Task Decoupled Low-rank Adaption for Parameter-Efficient Fine-Tuning

Anonymous ACL submission

Abstract

Parameter Efficient Fine-Tuning (PEFT) provides a powerful approach for adapting large pre-trained language models (PLMs) to specific domains. Among PEFT methods, LoRA-based methods have emerged as a leading approach. However, existing LoRA-based approaches primarily focus on modifying the architecture or the low-rank matrices, overlooking the significant influence of downstream tasks on LoRA effectiveness. In this paper, we first theoretically demonstrate that fine-tuning actually involves two tasks: domain inference fine-tuning and content inference fine-tuning. We also demonstrate that LoRA can be decoupled into two tasks. To address the limitations associated with coupled updates in LoRA, we introduce Task Decoupled LoRA (TD-LoRA), a novel approach that segregates LoRA tasks into distinct domain inference and content inference tasks. We employ cosine similarity between fine-tuned weights and pre-trained weights to approximate the transition from general knowledge to domain-specific knowledge. Additionally, TD-LoRA has the same memory requirements and comparable computational costs as LoRA. We conduct extensive experiments on various pre-trained models and demonstrate its effectiveness on the GLUE, E2E and MMLU benchmarks.

1 Introduction

Large Pre-trained language models (PLMs), such as GPT series (Radford et al., 2019; Brown et al., 2020), GLM (Zeng et al., 2023), LLaMA series (Touvron et al., 2023a,b), have achieved remarkable performance across a wide range of NLP tasks. Despite their strong general capabilities, PLMs still fall short in certain domains, such as programming, mathematics, biomedical, or finance (Wu et al., 2024). Fine-tuning is the primary technique for PLMs to adapt to specific downstream NLP tasks. Fine-tuning the whole PLMs requires tremendous computational cost and significant memory capacities. These requirements imposes significant limitations on both the research and application of the PLMs.

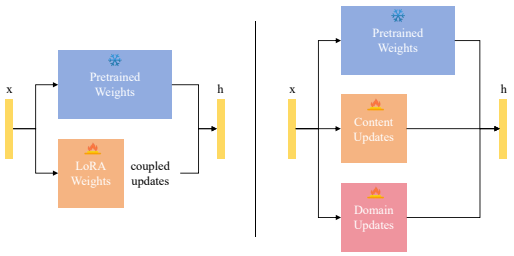


Figure 1: The structures of LoRA and TD-LoRA from the perspective of task decoupling. LoRA couples the tasks of fine-tuning in one branch. TD-LoRA has two branch in parallel, one branch for domain inference fine-tuning, and the other for content inference fine-tuning.

To address this problem, researchers propose an incredibly powerful method known as PEFT. PEFT updates only a small number of parameters, which significantly reduces both computational costs and memory requirements, while still managing to achieve performance levels comparable to those of full fine-tuning. PEFT methods encompass Adapter (Houlsby et al., 2019), prompt-tuning (Lester et al., 2021), Prefix-tuning (Li and Liang, 2021), LoRA (Hu et al., 2022), et al. Currently, LoRA is regarded as one of the most efficient techniques (Ding et al., 2023a), and a substantial number of PEFT strategies have been developed based on LoRA. LoRA operates by freezing the pre-trained weights W_0, and the weights to be updated, ΔW, are represented as low-rank decomposition matrices $A \in \mathbb{R}^{r \times d_2}$ and $B \in \mathbb{R}^{d_1 \times r}$.

In LoRA, rank r determines the number of trainable parameters. Experiments (Hu et al., 2022; Zhang et al., 2023b; Kopiczko et al., 2023; Ding et al., 2023b) demonstrate that, under the same LoRA-based method and PLMs, the optimal choice of rank r vary across different downstream tasks.

(Ding et al., 2023c) made experiments on 100 representative tasks and observed that the performance of PEFT is not consistent with their number of trainable parameters. Instead, the design of the structure for PEFT play a greater role. This phenomenon naturally raises a question:

Question 1: How do the downstream tasks affect the performance of LoRA-based methods?

Existing researches on downstream tasks in fine-tuning PLMs is focused on the area of catastrophic forgetting. (Kotha et al., 2023) factor a model into task inference and the capabilities on the tasks. (Lin et al., 2023) proposes that models gain speciality for the fine-tuning task and loss generality for other tasks during fine-tuning. Inspired by these works, we hypothesize that fine-tuning encompasses two parameter updating tasks: **1)** domain inference fine-tuning, which updates model's inference of the input domain; and **2)** content inference fine-tuning, which enhances domain-specific capabilities. We provide a theoretical proof that the fine-tuning process is composed of these two tasks in Section 3. Figure 1 illustrates the process of task decoupling in LoRA.

Therefore, considering the **Question 1**, we presents another essential question:

Question 2: Can we decouple the tasks of LoRA to improve the performance of LoRA?

To address the aforementioned two questions, we propose task decoupled LoRA, TD-LoRA, which is a novel LoRA structure with two branches to obtain domain-specific fine-tuning. We decouple the tasks of LoRA and improving the performance of the two tasks separately. Specifically, we introduce a new branch to LoRA adaption to achieve task decoupled in LoRA. To achieve the some memory requirements and comparable computational costs with LoRA, we utilize the cosine similarity between original LoRA adaption weight ΔW and the pre-training weight W_0 as the approximate prediction. As illustrated in Figure 2.

