大規模言語モデルに対するチューニング手法の調査：内部のアクセス性に基づく分類と比較

中島 京太郎 †, 金 輝燦 †, 平澤 寅庄 †, 榎本 大晟 †, 陳 宙斯 ‡, 小町 守 ‡
† 東京都立大学, ‡ 一橋大学
{nakajima-kyotaro@ed., kim-hwichan@ed., toshosan@, enomoto-taisei@ed.}tmu.ac.jp
{zhousi.chen, mamoru.komachi}@r.hit-u.ac.jp

概要

言語モデルを特定のタスクに対して最適化するチューニング手法は現在多くの場面で用いられている．チューニングには言語モデル内部のパラメータを更新する手法や，入力文を更新する手法など様々なアプローチが存在する．これらのチューニング手法は対象の言語モデルの内部のアクセス性によって使用できる手法が異なる．本研究では言語モデルのアクセス性に応じて分類されたチューニング手法を多角的に比較することで，それらの特徴や傾向を分析する．

1 はじめに

現在，大規模言語モデル（LLM）は幅広い分野において注目されている．LLM は特定のタスクに対して最適化することで幅広いタスクを解くことができる．モデルをタスクに対して最適化する技術を**チューニング**と呼ぶ．

チューニングのアプローチは大きく二つに分けられる．1 つ目のアプローチは言語モデル内部のパラメータを最適化する手法である．このアプローチの代表的なチューニング手法は微調整である [1]．微調整はモデル内部のパラメータを教師あり学習を介してタスクに最適化する手法である．2 つ目のアプローチは文脈内学習である．文脈内学習は入力文にタスクの情報を追加する．タスクについての指示文や具体的な事例を入力文に追加することで，モデルのパラメータを更新することなく，チューニングを実現する．

チューニング手法は**モデルのアクセス性**によって適用できるアプローチが異なる．例えばモデル内部へのアクセスを制限しているブラックボックスモデルは内部のパラメータにアクセスできない．そのためパラメータを最適化するアプローチは適用することができない．

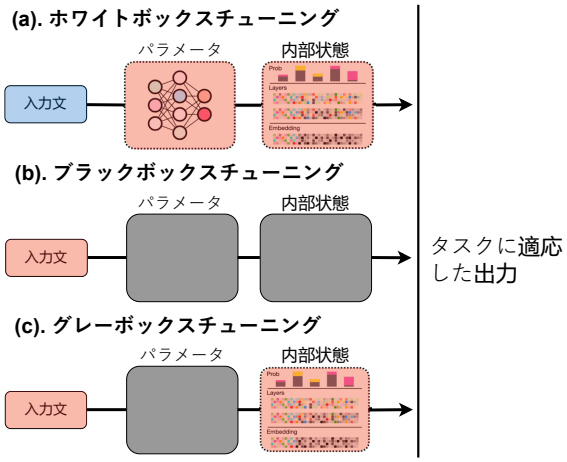

図 1 チューニング手法の各分類の図．チューニングの際にモデル内部のどの情報にアクセスするかで分類する．内部状態は入力文の埋め込みや各層の情報，生成確率などを意味する．各チューニングの分類は青色の部分を固定し，赤色の部分に更新を行う．チューニングの際，灰色の部分の情報にはアクセスしない．

本研究は，現状研究されているチューニング手法を適用できるモデルのアクセス性によって分類した．分類はモデル内の全てにアクセスできる**ホワイトボックスチューニング**，内部の情報に一切アクセスしない**ブラックボックスチューニング**，パラメータ以外にアクセスを行う**グレーボックスチューニング**である（図 1）．これらの分類に属する手法をそれぞれ多角的に比較する．また，各分類のチューニング手法はこれまでに公平な比較がされていない．そこで，本研究では各分類の代表的なチューニング手法を選択し，性能面とコスト面の 2 つの観点から比較する．そして，チューニング手法の分類と比較をもとに今後のチューニング手法の発展について考察を行う．

