# 生成文の短縮による言語モデルの計算量削減

海野圭矢　内田真人
早稲田大学
ku.sherlock2000@akane.waseda.jp　m.uchida@waseda.jp

## 概要

言語モデルの計算量削減を目的とした従来の手法は，1 トークンあたりの計算コストを削減することに焦点を当ててきた．本研究では，文章生成にかかるコストが生成トークン数にも影響されることに注目し，言語モデルの生成文を短縮することで計算量を削減する方法を検討する．これを実現する学習方法として，文章全体に長さに応じた報酬を与える強化学習と短い文章を誘発するトークンを正解データとする教師あり学習の 2 つを検証する．さらに，生成文の短縮率と性能の関係を調査することで，生成文の短縮が言語モデルの性能にどのように影響するかを示す．Phi-3-mini，Zamba2-2.7B における実験の結果，強化学習と教師あり学習の両手法において生成文の短縮に成功した．また，5%以下の性能低下のもとでは，Phi-3-mini は約 15%〜20%，Zamba2-2.7B は約 30%の短縮が可能であることが示された．

## 1　はじめに

大規模言語モデル（Large Language Models, LLMs）を活用したアプリケーションが急速に普及する中，LLMs が環境に与える影響への懸念が広がっている．2022 年に ChatGPT が登場したことを皮切りに，多くの LLMs が開発され，社会的な応用が拡大している．しかし，LLMs を開発および運用するためには，電力消費 [1, 2] や水資源の利用 [3]，二酸化炭素の排出 [2] といったさまざまな環境問題が伴う．

こうした状況の中，LLMs の計算量を削減するために枝刈り [4, 5] や量子化 [6, 7] などのモデル圧縮手法やプロンプト圧縮 [8, 9, 10] が提案されている．

多くの言語モデルは自己回帰モデルであるため，文章生成にかかる計算量は 1 トークンあたりの計算コストと生成トークン数によって決まる．そのため既存の計算量削減手法は，1 トークンあたりの計算コストを削減することに焦点を当てている．一方で，生成トークン数に注目した計算量削減手法は提案されていない．そこで本研究では，言語モデルに短い表現を学習させることで生成トークン数を削減する新たな計算量削減手法を検討する．具体的には，次を調査する．(1) 言語モデルに短い言語表現を学習させることが可能であるか．(2) 生成文の短縮がどの程度性能へ影響するか．

言語モデルに短い文章表現を学習させる方法として，生成した文章全体に対して長さに応じた損失を与える方法と，1 つのトークンに対してその後に続く文の長さに応じた損失を与える方法の 2 つが考えられる．本研究では，前者を実現する手法として強化学習に基づく手法を，後者を実現する手法として教師あり学習に基づく手法を検討する．

強化学習に基づく手法では，Reinforcement Learning from Human Feedback（RLHF）[11] を応用する．通常，RLHF は言語モデルに安全な出力やユーザが好む出力を学習させるために用いられるが，報酬関数を適切に設定することで，生成の質を維持したまま短い言語表現を学習させることができると考えられる．具体的には，学習中の言語モデル（Active Model）の生成文が，手法適用前の言語モデル（Reference Model）の生成文に類似しているほど，かつ短いほど高い報酬を与える報酬関数を定義する．

教師あり学習に基づく手法では，生成文内の 1 つのトークンに注目し，より短い文章を誘発するようなトークンを正解ラベルとして学習させる．具体的には，Reference Model の生成文内のトークンを別のトークンに置き換え，続きを Active Model に生成させることで，置き換え前後の文章の長さを比較する．生成文が短くなった場合は，置き換えたトークンを正解データとして学習させる．

Phi-3-mini，Zamba2-2.7B における実験の結果，強化学習および教師あり学習の両手法において，言語モデルの生成文を短縮することに成功した．また，ユーザ指示に基づくベンチマークテストの結果，生成文の短縮率と性能のトレードオフが観測さ