The contribution of the proposed methods are as follows:

- We first proposed that fine-tuning process encompasses two parameter updating tasks: domain inference fine-tuning and content inference fine-tuning.
- We provide a theoretical proof of the task decoupling in LoRA process. Based on this theory, we first analyze LoRA from the perspective of task decoupling.

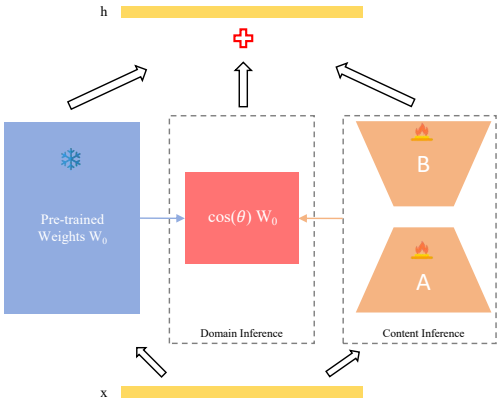


Figure 2: The structure of TD-LoRA. The fine-tuning adaption has two branch in parallel.

- We propose a novel LoRA, TD-LoRA, with two branches to obtain domain-specific fine-tuning. TD-LoRA has the some memory requirements and comparable computational costs with LoRA, and achieve better performance on multiple tasks.

2 Related work

2.1 LoRA-based Methods

Compared with full fine-tuning which requires tremendous computational cost, Low-rank adaption (LoRA) updates a small number of parameters and achieve comparable performance. LoRA become a prevalent strategy for fine-tuning LLMs. Based on the directions of improvement, we categorize the LoRA-based methods as follows. 1) (Aghajanyan et al., 2021; Edalati et al., 2022) focus on finding a better lower-rank matrix. These methods adjust the structure of adaption to get an optical LoRA weight. 2) (Valipour et al., 2023; Zhang et al., 2023b) adjust the rank of LoRA matrix. These methods adjust the diagonal matrix, indeed they make research on the a parameter r which represent the rank of LoRA. 3) (Zi et al., 2023; Zhang et al., 2023a) update pre-training weights according LoRA metric, that is they train an parameter to adjust the weight of pre-training weight. 4) (Dettmers et al., 2023; Xu et al., 2023b) propose quantization techniques to improve the high precision fine-tuning and inference of LoRA. 5) (Huang et al., 2023; Liu et al., 2023) combine multiple LoRA modules for cross-task transfer.

2.2 Task Decoupling

Task decoupling is a commonly used technique in computer vision (CV) domain, such as object de-

tection, scene text detection. Outstanding works used the decoupled features to get the representation need for CV tasks. CTPN (Tian et al., 2016) propose three branches to obtain distinct features to predict score, coordinates and offsets of anchors. Faster RCNN (Ren et al., 2015) has two branches for bounding box regression and classification. YOLOX (Ge et al., 2021) introduces decoupled head to the YOLO family to decoupling regression and classification. These works demonstrate the importance of decoupling tasks in CV domain. However, as mentioned in Section 1, the task decoupling of fine-tuning PLMs process has not been studied yet and requires close attention.

3 Theoretical Analysis of Fine-tuning Tasks

We are interested in the fine-tuning process from the perspective of downstream tasks. We will first study fine-tuning components through model weights, providing a theoretical analysis of the two tasks of fine-tuning. Then we demonstrate that LoRA can be divided into two branches to obtain domain-specific fine-tuning. Finally, we propose TD-LoRA to realized task decoupling in fine-tuning process.

3.1 Two Tasks of Fine-tuning Process

To identify the fine-tuning tasks from the perspective of downstream tasks, we study fine-tuning through model weights.

Specifically, for a pre-trained model fine-tuned on single downstream task, the distribution of fine-tuned model weights W can be represented as follows:

$$\begin{aligned} P(W|X,Y) &= \frac{P(Y|X,W)P(W|X)P(X)}{P(X,Y)} \\ &= \frac{P(Y|X,W)P(W|X)P(X)}{P(Y|X)P(X)} \\ &= \frac{P(Y|X,W)P(W|X)}{P(Y|X)} \end{aligned} \tag{1}$$

where $P(Y|X,W)$ denotes the W-conditioned content inference distribution, $P(W|X)$ denotes distribution of W given X.

$P(Y|X)$ is independent of W, Eq. 1 is approximated as:

$$P(W|X,Y) = P(Y|X,W)P(W|X) \tag{2}$$

We can observe that the distinction between model weights in two terms: $P(Y|X,W)$ and $P(W|X)$. The distribution of X varies across downstream tasks, thus $P(W|X)$ is different across downstream tasks. The optical W on one downstream tasks do not achieve similar performance on other tasks. This answers **Question 1**.

