A Survey for LLM Tuning Methods: Classifying Approaches Based on Model Internal Accessibility

Kyotaro Nakajima[†], **Hwichan Kim**[†], **Tosho Hirasawa**[†] **Taisei Enomoto**[†]
Zhousi Chen[‡], **Mamoru Komachi**[‡]
[†]Tokyo Metropolitan University, [‡]Hitotsubashi University
{nakajima-kyotaro@ed., kim-hwichan@ed., toshosan@, enomoto-taisei@ed.}tmu.ac.jp
{zhousi.chen, mamoru.komachi}@r.hit-u.ac.jp

Abstract

Recent large language models (LLMs) have significant inference potential. Tuning methods are techniques used to adapt these inference capabilities to specific tasks. However, unlike earlier, smaller models that allowed for efficient fine-tuning, modern LLMs function more like black boxes, disallowing access to their parameters and preventing traditional fine-tuning. Consequently, tuning studies have evolved to explore new approaches. In this survey, we categorize 36 tuning studies into a hierarchical structure. The root categories are as follows: 1) *white-box tuning* requires full or partial access to model parameters; 2) *black-box tuning* only involves modifying the task instructions within the input text; 3) *grey-box tuning* has limited internal access, such as input embeddings, intermediate layer states, or output log probabilities. We analyze tuning studies and discuss future trends based on the model properties these tuning techniques depend on.

1 Introduction

Before the advent of large language models (LLMs), pre-trained language models (PLMs) were a major focus in natural language processing (NLP) (Devlin et al., 2019; Brown et al., 2020; Raffel et al., 2023; Fedus et al., 2022; Zhang et al., 2020; Qiu et al., 2020). Tuning methods adapt the inference capabilities of PLMs to perform specific tasks. These smaller models cannot effectively solve tasks on their own until they are *tuned* for particular applications. A notable approach, known as fine-tuning, updates the model's parameters using gradients derived from specific tasks (Howard and Ruder, 2018). Fine-tuning adjusts all internal parameters of the model, proved to be an efficient technique for optimizing PLMs.

Recently, some LLMs do not allow access to their internals (i.e., any parameters or most activations). For instance, commercial LLMs like ChatGPT, GPT-4 (OpenAI et al., 2024), and Gemini

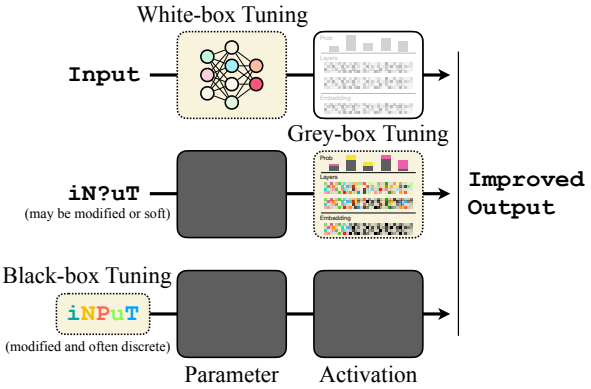

Figure 1: Classification of LLMs and tuning methods by their internal accessibility. We highlight tuning focuses in dotted boxes with colored fonts and shapes.

(Team and Anil, 2024) do not allow any access to their internal parameters. Without such access, traditional tuning approaches that involve updating parameters, such as fine-tuning, cannot be applied.

For LLMs with closed internals, in-context learning (Brown et al., 2020; Dong et al., 2024, ICL) is useful. ICL allows LLMs to adapt to specific tasks by incorporating an overview or examples of the tasks directly into the input, reducing the need for parameter tuning (von Oswald et al. (2023) and Deutch et al. (2024) suggested their equivalence). Additionally, fine-tuning with different hyperparameters tends to be more expensive compared to ICL. Tuning studies have gradually increased in aspects of modifying input and internal activations.

The applicability of tuning approaches varies depending on the level of access available to model internals. To account for the differences in tuning approaches, we utilize the three model classifications based on internal accessibility as proposed by Sun et al. (2024a), and summarize the existing tuning studies that can be applied to each category.

The categories of models are white-box, black-box and grey-box as shown in Figure 1. White-box models provide full access to their internals, in-

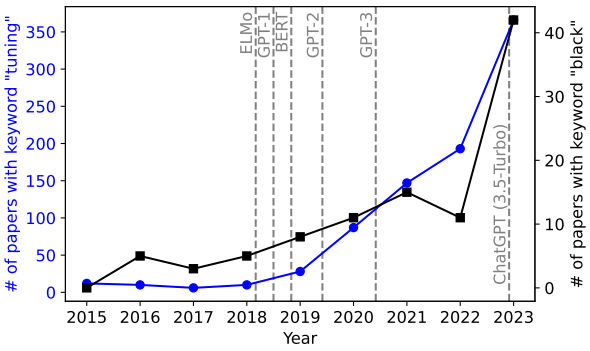

Figure 2: The transition in the number of papers w.r.t. keywords in titles. Analysis targets are the papers published in the ACL Anthology from 2015 to 2023. The two related keywords are set "tuning" and "black".

cluding parameters and all internal activations for backpropagation. In contrast, black-box models do not permit any access to their internals; the only available information is the text input and output. Additionally, there are models with partial inaccessibility, referred to as grey-box models. Grey-box models hide their parameters but reveal certain activations, such as input embeddings, layer states, and output log probabilities, to allow for tuning.

This paper covers following topics:

- We systematically categorize tuning studies involving LLMs in a hierarchy, as an extension of white-, black-, and grey-box categories.
- We discuss the features of each tuning approach, providing availability reference for the selection of a tuning approach w.r.t. a specific LLM internal accessibility.
- We outline future and refinement directions of LLMs and tuning method categories.

