Transformer 事前学習における最終層隠れ状態ジャンプの抑制

柴田 圭悟 [1] 矢野 一樹 [1] 高橋 良允 [1,2] 李 宰成 [1] 池田 航 [1] 鈴木 潤 [1,2,3]
[1] 東北大学 [2] 理化学研究所 [3] 国立情報学研究所 LLMC
shibata.keigo.p1@dc.tohoku.ac.jp is-failab-research@grp.tohoku.ac.jp

概要

本論文では，Transformer に基づく言語モデルの内部挙動を分析する．近年の多くの事前学習済みモデルでは，入力・出力隠れ状態間の角度距離の変化が中間層では小さい一方で，最終層付近では極端に大きくなる現象が観測されている．本研究では，この現象を「ジャンプ」と定義する．また，ジャンプの大きさを定量化する指標を導入し，多くの公開モデルにおける普遍性と，事前学習中にこの現象が増幅されることを示す．次に，事前学習時にジャンプを抑制する正則化手法（JREG）を提案する．Llama 系の 3 種類のモデルサイズでモデルを学習し評価した結果，モデル構造を変更することなく，ベースラインより高いタスク性能が得られた．

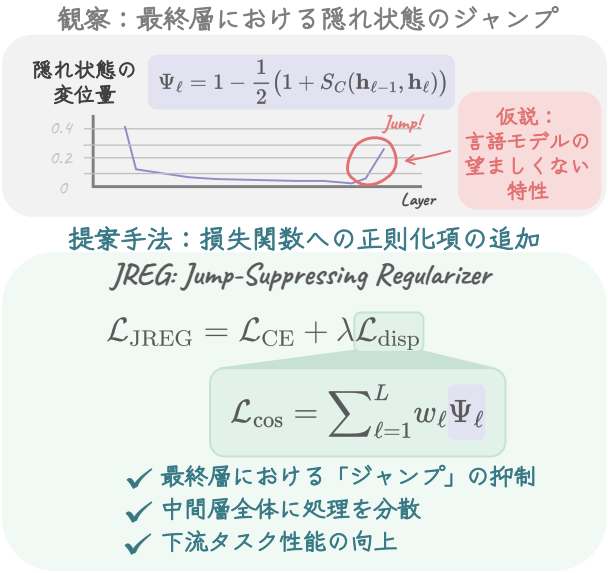


図 1 多くの公開言語モデルでは，中間層における隠れ状態の変化は小さい一方で，最終層において顕著なジャンプが観測される．このジャンプを事前学習時に抑制することで，中間層の能力活用を促し，性能向上が期待できる．

1 はじめに

Transformer に基づく言語モデル（Transformer-LMs）は，幅広い人工知能（AI）タスクにおいて卓越した性能を示している [1, 2]．この成功を背景として，これらのモデルが多様な指示に対して適切かつ流暢な応答を生成できる内部メカニズムを解明しようとする研究が活発化している [3, 4]．その結果，Transformer-LMs の内部挙動を解釈する研究は，近年の AI 研究における重要な分野の一つとなっている [5, 6]．

Transformer-LMs のモデル構造に関しては，事前学習済み言語モデルに冗長あるいは非効率なパラメータが多数存在することが報告されている一方 [7, 8, 9]，そうした冗長性が効果的な事前学習に寄与しているとする研究も存在する [10, 11, 12]．特に，Llama 系の事前学習済みモデルでは，中間層に冗長かつ非効率な Transformer 層が多く存在することが指摘されている [9]．これは，中間層が入力とほぼ同一方向の隠れ状態ベクトルを出力しており，角度距離の変化が極めて小さいことを意味する．一方で，最終層付近では角度距離が急激に変化する現象が観測されている [9, 7]．

本論文では，中間層における隠れ状態ベクトルの角度変化が小さい一方で，最終層付近において大きな角度変化が観測されるという層間の非対称的な挙動に着目する．このような層間の不均衡は，モデル全体の表現能力を制限し，パラメータの冗長性を増大させる可能性がある．我々は，この現象を，モデルが中間層を十分に活用せず，最終層に過度に依存している結果として生じていると仮説を立て，これを実験的に検証する．図 1 に本研究の概要を示す．まず，最終層付近におけるジャンプ強度を定量化する指標を導入する．次に，多くの公開されている事前学習済みモデルにおいてジャンプが広く観測され，学習の進行とともに顕著になることを示す．さらに，我々はこのジャンプを望ましくない性質と捉え，事前学習時に最終層周辺のジャンプを強く抑制する正則化手法 **Jump-Suppressing Regularizer**

表 1　6 つの公開モデルそれぞれの跳ね率.