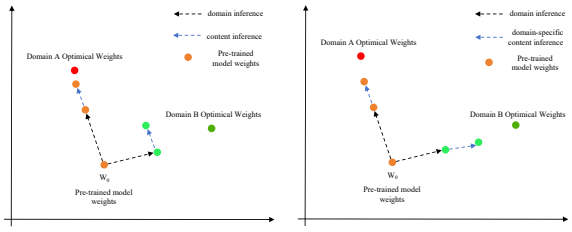


Figure 3: A schematic view of fine-tuning process with coupled tasks (Left) and decoupled tasks (Right). As training progresses, the model initially completes the domain inference task based on X, and subsequently adjusts content prediction according to both domain W and input X.

$P(W|X)$ represents the change of downstream tasks information, e.g., tasks changes in mathematics and biomedical. We refer to it as domain inference, as it represents the model's ability to identify downstream tasks from input X. $P(Y|X,W)$ stands for the prediction of content Y. We refer to it as content inference, representing the model's ability to generate outputs based on the input, following the process of domain inference.

$$P(W|X,Y) = \underbrace{P(Y|X,W)}_{\text{content inference}} \underbrace{P(W|X)}_{\text{domain inference}} \tag{3}$$

Essentially, the fine-tuning process updates two functions of the model: domain inference and content inference. The domain inference distribution undergoes changes across various downstream-specific datasets during the whole fine-tuning process. To achieve domain-specific fine-tuning, we must decouple these two tasks. Figure 3 shows a schematic view of fine-tuning process with couples tasks and decoupled tasks.

3.2 Task Decoupling in LoRA Process

According to Eq. 3, the objective of fine-tuned model weight W is:

$$\begin{aligned} W &= \underset{W}{\operatorname{argmax}} P(W|X,Y) \\ &= \underset{W}{\operatorname{argmax}} P(Y|X,W)P(W|X) \\ &= \underset{W}{\operatorname{argmax}} \log P(Y|X,W) + \log P(W|X) \end{aligned} \tag{4}$$

where $P(Y|X,W)$ denotes the fine-tuned model content inference, $P(W|X)$ denotes the fine-tuned model domain inference. Eq. 4 suggests that the

model weights are essentially composed of two parts: content inference and domain inference.

Similarly, we can obtain that the pre-trained model W_0 is:

$$\begin{aligned} W_0 &= \underset{W}{\operatorname{argmax}} P_0(Y|X,W)P_0(W|X) \\ &= \underset{W}{\operatorname{argmax}} P_0(Y|X,W)P_0(W|X) \end{aligned} \tag{5}$$

where $P_0(Y|X,W)$ denotes the pre-trained model content inference, $P_0(W|X)$ denotes the pre-trained model domain inference. $P_0(Y|X,W)$ and $P_0(W|X)$ is constant once the pre-trained model is confirmed.

For a pre-trained model weight $W_0 \in \mathbb{R}^{d_1 \times d_2}$, LoRA updates a low-rank matrix weight ΔW in fine-tuning process, i.e, $W = W_0 + \Delta W$. Researches (Dai et al., 2023; Garg et al., 2022; Li et al., 2023) approximate the standard attention to relaxed linear attention and (Kotha et al., 2023) demonstrate pre-trained models approach mixture regression.

Inspired by these works, we hypothesize that approximate the LoRA fine-tuned model can be approximately expressed as a combination of linear regression models:

$$\begin{aligned} P(Y|X,W) &= f(WX) = f(W_0X) + f(\Delta WX) \\ &= P(Y|X,W_0) + P(Y|X,\Delta W) \end{aligned} \tag{6}$$

W_0 is frozen in LoRA, $P(Y|X,W_0)$ and $P_0(Y|X,W_0)$ are independent of W. Combining Eq. 4, Eq. 5 and Eq. 6, during the process of fine-tuning a specific model using LoRA, the objective of model weight W is obtained by:

$$\begin{aligned} W &= \underset{W}{\operatorname{argmax}} P(Y|X,W)P(W|X) \\ &= \underset{W}{\operatorname{argmax}} (P(Y|X,W_0) + P(Y|X,\Delta W))P(W|X) \\ &= \underset{W}{\operatorname{argmax}} P(Y|X,W_0)P_0(W_0|X)\frac{P(W|X)}{P_0(W_0|X)} \\ &+ \underset{W}{\operatorname{argmax}} P(Y|X,\Delta W)P(W|X) \\ &= \underset{W}{\operatorname{argmax}} \frac{P(Y|X,W_0)}{P_0(Y|X,W_0)} W_0 \frac{P(W|X)}{P_0(W_0|X)} \\ &+ \underset{W}{\operatorname{argmax}} P(Y|X,\Delta W)P(\Delta W|X) \\ &= \underset{W}{\operatorname{argmax}} W_0 \frac{P(W|X)}{P_0(W_0|X)} + \Delta W \\ &= W_0 + \Delta W + \frac{P(W|X) - P_0(W_0|X)}{P_0(W_0|X)} W_0 \\ &= W_0 + \underbrace{\Delta W}_{\text{content}} + \underbrace{\frac{P(W|X) - P_0(W_0|X)}{P_0(W_0|X)} W_0}_{\text{domain}} \end{aligned} \tag{7}$$

Compared with LoRA, there are three parts in Eq. 7: the pre-trained model weights W_0, the LoRA weights ΔW and a new weights associated with the changes of domain inference. From the perspective of the proposed task decoupling, LoRA only updates the content inference of the model while disregarding the domain inference.

