RETHINKING PEFT IN LLMS: OPTIMAL LAYER FINE-TUNING FOR EFFICIENT ADAPTATION – AN EMPIRICAL STUDY ON NLU TASKS

Jingheng Pan
M.Sc. Student in Wirtschaftsinformatik
University of Hamburg
jingheng.pan@studium.uni-hamburg.de

ABSTRACT

Large Language Models (LLMs) have achieved strong performance across NLP tasks, but their growing size makes full fine-tuning computationally expensive. While Parameter-Efficient Fine-Tuning (PEFT) methods alleviate this cost by introducing a small number of trainable parameters while freezing the backbone model, existing approaches typically either modify all layers, achieving strong performance at high computational cost, or restrict tuning to the final layer, sacrificing performance for efficiency. We propose Optimal Layer Fine-Tuning (OLFT), a simple and effective strategy that fine-tunes only a single, task-specific layer. Motivated by prior interpretability studies, OLFT restricts candidate layers to the model's middle bucket, which has been shown to be critical for semantic representation and preliminary reasoning. A heuristic approach then selects the optimal layer by evaluating the performance of individually fine-tuned candidates. Experiments on NLU tasks show that OLFT achieves comparable or even superior performance to full-layer tuning while reducing training time and parameter count.

1 INTRODUCTION

In recent years, Large Language Models (LLMs) have demonstrated remarkable performance across a wide range of natural language processing (NLP) tasks (Zhao et al., 2023). However, this impressive performance comes at the cost of significantly increased model size and computational demands. Fine-tuning such models on new tasks can be computationally expensive and resource-intensive due to the large number of parameters involved.

A widely adopted solution to address this challenge is Parameter-Efficient Fine-Tuning (PEFT)(Mangrulkar et al., 2022), which introduces a small number of trainable parameters instead of updating all the parameters in the model. However, most PEFT methods still apply these modules either to all layers or only to the final layer. While full-layer tuning typically yields stronger performance, it still incurs considerable computational cost. Conversely, tuning only the final layer reduces training time and parameter count, but may fail to match the performance of the full-layer tuning. This trade-off raises an important question: Is there a single layer whose fine-tuning alone can match or even surpass the performance of full-layer tuning, while preserving the training cost of single-layer tuning?

CogSteer(Wang et al., 2024) explored the layer-wise functionality of LLMs by dividing the model into three equal groups: premature (first third of layers), middle (middle third), and mature (last third). Through correlation analysis between eye-tracking measures and the hidden states across layers, they found that the layers from middle buckets are strongly associated with semantic processing and preliminary reasoning. These findings point to a promising direction: rather than uniformly tuning all layers, it may be more effective to fine-tune those most functionally aligned with the target task.

In this work, we focus on Natural Language Understanding (NLU) tasks, which typically require semantic comprehension, reasoning, and inference capabilities. Motivated by the functional spe-

cialization of middle layers and the requirements of NLU tasks, we hypothesize that the optimal layer for fine-tuning lies in the middle bucket.

Inspired by this hypothesis and the interpretability findings, we propose **Optimal Layer Fine-Tuning (OLFT)**, a strategy that reduces computational cost by fine-tuning only a single, well-chosen layer. OLFT identifies this optimal layer via a heuristic selection process, focusing on layers in the middle bucket of the model. Despite its simplicity, OLFT achieves comparable or even superior performance to full-layer tuning on NLU tasks, while substantially reducing training time and the number of trainable parameters.

2 METHOD

In this section, we introduce **Optimal Layer Fine-Tuning (OLFT)**, a parameter-efficient strategy designed to reduce the cost of adapting large language models (LLMs) to downstream tasks. OLFT operates by identifying and fine-tuning only a single, task-specific layer, while keeping all other layers frozen. This approach relies on two key components: (1) a lightweight optimal layer selection process that identifies the most effective intervention layer by fine-tuning candidates directly on the validation set, and (2) an optimal layer fine-tuning step that applies PEFT modules exclusively to the selected layer using the full training dataset.

2.1 TRADITIONAL PARAMETER-EFFICIENT FINE-TUNING

Traditional parameter-efficient fine-tuning (PEFT)(Mangrulkar et al., 2022) methods, such as Bottleneck Adapters(Houlsby et al., 2019; Pfeiffer et al., 2020) and Prefix-Tuning(Li & Liang, 2021), are widely used for semantic adaptation. Bottleneck Adapters are "plug-in" modules that insert lightweight trainable components into the model, connected residually to the original architecture. Prefix-Tuning is a reparameterization-based method that prepends a sequence of trainable vectors to the attention mechanism of each transformer layer.

In existing PEFT methods, these modules are typically applied either to all layers or only the final layer. Full-layer tuning improves performance but demands significant computational resources. Tuning only the final layer is more efficient, but may not yield optimal task performance.

