Low-Rank Adaptation with Task-Relevant Feature Enhancement for Fine-tuning Language Models

Changqun Li[1], **Chaofan Ding**[1], **Kexin Luan**[1], **Xinhan Di**[1]

[1]AI Lab, Giant Network

lichangqun@ztgame.com, dingchaofan@ztgame.com, luankexin@ztgame.com, dixinhan@ztgame.com

Abstract

Fine-tuning pre-trained large language models in a parameter-efficient manner is widely studied for its effectiveness and efficiency. LoRA is one of the most widely used methods, which assumes that the optimization process is essentially low dimensional. Although LoRA has demonstrated commendable performance, there remains a significant performance gap between LoRA and full fine-tuning when learning new tasks. In this work, we propose Low-Rank Adaptation with Task-Relevant Feature Enhancement (LoRATRF) for enhancing task-relevant features from the perspective of editing neural network representations. To prioritize task-relevant features, a task-aware filter that selectively extracts valuable knowledge from hidden representations for the target or current task is designed. As the experiments on a vareity of datasets including NLU, commonsense reasoning and mathematical reasoning tasks demonstrates, our method reduces 33.71% parameters and achieves better performance on a variety of datasets in comparison with SOTA low-rank methods.

Introduction

Pre-trained language models (PLMs) have shown remarkable performance across a wide variety of downstream natural language processing tasks through fine-tuning on task-specific labeled data (Kenton and Toutanova 2019; Liu et al. 2019; Lewis et al. 2020). However, fine-tuning all model parameters (*full fine-tuning*) is prohibitively expensive. This issue is particularly salient with the ever-growing size of PLMs (e.g., BERT (Kenton and Toutanova 2019) with 330M parameters and GPT-3 (Brown et al. 2020) with 175B parameters).

To adapt general knowledge in pre-trained models to specific downstream tasks in a more parameter-efficient way, Parameter-Efficient Fine-Tuning (PEFT) methods have been proposed (Houlsby et al. 2019; Pfeiffer et al. 2021; Li and Liang 2021a; Lester, Al-Rfou, and Constant 2021; Ben Zaken, Goldberg, and Ravfogel 2022). For example, *adapter tuning* (Houlsby et al. 2019; Pfeiffer et al. 2021) inserts adapters to each layer of the pre-trained network. Inspired by the success of prompting methods that control PLMs through textual prompts (Brown et al. 2020), *prefix-tuning* (Li and Liang 2021a) and *prompt-tuning* (Lester, Al-Rfou, and Constant 2021) prepend an additional tunable prefix tokens to the input or hidden layers. Then, *LoRA* and its derivatives (Hu et al. 2022; Zhang et al. 2023a; Liu et al. 2024) decomposes the attention weight gradients into low-rank matrices. Some studies (Valipour et al. 2023; Zhang et al. 2023a; Ding et al. 2023) mainly focused on dynamically adjusting the rank of LoRA in different layers. The above methods does not explore task-relevant features from the perspective of editing neural network representations.

In this paper, we take a step towards addressing the performance gap question, by proposing a new PEFT method called Low-Rank Adaptation with Task-Relevant Feature Enhancement (LoRATRF). We propose to enhance task-relevant features from the perspective of editing neural network representations. To prioritize task-relevant features, we introduce a task-aware filter that selectively extracts valuable knowledge from hidden representations for the target or current task. We conduct extensive experiments on a wide range of tasks and models to demonstrate the effectiveness of our method. To sum up, our contributions are:

- We enhance the performance of LoRA from the perspective of neural network editing.
- We design a task-aware filter that can selectively extract valuable knowledge from the hidden representation of the current task, which can enhance the model's focus on crucial features.