2 Evidence

2.1 Number of Papers

The number of recently published papers highlights emerging trends in the field under this survey. Figure 2 illustrates the yearly progression in the number of papers published in the ACL Anthology[1] that include keywords related to LLM tuning in their titles. As shown in Figure 2, there has been a noticeable year-over-year increase in papers featuring the terms "tuning" and "black" in their titles. This trend suggests a growing interest in tuning methods and black-box models in recent years.

The rise of high-performance LLMs has likely driven a significant increase in research focused on tuning LLMs with inaccessible internals. Notably, the number of papers featuring both keywords has surged dramatically from 2022 to 2023. This trend is likely influenced by OpenAI's release of ChatGPT, an LLM that restricts access to its parameters, at the end of 2022. As more LLMs with inaccessible internals become available, research on tuning methods for these models is expected to continue advancing in the near future.

2.2 Model Development

LLMs originated from PLMs as small white-box recurrent models (Peters et al., 2018, ELMo) and quickly shifted to the Transformer structure, e.g. BERT (Devlin et al., 2019) and early GPT series. These Transformer-based models gradually evolved into leading LLMs. With the recent advancements in LLMs, a growing trend toward reduced accessibility to their internals is obvious.

White-box model. Full internal access enables backpropagation (Rumelhart et al., 1988). Many white-box models are available on open communities, like HuggingFace[2]. Representative examples are OPT (Zhang et al., 2022a) and llama series (Dubey et al., 2024) besides aforementioned PLMs.

Black-box model. Forbidding any access to the model's internal, the only information the user can utilize is the input text and the corresponding output text. The examples of black-box models include Gemini[3] and Grok[4].

Grey-box model. This category disallow access to parameters but permits access to other parts of the models. Specifically, a grey-box model refers to a model where certain components, like log probabilities or input embeddings, are accessible. GPT-3.5 with later series from OpenAI[5] and Jurassic-2 series from AI21 Labs[6] are examples of grey-box models because they disclose log probabilities.

3 Preliminary

This section introduces the techniques employed in the tuning approaches discussed in this paper.

[1] https://aclanthology.org/

[2] https://huggingface.co/

[3] https://gemini.google.com/

[4] https://help.x.com/en/using-x/about-grok

[5] Noticeably, limited fine-tuning is available. See https://platform.openai.com/docs/guides/fine-tuning.

[6] https://www.ai21.com

3.1 In-context Learning

ICL is a form of ability where information about downstream tasks is incorporated into the input text, allowing an LLM to be tuned without altering its parameters. This information, known as a prompt, may include task explanations and examples of input text paired with the expected output.

A challenge with ICL is that the prompt greatly affects performance. Crafting prompts that yield high performance demands substantial effort and specialized expertise (Jiang et al., 2022; Reynolds and McDonell, 2021; Zamfirescu-Pereira et al., 2023). Tuning approaches that utilize ICL seek to automatically generate and optimize these prompts.

Chain of Thought. Chain of Thought (CoT) (Wei et al., 2023) is a type of ICL. CoT involves adding demonstrations that include the key rationale behind the thought process to the prompt. This rationale enables an LLM to perform step-by-step reasoning, allowing it to tackle complex tasks, such as arithmetic problems, with high accuracy.

A setting where a few rationale-included demonstrations are added is called Few-shot CoT. In contrast, Kojima et al. (2023) proposed Zero-shot CoT, which requires no demonstrations. Zero-shot CoT achieves the CoT approach by prompting the LLM to generate the reasoning process independently. Specifically, Zero-shot CoT effectively guides the LLM to produce both the final answer and the rationale behind it simply by adding a self-motivating phrase "Let's think step by step." to the prompt.

3.2 Derivative Free Optimization

Derivative-Free Optimization (DFO) is a technique for searching for the optimal solution without using gradient information. Since DFO can be performed without accessing the model's parameters, it is well-suited as a tuning technique for LLMs with inaccessible parameters, such as black- and grey-box models. DFO encompasses a variety of approaches, with notable examples including genetic algorithm (GA) (Hansen et al., 2003) and bayesian optimization (BO) (Shahriari et al., 2016).

Genetic algorithm. GA is an optimization technique that searches for better solutions by retaining superior genes for subsequent generations, resembling biological evolution. Initially, a set of candidate solutions is created and evaluated. Only those of high performance are retained for the next generation (i.e., the next iteration). New candidates are then generated based on these retained candidates with mutation. By continuously evaluating the newly generated candidates and repeatedly preserving the superior ones for generations, the algorithm progressively explores and converges on the optimal solution.

Bayesian optimization. BO is a technique for searching for the optimal solution via evaluation and trials. It updates a probabilistic model to prioritize trials that are likely to yield high performance, thereby efficiently exploring the solution space.

The process works as follows: initially, a few data points (e.g., model inputs) are evaluated using an objective function (e.g., task performance). Based on these initial evaluations, a predictive model is constructed to estimate the objective function values for data within the search space. The probabilistic model and the predictive model then estimate and evaluate new data points that are likely to deliver high performance. These models are continuously updated and optimized based on the results of each evaluation. Through this iterative process, BO progressively explores and identifies the optimal solution.

4 Tuning Methods

Figure 3 provides an overview of the tuning methods explored in this paper. This paper primarily focuses on surveying tuning methods that are particularly useful for LLMs with internal accessibility.

4.1 White-box Tuning

White-box tuning is a genre that involves updating a model's internal parameters. These approaches calculate gradients using supervised data and optimizes the parameters through backpropagation.