2.2 EFFICIENT OPTIMAL LAYER SELECTION AND INTERVENTION

Based on the findings of CogSteer(Wang et al., 2024), which suggest that layers in the middle bucket of LLMs are particularly involved in token processing, information integration, and preliminary reasoning, we hypothesize that for NLU tasks, an optimal single layer exists within the middle bucket. Fine-tuning this layer alone could achieve a favorable balance between performance and efficiency: preserving much of the benefit of full-layer tuning while maintaining the resource efficiency of final-layer tuning.

To validate this hypothesis, we propose **Optimal Layer Fine-Tuning (OLFT)**, a simple yet effective strategy that identifies and fine-tunes only one optimal layer, while freezing all others. Specifically, we adopt a heuristic approach with two stages:

- **Optimal Layer Selection**: Each candidate layer within the middle bucket is fine-tuned independently by training on a small portion of data, like validation set. The performance of each layer is evaluated, and the layer achieving the best results is selected as the optimal intervention layer.
- **Optimal Layer Fine-Tuning**: We then fine-tune the model by applying the PEFT module exclusively to the selected optimal layer, using the full training dataset.

Despite intervening on only a single layer, OLFT enables effective task-specific adaptation. Additionally, since only 1/N of the model's layers are updated (where N is the total number of layers), both the number of trainable parameters and the computational cost are substantially reduced.

3 EXPERIMENT

We focus on evaluating OLFT in Natural Language Understanding (NLU) tasks, which demand strong semantic comprehension and reasoning abilities. These requirements align with the functional characteristics of middle layers observed in prior interpretability studies.

3.1 DATASET AND EVALUATION

To evaluate the performance of OLFT, we employ the GLUE(General Language Understanding Evaluation) benchmark(Wang et al., 2019), a widely used collection of tasks for assessing NLU systems. Specifically, we evaluate on eight standard GLUE tasks:

- **SST-2 (The Stanford Sentiment Treebank)**: A binary sentiment classification task (positive vs. negative). The validation set contains 872 examples, and the test set contains approximately 1.8k examples.
- **MRPC (Microsoft Research Paraphrase Corpus)**: A paraphrase identification task that requires determining whether two sentences are semantically equivalent. The validation set has 400 examples, and the test set has about 1.7k examples.
- **QQP (Quora Question Pairs)**: Another paraphrase identification task where the goal is to decide if two questions are semantically equivalent. The validation set has 40k examples, and the test set has around 391k examples.
- **MNLI (Multi-Genre Natural Language Inference)**: Given a premise and a hypothesis, the model must predict whether the premise entails, contradicts, or is neutral with respect to the hypothesis. There are two validation sets (MNLI-M and MNLI-MM), each with about 10k examples, and two corresponding test sets of the same size.
- **QNLI (Question Natural Language Inference)**: A task reformulated from SQuAD, where the goal is to determine if a context sentence contains the answer to a given question. Both the validation and test sets have about 5.5k examples.
- **RTE(Recognizing Textual Entailment)**: A binary entailment task asking whether a premise logically entails a hypothesis. The validation set contains 277 examples, and the test set has around 3k examples.
- **WNLI (Winograd NLI)**: Based on the Winograd Schema Challenge, this task involves resolving ambiguous pronouns. Models must determine if substituting the pronoun in a sentence preserves its meaning. The validation set has 71 examples, and the test set has 146 examples.

Each task includes training, validation, and test splits. We report F1 scores for QQP and MRPC, and accuracy for all other tasks.

3.2 MODEL AND BASELINE

We evaluate OLFT on three representative large language models (LLMs) with diverse architectures:

- **GPT-2 Large**: A 774M-parameter autoregressive transformer model from the GPT-2 family(Radford et al., 2019), trained on large-scale web corpora for open-domain text generation.
- **LLaMA2-7B**: A 7-billion-parameter decoder-only model developed by Meta AI(Touvron et al., 2023), trained on publicly available datasets and optimized for both generation and understanding tasks.
- **Mistral-7B**: A more recent open-source dense transformer model with 7 billion parameters from Mistral AI(Jiang et al., 2024), featuring architectural optimizations such as sliding window attention to improve efficiency and scalability.

These models are selected to span different parameter scales and architectural generations: from early-generation models like GPT-2 Large, to widely adopted modern models like LLaMA2-7B,

and to recently optimized architectures like Mistral-7B, allowing us to comprehensively evaluate the effectiveness and generalizability of OLFT.

As baselines, we consider two commonly used parameter-efficient fine-tuning (PEFT) configurations:

- Final-layer tuning: PEFT modules are applied exclusively to the final layer of the model.
- Full-layer tuning: PEFT modules are inserted into all layers.