れ，短縮率の上昇に伴い，性能が低下していく様子が見られた．一方，5%以下の性能低下のもとでは，Phi-3-mini で約 15%〜20%，Zamba2-2.7B で約 30%の短縮に成功した．さらに，強化学習に基づく手法は報酬関数のハイパーパラメータを操作することで，トレードオフを容易にコントロールできることが特徴的であり，教師あり学習に基づく手法は短縮率のコントロールが難しいものの，強化学習よりも性能への影響が少ないことが特徴的であることが示された．

本手法はモデル圧縮とは異なり，モデルサイズを変えないため，言語モデルが保有する知識に影響を与えることなく計算量を削減することができる．また，プロンプト圧縮とは異なり，言語モデルの出力を短縮するため，入力に対して出力が長い場面においても効果的である．

## 2 関連研究

### 2.1 プロンプト圧縮

プロンプト圧縮は，言語モデルへの入力プロンプトを短縮することで計算量を削減する手法である．Selective Context[8] は，perplexity が低いトークンを削除することで短縮プロンプトを生成する．perplexity はエントロピーに基づく尺度であるため，情報理論的にプロンプトの情報量を保つことができる．TCRA-LLM[9] は，要約ベースと意味ベースの 2 つの手法を提供している．要約ベース手法は，言語モデルによってプロンプトを要約する．意味ベース手法は，各トークンが文全体の意味に占める重要度を計算し，重要度が低いトークンを削除することでプロンプトを短縮する．

プロンプト圧縮は，入力が長い場合には有効な手法であるが，入力に対して出力が長い場合には効果が少ないと考えられる．本研究では，視点を変え，プロンプトではなく生成文を圧縮する．

### 2.2 RLHF

RLHF は，言語モデルに安全な出力やユーザが好む出力をさせる手法であり，通常は次の手順を踏む．

1. 教師ありファインチューニング
2. 報酬モデルの学習
3. Proximal Policy Optimization (PPO) による学習

PPO は，Trust Region Policy Optimization (TRPO) という自然方策勾配法の強化学習手法を簡潔にしたもので，以下で定義される報酬改善量を最大化する．

$$J(\theta_{\text{new}}) = \mathbb{E}\left[r(\theta_{\text{new}})A(s,a)\right] \tag{1}$$

$$r(\theta_{\text{new}}) = \frac{\pi_{\theta_{\text{new}}}(a|s)}{\pi_{\theta_{\text{old}}}(a|s)} \tag{2}$$

ここで，$\pi$ は方策の確率であり，言語モデルが出力するトークンの確率分布に対応する．$A$ はアドバンテージと呼ばれ，行動価値関数と状態価値関数の差で定義される．PPO では，パラメータ更新前後の比 $r(\theta_{\text{new}})$ をクリッピングすることで過剰な更新を避ける．

本研究では，生成文の質を保ち，かつ短い出力を好ましい出力として言語モデルを学習させる．なお，本研究ではチューニング済みのモデルを使用するため，手順 1,2 は省略する．

## 3 提案手法

### 3.1 強化学習に基づく手法

強化学習に基づく手法では，RLHF を適用する．報酬モデルを設定するにあたって重要なのは，生成の質を損なわない範囲で生成文を短くすることである．そこで次の報酬モデルを設定する．

$$r = (1-\lambda)\text{sim}(S_{\text{ref}}, S_{\text{act}}) + \lambda\frac{L_{S_{\text{ref}}} - L_{S_{\text{act}}}}{L_{S_{\text{ref}}}} \tag{3}$$

ここで，$S_{\text{ref}}, S_{\text{act}}$ は Reference Model，Active Model からの生成文，$L_{S_{\text{ref}}}, L_{S_{\text{act}}}$ は生成文の長さ，$\text{Sim}(\cdot,\cdot)$ はテキストの類似度を返す関数である．第一項は，Active Model が Reference Model と類似した文を生成するようにする働きを持ち，第二項は，生成文長の短縮率を報酬として与えることで，文の短縮率を高めさせる働きを持つ．本研究では，sentence transformer である all-mpnet-base-v2[12] によって生成文をベクトル化し，そのコサイン類似度を類似度関数として用いる．