The original LoRA single branch is not suitable, as indicated by Eq. 7. To achieve task decoupling weights updates, a new branch is required to facilitate domain inference fine-tuning. Through this process, domain-specific fine-tuning is achieved. Eq. 7 answers **Question 2** and provides insights into how decoupling tasks in LoRA can potentially enhance performance.

3.3 Task Decoupling LoRA (TD-LoRA)

It is an NP-hard problem to find $P(W|X)$ because W is a high-dimensional vector with a large number of parameters. Experiments in (Gueta et al., 2023) revealed that PLMs fine-tuned on the same downstream task are clustered together in weights space. This indicates that the cosine similarity between weights increases for more similar downstream tasks, whereas it decreases for less similar downstream tasks. The cosine similarity in weight spaces reflect whether X belongs to downstream tasks. So we approximate this NP-hard optimization objective by the cosine similarity of weights. The cosine similarity between model weights is inversely proportional to the change in the domain inference.

In TD-LoRA, the trainable low-rank matrix ΔW_{TD} is divided into two parts to decouple tasks:

$$\Delta W_{TD} = \Delta W_c + \Delta W_d \tag{8}$$

where ΔW_c performs the content inference fine-tuning, ΔW_d performs the domain inference fine-tuning.

To achieve the some memory requirements and comparable computational costs with LoRA, we utilize the cosine similarity between LoRA updates weights W and the pre-training weight W_0 as the approximate prediction of domain inference.

$$\Delta W_D = -\cos(W_0, \Delta W)W_0 \tag{9}$$

where the negative value of cosine similarity is attributed to the negative correlation between the cosine similarity of weights and the domain change, as we analyzed above.

This finding is supported by the experiments in (Kotha et al., 2023). It observed that the generated content of pre-trained models for similar tasks are most influenced by downstream tasks, while dissimilar tasks are less affected by downstream tasks.

Algorithm 1 TD-LoRA

Input: r: low rank; α: scale factor, T: the total number of training epochs, W_0: model parameters, A: low-rank metric, B: low-rank metric, τ: the global learning rate, X: fine-tuning dataset $\{(x_1, y_1), ..., (x_n, y_n)\}$.

W_0 is initialized with pre-trained weights. A is initialized by Kaiming Initialization, B = 0.
for epoch=$1, 2, ..., T$ **do**
  for $(x, y) \in X$ **do**
    $p \leftarrow W_0 x + BA$
    $A \leftarrow A - \tau \nabla_A l(p, y)$
    $B \leftarrow B - \tau \nabla_B l(p, y)$
    $\Delta W_d \leftarrow -\cos(W_0, W_0 + BA) W_0$
    $\Delta W_c \leftarrow BA$
    $W \leftarrow W_0 + \alpha \Delta W_c + \alpha \Delta W_d$
  end for
end for

Output: A,B

This demonstrates the inverse relationship between cosine similarity of weights and domain changes.

In Section 4.3.1, we compared cosine similarity with random matrix, additional trainable matrix and mixture methods. With cosine similarity, the memory requirements of TD-LoRA is some as LoRA, we only need to increase few computational costs to cover the cosine calculation between two matrices $W \in \mathbb{R}^{d_1 \times d_2}$. Table 3 shows the additional computational costs.

Algorithm 1 shows the details of TD-LoRA. Intuitively, in the direction of downstream tasks, the cosine similarity between W_0 and W is large, whereas in the direction of non-downstream tasks, the cosine similarity between the W_0 and W is small. Therefore, cosine similarity can be used to approximate the inference process of domain inference.

Finally, the overall weight updates of the proposed TD-LoRA can be expressed as:

$$W = W_0 + BA - \cos(W_0, W_0 + BA) W_0 \quad (10)$$

4 Experiments

In this section, we compare TD-LoRA with full fine-tuning and prior works of Prefix-tuning, LoRA, AdaLoRA. Our experiments cover natural language understanding tasks, natural language generation tasks and LLMs instruction tuning. Specifically, we fine-tuning RoBERTa-base and RoBERTa-large (Liu et al., 2019) on the General Language Understanding Evaluation (GLUE) benchmark (Wang et al., 2019), GPT-2 Medium (Radford et al., 2019) on E2E NLG Challenge benchmarks (E2E), LLaMA-7B and LLaMA2-7B on MMLU benchmark (Hendrycks et al., 2021). The pre-trained models are download from *HuggingFace*. We conduct the experiments on NVIDIA GeForce RTX 3090 GPU with 24GB memory.

4.1 Experimental Settings

Baseline. We compare TD-LoRA with the following methods:

- Full fine-tuning (Full FT) of PLMs involves training the entire model, including all layers and parameters. It is an effective methods for PLMs to adapt to downstream tasks. FT^{Top2} is a common variant that update the last two layers while freezing others.

- Prefix-tuning (Prefix) append a learnable vector $P = [P_1], [P_2], , [P_l]$ to the hidden states of the multi-head attention layer. Prefix-tuning enables the use of different prefixes for various downstream tasks.