4.1.1 Full Parameter Tuning

Full parameter tuning is a tuning approach used for white-box models, where all internal parameters of the model are updated. The most common technique under this approach is fine-tuning, which involves adjusting all the model's parameters to optimize performance on a specific task.

4.1.2 Parameter-Efficient Fine-Tuning

Balne et al. (2024) explored an efficient tuning approach that functions independently of the LLM. This approach, known as parameter-efficient fine-tuning (PEFT), aims to achieve improvement via

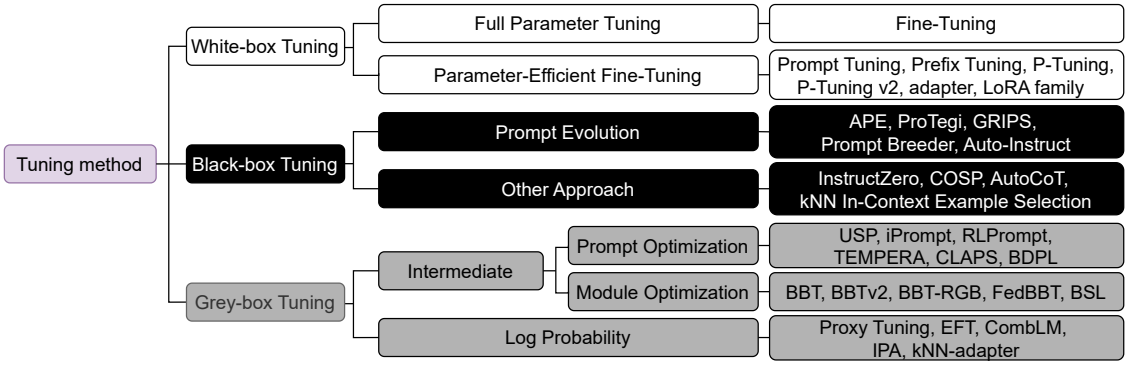

Figure 3: Our classification of tuning methods is based on the internal accessibility of LLMs. We further extends the three root categories into subgroups regarding the features of their subjected approaches.

minimal extra parameter updates. PEFT is beneficial for reducing the substantial computational costs associated with full-model tuning of LLMs.

A notable study in PEFT is prompt tuning (Lester et al., 2021). Prompt tuning involves refining an LLM by adding and optimizing a sequence of vector tokens, called a soft prompt, within the input embedding. During tuning, the LLM's parameters remain unchanged, while only the small set of parameters associated with the soft prompt are updated. At inference, the LLM treats the optimized soft prompt as a continuous-valued prompt. Similarly, P-tuning (Liu et al., 2022b) is another technique focused on optimizing continuous prompts.

Other approaches, like prefix tuning (Li and Liang, 2021) and P-tuning v2 (Liu et al., 2022c), extend the strategy by adding and optimizing sequences of vectors not just in the input embedding, but also at every layer of the LLM. In contrast, adapter tuning (Houlsby et al., 2019) inserts optimizable modules between LLM modules.

Low-Rank Adaptation (LoRA) (Hu et al., 2021) is a prominent tuning method within PEFT. LoRA focuses on learning the extent of change in the model's parameters before and after tuning. By applying matrix decomposition, it reduces the computation to a lower-dimensional space, thereby lowering computational costs. During inference, these parameter updates are incorporated into the linear layer of the model. Additionally, various studies have introduced LoRA variants, such as approaches that dynamically learn the rank for low-rank matrices (Zhang et al., 2023a; Valipour et al., 2023) and methods aimed at further reducing computational costs (Dettmers et al., 2023; Kim et al., 2024).

4.2 Black-box Tuning

Black-box tuning refers to optimization methods applied to LLMs without any internal access and necessarily relies on models' ICL capacity. In these scenarios, the only available information consists of the input sentences and their corresponding output sentences. Specifically, only the input sentences can be directly manipulated. This section discusses approaches that optimize input sentences based on feedback derived from the output sentences or other external information sources.

4.2.1 Prompt Evolution

Prompt evolution evolves prompts via GA. A LLM initially generates multiple candidate prompts, and the high-performing ones are selected. A new set of candidate prompts is then generated based on these selected prompts. This cycle of generating and selecting high-performance prompts is repeated iteratively, gradually refining the prompts to enhance performance.

Auto Prompt Engineer (APE) (Zhou et al., 2023b) utilizes the prompt evolution approach. In APE, candidate prompts are generated using a combination of labeled data and a meta-prompt designed for generating candidates. These prompts are evaluated on metrics, such as accuracy, to identify the most effective ones. The selected prompts are then rephrased by the LLM to generate a new set of prompts, continuing the iterative evolution process.

In the prompt evolution approach, innovations often focus on how candidates are generated and evaluated. ProTeGi (Pryzant et al., 2023) is a study that represents gradients in natural language and utilizes them to optimize prompts. In ProTeGi, the LLM generates the shortcomings of a prompt as a

natural language "gradient". The LLM then modifies the prompt based on each identified "gradient". These modified prompts are added to the candidate set and evaluated their performance. By iterative identification of prompt shortcomings, modifications, and selection of high-performing prompts, the prompts are gradually evolved.

Gradient-free Instructional Prompt Search (GRIPS) (Prasad et al., 2023) optimizes prompts by iteratively breaking them down into phrases and updating these phrases across multiple rounds. During each round, it performs phrase-level updates, retaining only the prompts that demonstrate effective improvements to carry forward to the next iteration. Another study, PromptBreeder (Fernando et al., 2024), adopts an approach that achieves high performance by simultaneously optimizing both the prompts and the meta-prompts used for generating candidate prompts. Auto-Instruct (Zhang et al., 2023b) evaluates candidate prompts using a fine-tuned white-box model, allowing for more accurate identification of effective prompts.