Model	ζ_L	ζ_{L-1}	ζ_{L-2}
Llama-3.2-1B	20.6	22.7	22.7
Llama-3.2-70B	10.1	15.7	16.2
Gemma-2B	22.2	28.2	28.2
DeepSeek-67B	7.16	8.48	8.87

(JREG) を提案する．JREG は間接的に中間層の能力活用を促進する．実験では，Llama 系アーキテクチャ（170M，1B，3.4B パラメータ）に対し，(1) 標準的なクロスエントロピー損失のみを用いた場合と，(2) JREG を追加した場合を比較する．ジャンプ挙動および複数の下流タスク性能を評価した結果，JREG はモデル構造を変更することなく，ジャンプ挙動を抑制しつつ性能向上を実現できることが確認された．

2　隠れ状態の移動軌跡

本節では，隠れ状態ベクトルの移動量 Ψ と，隠れ状態ベクトルの変化量を表す跳ね率 ζ を定義し，4 つのモデルサイズ・モデル設定が異なる公開されている事前学習済みモデルで移動量，跳ね率をそれぞれ計測し，最終層で隠れ状態が大きく変化する現象「ジャンプ」が生じていることを示す．また，事前学習中のチェックポイントが公開されている二つのモデルで，学習中の移動量，跳ね率を計測し，学習中にジャンプが増強されることを示す．

2.1　移動量の定義

定義 1 (隠れ状態ベクトルの移動量)**.** 層 ℓ における隠れ状態ベクトルの移動量 Ψ_ℓ を次のように定義する．

$$\Psi_\ell = 1 - \frac{1}{2}(1 + S_C(\boldsymbol{h}_{\ell-1}, \boldsymbol{h}_\ell)). \tag{1}$$

ここで，$S_C = (\boldsymbol{h}_{\ell-1} \cdot \boldsymbol{h}_\ell)/(||\boldsymbol{h}_{\ell-1}||_2||\boldsymbol{h}_\ell||_2)$ と定義する．$\Psi_\ell \in [0, 1]$ であり，値の大きさが，隠れ状態の変化量を表す．

定義 2 (隠れ状態ベクトルの跳ね率)**.** 層 ℓ における隠れ状態ベクトルの跳ね率 ζ_ℓ を次のように定義する．

$$\zeta_\ell = \sum_{k=\ell}^{L} \max(0, \Psi_k - \Psi_{k-1}) \times 100 \tag{2}$$

$\zeta_\ell \geq 0$ で，値が大きければ，ℓ 以降の層で移動量 Ψ の大きな増加を意味する．ζ_ℓ の最小値 0 は，$k \in \{\ell, \ell+1, \cdots, L\}$ において，$\Psi_k \leq \Psi_{k+1}$ になる場合に成立する．

2.2　事前学習済みモデルの挙動

本節では，§ 2.1 で定義した各層隠れ状態ベクトルの移動量 Ψ_ℓ，跳ね率 ζ_ℓ を 4 つのモデル，Llama3.2 1B, 70B [13], Gemma-2B [14], DeepSeek-67B [15] で計測して，全てのモデルでジャンプが生じていることを実験的に示す[1)]．図 2 に，各モデルで計測した隠れ状態ベクトルの移動量 Ψ_ℓ を，表 1 に，跳ね率 ζ_L, ζ_{L-1}, ζ_{L-2} を示す．LAMBADA データセット [16] を推論時に使用した．結果から，全ての公開されているモデルで最終層で隠れ状態ベクトルが大きく変化する現象が確認できる．