Related Work

Parameter-Efficient Fine-Tuning

Parameter-efficient fine-tuning (PEFT) is an approach of optimizing a small number of parameters when fine-tuning a large pre-trained backbone model and keeping the backbone model untouched for adaptation (Han et al. 2024). A branch of PEFT methods (Lester, Al-Rfou, and Constant 2021; Li and Liang 2021b; Liu et al. 2022) is to add some special trainable vectors. Representative works in this direction are Prefix tuning (Li and Liang 2021b), Prompt tuning (Lester, Al-Rfou, and Constant 2021) and P-tuning V2 (Liu et al. 2022). Another approach (Houlsby et al. 2019; Pfeiffer et al. 2021; Zhang et al. 2023b) is to insert additional neural modules to the backbone model, called Adapter. The

reparameterization-based methods have attracted much attention (Hu et al. 2022). This type of PEFT method is closely related to intrinsic dimension (Li et al. 2018; Aghajanyan, Gupta, and Zettlemoyer 2021). However, the cost to train the above methods is large as the amount of trainable parameter is not small enough.

LoRA and Its Variants

LoRA (Hu et al. 2022) is proven to be effective and yield stable results when applied to both relatively small pre-trained models and large language models (Dettmers et al. 2023; Hu et al. 2023). Then, some researchers explore more flexible and appropriate ranks, such as DyLoRA, AdaLoRA, and SoRA. Other works focus on the combination of LoRA and other approaches, such as AdaMix (Wang et al. 2022) and QLoRA (Dettmers et al. 2023). Besides, LoRAHub (Huang et al. 2023) and LoRAMoE (Wu, Huang, and Wei 2024) focus on how to merge multiple LoRA blocks that are fine-tuned on different tasks respectively. However, the above methods lacks efficiency on low-rank representation of complex reasoning tasks for LLMs.

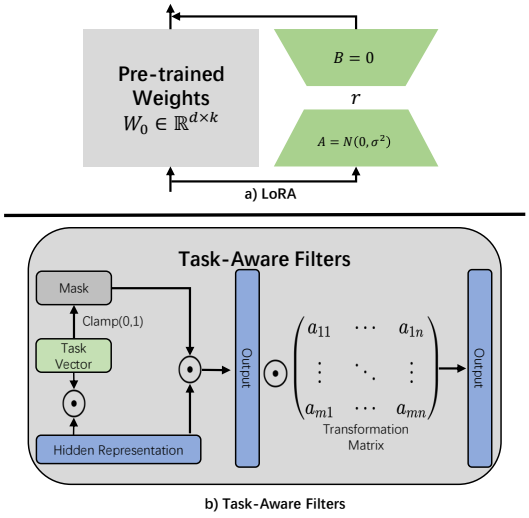

Figure 1: The overview of our approach. Task-aware filter can selectively extract valuable knowledge from hidden representations for the target task. $\odot$ refers to element-wise multiplication.

Methodology

Preliminaries

Low-Rank Adaptation (LoRA). LoRA (Hu et al. 2022) approximates the incremental update by decomposing it into the product of two low-rank matrices, constraining the update to a low-rank space and making the fine-tuning process more efficient. Through this approximate low-rank decomposition, we have:

$$\Delta W \approx BA, \tag{1}$$

where $B \in R^{m\times r}$, $A \in R^{r\times n}$, and the rank $r << min(m, n)$. In the forward propagation of LoRA, for an input representation x, the output after passing through the parameter matrix W_0 is

$$h = W_0 x + \Delta W x = W_0 x + BAx \tag{2}$$

To ensure that introducing LoRA at the initial stage does not impact the computation results of the model's forward propagation, it is crucial to ensure that BAx = 0. To achieve this, LoRA initializes A as a random Gaussian matrix and B as a zero matrix. During training, W_0 is frozen, and B and A are treated as trainable parameters. Upon completion of training, the parameter matrices A and B are merged into W_0 to form the final parameter matrix

$$W_{ft} = W_0 + BA \tag{3}$$

It is noteworthy that the final update is BA, which is constrained within a low-rank space.