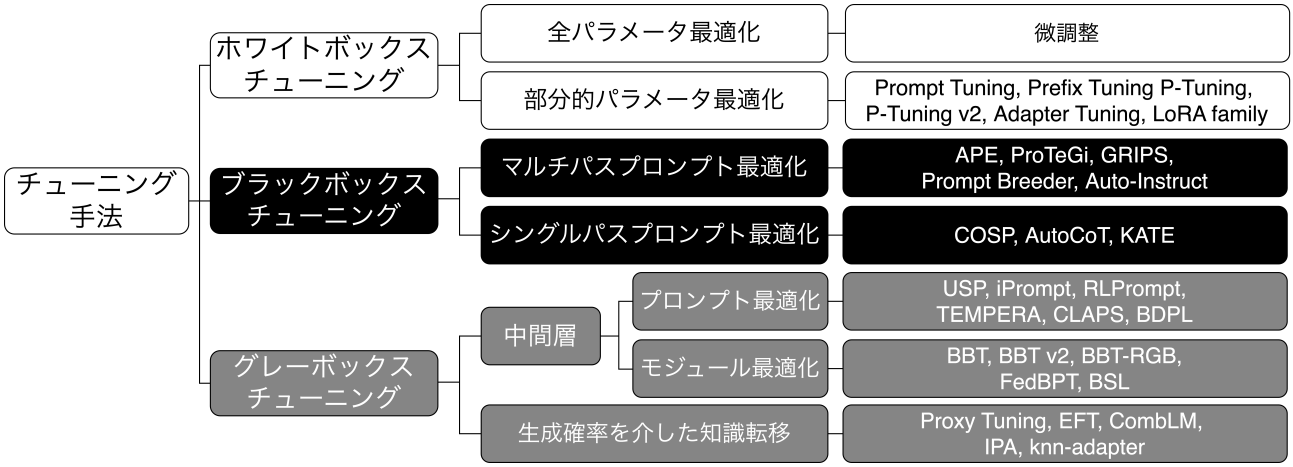

図 2　調査したチューニング手法の分類.

2　前提知識

2.1　文脈内学習

文脈内学習は適応するタスクの情報を入力文に組み込むことで，モデル内部のパラメータにアクセスすることなく言語モデルをチューニングする手法である [2]．タスクの情報はプロンプトと呼ばれる自然言語の形式でモデルに入力される．プロンプトはタスクの説明や入力文と期待される出力のペアの例などで構成される．文脈内学習の課題としてプロンプトがタスクの性能に大きく影響することが挙げられる [3]．高い性能を引き出すプロンプトの作成には多大な労力と専門的な知識が必要である．文脈内学習を元にしたチューニング手法はプロンプトを自動生成し，最適化することを目標としている．

2.2　Derivative-Free Optimization

Derivate-Free Optimization (DFO) は勾配情報を使用せずに最適解を探索する手法である．DFO はモデルのパラメータにアクセスせずに最適解を探索するため，ブラックボックスモデルやグレーボックスモデルのようなパラメータにアクセス不可能な LLM のチューニング手法として適している．DFO には様々なアプローチが存在し，遺伝的アルゴリズムやベイズ最適化が含まれる．

遺伝的アルゴリズム　遺伝的アルゴリズム [4] は生物の進化を模倣した最適化手法であり，優れた遺伝子を次世代に引き継ぐことで，より良い解を探索する．遺伝的アルゴリズムはまず解の候補となる集合を作成し，各候補を評価する．候補集合の中で性能が高いものだけを次世代（次のイテレーション）に残す．これらの選ばれた候補に言い換えなどの変化を加え，新たな候補集合を生成する．生成された候補を再び評価し，優れた解を繰り返し保存することで，世代を重ねながら最適化に向けて探索と収束を進めていく．

2.3　モデルのアクセス性による分類

本研究ではモデル内部のアクセス性に基づきホワイトボックスモデル，ブラックボックスモデル，グレーボックスモデルの 3 段階に分類を使用する [5]．

ホワイトボックスモデル　ホワイトボックスモデルはモデル内部の全てに対してアクセスできるようなモデルである．完全な内部アクセスによって誤差逆伝播法が可能となる．HuggingFace[1] などのオープンなコミュニティでは，多くのホワイトボックスモデルが利用できる．代表的な例として，LLaMA シリーズなどがある．

ブラックボックスモデル　ブラックボックスモデルはモデル内部に対して一切のアクセスを許可しない．モデルの内部へのアクセスが禁止されている場合，ユーザが利用できる情報は入力テキストと対応する出力テキストのみとなる．ブラックボックスモデルの例として，Gemini や Grok などがある．

