Transformer の算術演算能力の獲得機を解明するには，学習プロセスを厳密に制御し内部表現の変容を動的に追跡する必要がある．本研究では，加算演算に絞って難度が制御された加算演算ベンチマークを作成し，モデル構成（深さ・次元数）を操作した Transformer を一から学習させる．この過程で幾何学的構造の形成条件を特定することを目指す．

解析にあたっては数値的論理のみを純粋に抽出可能なデータセットが不可欠である．GSM8K [3] や MATH [5] 等の算術演算ベンチマークは自然言語の理解に加えてトークナイザによる数値の断片化が生じ，算術能力の獲得機序を厳密に分析するには適さない．そこで本研究では各数字のトークンを事前定義した統制加算データセット[1)]を構築した．データ形式は “123 + 45 = 168” のような単純な加算問題とし，1 桁から 3 桁までの全加算パターン（計 100 万問）を網羅することで桁数構成が学習機序に与える影響を精細に評価可能とした（表 1）.

このデータセットを元に，モデルパラメタを制御した Transformer を学習して学習過程のモデルを分析する．幾何学的解析として，数値トークンの埋め込みベクトルを一般化螺旋モデル [6] で近似し，その決定係数 R^2 を算出する．これにより，モデルが学習のどの段階で 10 進法のモジュラ性（螺旋構造）を獲得し，それが演算における記号的手続き能力の獲得といかに同期するかを定量的に評価する．

3 分析実験

本節では，Transformer モデルが算術演算能力を獲得する際の構造的条件や学習ダイナミクスおよび内部表現の幾何学的変容について分析する．

表 2: 加算能力獲得の実験結果: d_m, n_l, nd+nd はモデル次元数，層数，n 桁+n 桁の整数加算を示す.

モデル設定	桁別正答率 (入力 A + 入力 B)				
(d_m, n_l)	1 桁+1 桁	2 桁+2 桁	3 桁+3 桁	Avg.	(MSE)
Low Capacity / Shallow					
(32, 4)	0.00	7.04	8.21	7.85	(141.79)
(32, 8)	0.00	6.79	10.96	10.25	(120.84)
Comparable Capacity (Width vs Depth)					
(64, 4)	54.55	73.70	97.70	95.63	(2.03)
(32, 16)	**90.91**	**100.00**	**99.99**	**99.93**	**(0.26)**
High Capacity					
(64, 8)	90.91	100.0	99.99	99.99	(0.089)
(64, 16)	90.91	99.75	100.00	99.99	(0.10)

3.1 数理推論能力の性能に関する検証

モデル/学習設計: 学習に用いる Transformer の実装としては，PyTorch ライブラリの `torch.nn` クラスから `TransformerEncoderLayer` を使用し，学習および推論時には Attention ブロックのトークンの先読みによる推論を禁止することで，自己回帰的なデコーダーモデルとして動作させた．これにより，LLM で用いられる Transformer デコーダ[2)]と機構を同じにすることが可能である．モデルの埋め込み次元 $d_m \in \{32, 64\}$，FFN 次元 $d_{\text{ff}} \in \{128, 256\}$（通常 $4 \times d_m$），Transformer 層数 $n_l \in \{4, 8, 16\}$，ヘッド数 $h = 4$ を組み合わせて実験を行い，バッチサイズは 512，エポック数は 1000 を上限に設定し，評価損失の 50 エポック平均が $1e^{-5}$ を超えた場合は早期終了を行った．なお，学習・評価・テストセットは各桁数の構成ごとにそれぞれ 80%，10%，10%の割合でランダムに選択した.

モデル構造と演算能力の相関関係: 加算能力獲得の構造的要件を特定するため，同容量条件下で深さ（n_l）と次元（d_m）の影響を比較した（表 2）．分析の結果，以下の知見を得た.