### 3.2 教師あり学習に基づく手法

教師あり学習に基づく手法の概要図を図 1 に示す．言語モデルは自己回帰モデルであるため，トークンの選択はその後に続く文章に大きな影響を与える．この手法では，より短い文章を誘発するトークンの確率を上げ，より長い文章を誘発するトークンの確率を下げるように学習を行う．大まかには次の手順に従う．(1) Reference Model の生成文内の 1 つ

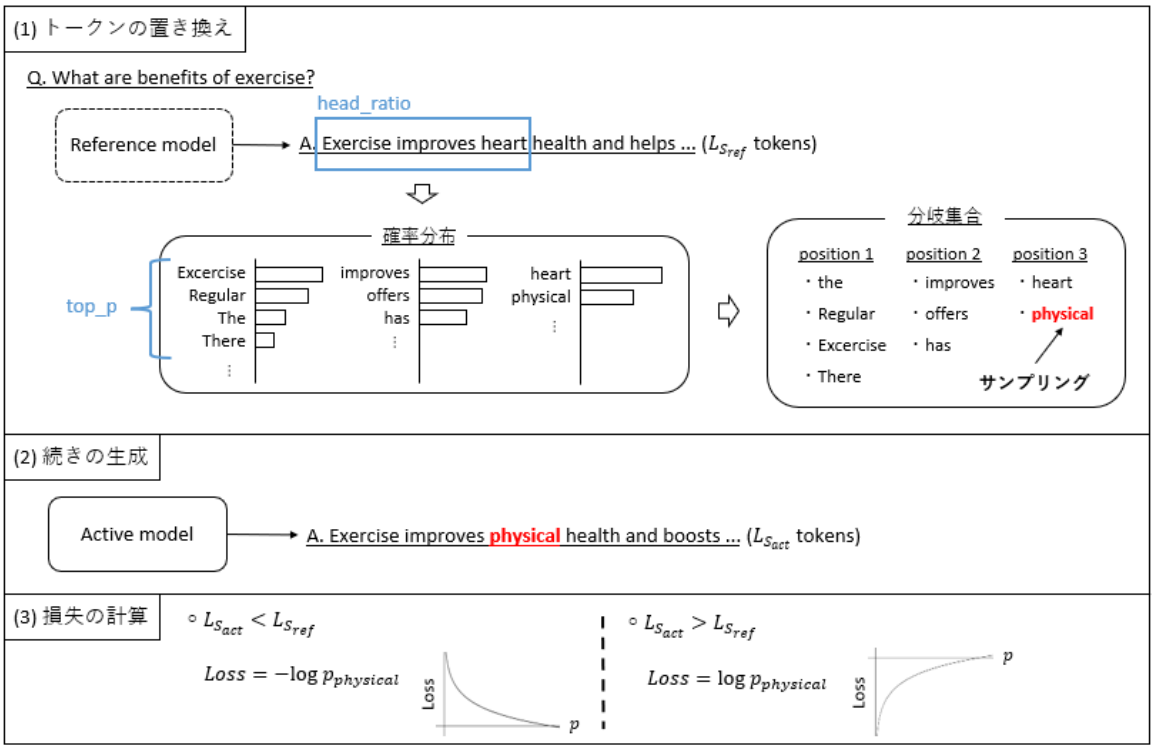



**図 1**　教師あり学習に基づく手法の概要図

のトークンを別のトークンに置き換える．(2) その続きを Active Model が生成する．(3) 置き換え前後で生成文が短くなった場合は，置き換えたトークンを正解データとして学習させる．反対に，生成文が長くなった場合は，不正解データとして出力確率を下げるように学習させる．

トークンの置き換えについて，はじめに，Reference Model が出力した確率分布から置き換えの候補となる分岐集合を得る．このとき分岐集合の作り方として，どの位置のトークンを候補にするか (head_ratio)，確率分布の上位何%のトークンを候補にするか (top_p) の 2 つを設定することができる．そして分岐集合からランダムサンプリングすることで，置き換え位置と置き換えトークンを決定する．