4.2.2 Other Approaches

There are various techniques for tuning prompts beyond the use of GA. For example, InstructZero (Chen et al., 2024) employs BO within DFO techniques. InstructZero optimizes a soft prompt using BO, and the optimized soft prompt is converted into a natural language prompt by a white-box model before being input into a black-box model.

Consistency-based Self-adaptive Prompting (COSP) (Wan et al., 2023a) is a study for generating high-performance CoT demonstrations using only unlabeled data. Initially, Zero-shot CoT is applied to the unlabeled data with a non-zero temperature, generating multiple answers and their corresponding rationales. COSP evaluates these answers using self-consistency (Wang et al., 2023), which involves taking a majority vote among the multiple answers to assess the confidence level of the LLM outputs. The most frequent answer is adopted as the final answer, and the proportion of this answer is used as a measure of the LLM's confidence. In COSP, Few-shot CoT is then executed using examples that have high-confidence answers.

Another approach involves selecting appropriate demonstrations from a dataset and adding them to the prompt. The kNN in-context example selection method (Liu et al., 2022a) constructs a high-performance demonstration set by selecting examples from the training data that are similar to the test examples. Auto-CoT (Zhang et al., 2022c) takes a different approach by clustering the unlabeled data and selecting a diverse set of demonstrations. Using Zero-shot CoT, answers and rationales are then generated and added to the selected demonstration set, forming a comprehensive prompt.

4.3 Grey-box Tuning

Grey-box tuning applies to LLMs whose activations are available. Apart from input text, accessible components may include each layer and the probability distribution during generation. In this paper, grey-box tuning approaches are divided into two categories: manipulating the input text or layers (as intermediate) or log probabilities.

4.3.1 Tuning via Intermediate

This umbrella concept covers two genres: one for optimizing prompts and the other for optimizing modules. Grey-box tuning offers greater flexibility in its application to tasks compared to black-box tuning. For black-box models, many tuning approaches rely solely on information from the output text to optimize prompts. In contrast, grey-box tuning can provide a more comprehensive evaluation of prompts by accessing log probabilities.

Prompt optimization. Prompt optimization is a tuning category that involves updating prompts. A notable example of grey-box tuning methods is Universal Self-Adaptive Prompting (USP) (Wan et al., 2023b). USP is an enhancement of COSP, discussed in Section 4.2.2, making it applicable to a broader range of tasks. USP adjusts the evaluation metrics based on the specific type of task to effectively evaluate the prompts. COSP relies on self-consistency, limiting its application to tasks where the output can be determined by a majority vote, such as classification tasks or arithmetic problems. In contrast, USP extends this approach to generative tasks by incorporating evaluation metrics based on log probabilities, allowing it to be used in a wider variety of contexts.

Since grey-box tuning does not allow access to internal parameters, some approaches utilize DFO, similar to those used in black-box models. One such study is iPrompt (Singh et al., 2023), which employs GA. In iPrompt, the LLM generates explanations of patterns found in the dataset, which are then used as prompts. The LLM is provided with several pieces of labeled data and generates explanations of the data patterns based on these

examples. Like prompt evolution approach, the effectiveness of these data explanations as prompts is then evaluated, and only the high-performing prompts are retained for further use.

In addition to DFO, some studies utilize reinforcement learning for prompt optimization. RLPrompt (Deng et al., 2022) generates prompts using words selected by a policy that selects optimal words from model's vocabulary. Another study, Test-tiMe Prompt Editing using Reinforcement leArning (TEMPERA) (Zhang et al., 2022b), learns a policy to determine which edits (e.g., deletion or swapping of phrases) to apply to the prompt. Unlike other approaches, TEMPERA achieves high performance by generating input-specific prompts, making them more effective.

Other studies focusing on input manipulation include Clustering and Pruning for Efficient Black-box Prompt Search (CLaPS) (Zhou et al., 2023a), which identifies impactful tokens and explores their combinations to optimize prompts. Black-box Discrete Prompt Learning (BDPL) (Diao et al., 2023), uses reinforcement learning to calculate gradients and optimize discrete prompts without access to the model's internal parameters.

Module optimization. We introduce grey-box tuning studies for scenarios where both the input and the internal layers of an LLM are accessible. Module optimization is a technique that focuses on optimizing modules added to the embeddings or layers of an LLM without accessing its parameters. There are two approaches within module optimization: (1) optimizing continuous-value prompts as modules and adding them before the input embeddings, (2) optimizing vector sequences as modules and incorporating them at each layer of the LLM.

A representative study of the approach that adds continuous-value prompts before input embeddings is Black-Box Tuning (BBT)[7](Sun et al., 2022b), which optimizes these prompts using evolutionary strategies. During inference, continuous-value prompts is added before the input text. In essence, BBT achieves a result similar to prompt tuning, as explained in Section 4.1.2, but without accessing the internal parameters. However, black-box models do not allow access to input embeddings and only accept natural language inputs, making BBT inapplicable. Another approach, similar to BBT, is FedBPT (Sun et al., 2023), which optimizes continuous-value prompts using federated learning (McMahan et al., 2023) to protect data privacy while tuning.

There are also approaches that add optimized vector sequences to each layer of LLMs, whereas BBT and FedBPT insert continuous-value prompts into the input text. In other words, the latter achieve effects similar to P-Tuning v2, as described in Section 4.1.2, but without accessing internal parameters. BBTv2 (Sun et al., 2022a), a derivative of BBT, optimizes vector sequences for each layer of the LLM using DFO. During inference, these optimized vector sequences are added to each layer of the LLM. Other studies, such as BBT-RGB (Sun et al., 2024b) and Black-box Prompt Tuning with Subspace Learning (BSL) (Zheng et al., 2024), also employ DFO to add optimized vector sequences to each layer, enhancing the LLM's performance without requiring access to its internal parameters.