深さの支配的寄与: 総パラメタ量（計算容量）が近似する (64, 4) と (32, 16) の比較において，後者の深層・狭幅構成が圧倒的に高い精度を達成した．これは，加算のような逐次的なステップを要するアルゴリズムの獲得において，一層あたりの次元数を広げるよりも，計算ステップを垂直方向に積層する深さを確保する方が構造的に適合していることを示唆している．つまり，Transformer における深さは多段階

1) 数値データの表現には 1 から 99 までの整数値をそれぞれ単一のトークンとして定義した上で，上位の桁から順に最大長となるよう切り出す **L2R（Left-to-Right）貪欲トークナイズ**を採用した．例えば，“123” は “12” と “3” の 2 トークンに分割される．この方式は，LLM が自然言語を処理する際の標準的なトークナイズ順序と整合するものである.

2) `TransformerDecoderLayer` はクロスアテンション機構 [7] を含み，既存の LLM の構成と異なるため使用しなかった.

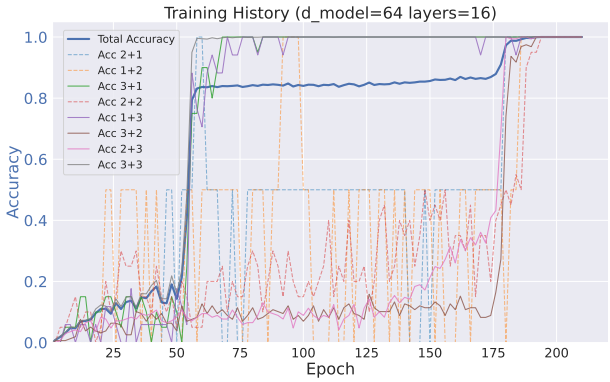

図 2: $d_m = 64, L = 16$ における桁別テスト精度推移

表 3: 桁数構成別 Grokking 発生特性の比較

構成	強度	特徴
1 桁+1 桁/2 桁+1 桁	低	学習事例不足により汎化困難
2 桁+2 桁	低	漸次的・緩慢な向上
3 桁+2 桁/2 桁+3 桁	強	明瞭かつ急峻な精度上昇
3 桁+3 桁	中	学習事例増加による漸次的向上
1 桁+3 桁/3 桁+1 桁	強	構造的変容を伴う急上昇
不均衡混合 (1v3 & 2v3)	**強**	**幾何学的構造の強力な誘導**

の推論回路を効率的に構成するために不可欠であると考えられる．

回路形成の閾値： $d_m = 32$ では $n_l = 8$ まで学習が停滞し，$n_l = 16$ で突如 Grokking が発生した．これは推論回路の形成に物理的な深さの閾値が存在することを示唆し，パラメタ数のみでは捉えきれない構造的要件を裏付けている．

桁別正答率と Grokking 現象: 予備実験の結果，R2L 方式では精度の漸増に留まるが，L2R 方式では顕著な Grokking が観測された．これは Transformer が上位桁からの情報を処理する過程で高度な内部構造を自己組織化している可能性を示唆する．

図 2 の通り，特定エポックを境に精度が段階的に遷移する Grokking 現象が観測された．3 桁+1 桁/3d 桁の習得を端緒に不均衡な組み合わせへと波及しており，特に不均衡な桁数データが内部表現を暗記から汎化へと相転移させる強力なトリガーであることが示唆された（表 3）．

3.2 螺旋モデルに基づく幾何学的解析

内部表現の変容を解明するため，Kantamneni ら [6] の手法に基づき，数値トークンの埋め込みを螺旋モデルで解析した．埋め込み $\mathbf{h}_a$ を周期 $T = 10$ を持つ基底へと射影し，決定係数 R^2 により構造の形成度を定量化した（詳細な数式は付録 A）．

表 4: 学習データ構成別の螺旋フィッティング率（R^2）と重要度の比較

学習データ構成	決定係数 R^2	重要度 ($T = 10$)
1 桁 + 1 桁（最小構成）	.031	0.52
2 桁 + 2 桁（同桁構成）	.127	0.47
3 桁 + 2 桁（単一不均衡）	.274	1.44
全組み合わせ	.261	0.91
不均衡混合 (1v3 & 2v3)	**.325**	**1.65**