次に損失を計算するため，損失関数 $L$ を以下のように定義する．

$$L = (1 - \lambda_1 - \lambda_2) \cdot a \cdot L_{\mathrm{CE}} + \lambda_1 D_{\mathrm{KL}}(p_{\mathrm{ref}} || p_{\mathrm{act}}) \quad (4)$$

$$+ \lambda_2 D_{\mathrm{KL}}(p_{\mathrm{act}} || p_{\mathrm{ref}}) \quad (5)$$

ここで，$a$ は $\{1, -1\}$ の二値変数で正解データの学習では 1 を，不正解データの学習では $-1$ を取る．$L_{\mathrm{CE}}$ は交差エントロピー損失，$p_{\mathrm{ref}}, p_{\mathrm{act}}$ は Reference Model，Active Model が出力する各トークンの確率分布である．KL 距離は生成の質を保つ役割があり，フォワード KL 距離は，学習によってトークンの生成確率が急激に上昇することを抑える働きを持ち，リバース KL 距離は，トークンの生成確率が急激に下降することを抑える働きを持つ．

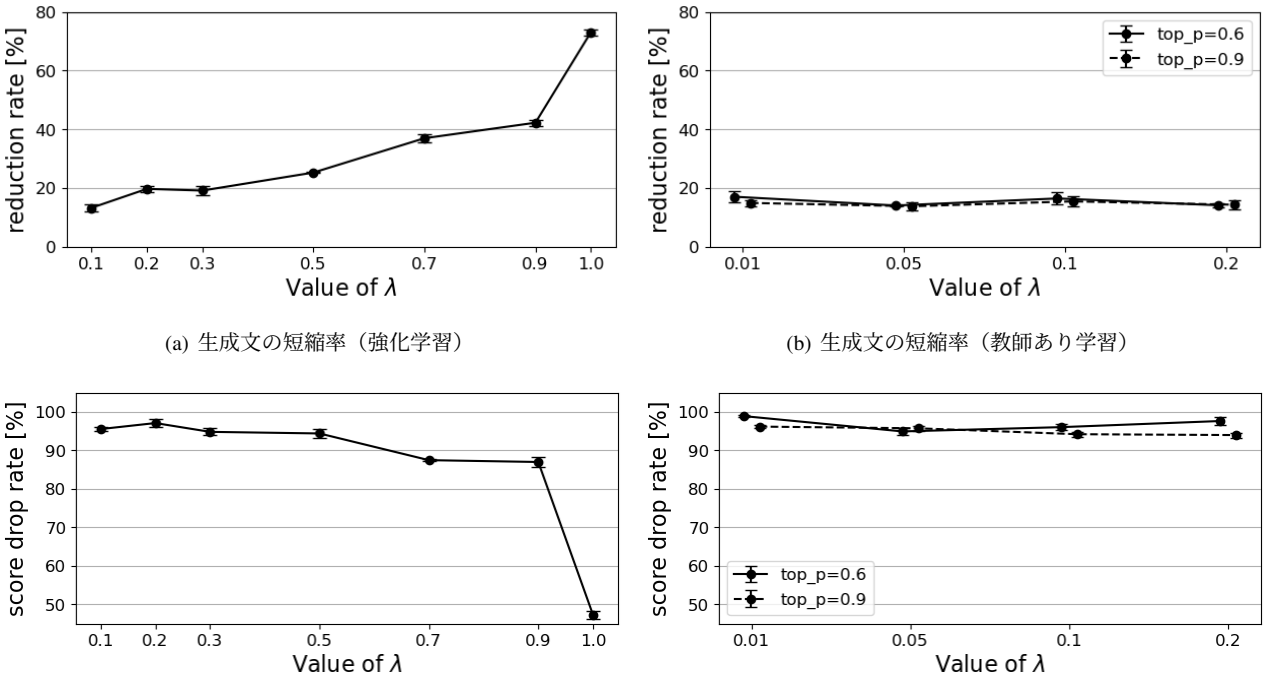
# 4　実験

## 4.1　実験設定

学習データには Alpaca データセット [13] から 4096 個のデータを用い，バッチサイズは 8，エポック数は 1 とした．

強化学習に基づく手法では，報酬モデル (3) におけるハイパーパラメータ $\lambda$ を，0.1 ~ 1.0 の間で変化させた．教師あり学習に基づく手法では，損失関数 (4) におけるハイパーパラメータ $\lambda_1, \lambda_2$ を $\lambda_1 = \lambda_2$ として実験を行った．また，head_ratio = 0.5，top_p = $\{0.6, 0.9\}$ として実験した．head_ratio = 0.5 としたのは，生成文の終盤のトークンを置き換えても短縮効果が弱いためである．

