Long Context への対応に向けたマルチスケール状態空間モデル

吉山大護 [1]　佐々木裕 [1]
[1] 豊田工業大学 知能数理研究室
{sd24449,yutaka.sasaki}@toyota-ti-ac.jp

概要

現在の大規模言語モデルの多くに使用されている注意機構は入力の長さに対して 2 乗に比例する計算量を持つ．これを改善するための有望なアプローチの一つとして，状態空間モデルを基にした Mamba がある．しかしながら，Mamba をはじめとする状態空間モデルを基にしたアーキテクチャには，計算量の優位性を発揮できる長大な文脈に対する性能低下が著しいという欠点がある．それを補うための手法として圧縮と統合を伴うマルチスケール処理を行う TreeMamba を提案する．長大な文脈を特徴とする LongBench-E ベンチマークにおける 13 個のタスクのうち，いくつかのタスクにおいて TreeMamba による性能向上が確認できた．

1　はじめに

現在の大規模言語モデル（Large Language Model; LLM）の多くに組み込まれている注意機構 [1] には，計算量が入力の長さの 2 乗に比例して大きくなるという問題がある．実際には，スライディングウィンドウを用いた注意機構による計算量の削減や，並列処理による生成に必要な所要時間の短縮が行われているため，計算量の問題は重要視されにくい．しかしながら依然として LLM の計算量は大きいため，性能を維持したまま計算量を小さくする研究が盛んに行われている．

LLM の性能を維持したまま計算量を小さくするためのアプローチの一つとして，状態空間モデル（State Space Model; SSM）がある．注意機構では各トークンに対してそれ以前のトークンを利用した計算をしていたが，SSM では決められたサイズの行列を入力されたトークンの数だけ更新していく形で計算を進める．そのため，SSM の計算量は入力の長さに比例しており，長い入力の処理する場合注意機構よりも計算量を抑えやすい．特に Mamba [2, 3] という SSM を基にしたアーキテクチャは，言語モデルとして用いた場合，一部の評価ベンチマークにおいて Transformer[1] をはじめとする注意機構を含む言語モデルに匹敵する性能を示している [2, 3]．

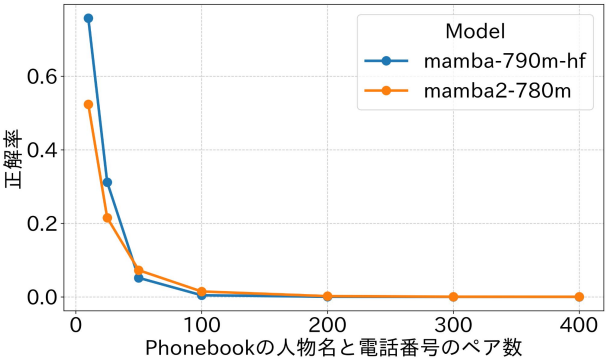

図 1　言語モデルの Mamba-790m-hf，Mamba2-780m，Pythia-410m を Phonebook ベンチマークで評価したときの正解率．

一方で，数千トークン以上の入力を必要とする長文処理ベンチマークでは，Mamba の性能が著しく損なわれ，Transformer に対する計算量の優位性を活かせないことが問題となっている．これは SSM の原理として，隠れ状態と呼ばれる固定の大きさの行列に入力の情報を保持するように更新していくが，入力が長すぎると隠れ状態の情報の保持に問題が発生してしまうためだと考えられている．実際に，Phonebook [4, 5] という入力に含まれる電話帳から特定の人物名を探して隣に書かれた電話番号を出力するといったベンチマークにおいて，電話帳の人物名と電話番号のペア数を変えながら正解率を測定したところ，図 1 のような結果となった．電話帳のペア数が 100 を超えると正解率はほぼ 0 になっていることが分かる．

Mamba の長い入力に対する性能を向上させるために，既存研究の LongMamba [6] では入力情報の圧縮というアプローチが行われている．これは，Mamba が長い入力に対して隠れ状態の崩壊を招くため，長い入力を避けて短くしてから処理すればよいという考えに基づいている．実際，圧縮による情

報量の減少はあるものの，いくつかのベンチマークで，そのままの長さで入力するよりも性能が向上することが確かめられている．

入力情報の圧縮に関するアプローチには，依然として研究の余地がある．例えば，圧縮していない入力と，圧縮した入力の両方を別々に処理し，それらの出力を適切に統合する手法が考えられる．異なる粒度の入力を処理した出力を組み合わせることでお互いの欠陥を補い合い，単一の入力を用いる従来手法よりも性能を向上させられる可能性がある．例えば圧縮した入力のみでは欠落してしまう局所的な情報を補完できるかもしれない．