グレーボックスモデル　グレーボックスモデルはパラメータへのアクセスを禁止しているが，モデルのパラメータ以外へのアクセスは許可されている．具体的には，生成確率や入力埋め込みなど特定の部分にアクセス可能なモデルを指す．OpenAI の GPT-3.5 以降のシリーズや AI21 Labs の Jurassic-2 シリーズは対数確率を公開しているため，グレーボックスモデルに該当する．

1)　https://huggingface.co/

3　各チューニング手法の分類

図 2 にチューニング手法の分類を示す．チューニング手法の各分類は以下のように分類する．

3.1　ホワイトボックスチューニング

ホワイトボックスチューニングはモデルの内部パラメータを更新する手法である．これらの手法では学習データを用いて損失関数の勾配を計算し，誤差逆伝播法を用いて内部のパラメータを最適化する．

3.1.1　全パラメータ最適化

フルパラメータ最適化はホワイトボックスモデルに適用させるチューニングのアプローチであり，モデル内部のパラメータ全てに対して更新を行うものである．このアプローチにおいて最も代表的な手法は微調整であり，特定のタスクに対してモデルを最適化するために全パラメータを更新する．

3.1.2　部分的パラメータ最適化

このカテゴリはモデル内部のパラメータを部分的に最適化するアプローチである [6, 7, 8, 9, 10, 11, 12, 13, 14]．このアプローチは Parameter-efficient Fine-tuning (PEFT) として知られ，最小限のパラメータの更新のみでチューニングすることを目指すものである．代表的な手法として Low-Rank Adaptation (LoRA) や Prompt-Tuning がある．

3.2　ブラックボックスチューニング

ブラックボックスチューニングとは言語モデルの内部にアクセスすることなく適用可能なチューニング手法である．ブラックボックスチューニングにおいてアクセス可能な情報は入力文とそれに対応する出力文のみである．特に直接操作可能な情報は入力文のみに限られる．ブラックボックスチューニングでは出力文やその他の外部情報を元にプロンプトを最適化する．

3.2.1　マルチパスプロンプト最適化

このカテゴリはプロンプトを反復的に最適化するアプローチである [15, 16, 17, 18, 19, 20]．カテゴリ内のチューニング手法の多くは遺伝的アルゴリズムを採用している．言語モデルははじめに複数のプロンプトの候補を生成し，その中で性能が高いプロンプトを選択する．選択した高性能のプロンプトを元に新たなプロンプトの候補集合を生成する．高性能なプロンプトを生成し選択するこのサイクルは反復的に繰り返され，プロンプトを徐々に洗練させて性能を向上させる．このカテゴリには Auto Prompt Engineer (APE) などが属する．

3.2.2　シングルパスプロンプト最適化

この分類には一度の処理で最適なプロンプトを作成することを目指す手法が属する [21, 22, 23]．具体的には，ホワイトボックスモデルから得られる入力文の埋め込み表現を利用して学習データ内から最適な事例をプロンプトに追加するアプローチがある．

3.3　グレーボックスチューニング

グレーボックスチューニングはパラメータ以外の情報にアクセスできる言語モデルに適用可能な手法である．具体的には入力文に加えて入力文の埋め込み表現や推論時の各トークンの生成確率にアクセスできる．本研究ではグレーボックスチューニングのアプローチをモデルの中間層に対して更新をかけるものと，生成確率を更新する 2 つに分類して説明を行う．

3.3.1　中間層最適化

中間層に対してのチューニング手法はプロンプトを最適化する手法とモジュールを最適化する手法の 2 つを含む．

プロンプト最適化　プロンプト最適化はグレーボックスチューニングにおけるプロンプトを更新することを目的とした分類である [24]．ブラックボックスチューニングに比べて利用可能な情報が多く，より柔軟にプロンプトを評価，探索できる．

モジュール最適化　モジュール最適化は行列（モジュール）を最適化し，モデルにモジュールを結合することでチューニングを行う [25]．モジュールは入力文の埋め込みの前に挿入する場合と，モデルの各層の前に挿入される 2 つの場合がある．

3.3.2　生成確率を介した知識転移

Log Probability は小規模なモデルをチューニングして得たタスクの知識を，生成確率を介して大規模なグレーボックスモデルに転移する [26]．具体的には小規模な言語モデルのチューニングし，チューニング前後の生成確率の変化を大規模なモデルに転移する．

表 1　チューニング手法の性能の比較.