学習モデルには Phi-3-mini-4k-instruct[14] および Zamba2-2.7B-instruct[15] を用いた．Phi-3-mini-4k-instruct は，Transformer-decoder で構成されるモデルであり，Zamba2-2.7B-instruct は，Transformer-decoder と Mamba2 層 [16] が 1:6 で混合されているモデルである．Phi-3-mini-4k-instruct では，Attention 層にお




(a) 生成文の短縮率（強化学習）

(b) 生成文の短縮率（教師あり学習）

(c) スコアの低下率（強化学習）

(d) スコアの低下率（教師あり学習）

**図 2**　MT-Bench における生成文の短縮率とスコアの低下率（Phi-3-mini-4k-instruct）

いて入力を Query，Key，Value へと変換する線形層（qkv_proj）を学習の対象とし，Zamba2-2.7B-instruct では，Mamba2 層において入力を状態空間モデルを構成する行列へと変換する線形層（in_proj）を学習の対象とした．

評価実験には MT-Bench[17] を用い，gpt-4o-08-06 によってスコア付けを行った．本手法適用前のモデルの性能は，Phi-3-mini-4k-instruct は平均スコア 7.47，平均トークン数 335.4，Zamba2-2.7B-instruct は平均スコア 6.20，平均トークン数 407.2 であった．

## 4.2　実験結果

Phi-3-mini-4k-instruct において，学習後のモデルの MT-Bench スコアと生成文の長さの変化を図 2 に示す．強化学習に基づく手法では，$\lambda$ が大きくなるにつれ生成文の短縮率が大きくなり（図 2(a)），その一方でスコアの低下率も大きくなる（図 2(c)）というトレードオフが観測された．また，短縮率が 20%程度である場合，スコアの低下は約 5%であり，短縮率が 40%程度でもスコアの低下が約 10%に留まった．教師あり学習に基づく手法では，$\lambda$ や top_p の値に依らず，短縮率は 15〜20%程度であった（図 4(b)）．しかし，スコアは top_p = 0.6 のときの方が高い傾向にあり，性能への影響がより少なかった（図 4(d)）．特に，$\lambda = 0.01$，top_p = 0.6 のときは，スコアの減少率が約 2%に抑えられ，強化学習に比べても性能への影響が少ない結果となった．

Zamba2-2.7B-instruct においても，5%以下の性能低下において，約 30%の短縮に成功した（付録 B）．

Phi-3-mini の学習の経過は付録 A に示す．

## 5　おわりに

本研究では，言語モデルの文章生成にかかるコストが 1 トークンあたりの計算量だけでなく生成トークン数にも影響されることに注目し，言語モデルの生成文を短縮する新たな計算量削減手法を検討した．文章全体に長さに応じた報酬を与える強化学習と短い文章を誘発するトークンを正解データとする教師あり学習の 2 つを検証し，どちらの手法においても生成文の短縮に成功した．ユーザ指示に基づく評価実験の結果，生成文の短縮率とスコアのトレードオフが観測され，Phi-3-mini では約 15%〜20%，Zamba2-2.7B では約 30%の短縮率であれば性能への影響が最小限であることが示された．

また，強化学習では，生成文の短縮率と性能低下のトレードオフを調整できることが特徴であり，教師あり学習では，強化学習に比べて性能への影響を抑えることが可能であると分かった．

# 参考文献


[1] IEA. Electricity 2024, 2024. Licence: CC BY 4.0.

[2] Nestor Maslej, Loredana Fattorini, Raymond Perrault, Vanessa Parli, Anka Reuel, Erik Brynjolfsson, John Etchemendy, Katrina Ligett, Terah Lyons, James Manyika, Juan Carlos Niebles, Yoav Shoham, Russell Wald, and Jack Clark. Artificial intelligence index report 2024, 2024.