本研究の貢献は，言語モデルとしての Mamba において圧縮していない入力と圧縮した入力の両方に対する出力を適切に統合することにより，Long Context に対する性能を向上させることを目指し TreeMamba という手法を提案した点と TreeMamba を長大な文脈を特徴とする LongBench-E [7] で評価した点である．

2 関連研究

2.1 状態空間モデル（SSM）

SSM は t 番目のトークン（時刻 t）に対応する入出力を $x_t, y_t \in \mathbb{R}^D$，隠れ状態を $h_t \in \mathbb{R}^{N\times D}$ とすると，$A_t \in \mathbb{R}^{N\times N}$, $B_t \in \mathbb{R}^N$, $C_t \in \mathbb{R}^N$, $D_t \in \mathbb{R}$ を用いて，

$$\begin{aligned} h_t &= A_t h_{t-1} + B_t x_t \\ y_t &= C_t h_t + D_t x_t \end{aligned} \tag{1}$$

のように隠れ状態 h_t の更新と出力 y_t の計算を行う．SSM は RNN と CNN の組み合わせと解釈することもでき，両者の利点を併せ持つことが知られている [8]．もともと状態空間モデルは連続時間の微分方程式で表現され，式 1 はそれを離散化したものである．そのため，時刻 t と $t-1$ の間隔を表す離散化ステップ Δ_t を導入し，状態空間モデルにおける離散化に従い，

$$\overline{A}_t = e^{\Delta_t A_t}, \quad \overline{B}_t \simeq \Delta_t B_t \tag{2}$$

のように定義される $\overline{A}_t$ や $\overline{B}_t$ を求め，式 1 における A_t, B_t として使用することがある．

2.2 Mamba

Mamba は，SSM を基にした言語モデルのアーキテクチャとして，計算量が入力の長さに対して線形でありながら，言語モデル用の複数の評価ベンチマークにおいて，Transformer に匹敵する高い性能を示したことにより非常に注目を集めている．Mamba の最大の特徴は，式 1 における C_t や式 2 における B_t, Δ_t を入力に応じて変化させるようにしたことにある．例えば，Δ_t が 0 に近いほど入力 x_t は隠れ状態に反映されず無視されることになり，逆に Δ_t の絶対値が大きいほど x_t の影響が強くなるという性質がある．したがって，従来の SSM を基にした言語モデルが機械的に隠れ状態を更新して情報を保持していたのに対し，Mamba は入力の重要度などを考慮した上で隠れ状態を更新して，より重要な情報のみを保持することが理論上は可能になった．

2.3 Mamba-2

Mamba の登場後，その理論的な背景と実装効率をさらに洗練させた改良版として，Mamba-2 [3] が提案された．Mamba-2 は Mamba の SSM の機構を構造化状態双対性（Structured State Space Duality; SSD）という観点から再構築することで，並列処理に最適化された形式で表現し，演算の効率化がなされた．この改良に伴い，Mamba-2 では，正規化の追加や SSM の状態次元数（式 1 の隠れ状態 h のサイズに関係する次元 N）を増加させ表現力を向上させるなどのハイパーパラメータに対する最適化も施されている．これにより，Mamba-2 は Mamba と比較して同等の性能をより少ないパラメータ数や計算資源で達成可能となっている．

2.4 LongMamba

LongMamba [6] は，事前学習済みモデルを用いて追加学習を行うことなく Mamba の推論方法を工夫し，長い入力に対する性能を向上させようとする手法である．LongMamba では，Mamba において各入力 x_t の影響度を決定する離散化ステップ Δ_t の大きさを重要度と考えて参照し，$\sum_{t=1}^{T} \Delta_t$ が学習時と同じになるように，文章の長さに応じて求められる閾値 $g(T)$ を設定して Δ_t の値を調節する．具体的には Δ_t が閾値 $g(T)$ を下回る場合は $\Delta_t = 0$ とし，x_t が隠れ状態の更新に影響を与えなくなる．また，$\sum_{t=1}^{T} \Delta_t$ については過去の情報に対する増幅・減衰度合と関係があるため，これを学習時の値とほぼ同じになるように調節することで，入力の長さによらず性能を発揮できるようにしている．これにより，Δ_t の値が小さい入力の影響を受けずに Δ_t の値が大きい重要な入力にのみ注目することができる．