4.3.2 Tuning via Log Probability

Grey-box LLMs can be refined not only through intermediate-based approaches but also by directly adjusting log probabilities. The task knowledge acquired by one tuned model is transferable to another general LLM via these log probabilities during inference.

Specifically, the changes in log probabilities after tuning a small white-box model can be transferred to a grey-box model during inference. Proxy-tuning (Liu et al., 2024a) is one such grey-box tuning technique. It starts with fine-tuning a white-box model on a specific downstream task. Then, the differences in log probabilities before and after tuning are calculated. Finally, these differences are applied to the log probabilities of the grey-box model, effectively transferring the learned task knowledge.

There are other studies that focus on the log probabilities of the tuned white-box model. Emulated Fine-Tuning (EFT) (Mitchell et al., 2023) adds the ratio (rather than the difference) of log probabilities before and after tuning to the log probabilities of the grey-box model. CombLM (Ormazabal et al., 2023) calculates the average or weighted sum of the log probabilities after tuning a white-box model and those of the grey-box model and performs inference based on these combined probabilities. Additionally, there are studies such as kNN-adapter (Huang et al., 2023), which manipulates log probabilities by referencing data similar to test examples within the training data. Furthermore, Inference-time Policy Adapters (IPA) (Lu et al., 2023), which

[7]This is a method's name and should not be confused with the meaning of the title for Section 4.2.

integrate policies learned through reinforcement learning in smaller language models into LLMs, can be considered one technique of transferring task knowledge between models of different sizes.

5 Discussion

5.1 The Cost of Tuning

This section examines the costs of tuning LLMs. White-box tuning requires more computational cost than inference when learning internal parameters. White-box tuning for LLMs demands substantial computational resources like GPUs, often requires multiple high-end GPUs, which are both expensive and scarce. For instance, when fine-tuning a model that has 175B parameters, such as GPT-3 (Brown et al., 2020), 1.2TB VRAM is required (Hu et al., 2021). We need to prepare massively GPUs to satisfy the VRAM requirements and a large amount of monetary expenses. Specifically, 38 NVIDIA V100 32GB GPUs ($4,000 USD per GPU[8]) are required for fine-tuning the 175B model and $152,000 USD is required in total. Using LoRA reduces this to 11 GPUs and a cost of around $44,000 USD. However, even with PEFT, tuning LLMs still incurs significant costs. The costs of local white-box tuning not only include the price of the GPUs but also the power consumption during computation.

Alternatively, instead of setting up private GPU servers, one can opt to rent and pay based on usage. The cost of Azure virtual machines[9] increases with the duration of usage. Black- and grey-box models can also be tuned on the LLM provider's servers, meaning that users do not need to prepare their own GPUs. Instead, the cost of tuning these LLMs is tied to the API usage, depending on the number of input and output tokens. Many black- and grey-box tuning approaches can be executed with inference only, so specifically for tuning GPT-4o, the cost is $5 USD per 1 million input tokens and $15 USD per 1 million output tokens. Tasks with extensive training data or generate many output tokens can lead to significant expenses. Advancements in tuning studies could help reduce the costs associated with training and running black- and grey-box models. Minimizing the number of input and output tokens could be a valuable contribution to research in tuning studies.

However, the cost of tuning is expected to decrease over time. One reason for this is the commercial competition among LLM providers. In July 2024, OpenAI introduced GPT-4o-mini, which offered much lower costs than existing LLMs while still maintaining high performance. This competition is likely to intensify, driving not only advancements in model performance but also reductions in usage costs.

Another reason is the advancement of Green AI (Schwartz et al., 2019). Green AI refers to environmentally friendly AI that focuses on creating efficient algorithms and hardware with lower power consumption. As Green AI continues to progress, it is expected that both LLM users and providers will benefit from lower operational computing costs. Lower energy consumption will also help to decrease the costs associated with tuning LLMs.

5.2 The Impact of Disclosing Model's Internal

By revealing the model's internals, a broader range of tuning approaches becomes possible. In white-box models, techniques that update parameters using gradients can be applied. Grey-box models can leverage log probabilities, allowing for the use of diverse loss functions.

However, from the providers' perspective, publishing LLM's internal also has its disadvantages. One major concern is the risk of the LLM's internal information being compromised or stolen.

Existing research has explored techniques to infer internal information from models. Fredrikson et al. (2015) demonstrated that it is possible to infer the data used for training based on the model's gradients. Additionally, Carlini et al. (2024) proposed a technique to identify specific details about an LLM, such as the number of dimensions in the hidden layer, by analyzing log probabilities.

There is a trade-off between model flexibility and the risk of information theft. Greater flexibility makes models accessible to more users, but models trained on sensitive data (e.g., private data) or LLMs that are costly to develop must be cautious about the potential theft of internal information from both security and commercial perspectives.

5.3 Further Model Development

New deep neural network architectures have been gaining attention in recent years, which may influence applicable tuning techniques. For example, Kolmogorov-Arnold Networks (KAN) (Liu et al., 2024c,b) was introduced as a new network structure to replace MLP (Cybenko, 1989; Hornik et al.,

[8] As of October 2024.

[9] https://azure.microsoft.com/en-us/pricing/details/virtual-machines/windows/

1989). Unlike traditional models, KAN does not use linear layer weights but instead learns nonlinear layers. Consequently, white-box tuning techniques like LoRA, which modify linear layers, cannot be applied to LLMs using KAN.