Motivation

To further improve performance, we propose Low-Rank Adaptation with Task-Relevant Featur Enhancement (LoRATRF). Figure 1 gives an overview of our approach, where the task-aware filter identifies task-relevant features within hidden representations and adaptively integrates these features back into the representations.

Task-Aware Filters

Firstly, we introduce task-aware filters (Zou et al. 2023) that are able to select task-relevant features in the output and then reincorporate them into the output representation.

Specifically, we design a learnable task vector $t_\xi \in R^d$ and employ it to perform the matrix product with the representation vector h^l in each Transformer layer l, subsequently, the resulting product is clamped to [0,1]. This selective mechanism preferentially retains tokens that exhibit high similarity to t_ξ, while effectively attenuating others via a soft masking procedure.

$$\tilde{h}_i^l = sim(h_i^l, t_\xi) \cdot h_i^l \tag{4}$$

Here t_ξ acts as a task embedding that encodes what kind of tokens are important for the task, and each token h_i^l is reweighted by its relevance (measured by cosine similarity) with the task embedding, thereby simulating token prioritization based on task-related importance.

Secondly, we integrate a transformation matrix $T \in R^{d\times d}$ to execute linear transformations on the reweighted representation [1], permitting finer adjustments and augmentations that are tailored to the task:

$$\hat{h}_i^l = \tilde{h}_i^l \cdot T \tag{5}$$

The transformation matrix T is designed to linearly transform the selected and reweighted token representations in a task-adaptive manner. This approach ensures that the resulting representation is discriminative and adaptable, prioritizing the information elements that is most relevant to the task. Finally, this refined information $\hat{h}_i^l$ is added to the original representation to achieve comprehensive enhancement.

[1] To promote parameter efficiency, we approximate the transformation matrix T using the product of two low-rank matrices.

Method	# Params	MNLI Acc	SST-2 Acc	CoLA Mcc	QQP Acc/F1	QNLI Acc	RTE Acc	MRPC Acc	STS-B Corr	All Ave.
Full FT	184M	89.90	95.63	69.19	**92.40/89.80**	94.03	83.75	89.46	91.60	88.42
BitFit	0.1M	89.37	94.84	66.96	88.41/84.95	92.24	78.70	87.75	91.35	86.06
HAdapter	1.22M	90.13	95.53	68.64	91.91/89.27	94.11	84.48	89.95	91.48	88.39
PAdapter	1.18M	90.33	95.61	68.77	92.04/89.40	94.29	85.20	89.46	91.54	88.52
LoRA$_{r=8}$	1.33M	90.65	94.95	69.82	91.99/89.38	93.87	85.20	89.95	91.60	88.60
AdaLoRA	1.27M	90.76	**96.10**	71.45	92.23/89.74	94.55	**88.09**	90.69	91.84	89.49
LoRATRF	0.88M	**90.87**	95.99	**71.61**	92.00/89.75	**94.69**	87.73	**91.18**	**91.89**	**89.52**

Table 1: Results with DeBERTaV3-base on GLUE development set. Best performances are highlighted in bold. Full FT, HAdapter and PAdapter represent full fine-tuning, Houlsby adapter, and Pfeiffer adapter respectively. We report baseline results directly from (Zhang et al. 2023a). We run the experiment on 5 different random seeds and report the mean.

Experimental Setup

Datasets

General Language Understanding Evaluation: GLUE (Wang et al. 2019) is a generalized natural language understanding assessment benchmark that includes a variety of tasks such as natural language inference, sentiment analysis, and sentence similarity evaluation, from which we select eight tasks for systematic evaluation, including Corpus of Linguistic Acceptability (CoLA), Multi-Genre Natural Language Inference (MNLI), Microsoft Research Paraphrase Corpus (MRPC), Question Natural Language Inference (QNLI), Quora Question Pairs (QQP), Recognizing Textual Entailment (RTE), Stanford Sentiment Treebank (SST-2), Semantic Textual Similarity Benchmark (STS-B).