2.5 Multi-Scale SSM

MS-SSM[9] は，文章だけではなく画像や音声などの入力も対象として，複数の粒度で捉えることを目的とした手法である．これは文章，画像の画素値，音声信号などの入力を解析しようとした際に，局所的な特徴から大局的な特徴までの異なる粒度で捉える必要があると考えられるためである．この手法は，明示的に Mamba が複数の粒度で処理ができるように，各入力 x_t にそれ以前の入力を足しこむ畳み込みを，多段階に分けて段々捉える粒度（範囲）が大きくなるように適用し複数の入力を作成する．そして，それによって得られた複数の入力を同じパラメータで構成された SSM に独立に入力し，重みづけ和でそれらを統合する．重みは複数の入力を作成する前の x_t を線形層に通すことで得るようになっている．しかしながら自然言語処理タスクでの評価は Long-Range Arena [10] のみと非常に限定的であるため，マルチスケール処理を行う SSM に関してさらなる検証が必要である．また，この手法は作成した入力の長さが全て同じであるため，計算量が数倍になってしまう点が問題である．

3 提案手法

本研究では，長いトークン列を間引いてより広い粒度で情報を捉えるための入力を作り出し，その入力から得られる出力を統合することにより，複数の粒度を扱うマルチスケール処理を行う TreeManba を提案する．TreeMamba は図 2 に示すように，通常の Mamba に，段階的な処理，各段階ごとに入力を間引く処理，それぞれの段階で得られた出力を統合する処理，の 3 つの機構からなるマルチスケール処理を追加したモデルである．圧縮は等間隔に 1 つのトークンのみを残しそのほかのトークンは間引くように行い，統合は要素ごとの和で行う．Mamba は，容量に限界のある隠れ状態に入力情報を取り入れ更新しながら保持する仕組みのため，位置的に最初の方にある入力ほど，次のトークンを予測するための確率分布への影響度が指数関数的に減衰するという特徴を持つ．基本的に，予測するトークンとの距離と近いため，位置的に後ろの方にあるトークンの影響度が高くなる傾向がある．しかしながら，長大な文脈の最初の方に重要な情報があった場合に，致命的な見落としが発生することにつながる．TreeMamba では，入力を間引くことにより離れたトークン同士を

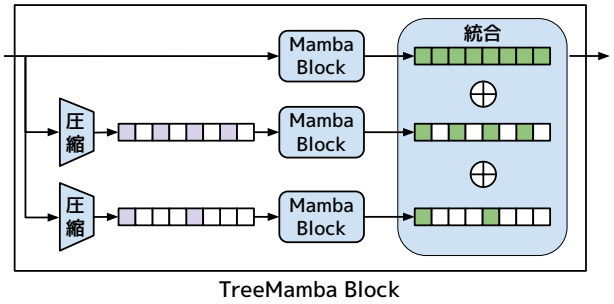

図 2 TreeMamba の模式図．シーケンス方向に対して間引いた入力を作成し，それぞれの出力を要素ごとの和によって統合する．

相対的に近づけ，最初の方にあるトークンが重要であった場合に，予測に対するその影響度を向上させることができるようにする狙いがある．また，必ずしも最初の方にあるトークンが重要とは限らないため，複数の入力から得られる出力を適切に統合することにより，より汎用的に長大な文脈に対応することができるようにする狙いもある．それでいて，間引いて作成される入力の長さが短くなっているため，MS-SSM ほど計算負荷を増やすことがないという利点もある手法となっている．

4 実験

TreeMamba の長大な文脈に対する性能を評価するために，標準的に用いられているベンチマークデータセットである LongBench-E [7] を用いて実験を行った．LongBench-E では数千トークンに及ぶ長大な文脈とそれに関する質問からなる入力に対して，どれだけ正確に回答を生成できるかが評価される．比較に用いた言語モデルは事前学習済みモデルで初期化した後に Common Pile データセット [11] で

表 1 Mamba-790m-hf をベースラインとして LongBench-E で評価した結果．スコアは全て高いほど良い性能を表す．

タスク名	Mamba	提案手法
HotpotQA (質問)	0.0430	**0.0556**
2WikiMultihopQA (質問)	0.0675	**0.0756**
MultiFieldQA (質問)	0.0882	**0.1203**
Qasper (質問)	**0.0588**	0.0443
GovReport (要約)	**0.1810**	0.1047
MultiNews (要約)	**0.1341**	0.0324
TriviaQA (質問)	0.2354	**0.4272**
SAMSum (要約)	**0.0233**	0.0055
TREC (分類)	0.2233	**0.3667**
PassageRetrieval (引用)	0.0342	**0.0535**
PassageCount (計上)	**0.0171**	0.0011
LCC (プログラムの生成)	0.3234	**0.3940**
RepoBench (プログラムの生成)	0.2820	**0.2964**