The more a LLM's internals are accessed during tuning, the more vulnerable it becomes to changes in the LLM's structure, and the higher the implementation costs. Grey-box tuning, which requires minimal access to the model's internals, is more robust to changes in the LLMs' architecture. Black-box tuning, which does not access the internal at all, is even more robust. As alternative architectures, such as Mamba (Gu and Dao, 2024), are being explored, research into black- and grey-box tuning studies is becoming increasingly important.

5.4 Refinement of Tuning Methods

Diversity of outputs in black-box tuning. In the black-box tuning studies reviewed in this paper, many approaches involve repeatedly generating candidate prompts and selecting the optimal one (Prompt evolution is discussed in Section 4.2.1). A key challenge in these approaches is the diversity of the prompt candidates generated by the LLMs. These approaches assumes that effective prompts exist within the pool of candidate prompts (i.e., the prompt search space). The breadth of this search space depends on the diversity of the prompts generated. If the LLM's output diversity is low, the candidate prompts will be too similar to one another, reducing the chance of finding effective prompts within the search space. To comprehensively explore prompt representations, it is necessary for the LLM to have high output diversity.

Several studies aim to increase the diversity of LLM outputs. Auto-Instruct prepares seven meta-prompts to generate candidate prompts, with prompt candidates generated using random meta-prompts. For future development in this approach, enhancing output diversity is expected to become more important (Vijayakumar et al., 2018; Lahoti et al., 2023), as well as evaluating diversity (Li et al., 2016; Zhu et al., 2018; Shen et al., 2019).

Input-dependent approach. The future of research in prompt optimization includes developing methods that automate the creation of input-dependent prompts, such as TEMPERA. Wu et al. (2022) highlight the effectiveness of generating distinct prompts for each input sentence.

However, much of the current research tends to rely on fixed prompts for each task, with input-dependent techniques being relatively uncommon. Existing tuning studies that optimize prompts still have room for improvement when it comes to adapting prompts based on the input text.

Knowledge transfer methods. Recently, approaches that transfer task knowledge acquired from a small white-box model to large grey-box models become more prevalent. Examples of such approaches include Proxy-tuning, EFT, and CombLM. The feature of these approaches is their ability to enhance the performance of large grey-box models using task knowledge from a smaller white-box model. A key advantage of the approach is that they require only minimal computational cost during tuning.

In practice, proxy-tuning has produced results that are nearly as effective as directly tuning an LLM. For instance, in a Question-Answering task, directly tuning Llama-2 70B resulted in an accuracy of 63.1, while transferring knowledge from tuning Llama-2 7B to Llama-2 70B achieved a close accuracy of 62.7. This demonstrates that by training the smaller 7B model, it is possible to achieve results comparable to those obtained from training the larger 70B model, highlighting the parameter efficiency of the knowledge transfer approach.

However, this approach is only feasible when the source and target models are similar. For example, in proxy-tuning, it is crucial that the models share a common vocabulary between their tokenizers.

The limitation that knowledge transfer can only be applied between models of the same type poses a challenge for this approach. Developing techniques that enable knowledge transfer between models of different types is becoming increasingly important.

6 Conclusion

This paper surveys tuning studies and classify them by model category. The model category is based on the accessibility of their internals: white-box models, which allow full access to internal parameters; black-box models, which allow access to only the input and output; and grey-box models, which offer partial access to their internals.

Based on trends observed in the surveyed studies, we identify challenges and considerations for future research on tuning techniques. We aim to engage in more detailed discussions by comparing the performance and costs of various tuning approaches.

Acknowledgment

This work was partly supported by JST, PRESTO Grant Number JPMJPR2366, Japan.

References

Charith Chandra Sai Balne, Sreyoshi Bhaduri, Tamoghna Roy, Vinija Jain, and Aman Chadha. 2024. Parameter efficient fine tuning: A comprehensive analysis across applications. *Preprint*, arXiv:2404.13506.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared 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 M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Nicholas Carlini, Daniel Paleka, Krishnamurthy Dj Dvijotham, Thomas Steinke, Jonathan Hayase, A. Feder Cooper, Katherine Lee, Matthew Jagielski, Milad Nasr, Arthur Conmy, Itay Yona, Eric Wallace, David Rolnick, and Florian Tramèr. 2024. Stealing part of a production language model. *Preprint*, arXiv:2403.06634.

Lichang Chen, Jiuhai Chen, Tom Goldstein, Heng Huang, and Tianyi Zhou. 2024. InstructZero: Efficient instruction optimization for black-box large language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 6503–6518. PMLR.

George V. Cybenko. 1989. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2:303–314.

Mingkai Deng, Jianyu Wang, Cheng-Ping Hsieh, Yihan Wang, Han Guo, Tianmin Shu, Meng Song, Eric Xing, and Zhiting Hu. 2022. RLPrompt: Optimizing discrete text prompts with reinforcement learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3369–3391, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLora: Efficient finetuning of quantized llms. *Preprint*, arXiv:2305.14314.

Gilad Deutch, Nadav Magar, Tomer Bar Natan, and Guy Dar. 2024. In-context learning and gradient descent revisited. *Preprint*, arXiv:2311.07772.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. *Preprint*, arXiv:1810.04805.

Shizhe Diao, Zhichao Huang, Ruijia Xu, Xuechun Li, Yong Lin, Xiao Zhou, and Tong Zhang. 2023. Black-box prompt learning for pre-trained language models. *Preprint*, arXiv:2201.08531.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. 2024. A survey on in-context learning. *Preprint*, arXiv:2301.00234.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic,

Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzmán, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vítor Albiero, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. 2024. The Llama 3 herd of models. *Preprint*, arXiv:2407.21783.