Mathematical Reasoning: (1) **GSM8K** (Cobbe et al. 2021) dataset consists of high quality linguistically diverse grade school math word problems, (2) **SVAMP** (Patel, Bhattamishra, and Goyal 2021) benchmark consists of one-unknown arithmetic word problems, (3) **AddSub** (Hosseini et al. 2014) is a specialized dataset designed for evaluating algorithms.

Commonsense Reasoning: (1) BoolQ (Clark et al. 2019) dataset is a question-answering dataset for yes/no questions containing 15942 examples. (2) PIQA (Bisk et al. 2020) dataset of questions with two solutions requiring physical commonsense to answer; (3) SIQA (Sap et al. 2019) focuses on reasoning about people's actions and their social implications; (4) HellaSwag (Zellers et al. 2019) is a challenging dataset, which contains questions to select the best endings to complete sentences.

Baselines

We compare our methods to Full fine-tuning, Bitfit, Adapter tuning, LoRA and AdaLoRA. **Bitfit** (Ben Zaken, Goldberg, and Ravfogel 2022) fine-tunes bias vectors. **Houlsby adapter** (Houlsby et al. 2019) is inserted between the self-attention module and the FFN module. **Pfeiffer adapter** (Pfeiffer et al. 2021) inserts the adapter after FFN modules and LayerNorm modules. **LoRA** (Hu et al. 2022) parameterizes incremental updates by two small matrices. **AdaLoRA** (Zhang et al. 2023a) expresses the low-rank multiplication of LoRA. In empirical, we find that applying LoRA to Wv, W_{f_1} and W_{f_2} matrices can achieve the best performance (Please see **Different Choices of Modules to Adapt** Section).

Implementation Details

We implement our method for fine-tuning DeBERTaV3-base (He, Gao, and Chen 2022) and large language model LLaMA-7B (Touvron et al. 2023). LoRA (Hu et al. 2022) scales ΔW by α/r where α is a constant in r. As a result, the magnitude of output can be consistent given different r. It reduces the efforts of retuning learning rate when varying r. Typically α is set as 16 or 32 and never tuned.

Evaluation

For the GLUE benchmark, we report both accuracy and F1 for QQP in GLUE. For STS-B, we report the average correlation. For CoLA, we report Matthews correlation. For all remaining sub-tasks in GLUE, we report accuracy. For mathematical and commonsense reasoning datasets, we report accuracy.

Main Results

Natural Language Understanding

We compare LoRATRF with various baselines. Table 1 shows experimental results on the GLUE development set. We see that LoRATRF achieves better or on par performance compared with existing approaches on all datasets. For example, our method attains an accuracy of 71.61% on CoLA, surpassing AdaLoRA baseline by 0.2%, all while utilizing fewer parameters (0.88M compared to 1.27M). Among the baseline methods, the AdaLoRA method performs the best, which may be because it designs a method to dynamically allocate the rank of LoRA in different layers based on their importance. For all tasks except QQP, our method demonstrates different degrees of performance enhancement. These experiments verify the general applicability of our method to the NLU tasks.

Commonsense and Mathematical Reasoning

We also conduct experiments on the Mathematical and Commonsense Reasoning task using LLaMA-7B (Touvron et al.

LLM	Method	GSM8K	AddSub	SVAMP	HellaSwag	BoolQ	PIQA	SIQA
GPT-3.5		56.4	85.3	69.9	78.5	73.1	85.4	68.5
LLaMA-7B	Prefix	24.4	57.0	38.1	42.1	64.3	76.8	73.9
	Series	33.3	80.0	52.3	67.9	63.0	79.2	76.3
	Parallel	35.3	**86.6**	49.6	69.8	67.9	76.4	78.8
	LoRA	37.5	83.3	52.1	78.1	68.9	80.7	77.4
	DoRA	38.4	84.2	52.7	**84.8**	68.5	82.9	**79.6**
	LoRATRF	**38.6**	84.1	**53.0**	82.4	**69.6**	**83.5**	78.4

Table 2: Comparison results of different methods based on LLaMA-7B on reasoning datasets. We report some baseline results directly from (Liu et al. 2024) and (Hu et al. 2023).