表 2 Mamba2-780m をベースラインとして LongBench-E で評価した結果．スコアは全て高いほど良い性能を表す．

タスク名	Mamba2	TreeMamba
HotpotQA (質問)	0.0453	**0.0458**
2WikiMultihopQA (質問)	0.0602	**0.0657**
MultiFieldQA (質問)	**0.1103**	0.1002
Qasper (質問)	**0.0353**	0.0332
GovReport (要約)	**0.1272**	0.1200
MultiNews (要約)	**0.1164**	0.1101
TriviaQA (質問)	0.4424	**0.4873**
SAMSum (要約)	0.0575	**0.0820**
TREC (分類)	0.4033	**0.4133**
PassageRetrieval (引用)	0.0623	**0.0627**
PassageCount (計上)	0.0083	**0.0204**
LCC (プログラムの生成)	0.5176	**0.5399**
RepoBench (プログラムの生成)	0.4447	**0.4536**

追加の事前学習を行ったものを使用した．事前学習済みの Mamba-790m-hf と，Mamba2-780m を軸に構築した TreeMamba において導入したパラメータを学習するためであるが，比較のため Mamba-790m-hf と Mamba2-780m にも同様に学習を施した．

結果として，Mamba-790m-hf をベースラインとした場合，図 1 に示したように，正確性の求められる要約タスクである GovReport，MultiNews，SAMSum や計上タスクである PassageCount では提案手法が性能低下を招いたものの，Qasper を除くその他の全てのタスクにおいて，性能向上を示した．Qasper に関しても要約タスクや計上タスクほど極端な性能低下は起きていない．Mamba2-780m をベースラインとした場合は，図 2 に示したように，MultiFieldQA，Qasper，GovReport，MultiNews で少し性能低下がみられたものの，それ以外のタスクでは性能向上を示した．いずれの場合においても，いくつかのタスクで性能改善がみられたため長大な文脈に対する処理能力に TreeMamba の効果があることが分かった．しかしながら，特に Mamba-790m-hf における要約タスクや計上タスクにおける性能低下が著しいため，圧縮や統合により正確性が低下している可能性がある．

LongBench-E だけでなく，平均入力長が 100 トークン未満のタスクを用いた性能評価も行った．評価に用いるモデルは LongBench-E の時と同じである．評価に用いたデータセットは LAMBADA [12]，HellaSwag [13]，PIQA [14]，ARC-challenge，ARC-easy [15]，WinoGrande [16]，OpenBookQA [17] である．便宜的にこれら 7 つのタスクを短文タスクと呼称する．結果は図 3 と図 4 に示した．Mamba-790m-hf を

表 3 Mamba-790m-hf をベースラインとしたときの短文タスクによる実験結果．スコアは全て高いほど良い性能を表す．

タスク名	Mamba	TreeMamba
ARC-challenge	**0.2619**	0.2560
ARC-easy	**0.6010**	0.5825
HellaSwag	**0.4148**	0.4042
LAMBADA	**0.5769**	0.5352
OpenBookQA	**0.2320**	0.2220
PIQA	**0.7127**	0.7062
WinoGrande	**0.5549**	0.5533

表 4 Mamba2-780m をベースラインとしたときの短文タスクによる実験結果．スコアは全て高いほど良い性能を表す．

タスク名	Mamba2	TreeMamba
ARC-challenge	**0.2594**	**0.2594**
ARC-easy	0.6014	**0.6019**
HellaSwag	0.4128	**0.4201**
LAMBADA	0.5997	**0.6074**
OpenBookQA	0.2200	**0.2400**
PIQA	0.7171	**0.7209**
WinoGrande	0.5904	**0.6022**

ベースラインとした TreeMamba では全体的にわずかな性能低下が発生し，逆に Mamba2-780m をベースラインとした TreeMamba では全体的にわずかな性能向上となった．これは Mamba2 の方が学習の安定性が高く学習が上手くいきやすいからだと考えている．この結果から，TreeMamba では短い入力に対する処理能力を損なっていないと結論付けられる．

5 おわりに

本研究では，計算効率に優れる Mamba アーキテクチャの長大な文脈に対する性能を向上させるために，マルチスケール処理を行う TreeMamba を提案した．図 1 と図 2 に示した LongBench-E による実験の結果から，正確性が重要な要約タスクと計上タスクなどで明確な性能低下を招く場合も見受けられたが，それ以外のほとんどのタスクにおいて提案手法の有効性が確認された．また，図 3 と図 4 に示した短文タスクの結果から，TreeMamba が短い入力に対する処理能力をほとんど損なっていないことが確認できた．