2.3　学習の進行に伴う挙動の分析

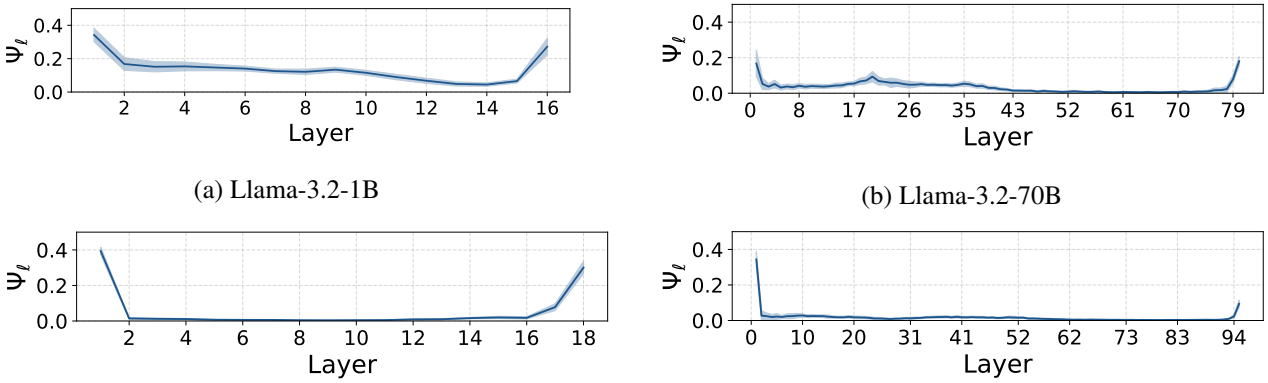
学習中のチェックポイントが利用可能なオープンモデルである Pythia, OLMo1 の 20%, 40%, 60%, 80%, 100%地点における隠れ状態の移動量から，学習進行とジャンプの関係を示す．各チェックポイントにおける移動量 Ψ を 図 4 に示す．Pythia 1B の跳ね率 ζ_L は，学習の進行に伴って，1.03, 2.79, 5.37, 10.4, 11.2 と単調増加しており，この傾向は他の全てのモデルでも確認された．つまり，ジャンプは学習の進行に伴って大きくなる．

3　提案手法

§ 2.2 において，隠れ状態ベクトルは角度距離の意味で，中間層では小さく，最終層で極端に大きな変化，ジャンプをすることが実験的に示された．先行研究では，(1) このような角度距離の変化が小さい層は削除して推論させてもモデルの性能に影響しない [9, 8], (2) 中間層における入出力隠れ状態間の角度距離の変化が大きくなるようなアーキテクチャの変更を施すことで，モデルの性能の向上が可能であることが示されている [17, 18]．この結果は，中間層に冗長性と中間層における隠れ状態の角度距離変化が小さいこととの関係性が示唆される．

本論文では，隠れ状態が中間層で変化せず，最終層で極端に大きく変化する現象は、モデルが中間層を十分に活用せず，最終層に過度に依存している結果として生じていると仮説を立てる．これを，隠れ状態ベクトルの移動距離に対する正則化を追加して事前学習を行い，モデルの性能を評価することで検証する．$\mathcal{L}_{\text{CE}}$ を交差エントロピーで定義される損失関数とする．この交差エントロピーは出力の性能の

1)　他の事前学習済みモデルの移動量は § C.



(a) Llama-3.2-1B　(b) Llama-3.2-70B

(c) Gemma-2B　(d) DeepSeek-67B

図 2　LAMBADA データセットから抽出した 100 サンプルに対する次単語予測における，各層ごとの隠れ状態の変位 Ψ_ℓ．すべてのモデルアーキテクチャに共通して，最終層における変位は中間層のそれよりも大きくなる傾向がある．