2023). We use the same rank size (r=32) as the baseline method and only apply the LoRA module to output projection (W_o) in the self-attention, and two weight matrices (W_{f_1}, W_{f_2}) in two-layer FFNs. Comparison results are reported in Table 2. Notably, in the LLaMA-7B model, where DoRA exceeds the performance of other baselines, which may be due to DoRA decomposes weights for enhanced learning capacity. LoRATRF attains an accuracy of 85.6% on HellaSwag, surpassing DoRA baseline by 0.8%. To sum up, LoRATRF achieves the best performance in GSM8K. The results show that our method maintains its effectiveness in LLM, and further illuminate the significance of enhancing task-related features.

Quantitative Analysis

Different Choices of Modules to Adapt

We study the choices of modules to adapt for our method on SVAMP. We choose possible modules to adapt within query/key/value projection (W_q, W_k, W_v), output projection (W_o) in the self-attention, and two weight matrices (W_{f_1}, W_{f_2}) in two-layer FFNs. We hold the number of trainable parameters at the same level. Figure 2 shows the performance when fine-tuning specific modules, which demonstrates that adapting W_{f_1}, W_{f_2} and W_o yields the highest performance.

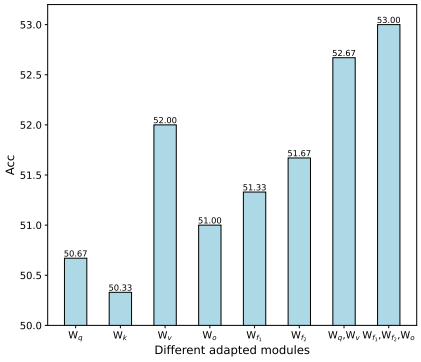

Figure 2: Testing performance of LLaMA-7B on SVAMP with different adapted modules.

Robustness of LoRATRF towards different rank settings

This section explores the impact of various rank configurations on LoRATRF and LoRA by adjusting r within the set $\{4, 8, 16, 32\}$ and assessing the performance of the fine-tuned LLaMA-7B on commonsense reasoning dataset HellaSwag. The average accuracies of LoRA and LoRATRF across different ranks are depicted in Figure 3. From Figure 3, we can observe that LoRATRF consistently surpasses LoRA across all rank configurations. Notably, the performance gap widens for ranks below 8, where LoRA's average accuracies drop to 59.3% for r = 8 and 51.2% for r = 4. In contrast, LoRATRF retains a notable accuracy of 73.9% for r = 8 and 70.5% for r = 4, demonstrating its resilience and consistently superior performance over LoRA regardless of the rank setting.

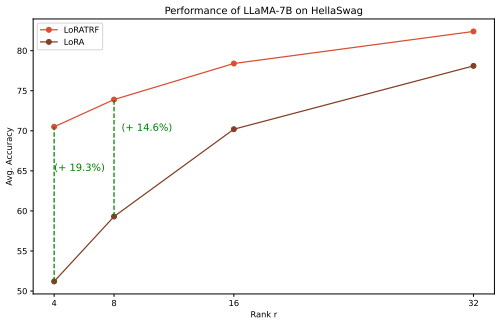

Figure 3: Testing performance of LLaMA-7B on HellaSwag with different rank settings.

Conclusion

In this work, we propose Low-Rank Adaptation with Task-Relevant Feature Enhancement, a novel approach aimed at bridging the performance gap between LoRA and full fine-tuning on complex tasks. We introduce task-aware filters to improve performance by prioritizing task-relevant features. Experiments on diverse benchmarks with different settings confirm the effectiveness of our method.

References

Aghajanyan, A.; Gupta, S.; and Zettlemoyer, L. 2021. Intrinsic Dimensionality Explains the Effectiveness of Language Model Fine-Tuning. In Zong, C.; Xia, F.; Li, W.; and Navigli, R., eds., *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)*, 7319–7328. Online: Association for Computational Linguistics.