William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch Transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Preprint*, arXiv:2101.03961.

Chrisantha Fernando, Dylan Sunil Banarse, Henryk Michalewski, Simon Osindero, and Tim Rocktäschel. 2024. Promptbreeder: Self-referential self-improvement via prompt evolution.

Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. 2015. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, CCS '15, page 1322–1333, New York, NY, USA. Association for Computing Machinery.

Albert Gu and Tri Dao. 2024. Mamba: Linear-time sequence modeling with selective state spaces. *Preprint*, arXiv:2312.00752.

Nikolaus Hansen, Sibylle D. Müller, and Petros Koumoutsakos. 2003. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evolutionary Computation*, 11(1):1–18.

K. Hornik, M. Stinchcombe, and H. White. 1989. Multilayer feedforward networks are universal approximators. *Neural Netw.*, 2(5):359–366.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. *Preprint*, arXiv:1902.00751.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Yangsibo Huang, Daogao Liu, Zexuan Zhong, Weijia Shi, and Yin Tat Lee. 2023. kNN-Adapter: Efficient domain adaptation for black-box language models. *Preprint*, arXiv:2302.10879.

Ellen Jiang, Kristen Olson, Edwin Toh, Alejandra Molina, Aaron Donsbach, Michael Terry, and Carrie J Cai. 2022. PromptMaker: Prompt-based prototyping with large language models. In *Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI EA '22, New York, NY, USA. Association for Computing Machinery.

Hwichan Kim, Shota Sasaki, Sho Hoshino, and Ukyo Honda. 2024. A single linear layer yields task-adapted low-rank matrices. *Preprint*, arXiv:2403.14946.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2023. Large language models are zero-shot reasoners. *Preprint*, arXiv:2205.11916.

Preethi Lahoti, Nicholas Blumm, Xiao Ma, Raghavendra Kotikalapudi, Sahitya Potluri, Qijun Tan, Hansa Srinivasan, Ben Packer, Ahmad Beirami, Alex Beutel, and Jilin Chen. 2023. Improving diversity of demographic representation in large language models via collective-critiques and self-voting. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10383–10405, Singapore. Association for Computational Linguistics.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California. Association for Computational Linguistics.

Xiang Lisa Li and Percy Liang. 2021. Prefix-Tuning: Optimizing continuous prompts for generation. In *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)*, pages 4582–4597, Online. Association for Computational Linguistics.

Alisa Liu, Xiaochuang Han, Yizhong Wang, Yulia Tsvetkov, Yejin Choi, and Noah A. Smith. 2024a. Tuning language models by proxy. *Preprint*, arXiv:2401.08565.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022a. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022b. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 61–68, Dublin, Ireland. Association for Computational Linguistics.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Lam Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022c. P-Tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *Preprint*, arXiv:2110.07602.

Ziming Liu, Pingchuan Ma, Yixuan Wang, Wojciech Matusik, and Max Tegmark. 2024b. KAN 2.0:

Kolmogorov-arnold networks meet science. *Preprint*, arXiv:2408.10205.

Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljačić, Thomas Y. Hou, and Max Tegmark. 2024c. KAN: Kolmogorov-arnold networks. *Preprint*, arXiv:2404.19756.

Ximing Lu, Faeze Brahman, Peter West, Jaehun Jung, Khyathi Chandu, Abhilasha Ravichander, Prithviraj Ammanabrolu, Liwei Jiang, Sahana Ramnath, Nouha Dziri, Jillian Fisher, Bill Lin, Skyler Hallinan, Lianhui Qin, Xiang Ren, Sean Welleck, and Yejin Choi. 2023. Inference-time policy adapters (IPA): Tailoring extreme-scale LMs without fine-tuning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6863–6883, Singapore. Association for Computational Linguistics.

H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. 2023. Communication-efficient learning of deep networks from decentralized data. *Preprint*, arXiv:1602.05629.

Eric Mitchell, Rafael Rafailov, Archit Sharma, Chelsea Finn, and Christopher D. Manning. 2023. An emulator for fine-tuning large language models using small language models. *Preprint*, arXiv:2310.12962.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. GPT-4 technical report. *Preprint*, arXiv:2303.08774.

Aitor Ormazabal, Mikel Artetxe, and Eneko Agirre. 2023. CombLM: Adapting black-box language models through small fine-tuned models. *Preprint*, arXiv:2305.16876.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Confer-*

ence of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers), pages 2227–2237. Association for Computational Linguistics.

Archiki Prasad, Peter Hase, Xiang Zhou, and Mohit Bansal. 2023. GrIPS: Gradient-free, edit-based instruction search for prompting large language models. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3845–3864, Dubrovnik, Croatia. Association for Computational Linguistics.

Reid Pryzant, Dan Iter, Jerry Li, Yin Tat Lee, Chenguang Zhu, and Michael Zeng. 2023. Automatic prompt optimization with "Gradient Descent" and beam search. *Preprint*, arXiv:2305.03495.

XiPeng Qiu, TianXiang Sun, YiGe Xu, YunFan Shao, Ning Dai, and XuanJing Huang. 2020. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, 63(10):1872–1897.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer. *Preprint*, arXiv:1910.10683.

Laria Reynolds and Kyle McDonell. 2021. Prompt programming for large language models: Beyond the few-shot paradigm. *Preprint*, arXiv:2102.07350.

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. 1988. *Learning representations by back-propagating errors*, page 696–699. MIT Press, Cambridge, MA, USA.

Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. 2019. Green AI. *Preprint*, arXiv:1907.10597.

Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P. Adams, and Nando de Freitas. 2016. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175.