みを考慮しており、内部状態を最適化するものではない．そこで本論文では，$\mathcal{L}_{\text{CE}}$ に加えて，最終層における移動量を抑制する損失関数 $\mathcal{L}_{\text{disp}}$ を導入する．

$$\mathcal{L}_{\text{disp}} = \sum_{\ell=1}^{L} w_\ell \Psi_\ell, \tag{3}$$

ここで w_ℓ は，$\boldsymbol{w} = \texttt{softmax}(\alpha \boldsymbol{l})$ の ℓ 番目の要素で，$\boldsymbol{l} = (1, 2, 3, \cdots, L)$ である．また，α はハイパーパラメータで，α が大きくなると，後ろの層に大きなペナルティがかかる．$\alpha = 0$ で全ての層に均等に正則化がかかり，$\alpha \to \infty$ で，最終層の重み w_L のみ 1 で他の層には正則化がかからない．提案手法の Jump-Suppressing Regularizer (JREG) の損失関数を次のように定義する．

$$\mathcal{L}_{\text{JREG}} = \mathcal{L}_{\text{CE}} + \lambda \mathcal{L}_{\text{disp}}, \tag{4}$$

λ はハイパーパラメータで，二つの損失関数 $\mathcal{L}_{\text{CE}}$, $\mathcal{L}_{\text{disp}}$ の相対的な重要度を決定する．

4　実験

4.1　事前学習

モデル設定・訓練設定　三つの異なるモデルサイズ（170M, 1B, 3.4B）の Llama ベースのモデルをスクラッチから事前学習を行う．JREG の最適なハイパーパラメータである α を決定するために，170M モデルで $\alpha \in \{0.0, 0.1, 0.3, 0.5, 1.0, 3.0\}$ で実験し，最も高い性能が出た α の値を 1B, 3.4B の設定に用いる．事前学習データには，FineWeb-Edu [19] から抽出した 100B トークンを用い，200K ステップで訓練を行う[2]．詳細なモデル設定，訓練設定は § A, § B に示す．

2)　同一トークン数で訓練した場合でも，訓練ステップ数が多いモデルの方が，より顕著なジャンプが見られる (§ D).

評価方法　Top-1 の次単語予測の精度を測る LAMBADA [16], QA タスクの BoolQ [21], ARC-easy(ARC-e) [22], HellaSwag [23], PIQA [24], RACE [25], SocialIQa [26], SciQ [27], SWAG [28] の予測精度の平均値で下流タスク性能を評価する．また，跳ね率 $\zeta_L, \zeta_{L-1}, \zeta_{L-2}$ の改善も評価する．

4.2　Supervised Fine-tuning (SFT)

訓練設定　§ 4.1 で事前学習した 3.4B モデルに対して updating all parameters で SFT を行う．学習データセットは Tulu-v1 SFT mixture[29] である．詳細な訓練設定は § B.2 に示す．

評価方法　GPT-4 を judge model として，三つのベンチマーク，MT-bench [30], Vicuna bench [30], WizardLM testset [31] で指示追従能力を測る．

5　結果

以降の結果は JREG のハイパーパラメータである λ は 1.0 に固定している．[3]

5.1　事前学習

表 2 に下流タスク性能を示す．170M モデルにおいて，全ての α の設定でベースラインの下流タスク平均性能である 47.5 を上回っている．最も高い平均性能を示したモデルは $\alpha = 1.0, 3.0$ の 48.5 である．1B モデルでは，170M モデルで最も高い性能であった $\alpha = 1.0, 3.0$ で事前学習を行う．170M モデルの場

オープンな事前学習済みモデル（例：Pythia は 14.3K ステップ [20]）に近い設定にするため，200K ステップとした．

3)　170M モデルで $\lambda \in \{1.0, 2.0, 3.0\}$ で実験を行い，最も性能が高かった $\lambda = 1.0$ を他のモデルサイズでも一貫して採用する．

表 2　下流タスクにおける性能評価結果．α は JREG のハイパーパラメータを表す．各行は各ベンチマークデータセットにおけるスコアを示しており，右端の列にはその平均値を示す．上段の表は，170M モデルを用いて $\mathcal{L}_{\text{CE}}$（ベースライン）および $\mathcal{L}_{\text{JREG}}$（提案手法）で学習したモデルの結果を示し，下段の表は 1B および 3.4B モデルで学習した結果を示す．