Ben Zaken, E.; Goldberg, Y.; and Ravfogel, S. 2022. BitFit: Simple Parameter-efficient Fine-tuning for Transformer-based Masked Language-models. In Muresan, S.; Nakov, P.; and Villavicencio, A., eds., *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 1–9. Dublin, Ireland: Association for Computational Linguistics.

Bisk, Y.; Zellers, R.; Gao, J.; Choi, Y.; et al. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 7432–7439.

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, 1877–1901.

Clark, C.; Lee, K.; Chang, M.-W.; Kwiatkowski, T.; Collins, M.; and Toutanova, K. 2019. BoolQ: Exploring the Surprising Difficulty of Natural Yes/No Questions. In Burstein, J.; Doran, C.; and Solorio, T., eds., *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2924–2936. Minneapolis, Minnesota: Association for Computational Linguistics.

Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Dettmers, T.; Pagnoni, A.; Holtzman, A.; and Zettlemoyer, L. 2023. QLoRA: Efficient Finetuning of Quantized LLMs. *arXiv preprint arXiv:2305.14314*.

Ding, N.; Lv, X.; Wang, Q.; Chen, Y.; Zhou, B.; Liu, Z.; and Sun, M. 2023. Sparse Low-rank Adaptation of Pre-trained Language Models. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 4133–4145. Singapore: Association for Computational Linguistics.

Han, Z.; Gao, C.; Liu, J.; Zhang, S. Q.; et al. 2024. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*.

He, P.; Gao, J.; and Chen, W. 2022. DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing. In *The Eleventh International Conference on Learning Representations*.

Hosseini, M. J.; Hajishirzi, H.; Etzioni, O.; and Kushman, N. 2014. Learning to Solve Arithmetic Word Problems with Verb Categorization. In Moschitti, A.; Pang, B.; and Daelemans, W., eds., *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 523–533. Doha, Qatar: Association for Computational Linguistics.

Houlsby, N.; Giurgiu, A.; Jastrzebski, S.; Morrone, B.; De Laroussilhe, Q.; Gesmundo, A.; Attariyan, M.; and Gelly, S. 2019. Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning*, 2790–2799. PMLR.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*.

Hu, Z.; Wang, L.; Lan, Y.; Xu, W.; Lim, E.-P.; Bing, L.; Xu, X.; Poria, S.; and Lee, R. 2023. LLM-Adapters: An Adapter Family for Parameter-Efficient Fine-Tuning of Large Language Models. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 5254–5276. Singapore: Association for Computational Linguistics.

Huang, C.; Liu, Q.; Lin, B. Y.; Pang, T.; Du, C.; and Lin, M. 2023. Lorahub: Efficient cross-task generalization via dynamic lora composition. *arXiv preprint arXiv:2307.13269*.

Kenton, J. D. M.-W. C.; and Toutanova, L. K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL-HLT*, 4171–4186.

Lester, B.; Al-Rfou, R.; and Constant, N. 2021. The Power of Scale for Parameter-Efficient Prompt Tuning. In Moens, M.-F.; Huang, X.; Specia, L.; and Yih, S. W.-t., eds., *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 3045–3059. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics.

Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; and Zettlemoyer, L. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7871–7880.

Li, C.; Farkhoor, H.; Liu, R.; and Yosinski, J. 2018. Measuring the Intrinsic Dimension of Objective Landscapes. In *International Conference on Learning Representations*.

Li, X. L.; and Liang, P. 2021a. 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)*, 4582–4597.

Li, X. L.; and Liang, P. 2021b. Prefix-Tuning: Optimizing Continuous Prompts for Generation. In Zong, C.; Xia, F.; Li, W.; and Navigli, R., eds., *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)*, 4582–4597. Online: Association for Computational Linguistics.