- LoRA is the most common approach for PLMs. LoRA freezes the pre-trained weights W_0 and the update weights ΔW is low-rank decomposition matrices A and B. We reproduced LoRA, since several results on the GLUE development set were missed. To ensure a fair comparison, we initialize the model to the pre-trained model on all tasks.

- AdaLoRA extends LoRA by dynamically adjusting the rank r according to importance of modules. AdaLoRA parameterizes the incremental updates in the form of singular value decomposition. AdaLoRA adaptively allocates the parameter budget among weight matrices according to their importance score.

Models and Datasets. We conduct extensive experiments on multiple pre-trained models and benchmarks.

- GLUE Benchmark consists of 8 natural language understanding tasks. These tasks encompass a range of challenges, including single-sentence classification, pairwise text classification, and regression. Following most previous PEFT works (Hu et al., 2022;

	Methods	# TPs	CoLA Acc	SST-2 Acc	MRPC Acc/F1	STS-B P/S	QQP Acc/F1	NNLI Acc	QNLI Acc	RTE Acc
base	Full FT	124.6M	59.07	92.89	88.24/91.58	90.87/90.61	90.81/87.72	86.27	91.07	72.20
	Prefix	0.96M	59.31	93.81	87.25/91.03	88.48/88.32	87.75/84.09	85.21	90.77	54.51
	LoRA(Re)	0.3M	63.50	**95.071**	88.48/91.59	**90.91/90.51**	90.75/87.69	87.32	92.73	**81.23**
	AdaLoRA	1.03M	59.82	93.92	87.99/91.33	90.83/90.73	88.58/84.98	86.26	91.43	70.04
	TD-LoRA	0.3M	**65.61**	94.84	**90.68/92.14**	90.65/90.39	**90.81/87.82**	**87.39**	**93.23**	79.06
large	Full FT	355.3M	65.78	95.54	89.22/92.28	91.74/91.76	89.30/86.68	89.42	93.61	81.23
	Prefix	2.03M	59.01	95.76	88.24/91.37	90.92/91.07	88.88/85.45	89.30	93.32	74.01
	LoRA(Re)	0.8M	**68.03**	95.99	90.20/92.91	92.0/91.75	**91.01/88.07**	**90.54**	94.75	85.56
	AdaLoRA	2.23M	65.85	94.95	89.46/92.34	92.05/91.80	89.60/86.30	90.36	94.62	77.98
	TD-LoRA	0.8M	67.03	**96.10**	**90.93/93.40**	**92.19/91.9**	90.99/88.14	90.49	**95.24**	**87.00**

Table 1: Results on the GLUE benchmark with RoBERTa-Base and RoBERTa-Large. In the table, "TPs"denotes the number of trainable parameters, "P/S"represents the Pearson/Spearman correlation. Specifically, we report Matthews correlation for COLA, the averaged matched accuracy for MNLI. Results of Full Fune-tuning, Prefix-tuning and AdaLoRA are sourced from (Xu et al., 2023a). We reproduced LoRA (LoRA(Re)) since several results on the GLUE development set were missed. We denote the best result in bold and underline the second best result(except the LoRA results from (Hu et al., 2022)). "-"denotes the missed result in related paper.

Fu et al., 2023; Kopiczko et al., 2023; Xu et al., 2023a), we employ RoBERTa-base and RoBERTa-large as backbone model.

- E2E NLG Challenge benchmarks is a common used benchmark for NLG models, such as GPT-2. It consists of 42,000 training, 4,600 validation, and 4,600 test examples from the restaurant domain. We fine-tuning GPT-2 Medium on E2E to demonstrate that TD-LoRA still performs better on NLG tasks and NLG models.

- MMLU benchmark (MMLU) is widely used as a benchmarks for LLMs evaluation. It covers 57 tasks including science, humanities, econometric and more. Alpaca dataset (Taori et al., 2023) consists of 52,000 instructions and demonstrations generated by OpenAI's text-davinci-003 engine. It is a popular instruction tuning dataset for LLMs. We download Alpaca dataset from $HuggingFace$ and fine-tuned LLaMA-7B, LLaMA2-7B. We evaluate the fine-tuned model on MMLU to compare the performance of TD-LoRA and LoRA.

Implementation Details. We propose a task decoupling PEFT based on LoRA. TD-LoRA employ cosine similarity to approximate the domain inference function. The trainable parameters are the same as in LoRA. Therefore, we mainly compare the performance of TD-LoRA and LoRA under the same experimental setup. In GLUE benchmark experiments, we use the experimental setup from (Hu et al., 2022). In the E2E benchmark experiments, we also use the experimental setup from (Hu et al., 2022). In instruction tuning experiments, we use the some experimental setup for TD-LoRA and LoRA. We set the learning rate 10^{-4}, epochs 3, batch size 128, LoRA rank 8, LoRA alpha 16, LoRA trainable matrices $[Q, V]$. In the ablation study, we conduct experiments on RoBERTa-large and LLaMA-7B. The downstream task is RTE. The experimental setup is the some as experiments in GLUE benchmark. Without loss of generality, the training settings of experiments similar to those of LoRA. LLaMA is the most commonly used open-source large language model. We fine-tuning LLaMA to further demonstrate the effectiveness of TD-LoRA on LLMs.