Size	Method	α	ARC-e	BoolQ	HellaSwag	LAMBADA	PIQA	RACE	SocialIQA	SciQ	SWAG	**avg**
170M	Baseline	-	54.9	57.5	32.1	32.0	64.0	29.1	38.4	80.2	39.7	47.5
	JREG (ours)	0.0	55.7	58.3	32.4	**32.5**	**65.9**	29.6	38.6	81.3	**40.2**	48.3
		0.1	56.1	60.5	32.2	29.8	65.6	29.6	39.2	80.9	39.9	48.2
		0.3	56.0	59.2	32.2	31.7	65.0	29.1	38.1	**82.2**	40.0	48.0
		0.5	57.0	57.1	**32.5**	30.7	65.0	29.3	38.6	**82.2**	40.0	48.0
		1.0	**57.2**	60.0	32.1	31.7	65.2	**29.9**	38.8	81.1	**40.2**	**48.5**
		3.0	**57.2**	60.0	32.4	31.6	65.1	28.9	**40.0**	81.7	39.9	**48.5**
1B	Baseline	-	68.5	**61.4**	42.9	46.6	**72.4**	32.9	41.0	89.5	47.3	55.8
	JREG (ours)	1.0	69.4	**61.4**	**43.1**	**47.2**	71.8	**35.3**	**42.1**	88.6	**47.6**	**56.2**
		3.0	**70.6**	59.3	42.6	45.6	71.8	34.9	41.4	**91.2**	47.3	56.1
3.4B	Baseline	-	75.6	59.4	49.7	55.9	**76.5**	36.3	**43.9**	**93.4**	51.3	60.2
	JREG (ours)	1.0	**77.9**	**61.6**	**50.3**	**57.2**	75.2	**37.8**	42.5	93.0	**51.5**	**60.8**

表 3　ベースラインモデルおよび JREG により学習したモデルについて，式 2 で定義したジャンプ率（$\zeta_L, \zeta_{L-1}, \zeta_{L-2}$）の比較．170M および 1B モデルにおける異なるハイパーパラメータ α の設定に対する結果を示す．JREG は最終層付近におけるジャンプ率を効果的に抑制している．

Model		α	ζ_L	ζ_{L-1}	ζ_{L-2}
170M	Baseline	-	7.24	7.63	7.63
	JREG (ours)	0.0	0.84	1.28	1.76
		0.1	0.48	0.72	1.05
		0.3	**0.00**	0.05	0.15
		0.5	**0.00**	**0.00**	**0.00**
		1.0	**0.00**	**0.00**	**0.00**
		3.0	**0.00**	**0.00**	1.55
1B	Baseline	-	5.42	5.66	5.66
	JREG (ours)	0.5	**0.00**	**0.00**	**0.00**
		1.0	**0.00**	**0.00**	**0.00**
		3.0	**0.00**	**0.00**	**0.00**
3.4B	Baseline	-	5.21	6.15	6.25
	JREG (ours)	1.0	**0.00**	**0.00**	**0.00**

合と同様に，全ての設定でベースラインモデルの平均性能 55.8 を上回っており，$\alpha = 1.0$ で最も高い性能 56.2 であった．モデルサイズが異なると，最適な α が異なることがわかる．3.4B モデルでも同様に，1B モデルで最高性能であった $\alpha = 1.0$ で事前学習を行い，ベースラインの平均性能である 60.2 を上回る 60.8 であった．

表 3 に跳ね率 $\zeta_L, \zeta_{L-1}, \zeta_{L-2}$ を示す．全てのモデルサイズでベースラインの跳ね率は 0 より大きな値であり，§ 2.2 で示した公開されている事前学習済みモデルと同じ傾向である．一方で JREG で事前学習したモデルでは，全ての α の設定で跳ね率はベースラインより小さく，170M モデルでは，$\alpha \geq 0.5$, 1B と 3.4B では全ての設定で跳ね率は 0.0 である．

表 4　3.4B のベースラインモデルおよび JREG（$\alpha = 1.0$）における各種ベンチマーク性能の結果．

Model	Vicuna Bench	WizardLM testset	MT Bench	avg
Baseline	5.89	4.10	2.84	4.28
JREG ($\alpha = 1.0$)	**6.36**	**4.23**	**3.40**	**4.67**