[3] Pengfei Li, Jianyi Yang, Mohammad A. Islam, and Shaolei Ren. Making ai less "thirsty": Uncovering and addressing the secret water footprint of ai models. **arXiv preprint arXiv:2304.03271**, 2023.

[4] Elias Frantar and Dan Alistarh. Sparsegpt: Massive language models can be accurately pruned in one-shot. In **International Conference on Machine Learning**, pp. 10323–10337. PMLR, 2023.

[5] Mingjie Sun, Zhuang Liu, Anna Bair, and J Zico Kolter. A simple and effective pruning approach for large language models. In **The Twelfth International Conference on Learning Representations**, 2024.

[6] Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. OPTQ: Accurate quantization for generative pre-trained transformers. In **The Eleventh International Conference on Learning Representations**, 2023.

[7] Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. Awq: Activation-aware weight quantization for on-device llm compression and acceleration. **Proceedings of Machine Learning and Systems**, Vol. 6, pp. 87–100, 2024.

[8] Yucheng Li. Unlocking context constraints of llms: Enhancing context efficiency of llms with self-information-based content filtering. **arXiv preprint arXiv:2304.12102**, 2023.

[9] Junyi Liu, Liangzhi Li, Tong Xiang, Bowen Wang, and Yiming Qian. Tcra-llm: Token compression retrieval augmented large language model for inference cost reduction. In **Conference on Empirical Methods in Natural Language Processing**, 2023.

[10] Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. LLMLingua: Compressing prompts for accelerated inference of large language models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, **Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing**, pp. 13358–13376, Singapore, December 2023. Association for Computational Linguistics.

[11] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. **Advances in neural information processing systems**, Vol. 35, pp. 27730–27744, 2022.

[12] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In **Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing**. Association for Computational Linguistics, 11 2019.

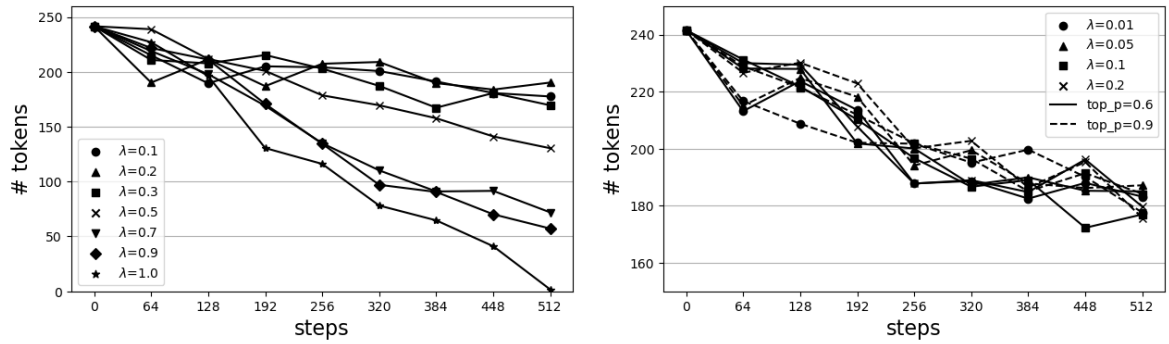
[13] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

[14] Microsoft. Phi-3 technical report: A highly capable language model locally on your phone. **arXiv preprint arXiv:2404.14219**, 2024.

[15] Paolo Glorioso, Quentin Anthony, Yury Tokpanov, Anna Golubeva, Vasudev Shyam, James Whittington, Jonathan Pilault, and Beren Millidge. The zamba2 suite: Technical report, 2024.