4.2 Results

4.2.1 GLUE Benchmark

We first conduct an evaluation on GLUE benchmark, a widely recognized benchmark for PEFT methods. Similarly to LoRA, we apply RoBERTa-base and RoBERTa-large as backbones. We initialize the model to the pre-trained model on all tasks and reproduced LoRA results using some training settings to obtain several missing results in paper.

Table 1 shows experimental results on the GLUE development set. The results indicate that TD-LoRA outperforms LoRA and other representative fine-tuning methods in more than half of the tasks, and it ranks in the top 2 of all tasks. This demonstrates the effectiveness of the proposed task decoupling algorithm.

4.2.2 E2E Benchmark

GPT-2 Medium is the 355M parameter version of GPT-2, a transformer-based language model. We

Model & Method	# TPs	E2E NLG Challenge				
		BLEU	NIST	MET	ROUGE-L	CIDEr
GPT-2 M (FT)	354.92M	68.2	8.62	46.2	71.0	2.47
GPT-2 M (FT^{Top2})	25.19M	68.1	8.59	46.0	70.8	2.41
GPT-2 M (LoRA)	0.35M	68.69	8.65	46.37	71.12	2.48
GPT-2 M (TD-LoRA)	0.35M	**69.93**	**8.80**	**46.59**	**71.55**	**2.53**

Table 2: GPT-2 medium (M) with Full fine-tuning, LoRA and TD-LoRA on the E2E benchmark. Results of FT and FT^{Top2} are sourced from (Hu et al., 2022). We denote the best result in bold. For all metrics, TD-LoRA outperforms LoRA without increasing trainable parameters.

make experiments on GPT-2 Medium to evaluate the performance of TD-LoRA on NLG models. We use the experimental setup provided by (Hu et al., 2022) both for LoRA and TD-LoRA. Table 2 shows TD-LoRA performs better than LoRA for all metrics. This indicates TD-LoRA also also performs effectively with transformer-based NLG models.

4.2.3 Instruction Tuning

RoBERTa-base (125M), RoBERTa-large (355M) and GPT-2 M (354.92M) are relative small models. LLaMA is the widely used open-source large lauguage model. We apply LLaMA-7B and LLaMA2-7B for instruction tuning and perform fine-tuning using both LoRA and TD-LoRA. We apply LoRA and TD-LoRA on all 32 LLaMA decoder layers. We choose Alpaca datasets as fine-tuning dataset. This dataset is used to conduct instruction-tuning for LLMs, making LLMs follow instruction better. We evaluate the fine-tuned models on MMLU benchmarks. Results are shown in Table 4.

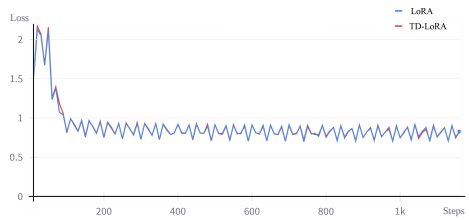
To calculate the additional computational costs of TD-LoRA, we compute the total FLOPs of the fine-tuning process of LLaMA, shown in Table 3. The additional computational costs are relative small.

4.3 Ablation Study

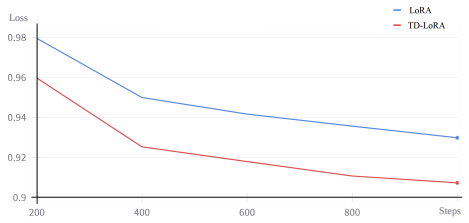
4.3.1 Domain Inference Methods

TD-LoRA appends a branch for domain inference task and employs cosine similarity to accomplish the domain inference task. We conducted experiments to explore different method on the domain inference branch. We compare the performance of various methods: cosine similarity, random matrix, additional trainable matrix and mixture of LoRA and TD-LoRA.

Table 5 shows the results of these methods. Compared to random matrix, additional trainable matrix and mixture methods, cosine similarity achieved superior results. This indicates that cosine similarity more effectively approximate domain changes in fine-tuning process.



(a) Train Loss



(b) Eval Loss

Figure 4: Loss curves of LLaMA-7B in the training process. The fine-tuning data is Alpaca dataset.

4.3.2 Optimization Process

To illustrate the detailed process of fine-tuning pre-trained model using TD-LoRA, we fine-tuned LLaMA-7b on the Alpaca dataset. We recorded the train loss value every 10 steps and recorded the evaluation loss values every 200 steps. We show the loss curves in Figure 4.

It can be observed that the train loss of TD-LoRA is very close to that of LoRA, but the evaluation loss of TD-LoRA is smaller than that of LoRA and decreases faster. This indicates that during training process, TD-LoRA is better able to capture different optimization directions for different domains, rather than simply fitting the training data. LoRA couple the domain inference and content inference updates. It appears that LoRA is exhibiting characteristics of overfitting to the training data.