5.2　SFT

表 4 に，3.4B モデルを SFT した後の三つのベンチマークにおける性能を示す．全てのベンチマークで，JREG がベースラインを上回り，平均性能はベースラインで 4.28，JREG は 4.67 であった．これは，JREG が事前学習時だけでなく，SFT 後のモデル性能にも良い影響を与えていることを示している．

6　おわりに

本研究では，Transformer-LMs において，最終層付近で入力と出力の隠れ状態ベクトル間の角度距離が大きく変化する「ジャンプ」を定義し，その影響を分析した．この挙動は，事前学習済みモデルおよびスクラッチから学習したモデルのいずれにおいても観測されることを確認した．さらに，事前学習時に最終層周辺の隠れ状態移動量を抑制する正則化手法 JREG を提案し，Llama アーキテクチャの複数のモデルサイズにおいて，ジャンプ挙動の低減，中間層の相対的寄与の増加，およびモデル構造を変更しない性能向上を一貫して達成できることを示した．

謝辞

本研究は，JST ムーンショット型研究開発事業 JPMJMS2011-35 (fundamental research)，文部科学省の補助事業「生成 AI モデルの透明性・信頼性の確保に向けた研究開発拠点形成」，および，国家戦略分野の若手研究者及び博士後期課程学生の育成事業（博士後期課程学生支援）JPMJBS2421 の助成を受けたものである．また，本研究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，および，産総研及び AIST Solutions が提供する ABCI 3.0 を「ABCI 3.0 開発加速利用」の支援を受けて利用した．

参考文献

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report, 2023.

[2] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and et al. The llama 3 herd of models, 2024.

[3] Curt Tigges, Michael Hanna, Qinan Yu, and Stella Biderman. LLM circuit analyses are consistent across training and scale. In **The Thirty-eighth Annual Conference on Neural Information Processing Systems**, 2024.

[4] Alessandro Stolfo, Ben Peng Wu, Wes Gurnee, Yonatan Belinkov, Xingyi Song, Mrinmaya Sachan, and Neel Nanda. Confidence regulation neurons in language models. In **The Thirty-eighth Annual Conference on Neural Information Processing Systems**, 2024.

[5] Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. Eliciting latent predictions from transformers with the tuned lens, 2023.

[6] Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. Analyzing feed-forward blocks in transformers through the lens of attention maps. In **The Twelfth International Conference on Learning Representations**, 2024.

[7] Georgy Tyukin, Gbetondji J-S Dovonon, Jean Kaddour, and Pasquale Minervini. Attention is all you need but you don't need all of it for inference of large language models, 2024.

[8] Xin Men, Mingyu Xu, Qingyu Zhang, Qianhao Yuan, Bingning Wang, Hongyu Lin, Yaojie Lu, Xianpei Han, and Weipeng Chen. ShortGPT: Layers in large language models are more redundant than you expect. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, **Findings of the Association for Computational Linguistics: ACL 2025**, pp. 20192–20204, Vienna, Austria, July 2025. Association for Computational Linguistics.

[9] Qi Sun, Marc Pickett, Aakash Kumar Nain, and Llion Jones. Transformer layers as painters. In Toby Walsh, Julie Shah, and Zico Kolter, editors, **AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA**, pp. 25219–25227. AAAI Press, 2025.

[10] Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, **Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)**, pp. 7319–7328, Online, August 2021. Association for Computational Linguistics.

[11] Bingqing Song, Boran Han, Shuai Zhang, Jie Ding, and Mingyi Hong. Unraveling the gradient descent dynamics of transformers. In **The Thirty-eighth Annual Conference on Neural Information Processing Systems**, 2024.

[12] Vedang Lad, Jin Hwa Lee, Wes Gurnee, and Max Tegmark. The remarkable robustness of llms: Stages of inference?, 2025.

[13] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and et.al. The llama 3 herd of models, 2024.

[14] Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, and et al. Gemma: Open models based on gemini research and technology, 2024.

[15] DeepSeek-AI, Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, and et al. Deepseek llm: Scaling open-source language models with longtermism, 2024.

[16] Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The LAMBADA dataset: Word prediction requiring a broad discourse context. In **Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers**. The Association for Computer Linguistics, 2016.