3.2.1 埋め込み空間における螺旋構造の形成

学習データの構成が幾何学的構造に与える影響を評価するため，周期 $T = 10$ のスペクトル重要度（PSD）と決定係数 R^2 を算出した（表 4）．

分析の結果，同桁同士の学習（1d+1d, 2d+2d）では R^2 が極めて低く，明瞭な構造が形成されないことが判明した．これは，データセットが単純な場合，モデルが大域的な規則の獲得ではなく，個別の計算結果の暗記（Lookup Table 的な保持）によって損失を低減できるためであると考えられる．

一方，不均衡な桁数ペアを混合したデータでは，R^2（.325）および重要度（1.65）共に最高値を記録した．これは，多様な繰り上がりパターンを含む不均衡データが，モデルに対して十進法の体系的な周期構造（螺旋多様体）を共通の数理的基底として獲得することを強制した結果であると解釈できる．この幾何学的アンカーとしての性質は，効果的な学習戦略の設計において極めて重要な示唆を与える．

4 Grokking 誘導型学習戦略

前節の解析により，算術能力の獲得と埋め込み空間における螺旋構造の形成との関連性が示された．本節では，この知見を基に学習順序を制御することで効率的に高精度な演算能力を獲得させる Grokking 誘導型学習戦略を提案し，その効果を検証する．

4.1 学習手順と検証方法

本戦略は以下の 2 フェーズで構成される．フェーズ 2 において埋め込み層の更新（Unfrozen）を許可するか否かが，最終的な幾何学的洗練度と精度に与える影響を検証した．

フェーズ 1：構造形成 Grokking を強く誘発する特定のデータサブセット（不均衡混合データ）のみを用いて初期学習を行う．これにより，モデル内部に演算の基盤となる幾何学的多様体を早期に自己組織化させる．

表 5: 学習戦略による最終性能と螺旋構造 ($T = 10$) への適合率の比較 ($d_m = 64, L = 4$).

学習戦略	決定係数 R^2	重要度	平均精度 (%)
従来法 (一括学習)	.261	0.91	95.63
提案法 A (誘導 + 固定)	.325	1.65	98.42
提案法 B (誘導 + 非固定)	**.372**	**1.74**	**99.97**

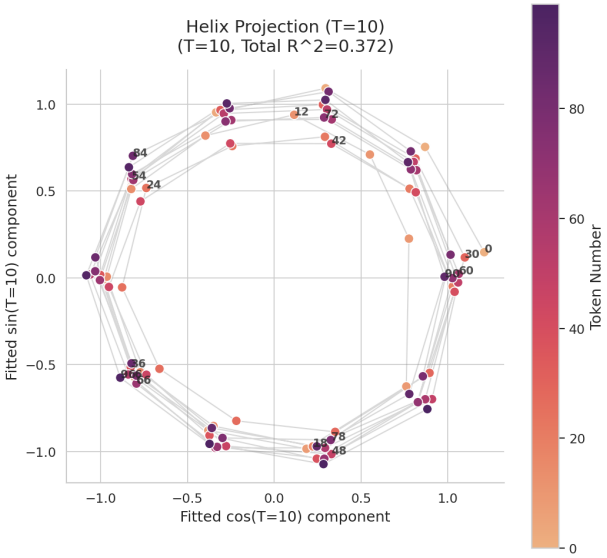

図 3: 提案手法 B における埋め込み空間の螺旋射影 ($T = 10$). 決定係数 $R^2 = 0.372$ に達しており，数値トークンが 10 進法の周期性に基づき理想的な円環構造を形成していることが視覚的にも確認できる．