今後の課題として，比較対象となる LongMamba や MS-SSM を交えた実験を行うことや，提案手法の圧縮や統合の手法としてより最適なものがないか色々なパターンを試す必要がある．LongMamba や MS-SSM の手法を TreeMamba に適用した場合についても調査が必要である．

謝辞

本研究を進めるにあたり，多大なるご指導とご助言を賜りました指導教員の佐々木裕教授に，心より感謝申し上げます．研究の方向性の提示に加え，論文執筆においても多くのご指導をいただきました．知能数理研究室の皆様には，日常のゼミやそれ以外の場においても議論の機会をいただき，これらの議論を通じて本研究を一層深めることができました．心より感謝申し上げます．

参考文献

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. **Advances in neural information processing systems**, Vol. 30, , 2017.

[2] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. **arXiv preprint arXiv:2312.00752**, 2023.

[3] Tri Dao and Albert Gu. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In **International Conference on Machine Learning (ICML)**, 2024.

[4] Samy Jelassi, David Brandfonbrener, Sham M. Kakade, and Eran Malach. Repeat after me: Transformers are better than state space models at copying, 2024.

[5] Roger Waleffe, Wonmin Byeon, Duncan Riach, Brandon Norick, Vijay Korthikanti, Tri Dao, Albert Gu, Ali Hatamizadeh, Sudhakar Singh, Deepak Narayanan, Garvit Kulshreshtha, Vartika Singh, Jared Casper, Jan Kautz, Mohammad Shoeybi, and Bryan Catanzaro. An empirical study of mamba-based language models, 2024.

[6] Zhifan Ye, Kejing Xia, Yonggan Fu, Xin Dong, Jihoon Hong, Xiangchi Yuan, Shizhe Diao, Jan Kautz, Pavlo Molchanov, and Yingyan Celine Lin. Longmamba: Enhancing mamba's long-context capabilities via training-free receptive field enlargement. In **International Conference on Learning Representations**, 2025.

[7] Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, et al. Longbench: A bilingual, multitask benchmark for long context understanding. In **Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 3119–3137, 2024.

[8] Albert Gu, Isys Johnson, Karan Goel, Khaled Saab, Tri Dao, Atri Rudra, and Christopher Ré. Combining recurrent, convolutional, and continuous-time models with linear state space layers. **Advances in neural information processing systems**, Vol. 34, pp. 572–585, 2021.

[9] Mahdi Karami, Ali Behrouz, Peilin Zhong, Razvan Pascanu, and Vahab Mirrokni. MS-SSM: A multi-scale state space model for efficient sequence modeling. In **Second Conference on Language Modeling**, 2025.

[10] Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. Long range arena : A benchmark for efficient transformers. In **International Conference on Learning Representations**, 2021.

[11] Nikhil Kandpal, Brian Lester, Colin Raffel, Sebastian Majstorovic, Stella Biderman, Baber Abbasi, Luca Soldaini, Enrico Shippole, A. Feder Cooper, Aviya Skowron, Shayne Longpre, Lintang Sutawika, Alon Albalak, Zhenlin Xu, Guilherme Penedo, Loubna Ben, Elie Bakouch, John David, Honglu Fan, Dashiell Stander, Guangyu Song, Aaron Gokaslan, John Kirchenbauer, Tom Goldstein, Brian R, Bhavya Kailkhura, and Tyler Murray. The Common Pile v0.1: An 8TB Dataset of Public Domain and Openly Licensed Text. **arXiv preprint**, 2025.

[12] Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The LAMBADA dataset: Word prediction requiring a broad discourse context. In Katrin Erk and Noah A. Smith, editors, **Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 1525–1534, Berlin, Germany, August 2016. Association for Computational Linguistics.

[13] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In Anna Korhonen, David Traum, and Lluís Màrquez, editors, **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics**, pp. 4791–4800, Florence, Italy, July 2019. Association for Computational Linguistics.

[14] Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. Piqa: Reasoning about physical commonsense in natural language. In **Thirty-Fourth AAAI Conference on Artificial Intelligence**, 2020.

[15] 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. **arXiv:1803.05457v1**, 2018.

[16] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: an adversarial winograd schema challenge at scale. **Commun. ACM**, Vol. 64, No. 9, p. 99- 106, August 2021.

[17] Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, **Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing**, pp. 2381–2391, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.