These two baselines represent the two ends of the efficiency-performance trade-off spectrum: final-layer tuning prioritizes efficiency over performance, while full-layer tuning prioritizes performance over efficiency. By comparing against both configurations, we aim to demonstrate that OLFT achieves a favorable balance, maintaining strong task performance while substantially reducing training cost.

3.3 Evaluation on GLUE Benchmark

Model	VAL-SET								
	MNLI-M	**MNLI-MM**	**MRPC**	**QNLI**	**QQP**	**RTE**	**SST-2**	**WNLI**	**Avg.**
GPT2-L	78.0 *l-19*	79.5 *l-19*	85.5 *l-20*	83.3 *l-19*	80.6 *l-19*	71.1 *l-19*	92.9 *l-19*	53.5 *l-19*	78.1
	59.4	60.5	80.4	67.4	72.2	63.9	86.9	53.5	68.0
	82.1	83.5	83.1	85.2	82.6	70.4	93.6	53.5	79.2
LLaMA2-7B	86.4 *l-14*	87.1 *l-14*	86.5 *l-14*	89.3 *l-14*	83.3 *l-14*	75.5 *l-14*	95.8 *l-14*	56.4 *l-19*	82.5
	58.7	61.4	83.3	63.3	80.2	66.4	93.2	56.3	70.4
	89.0	89.3	86.3	91.9	85.6	65.3	96.7	56.3	82.5
Mistral-7B	87.3 *l-12*	88.1 *l-12*	86.9 *l-12*	91.4 *l-14*	84.6 *l-12*	80.1 *l-12*	95.8 *l-14*	56.3 *l-12*	83.8
	76.3	70.6	82.2	78.9	81.6	65.7	93.2	54.9	75.4
	89.5	89.7	82.2	81.7	78.1	58.9	96.7	56.3	79.1
Model	**TEST-SET**								
	MNLI-M	**MNLI-MM**	**MRPC**	**QNLI**	**QQP**	**RTE**	**SST-2**	**WNLI**	**Avg.**
GPT2-L	79.3 *l-19*	79.3 *l-19*	83.0 *l-20*	84.1 *l-19*	65.6 *l-19*	64.6 *l-19*	92.4 *l-19*	58.9 *l-19*	75.8
	59.9	59.9	80.2	67.1	58.6	57.4	87.5	51.4	65.3
	82.6	83.0	82.7	85.6	65.6	62.6	93.5	61.6	77.1
LLaMA2-7B	82.9 *l-14*	86.3 *l-14*	83.4 *l-14*	88.5 *l-14*	68.8 *l-14*	74.7 *l-14*	95.2 *l-14*	64.4 *l-19*	80.5
	58.6	61.0	80.8	62.6	65.4	61.0	91.6	64.4	68.2
	89.5	88.8	80.5	92.1	71.6	58.2	93.5	55.5	78.7
Mistral-7B	87.1 *l-12*	87.5 *l-12*	86.6 *l-12*	91.7 *l-14*	70.5 *l-12*	81.0 *l-12*	95.9 *l-14*	65.8 *l-12*	83.2
	71.0	70.7	80.0	79.4	66.6	62.5	84.7	59.6	71.8
	89.7	89.4	80.4	81.3	62.7	52.3	97.3	65.1	77.3

Table 1: **Evaluation on GLUE Benchmark.** MRPC and QQP are reported using F1, while the other tasks are reported using Accuracy. A green box indicates that the OLFT outperforms the full-layer tuning, an orange box denotes comparable performance, and a blue box indicates slightly lower performance. The gray box indicate the performance of final-layer tuning, and the values without background color correspond to the performance of full-layer tuning.

Table 1 reports the performance of OLFT across eight GLUE tasks. On the test set, both LLaMA2-7B and Mistral-7B outperform the full-layer tuning baseline on average, with Mistral-7B achieving an absolute increase of +5.9 (83.2 vs. 77.3) and LLaMA2-7B achieving an absolute increase of +1.8 (80.5 vs. 78.7). On the validation set, Mistral-7B consistently outperforms full-layer tuning with a +4.7 advantage (83.8 vs. 79.1), while LLaMA2-7B maintains equivalent performance (82.5 vs. 82.5). For GPT2-L, OLFT achieves comparable results on both validation and test sets, with performance gaps of 1.1 (validation) and 1.3 (test) relative to full-layer tuning, corresponding to a difference of less than 2%.

Compared to final-layer tuning, OLFT consistently delivers substantial performance gains across all models. On the test set, OLFT improves the average GLUE score by +10.5 on GPT2-L (75.8 vs. 65.3), +12.3 on LLaMA2-7B (80.5 vs. 68.2), and +11.4 on Mistral-7B (83.2 vs. 71.8). These results confirm that fine-tuning a carefully selected single layer significantly outperforms the final-layer tuning while preserving parameter efficiency, and achieves comparable or even superior performance to full-layer tuning.