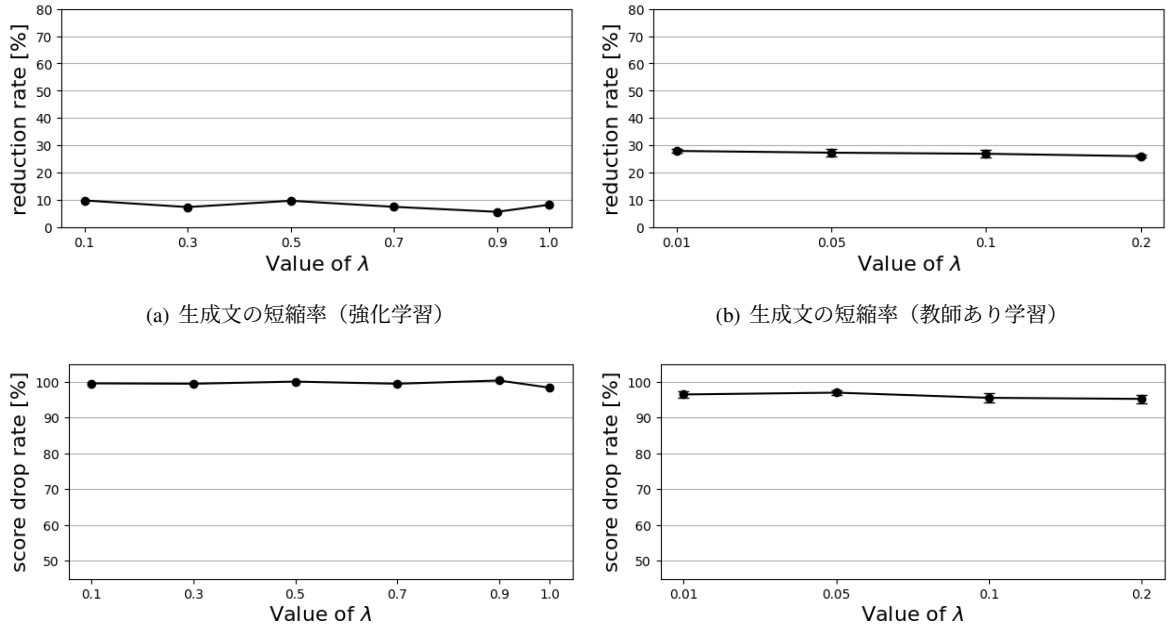
[16] Tri Dao and Albert Gu. Transformers are ssms: generalized models and efficient algorithms through structured state space duality. In **Proceedings of the 41st International Conference on Machine Learning**, ICML'24. JMLR.org, 2025.

[17] 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, 2023.




(a) 強化学習に基づく手法　　(b) 教師あり学習に基づく手法

**図 3**　学習の経過．50 個の検証データに対する平均出力トークン数．




(a) 生成文の短縮率（強化学習）　　(b) 生成文の短縮率（教師あり学習）

(c) スコアの低下率（強化学習）　　(d) スコアの低下率（教師あり学習）

**図 4**　MT-Bench における生成文の短縮率とスコアの低下率（Zamba2-2.7B-instruct）

# A　学習の経過

学習の経過を図 3 に示す．64 ステップごとに検証データセットを用いて，学習中のモデルの生成文の長さの変化を測定した．検証データには Alpaca データセットから学習には含まれていない 50 個のインストラクションを用いた．強化学習に基づく手法では，学習が進むにつれて生成文の長さが減少し，$\lambda$ が大きいほど生成文の長さが短くなった．教師あり学習に基づく手法でも同様に，学習が進むにつれてモデルの生成文の長さが短くなったものの，top_p や $\lambda$ の値に関わらず，減少の傾向は類似していた．

# B　Zamba2-2.7B-instruct の結果

Zamba2-2.7B-instruct における MT-Bench スコアと生成文の長さの変化を図 4 に示す．強化学習に基づく手法では，Phi-3-mini のようなトレードオフを観測することはできなかったものの，生成文の短縮に成功した．教師あり学習に基づく手法では，top_p を 0.6 に固定して実験を行った．Phi-3-mini-4k-instruct における結果と同様に，教師あり学習に基づく手法では $\lambda$ の値に関わらず短縮率は一定であった．短縮率は 30%近くあり，スコアの低下率は 5%以下に抑えられた．

強化学習に基づく手法が Phi-3-mini のような結果にならなかった要因として，強化学習のハイパーパラメータが多いことが挙げられる．このため，モデルごとのチューニングが容易ではなく，試行錯誤を伴う．したがって，強化学習に基づく手法の結果については改善の余地があると考えられる．