分類	手法	SST-2	AGnews
ホワイト	LoRA	96.8	94.0
ブラック	APE	95.1	75.2
	KATE	89.1	81.4
グレー	USP	95.0	71.0
	Proxy-Tuing	95.4	93.2

表 2　SST-2 におけるチューニング手法のコストの比較.

分類	手法	メモリ量 (MB)	実行時間
ホワイト	LoRA	83,968	7 時間 20 分
ブラック	APE	55,790	0 時間 40 分
	KATE	60,650	1 時間 30 分
グレー	USP	50,380	0 時間 25 分
	Proxy-Tuing	50,802	2 時間 50 分

4　各チューニング手法に対する考察

表 1 に各カテゴリに属する代表的なチューニング手法の性能を示す．モデルは llama2-13B を使用し，タスクは評価極性分類 (SST-2) と文書分類 (AGnews) の 2 種類の分類タスクを用いる．また表 2 に SST-2 に対するチューニング時にかかるコスト（メモリ量と時間）を示す．この実験結果と各分類の特徴をもとに各チューニング手法について議論する．

4.1　ホワイトボックスチューニングに対する考察

ホワイトボックスチューニングは各チューニング手法の分類の中で最も性能が高い．一方でチューニング時に必要なメモリ量や時間は他の手法の中で最も高い．これよりホワイトボックスチューニングは高性能かつ高コストであるとわかる．

ホワイトボックスチューニングはコストに加えて，モデルの変化に対する頑健性が低いことが課題である．近年，新しい深層ニューラルネットワーク構造が注目を浴びており，ネットワーク構造の変化によって適用可能なチューニング手法も変化する．例えば Kolmogorov-Arnold Networks (KAN) [27] は現在の多くのモデルに使われている多層パーセプトロンの代替となるようなネットワーク構造である．従来の構造とは異なり，KAN は線形層の重みを使用せずに非線形層を学習する．そのため線形層を修正する LoRA のようなホワイトボックスチューニングの手法は，KAN を使用する言語モデルには適用できない．

4.2　ブラックボックスチューニングに対する考察

ブラックボックスチューニングは SST-2 においてその他の手法に近い性能を発揮する．一方で AGnews では SST-2 に比べ性能が劣化している．ブラックボックスチューニングはタスクの難易度によって有用性が変わると言える．ブラックボックスチューニングは最適化が可能なのは自然言語のプロンプトのみである．自然言語のプロンプトに含むことができるタスクの情報はパラメータの更新によって得られるものよりも限定的であり，複雑なタスクの場合は十分にタスクの情報を言語モデルに伝えられていないことが性能低下の要因だと考えられる．

ブラックボックスチューニングの利点はコストである．メモリ量，時間の両方でホワイトボックスチューニングのコストを下回る．また入出力のみにしかアクセスしないため，モデル内部の変化に対して影響を受けにくく，モデルの変化に対する頑健性も高い．

4.3　グレーボックスチューニングに対する考察

グレーボックスチューニングはプロンプトを最適化する USP に関しては基本的にブラックボックスチューニングと同様の結果である．生成確率を介して小規模言語モデルから知識を転移する Proxy-Tuning は性能，コストの両方において優れている．

しかし知識の転移は同じ種類のモデル間でのみしか実行できないことが多々ある．例えば Proxy-Tuning は転移するモデル間で共通の語彙を持つ必要がある．知識転移は性能とコストの観点の両方で有用なアプローチである．そのため異なるモデル間での語彙の対応づけなど，知識転移を可能にする技術の開発が重要になってくると考えられる．

5　おわりに

本研究ではアクセス性ごとに分類した各モデルの種類において，それぞれどのようなチューニング手法が適用できるかを調査した．また調査したチューニング手法を比較することで現状のチューニング手法の研究の傾向や，将来的にどのような観点が重要になるかを議論した．

参考文献

[1] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. In **ACL 2018**. ACL, July 2018.

[2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In **NeurIPS 2020**. Curran Associates, Inc., 2020.

[3] Ellen Jiang, Kristen Olson, Edwin Toh, Alejandra Molina, Aaron Donsbach, Michael Terry, and Carrie J Cai. PromptMaker: Prompt-based prototyping with large language models. In **CHI EA '22'**. ACM, 2022.