Tianxiao Shen, Myle Ott, Michael Auli, and Marc'Aurelio Ranzato. 2019. Mixture models for diverse machine translation: Tricks of the trade. *Preprint*, arXiv:1902.07816.

Chandan Singh, John X. Morris, Jyoti Aneja, Alexander Rush, and Jianfeng Gao. 2023. Explaining data patterns in natural language with language models. In *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 31–55, Singapore. Association for Computational Linguistics.

Haotian Sun, Yuchen Zhuang, Wei Wei, Chao Zhang, and Bo Dai. 2024a. Bbox-Adapter: Lightweight adapting for black-box large language models. *Preprint*, arXiv:2402.08219.

Jingwei Sun, Ziyue Xu, Hongxu Yin, Dong Yang, Daguang Xu, Yiran Chen, and Holger R. Roth. 2023. FedBPT: Efficient federated black-box prompt tuning for large language models. *Preprint*, arXiv:2310.01467.

Qiushi Sun, Chengcheng Han, Nuo Chen, Renyu Zhu, Jingyang Gong, Xiang Li, and Ming Gao. 2024b. Make prompt-based black-box tuning colorful: Boosting model generalization from three orthogonal perspectives. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 10958–10969, Torino, Italia. ELRA and ICCL.

Tianxiang Sun, Zhengfu He, Hong Qian, Yunhua Zhou, Xuanjing Huang, and Xipeng Qiu. 2022a. BBTv2: Towards a gradient-free future with large language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3916–3930, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Tianxiang Sun, Yunfan Shao, Hong Qian, Xuanjing Huang, and Xipeng Qiu. 2022b. Black-box tuning for language-model-as-a-service. *Preprint*, arXiv:2201.03514.

Gemini Team and Rohan Anil. 2024. Gemini: A family of highly capable multimodal models. *Preprint*, arXiv:2312.11805.

Mojtaba Valipour, Mehdi Rezagholizadeh, Ivan Kobyzev, and Ali Ghodsi. 2023. DyLoRA: Parameter-efficient tuning of pre-trained models using dynamic search-free low-rank adaptation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3274–3287, Dubrovnik, Croatia. Association for Computational Linguistics.

Ashwin K Vijayakumar, Michael Cogswell, Ramprasath R. Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2018. Diverse beam search: Decoding diverse solutions from neural sequence models. *Preprint*, arXiv:1610.02424.

Johannes von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. 2023. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 35151–35174. PMLR.

Xingchen Wan, Ruoxi Sun, Hanjun Dai, Sercan Arik, and Tomas Pfister. 2023a. Better zero-shot reasoning with self-adaptive prompting. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 3493–3514, Toronto, Canada. Association for Computational Linguistics.

Xingchen Wan, Ruoxi Sun, Hootan Nakhost, Hanjun Dai, Julian Eisenschlos, Sercan Arik, and Tomas

Pfister. 2023b. Universal self-adaptive prompting. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7437–7462, Singapore. Association for Computational Linguistics.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. *Preprint*, arXiv:2203.11171.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

Zhuofeng Wu, Sinong Wang, Jiatao Gu, Rui Hou, Yuxiao Dong, V.G.Vinod Vydiswaran, and Hao Ma. 2022. IDPG: An instance-dependent prompt generation method. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5507–5521, Seattle, United States. Association for Computational Linguistics.

J.D. Zamfirescu-Pereira, Richmond Y. Wong, Bjoern Hartmann, and Qian Yang. 2023. Why johnny can't prompt: How non-AI experts try (and fail) to design llm prompts. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, New York, NY, USA. Association for Computing Machinery.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Nikos Karampatziakis, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. 2023a. AdaLoRA: Adaptive budget allocation for parameter-efficient fine-tuning. *Preprint*, arXiv:2303.10512.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022a. OPT: Open pre-trained transformer language models. *Preprint*, arXiv:2205.01068.

Tianjun Zhang, Xuezhi Wang, Denny Zhou, Dale Schuurmans, and Joseph E. Gonzalez. 2022b. TEMPERA: Test-time prompting via reinforcement learning. *Preprint*, arXiv:2211.11890.

Zhengyan Zhang, Xu Han, Hao Zhou, Pei Ke, Yuxian Gu, Deming Ye, Yujia Qin, Yusheng Su, Haozhe Ji, Jian Guan, Fanchao Qi, Xiaozhi Wang, Yanan Zheng, Guoyang Zeng, Huanqi Cao, Shengqi Chen, Daixuan Li, Zhenbo Sun, Zhiyuan Liu, Minlie Huang, Wentao Han, Jie Tang, Juanzi Li, Xiaoyan Zhu, and Maosong Sun. 2020. CPM: A large-scale generative chinese pre-trained language model. *Preprint*, arXiv:2012.00413.

Zhihan Zhang, Shuohang Wang, Wenhao Yu, Yichong Xu, Dan Iter, Qingkai Zeng, Yang Liu, Chenguang Zhu, and Meng Jiang. 2023b. Auto-Instruct: Automatic instruction generation and ranking for black-box language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9850–9867, Singapore. Association for Computational Linguistics.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2022c. Automatic chain of thought prompting in large language models. *Preprint*, arXiv:2210.03493.

Yuanhang Zheng, Zhixing Tan, Peng Li, and Yang Liu. 2024. Black-box prompt tuning with subspace learning. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32:3002–3013.

Han Zhou, Xingchen Wan, Ivan Vulić, and Anna Korhonen. 2023a. Survival of the most influential prompts: Efficient black-box prompt search via clustering and pruning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13064–13077, Singapore. Association for Computational Linguistics.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2023b. Large language models are human-level prompt engineers. *Preprint*, arXiv:2211.01910.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. *Preprint*, arXiv:1802.01886.