フェーズ 2：全データ学習　形成された構造に基づき，全データセットを用いた学習へ移行する．

固定:　埋め込み層を固定し，幾何学的構造を保ったまま後続層を適応させる．

非固定:　埋め込み層を固定せず，幾何学的構造の微調整を許容する．

4.2　実験結果

表 5 に，フェーズ 2 終了時における各指標の比較を示す．分析の結果，フェーズ 2 において埋め込みを非固定とした手法が，螺旋構造へのフィッティング率（R^2）および演算性能の双方で最高値を記録した．これは，初期フェーズで誘導された螺旋構造が，その後の全データ適合過程において破壊されるのではなく，むしろ幾何学的にさらに洗練されたことを示している．

図 3 に示す通り，数値トークンは 10 を周期とする極めて鮮明な螺旋構造を形成している．この幾何学的一貫性は，3 桁計算における高い汎化性能を支える本質的な幾何学的基盤であり，モデルが個別の暗記を排して，体系的な演算規則を記述可能な内部表現を構築したことを強く示唆している．

表 6: 桁別正答率比較 (%) ($d_m = 64, L = 4$)

	従来法			**提案法 B**		
A \B	1d	2d	3d	1d	2d	3d
1d	54.55	45.98	85.16	0.00	23.44	**99.44**
2d	66.67	73.70	91.01	2.44	**88.61**	**99.89**
3d	84.48	85.08	97.70	**99.89**	**99.83**	**100.00**

4.3　桁別正答率の詳細比較と機序の考察

表 6 に提案法 B とベースラインとの桁組合せ別の正答率を示す．特筆すべきは，提案手法が 3 桁の組み合わせにおいて **100%** という完璧な汎化を達成した点である．一方，1d+1d において精度が **0.00%** を記録する等，低桁数領域で特異な性能低下が確認された．これは，特定の幾何学的アンカーを先行学習させたことにより，モデルが個別の数値を暗記する局所的な解を完全に放棄し，十進法の体系的演算規則を司る大域的な螺旋多様体へとパラメタを高度に専門化させた結果であると解釈できる．この挙動は，暗記から汎化へのトレードオフを劇的に反映したものであり，本提案戦略が数理的本質の獲得において強力な手法となることを示唆している．

5　おわりに

本研究では，統制された合成データセットを用いた実験により，Transformer の加算能力獲得にはモデルの深さと不均衡な桁数構成が不可欠であることを解明した．また，幾何学的解析を通じて，演算能力の獲得が埋め込み空間における螺旋構造の形成と同期していることを特定し，これを意図的に誘導する Grokking 誘導型学習戦略を提案した．実験では，本手法が従来の学習法では到達困難であった高桁数計算における完全な汎化を実現することを確認した．今後は，本知見を乗算や多段階推論タスクへ拡張し，LLM における数理的論理獲得の一般原理の究明を目指す．

参考文献

[1] Nouha Dziri, Ximing Lu, Peter West, Chandra Bhagavatula, Ari Holtzman, and Yejin Choi. Faith and fate: Limits of transformers on compositionality. In **Advances in Neural Information Processing Systems**, Vol. 36, 2023.

[2] Cem Anil, Yuhuai Wu, Anders Andreassen, Aitor Lewkowycz, Vedant Misra, Vinay Ramasesh, Ambrose Bluemke, Behnam Behnam, Kelly d'Hoffschmidt, and Chin Macua. Exploring length generalization in large language models. **arXiv preprint arXiv:2207.07828**, 2022.

[3] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. **arXiv preprint arXiv:2110.14168**, 2021.

[4] Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. Grokking: Generalization beyond overfitting on small algorithmic datasets. **arXiv preprint arXiv:2201.02177**, 2022.

[5] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. In J. Vanschoren and S. Yeung, editors, **Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks**, Vol. 1, 2021.

[6] Subhash Kantamneni and Max Tegmark. Language models use trigonometry to do addition, 2025.

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, **Advances in Neural Information Processing Systems**, Vol. 30. Curran Associates, Inc., 2017.