A consistent observation is that the optimal layers selected by our heuristic method consistently fall within the middle bucket, which supports the hypothesis proposed in the *Section* 2 that middle layers are particularly involved in token processing, information integration, and preliminary reasoning, which are critical for success on NLU tasks. Moreover, these optimal layers remain stable across both validation and test splits. For instance, GPT2-L selects L19 as optimal for most GLUE tasks (except MRPC: L20). Similarly, LLaMA2-7B consistently selects L14 (except WNLI: L19), and Mistral-7B selects L12 for most tasks (except SST-2 and QNLI: Layer 14). These findings validate the reliability of our layer selection strategy and reveal that the optimal layer is not only model-specific but also task-dependent.

3.4 Efficiency Analysis

Model	Time/min	Time%	Params/M	Params%	Avg. GLUE↑
GPT2-L → *full*	33	100	14.8	100	77.1
→ *single*	13	39.4	0.4	2.7	75.8
LLaMA2-7B → *full*	205	100	1.3	100	78.7
→ *single*	85	41.5	0.04	3.1	80.4
Mistral-7B → *full*	36	100	134.5	100	77.3
→ *single*	25	69.4	4.2	3.1	83.1

Table 2: **Efficiency Comparison between *OLFT* and *Full-layer Tuning*.** A green box indicates that the single-layer tuning outperforms the full-layer tuning, while an orange box denotes comparable performance.

Table 2 presents a comparison between full-layer and single-layer fine-tuning in terms of training time, parameter count, and performance. Across all three models, single-layer fine-tuning substantially reduces training cost down to 39.4% for GPT2-L, 41.5% for LLaMA2-7B, and 69.4% for Mistral-7B, while updating only about 3% of the parameters used in full-layer tuning.

Despite this significant reduction in training cost, the performance remains competitive. As previously shown in GLUE experiments, OLFT achieves comparable or even better results than full-layer tuning on most tasks. This demonstrates that OLFT provides a strong balance between efficiency and effectiveness, making it a practical alternative for low-resource or large-scale fine-tuning scenarios.

4 Conclusion

In this work, we propose Optimal Layer Fine-Tuning (OLFT), a simple yet effective strategy for parameter-efficient adaptation of large language models. Unlike conventional PEFT methods that apply modifications to all or the final layers, OLFT fine-tunes only a single, task-specific layer—typically located in the middle bucket of the model, which prior interpretability studies have identified as critical for semantic representation and reasoning.

To support our hypothesis, we introduce a heuristic layer selection approach that efficiently identifies the optimal layer by fine-tuning each candidate in the middle bucket on validation set. Through experiments on natural language understanding (GLUE benchmark), which require strong semantic comprehension and reasoning abilities, we demonstrate that OLFT not only reduces training cost and the number of trainable parameters, but also achieves comparable or even superior performance to full-layer, while significantly surpassing final-layer tuning.

Our findings further reveal that optimal layers are both model- and task-dependent, yet consistently fall within the middle layers across architectures. These results highlight OLFT as a practical and effective alternative for efficient fine-tuning of large language models.

REFERENCES

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. *ArXiv*, abs/1902.00751, 2019. URL https://api.semanticscholar.org/CorpusID:59599816.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint: 2401.04088*, 2024. URL https://arxiv.org/pdf/2401.04088.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4582–4597, 2021. URL https://api.semanticscholar.org/CorpusID:230433941.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. PEFT: State-of-the-art parameter-efficient fine-tuning methods. https://github.com/huggingface/peft, 2022.

Jonas Pfeiffer, Andreas Rücklé, Clifton A. Poth, Aishwarya Kamath, Ivan Vulic, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. Adapterhub: A framework for adapting transformers. *ArXiv*, abs/2007.07779, 2020. URL https://api.semanticscholar.org/CorpusID:220525782.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 2019. URL https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. LLAMA 2: Open foundation and fine-tuned chat models. *arXiv preprint: 2307.09288*, 2023. URL https://arxiv.org/pdf/2307.09288.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *The Seventh International Conference on Learning Representations (ICLR)*, pp. 1–20, 2019. URL https://openreview.net/pdf?id=rJ4km2R5t7.

Xintong Wang, Jingheng Pan, Liang Ding, Longyue Wang, Longqin Jiang, Xingshan Li, and Christian Biemann. Cogsteer: Cognition-inspired selective layer intervention for efficiently steering large language models. 2024. URL https://api.semanticscholar.org/CorpusID:276421262.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Z. Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jianyun Nie, and Ji rong Wen. A survey of large language models. *ArXiv*, abs/2303.18223, 2023. URL https://api.semanticscholar.org/CorpusID:257900969.