Model	PEFT Method	Epoch	FLOPs (TFLOPs)	Training Times (min)
LLaMA-7B	LoRA	3	1345015.310	466 min
	TD-LoRA	3	1345028.006(↑ 0.001%)	486 min(↑ 4.3%)
LLaMA2-7B	LoRA	3	1351931.086	234 min
	TD-LoRA	3	1351972.977(↑ 0.003%)	245 min(↑ 4.7%)

Table 3: Total FLOPs and training time of fine-tuning with LoRA and TD-LoRA. The additional computational costs occurs during the calculation of cosine similarity.

Model	PEFT Method	# TPs	MMLU Acc
LLaMA-7B	Non	0M	35.23
	LoRA	4.19M	35.54
	TD-LoRA	4.19M	**36.41**
LLaMA2-7B	Non	0M	45.30
	LoRA	4.19M	45.19
	TD-LoRA	4.19M	**45.61**

Table 4: Results of 5-shot MMLU accuracy. Fine-tuning dataset is Alpaca dataset. "Non "denotes the pre-trained model without fine-tuning. The best results are in bold. TD-LoRA outperforms LoRA on MMLU benchmark.

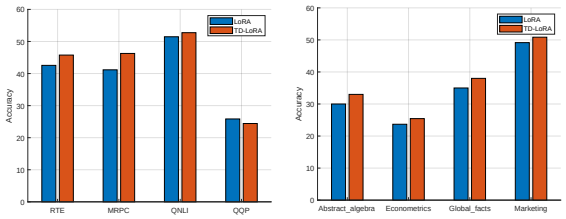
Method	Add TPs	RTE Acc(%)
Non	0M	85.56
Random	0M	52.71
Trainable	0.8M	85.92
Cosine	0M	**87.00**
Non + Cosine	0M	85.20

Table 5: Results of different domain inference methods. Experiments are conducted on RoBERTa-large, with RTE tasks. "Add TPs "denotes the additional trainable parameters compared to LoRA. "Non "denotes models trained with LoRA. "Non + Cosine "denotes training with LoRA first, followed by training with TD-LoRA.

4.3.3 Cross Domains Performance

The optimization process of TD-LoRA is the domain-specific fine-tuning. To shown TD-LoRA obtain better domain-specific fine-tuning, we evaluate the cross domain performance of TD-LoRA. We fine-tuned RoBERTa-large with LoRA and TD-LoRA on RTE tasks, and evaluated the performance of the fine-tuned models on four additional tasks in GLUE. Similarly, we fine-tuned LLaMA-7B on the Alpaca dataset and evaluated the performance of the fine-tuned models on four tasks across different domains in MMLU.

The results are shown in Figure 5. TD-LoRA achieve better cross domain performance than LoRA. The results demonstrate the effectiveness of task decoupling operation.



(a) Roberta-large (b) LLaMA-7b

Figure 5: Cross domains performance of LoRA and TD-LoRA. The blue bars represent results of LoRA, the red bars represent results of TD-LoRA. Figure (a) shows results of pre-trained Roberta-large fine-tuned on RTE task. SST2, MRPC, QNLI and QQP are four tasks in GLUE. Figure (b) shows results of LLaMA-7B fine-tuned on Alpaca data. Abstract_algebra, econometrics, global_facts, marketing are four tasks in MMLU. TD-LoRA helps Roberta-large performs better cross tasks.

5 Conclusion

The performance of the LoRA-based method with the same structure is significantly influenced by downstream tasks. We first analyze the fine-tuning process from the perspective of downstream tasks. We provide an theoretical proof that fine-tuning process encompasses two parameter updating tasks: domain inference fine-tuning and content inference fine-tuning. We first analysis task decoupling in LoRA process and provide a theoretical proof that LoRA also can be decoupled. Based on this theory, we analyze LoRA from the view of task decoupling and propose TD-LoRA. TD-LoRA has two branches to obtain domain-specific fine-tuning. We apply cosine similarity to approximate domain inference task, thus achieve some memory requirements and comparable computational costs with LoRA. Extensive experiments demonstrate the effectiveness of TD-LoRA. In the future, a more optimal function to approximate domain inference is worth researching on.

6 Limitations

It is an NP-hard problem to find $P(W|X)$ because W is a high-dimensional vector with a large number of parameters. We employ cosine similarity between LoRA updates weights and pre-trained weights to approximate domain inference. This lacks a certain level of accuracy. A more accurate domain inference requires further research. Due to limitations in computational resources, we did not conduct experiments on larger models such as LLaMA-13B or LLaMA-70b. The effectiveness of TD-LoRA on bigger LLMs remains to be verified. However, this is precisely the significance of our research on PEFT methods.

References

Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. 2021. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 7319–7328. Association for Computational Linguistics.

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. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Shuming Ma, Zhifang Sui, and Furu Wei. 2023. Why can GPT learn in-context? language models secretly perform gradient descent as meta-optimizers. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 4005–4019. Association for Computational Linguistics.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *CoRR*, abs/2305.14314.

Ning Ding, Xingtai Lv, Qiaosen Wang, Yulin Chen, Bowen Zhou, Zhiyuan Liu, and Maosong Sun. 2023a. Sparse low-rank adaptation of pre-trained language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 4133–4145. Association for Computational Linguistics.

Ning Ding, Xingtai Lv, Qiaosen Wang, Yulin Chen, Bowen Zhou, Zhiyuan Liu, and Maosong Sun. 2023b. Sparse low-rank adaptation of pre-trained language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 4133–4145. Association for Computational Linguistics.

Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, Jing Yi, Weilin Zhao, Xiaozhi Wang, Zhiyuan Liu, Hai-Tao Zheng, Jianfei Chen, Yang Liu, Jie Tang, Juanzi Li, and Maosong Sun. 2023c. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nat. Mac. Intell.*, 5(3):220–235.

Ali Edalati, Marzieh S. Tahaei, Ivan Kobyzev, Vahid Partovi Nia, James J. Clark, and Mehdi Rezagholizadeh. 2022. Krona: Parameter efficient tuning with kronecker adapter. *CoRR*, abs/2212.10650.

Zihao Fu, Haoran Yang, Anthony Man-Cho So, Wai Lam, Lidong Bing, and Nigel Collier. 2023. On the effectiveness of parameter-efficient fine-tuning. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pages 12799–12807. AAAI Press.

Shivam Garg, Dimitris Tsipras, Percy Liang, and Gregory Valiant. 2022. What can transformers learn in-context? A case study of simple function classes. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. 2021. YOLOX: exceeding YOLO series in 2021. *CoRR*, abs/2107.08430.

Almog Gueta, Elad Venezian, Colin Raffel, Noam Slonim, Yoav Katz, and Leshem Choshen. 2023. Knowledge is a region in weight space for fine-tuned language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 1350–1370. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

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. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Chengsong Huang, Qian Liu, Bill Yuchen Lin, Tianyu Pang, Chao Du, and Min Lin. 2023. Lorahub: Efficient cross-task generalization via dynamic lora composition. *CoRR*, abs/2307.13269.

Dawid Jan Kopiczko, Tijmen Blankevoort, and Yuki Markus Asano. 2023. Vera: Vector-based random matrix adaptation. *CoRR*, abs/2310.11454.

Suhas Kotha, Jacob Mitchell Springer, and Aditi Raghunathan. 2023. Understanding catastrophic forgetting in language models via implicit inference. *CoRR*, abs/2309.10105.

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, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 3045–3059. 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, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 4582–4597. Association for Computational Linguistics.

Yingcong Li, Muhammed Emrullah Ildiz, Dimitris Papailiopoulos, and Samet Oymak. 2023. Transformers as algorithms: Generalization and stability in in-context learning. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 19565–19594. PMLR.

Yong Lin, Lu Tan, Hangyu Lin, Zeming Zheng, Renjie Pi, Jipeng Zhang, Shizhe Diao, Haoxiang Wang, Han Zhao, Yuan Yao, and Tong Zhang. 2023. Speciality vs generality: An empirical study on catastrophic forgetting in fine-tuning foundation models. *CoRR*, abs/2309.06256.

Qidong Liu, Xian Wu, Xiangyu Zhao, Yuanshao Zhu, Derong Xu, Feng Tian, and Yefeng Zheng. 2023. Moelora: An moe-based parameter efficient fine-tuning method for multi-task medical applications. *CoRR*, abs/2310.18339.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. 2015. Faster R-CNN: towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 91–99.

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

Zhi Tian, Weilin Huang, Tong He, Pan He, and Yu Qiao. 2016. Detecting text in natural image with connectionist text proposal network. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII*, volume 9912 of *Lecture Notes in Computer Science*, pages 56–72. Springer.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas

Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

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, EACL 2023, Dubrovnik, Croatia, May 2-6, 2023*, pages 3266–3279. Association for Computational Linguistics.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Chengyue Wu, Yukang Gan, Yixiao Ge, Zeyu Lu, Jiahao Wang, Ye Feng, Ping Luo, and Ying Shan. 2024. Llama pro: Progressive llama with block expansion. *CoRR*, abs/2401.02415.

Lingling Xu, Haoran Xie, Si-Zhao Joe Qin, Xiaohui Tao, and Fu Lee Wang. 2023a. Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment. *CoRR*, abs/2312.12148.

Yuhui Xu, Lingxi Xie, Xiaotao Gu, Xin Chen, Heng Chang, Hengheng Zhang, Zhengsu Chen, Xiaopeng Zhang, and Qi Tian. 2023b. Qa-lora: Quantization-aware low-rank adaptation of large language models. *CoRR*, abs/2309.14717.

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, Weng Lam Tam, Zixuan Ma, Yufei Xue, Jidong Zhai, Wenguang Chen, Zhiyuan Liu, Peng Zhang, Yuxiao Dong, and Jie Tang. 2023. GLM-130B: an open bilingual pre-trained model. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Mingyang Zhang, Hao Chen, Chunhua Shen, Zhen Yang, Linlin Ou, Xinyi Yu, and Bohan Zhuang. 2023a. Pruning meets low-rank parameter-efficient fine-tuning. *CoRR*, abs/2305.18403.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. 2023b. Adaptive budget allocation for parameter-efficient fine-tuning. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Bojia Zi, Xianbiao Qi, Lingzhi Wang, Jianan Wang, Kam-Fai Wong, and Lei Zhang. 2023. Delta-lora: Fine-tuning high-rank parameters with the delta of low-rank matrices. *CoRR*, abs/2309.02411.