A　一般化螺旋モデルの数学的詳細

数値トークン $a \in \{0, \ldots, 99\}$ の d 次元埋め込みベクトル $\mathbf{h}_a \in \mathbb{R}^d$ に対し，以下の基底ベクトル $\mathbf{b}(a) \in \mathbb{R}^{2k+1}$ を構成する:

$$\mathbf{b}(a) = [a, \phi_1(a), \psi_1(a), \ldots, \phi_k(a), \psi_k(a)]^\top \quad (1)$$

ここで，$\phi_i(a) = \sin(2\pi a/T_i)$，$\psi_i(a) = \cos(2\pi a/T_i)$ である．ここで，$\mathcal{T}$ は埋め込みベクトルの周期を表し，$\mathcal{T} = 10$ は十進法の構造を反映している．

埋め込み行列 $\mathbf{H} = [\mathbf{h}_0, \ldots, \mathbf{h}_{99}]^\top$ および基底行列 $\mathbf{B} = [\mathbf{b}(0), \ldots, \mathbf{b}(99)]^\top$ を用いる．中心化された埋め込み行列 $\bar{\mathbf{H}}$ に対し，螺旋基底への変換行列 $\mathbf{C} \in \mathbb{R}^{(2k+1)\times d}$ は，以下のフロベニウスノルムを最小化する最小二乗法により推定される ：

$$\hat{\mathbf{C}} = \arg\min_{\mathbf{C}} \|\bar{\mathbf{H}} - \mathbf{B}\mathbf{C}\|_F^2 \quad (2)$$

モデルの適合度（螺旋構造の形成度）は，以下の決定係数 R^2 によって定量化される ：

$$R^2 = 1 - \frac{\sum_a \|\mathbf{h}_a - \hat{\mathbf{C}}^\top \mathbf{b}(a)\|^2}{\sum_a \|\mathbf{h}_a - \boldsymbol{\mu}\|^2} \quad (3)$$

ここで，$\boldsymbol{\mu} = \frac{1}{100}\sum_a \mathbf{h}_a$ は埋め込みベクトルの平均である．

B　実装および実験設定の詳細

本研究におけるモデルの実装および学習設定の詳細を述べる．

B.1　計算環境

実験には NVIDIA RTX A6000 GPU を 4 基搭載した計算サーバを使用し，PyTorch の Distributed Data Parallel (DDP) により分散学習を実施した．

B.2　モデル構成とハイパーパラメタ

モデルは Transformer Encoder をベースとし，学習および推論時には `generate_causal_mask` により下三角行列のマスクを適用することで，自己回帰的なデコーダーモデルとして動作させた．具体的なハイパーパラメタの設定を表 7 に示す．

B.3　語彙およびトークナイズ

数値データのトークナイズには，上位の桁から順に切り出す L2R（Left-to-Right）方式を採用し，1 トークンあたり最大 2 桁（0 から 99）を許容した．語彙（Vocabulary）の構成は以下の通りである．

表 7: モデル構成および学習ハイパーパラメタの設定

カテゴリ / 項目	設定値
モデル構成	
層数 (`num_layers`)	4
モデル次元 (`d_model`)	64
ヘッド数 (`nhead`)	4
FFN 次元 (`dim_feedforward`)	256
最大シーケンス長 (`max_len`)	20
ドロップアウト率	0.1
活性化関数	GeLU
位置エンコーディング	Sinusoidal
学習設定	
オプティマイザ	AdamW
学習率 (`learning_rate`)	1.0×10^{-4}
GPU あたりバッチサイズ	512
総バッチサイズ (4 GPU)	2048
最大エポック数	1000
損失関数	Cross Entropy (`<PAD>` を無視)
勾配クリッピング	1.0

- **数値トークン：** `0`, `1`, `2`, `...`, `99`
- **演算記号・記号：** `+`, `=`
- **特殊トークン：** `<PAD>`, `<BOS>`, `<EOS>`