[4] Nikolaus Hansen, Sibylle D. Müller, and Petros Koumoutsakos. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). **Evolutionary Computation**, 2003.

[5] Haotian Sun, Yuchen Zhuang, Wei Wei, Chao Zhang, and Bo Dai. BBOX-ADAPTER: lightweight adapting for black-box large language models. In **ICML 2024**, ICML'24. JMLR, 2025.

[6] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In **EMNLP 2021**. ACL, November 2021.

[7] Xiang Lisa Li and Percy Liang. Prefix-Tuning: Optimizing continuous prompts for generation. In **ACL-IJCNLP 2021**, Online, August 2021. ACL.

[8] Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-Tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In **ACL 2022**. ACL, May 2022.

[9] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In **ICML 2019**. PMLR, Jun 2019.

[10] Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In **ICML 2022**, 2022.

[11] Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adaptive budget allocation for parameter-efficient fine-tuning. In **ICLR 2023**, 2023.

[12] Mojtaba Valipour, Mehdi Rezagholizadeh, Ivan Kobyzev, and Ali Ghodsi. DyLoRA: Parameter-efficient tuning of pre-trained models using dynamic search-free low-rank adaptation. In **EACL 2023**. ACL, May 2023.

[13] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs. In **NeurIPS 2023**, 2023.

[14] Hwichan Kim, Shota Sasaki, Sho Hoshino, and Ukyo Honda. A single linear layer yields task-adapted low-rank matrices. In **LREC-COLING 2024**. ELRA and ICCL, May 2024.

[15] Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. Large language models are human-level prompt engineers. In **ICLR 2023**, 2023.

[16] Reid Pryzant, Dan Iter, Jerry Li, Yin Lee, Chenguang Zhu, and Michael Zeng. Automatic prompt optimization with "gradient descent" and beam search. In **EMNLP 2023**. ACL, December 2023.

[17] Archiki Prasad, Peter Hase, Xiang Zhou, and Mohit Bansal. GrIPS: Gradient-free, edit-based instruction search for prompting large language models. In **EACL 2023**. ACL, May 2023.

[18] Chrisantha Fernando, Dylan Banarse, Henryk Michalewski, Simon Osindero, and Tim Rocktäschel. Promptbreeder: self-referential self-improvement via prompt evolution. In **ICML 2024**, ICML'24. JMLR, 2025.

[19] Zhihan Zhang, Shuohang Wang, Wenhao Yu, Yichong Xu, Dan Iter, Qingkai Zeng, Yang Liu, Chenguang Zhu, and Meng Jiang. Auto-Instruct: Automatic instruction generation and ranking for black-box language models. In **EMNLP 2023 Findings**. ACL, December 2023.

[20] Lichang Chen, Jiuhai Chen, Tom Goldstein, Heng Huang, and Tianyi Zhou. InstructZero: Efficient instruction optimization for black-box large language models. In **ICML 2024**. PMLR, Jul 2024.

[21] Xingchen Wan, Ruoxi Sun, Hanjun Dai, Sercan Arik, and Tomas Pfister. Better zero-shot reasoning with self-adaptive prompting. In **ACL 2023 Findings**. ACL, July 2023.

[22] Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. What makes good in-context examples for GPT-3? In **DeeLIO 2022**. ACL, May 2022.

[23] Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. Automatic chain of thought prompting in large language models. In **ICLR 2023**, 2023.

[24] Xingchen Wan, Ruoxi Sun, Hootan Nakhost, Hanjun Dai, Julian Eisenschlos, Sercan Arik, and Tomas Pfister. Universal self-adaptive prompting. In **EMNLP 2023**. ACL, 2023.

[25] Tianxiang Sun, Yunfan Shao, Hong Qian, Xuanjing Huang, and Xipeng Qiu. Black-box tuning for language-model-as-a-service. In **ICML 2022**, PMLR. PMLR, Jul 2022.

[26] Alisa Liu, Xiaochuang Han, Yizhong Wang, Yulia Tsvetkov, Yejin Choi, and Noah A. Smith. Tuning language models by proxy. In **COLM 2024**, 2024.

[27] Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljačić, Thomas Y. Hou, and Max Tegmark. KAN: Kolmogorov-arnold networks, 2024.