[17] Wenfang Sun, Xinyuan Song, Pengxiang Li, Lu Yin, Yefeng Zheng, and Shiwei Liu. The curse of depth in large language models, 2025.

[18] Pengxiang Li, Lu Yin, and Shiwei Liu. Mix-LN: Unleashing the power of deeper layers by combining pre-LN and post-LN. In **The Thirteenth International Conference on Learning Representations**, 2025.

[19] Guilherme Penedo, Hynek Kydlícek, Loubna Ben Allal, Anton Lozhkov, Margaret Mitchell, Colin A. Raffel, Leandro von Werra, and Thomas Wolf. The fineweb datasets: Decanting the web for the finest text data at scale. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, **Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024**, 2024.

[20] Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, and Edward Raff et.al. Pythia: A suite for analyzing large language models across training and scaling. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, **International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA**, Vol. 202 of **Proceedings of Machine Learning Research**, pp. 2397–2430. PMLR, 2023.

[21] Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, **Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)**, pp. 2924–2936. Association for Computational Linguistics, 2019.

[22] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge, 2018.

[23] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In Anna Korhonen, David R. Traum, and Lluís Màrquez, editors, **Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers**, pp. 4791–4800. Association for Computational Linguistics, 2019.

[24] Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: reasoning about physical commonsense in natural language. In **The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020**, pp. 7432–7439. AAAI Press, 2020.

[25] Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard H. Hovy. RACE: large-scale reading comprehension dataset from examinations. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel, editors, **Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017**, pp. 785–794. Association for Computational Linguistics, 2017.

[26] Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. Social iqa: Commonsense reasoning about social interactions. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, **Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019**, pp. 4462–4472. Association for Computational Linguistics, 2019.

[27] Johannes Welbl, Nelson F. Liu, and Matt Gardner. Crowdsourcing multiple choice science questions. In Leon Derczynski, Wei Xu, Alan Ritter, and Tim Baldwin, editors, **Proceedings of the 3rd Workshop on Noisy User-generated Text, NUT@EMNLP 2017, Copenhagen, Denmark, September 7, 2017**, pp. 94–106. Association for Computational Linguistics, 2017.

[28] Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. SWAG: A large-scale adversarial dataset for grounded commonsense inference. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, **Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018**, pp. 93–104. Association for Computational Linguistics, 2018.

[29] Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Raghavi Chandu, David Wadden, Kelsey MacMillan, Noah A. Smith, Iz Beltagy, and Hannaneh Hajishirzi. How far can camels go? exploring the state of instruction tuning on open resources. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, **Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023**, 2023.

[30] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, **Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023**, 2023.

[31] Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. Wizardlm: Empowering large pre-trained language models to follow complex instructions. In **The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024**. OpenReview.net, 2024.

[32] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, and Chenxu Lv et.al. Qwen3 technical report, 2025.

[33] Mostafa Elhoushi, Akshat Shrivastava, Diana Liskovich, Basil Hosmer, Bram Wasti, Liangzhen Lai, Anas Mahmoud, Bilge Acun, Saurabh Agarwal, Ahmed Roman, and et.al. Layerskip: Enabling early exit inference and self-speculative decoding. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, **Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024**, pp. 12622–12642. Association for Computational Linguistics, 2024.

A　モデル設定

表 5　モデル設定.

	170M	1B	3.4B
Layers	12	16	30
Model Dim	768	2048	3072
FFN Dim	2048	5376	8192
Attention Heads	12	16	24
Key / Value Heads	12	16	24
Activation		SwiGLU	
Vocabulary Size		32000	

B　訓練設定

B.1　事前学習

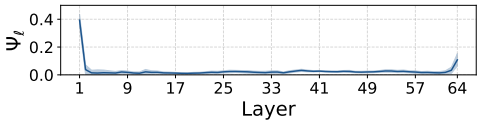
表 6　事前学習設定.