Liu, S.-Y.; Wang, C.-Y.; Yin, H.; Molchanov, P.; Wang, Y.-C. F.; Cheng, K.-T.; and Chen, M.-H. 2024. Dora: Weight-decomposed low-rank adaptation. *arXiv preprint arXiv:2402.09353*.

Liu, X.; Ji, K.; Fu, Y.; Tam, W.; Du, Z.; Yang, Z.; and Tang, J. 2022. P-Tuning: Prompt Tuning Can Be Comparable to Fine-tuning Across Scales and Tasks. In Muresan, S.; Nakov, P.; and Villavicencio, A., eds., *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 61–68. Dublin, Ireland: Association for Computational Linguistics.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Patel, A.; Bhattamishra, S.; and Goyal, N. 2021. Are NLP Models really able to Solve Simple Math Word Problems? In Toutanova, K.; Rumshisky, A.; Zettlemoyer, L.; Hakkani-Tur, D.; Beltagy, I.; Bethard, S.; Cotterell, R.; Chakraborty, T.; and Zhou, Y., eds., *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2080–2094. Online: Association for Computational Linguistics.

Pfeiffer, J.; Kamath, A.; Rücklé, A.; Cho, K.; and Gurevych, I. 2021. AdapterFusion: Non-Destructive Task Composition for Transfer Learning. In Merlo, P.; Tiedemann, J.; and Tsarfaty, R., eds., *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 487–503. Online: Association for Computational Linguistics.

Sap, M.; Rashkin, H.; Chen, D.; Le Bras, R.; and Choi, Y. 2019. Social IQa: Commonsense Reasoning about Social Interactions. In Inui, K.; Jiang, J.; Ng, V.; and Wan, X., eds., *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 4463–4473. Hong Kong, China: Association for Computational Linguistics.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Valipour, M.; Rezagholizadeh, M.; Kobyzev, I.; and Ghodsi, A. 2023. DyLoRA: Parameter-Efficient Tuning of Pre-trained Models using Dynamic Search-Free Low-Rank Adaptation. In Vlachos, A.; and Augenstein, I., eds., *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, 3274–3287. Dubrovnik, Croatia: Association for Computational Linguistics.

Wang, A.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; and Bowman, S. R. 2019. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019*.

Wang, Y.; Agarwal, S.; Mukherjee, S.; Liu, X.; Gao, J.; Awadallah, A. H.; and Gao, J. 2022. AdaMix: Mixture-of-Adaptations for Parameter-efficient Model Tuning. In Goldberg, Y.; Kozareva, Z.; and Zhang, Y., eds., *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 5744–5760. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics.

Wu, X.; Huang, S.; and Wei, F. 2024. MoLE: Mixture of LoRA Experts. In *The Twelfth International Conference on Learning Representations*.

Zellers, R.; Holtzman, A.; Bisk, Y.; Farhadi, A.; and Choi, Y. 2019. HellaSwag: Can a Machine Really Finish Your Sentence? In Korhonen, A.; Traum, D.; and Màrquez, L., eds., *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 4791–4800. Florence, Italy: Association for Computational Linguistics.

Zhang, Q.; Chen, M.; Bukharin, A.; He, P.; Cheng, Y.; Chen, W.; and Zhao, T. 2023a. Adaptive Budget Allocation for Parameter-Efficient Fine-Tuning. In *The Eleventh International Conference on Learning Representations*.

Zhang, Y.; Wang, P.; Tan, M.; and Zhu, W. 2023b. Learned Adapters Are Better Than Manually Designed Adapters. In Rogers, A.; Boyd-Graber, J.; and Okazaki, N., eds., *Findings of the Association for Computational Linguistics: ACL 2023*, 7420–7437. Toronto, Canada: Association for Computational Linguistics.

Zou, A.; Phan, L.; Chen, S.; Campbell, J.; Guo, P.; Ren, R.; Pan, A.; Yin, X.; Mazeika, M.; Dombrowski, A.-K.; et al. 2023. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*.