Configuration	170M	1B	3.4B
lr	9×10^{-4}	7×10^{-4}	5×10^{-4}
local batch	128	64	32
global batch		512	
sequence len		1024	
weight decay		0.1	
epsilon		1×10^{-8}	
Optimizer		AdamW ($\beta_1 = 0.9, \beta_2 = 0.95$)	
clip		1.0	
scheduler		cosine	
warmup		1000	
lr_min_ratio		0.1	
cycle_length		1.0	

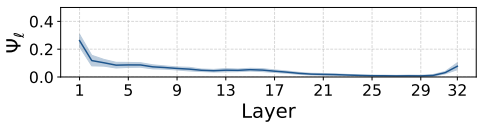
B.2　SFT

学習率の最大値は 2×10^{-5} とし，線形スケジューラを用いる．最適化手法には AdamW（$\beta_1 = 0.9$，$\beta_2 = 0.95$）を使用する．グローバルバッチサイズは 512 とし，総訓練ステップ数は 957 ステップである．

C　他の事前学習済みモデルの挙動



(a) Qwen3-32B [32]



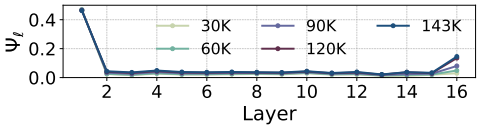
(b) layerskip-llama3.2-8B [33]

図 3　他の事前学習済みモデルの移動量 Ψ

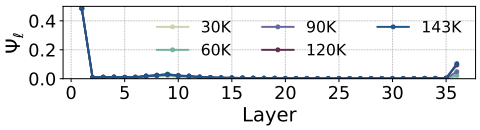
D　ステップ数とジャンプの関連性

本節では，訓練ステップ数がジャンプ挙動に与える影響を明らかにするために，チェックポイントを用いたオープンモデルの分析と，スクラッチ学習モデルにおいて訓練トークンを同一にした条件下での訓練ステップ数の影響の二点を実験的に調査する．

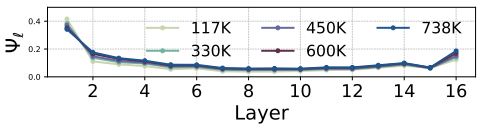
各チェックポイントにおける移動量 Ψ を 図 4 に示す．詳細な議論は § 2.3 を参照.



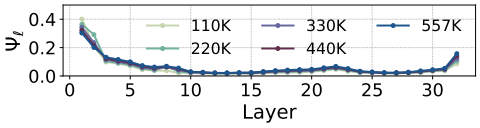
(a) Pythia 1B



(b) Pythia 12B



(c) OLMo 1B



(d) OLMo 7B

図 4　チェックポイントごとの隠れ状態移動量の解析.

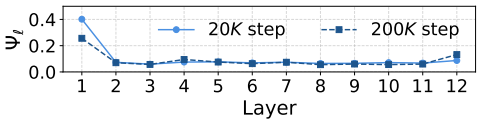


図 5　100B トークンで事前学習した 170M モデルにおける隠れ状態の移動量.（薄青：20K ステップ，濃青：200K ステップ）.

表 7　各チェックポイントにおける跳ね率.

Model	Size	20%	40%	60%	80%	100%
Pythia	1B	1.03	2.79	5.37	10.4	11.2
	12B	1.10	2.23	4.55	9.11	10.0
OLMo	1B	6.06	7.84	8.84	10.3	11.9
	7B	4.94	6.40	7.63	9.14	10.3

次に，総訓練ステップ数が与える影響について，170M パラメータの Llama ベースモデルを用いて，訓練ステップ数を 20K, 200K で訓練を行った二つのモデルで比較することで検証する．二つのモデルとも，学習トークン数は 100B であり，バッチサイズを変更することで総訓練ステップ数を変化させている．モデル設定，訓練設定（バッチサイズ以外）はそれぞれ § A，§ B と同一である．

図 5 に，20,000 ステップおよび 200,000 ステップで訓練したモデルの隠れ状態変位を示す．得られたジャンプレートは，20,000 ステップで $\zeta_L = 1.93$，200,000 ステップで $\zeta_L = 7.24$ であった．この結果から，同一のトークン予算の下でも，update step 数が多いほどジャンプレートが高くなり，隠れ状態変位の挙動がオープンな事前学習済みモデルにより近づくことが示される．