TeamLoRA: Boosting Low-Rank Adaptation with Expert Collaboration and Competition

Tianwei Lin[1,2], **Jiang Liu**[1,2], **Wenqiao Zhang**[1,†], **Yang Dai**[1], **Haoyuan Li**[2], **Zhelun Yu**[2], **Wanggui He**[2], **Juncheng Li**[1,†], **Jiannan Guo**[1], **Hao Jiang**[2], **Siliang Tang**[1], **Yueting Zhuang**[1],
[1]Zhejiang University, [2]Alibaba Group,
Correspondence[†]**:** {wenqiaozhang, junchengli}@zju.edu.cn

Abstract

While Parameter-Efficient Fine-Tuning (PEFT) methods like Low-Rank Adaptation (LoRA) effectively address resource constraints during fine-tuning, their performance often falls short, especially in multidimensional task scenarios. To address this issue, one straightforward solution is to introduce task-specific LoRA as domain experts, leveraging the modeling of multiple capabilities of experts and thus enhancing the general capability of multi-task learning. Although promising, these additional components often add complexity to the training and inference process, contravening the efficiency that PEFT is designed to deliver. Considering this, we introduce an innovative PEFT method, **TeamLoRA**, consisting of a collaboration and competition module for LoRA experts, thus achieving the right balance of effectiveness and efficiency: **(i)** For *collaboration*, we introduce a novel knowledge sharing and organization mechanism designed to optimize hierarchical learning while enhancing the efficiency of model training and inference. **(ii)** For *competition*, we propose leveraging a game-theoretic interaction mechanism for experts, encouraging experts to transfer their domain-specific knowledge while facing diverse downstream tasks, thus enhancing the performance. By doing so, TeamLoRA elegantly connects the experts as a "*Team*" with internal collaboration and competition, enabling a faster and more accurate PEFT paradigm. Meanwhile, we curate a **Comprehensive Multi-Task Evaluation (CME)** benchmark to thoroughly assess the capability of multi-task learning. Experiments conducted on our CME and other benchmarks indicate the effectiveness and efficiency of TeamLoRA. Our project is available at `https://github.com/DCDmllm/TeamLoRA`.

1 Introduction

Instruction fine-tuning of Large Language Models (LLMs) (Achiam et al., 2023a; Reid et al., 2024; Cai et al., 2024; Yang et al., 2024) and Multimodal

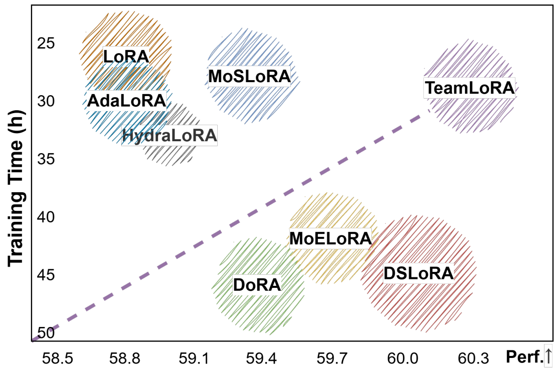

Figure 1: Visualization of various PEFT methods in terms of speed and performance on the CME benchmark — where the area indicates the proportion of trainable parameters — reveals that TeamLoRA not only achieves **superior performance** but also demonstrates **higher efficiency** compared to multi-LoRA methods.

Large Language Models (MLLMs) (Li et al., 2022, 2023c; Huang et al., 2023; Achiam et al., 2023b; Zhang et al., 2024a; Yuan et al., 2024; Zhang et al., 2025; Bai et al., 2025) has achieved impressive proficiency in NLP and multimodal tasks by effectively adapting *task-agnostic* foundations to *task-specific* domains. However, full fine-tuning billions of parameters poses significant challenges in resource-constrained scenarios, thereby limiting the application of LLMs across various downstream tasks. Therefore, Parameter-Efficient Fine-Tuning (PEFT) techniques (Han et al., 2024) emerge with the aim of reducing the cost by fine-tuning a small subset of parameters, offering a streamlined approach for domain adaptation. Among these methods, Low-Rank Adaptation (LoRA) (Hu et al., 2022), a popular PEFT approach, fine-tunes models by adapting lightweight auxiliary modules $\{\boldsymbol{A}, \boldsymbol{B}\}$ on top of pre-trained weights $\boldsymbol{W}_0$, where $\boldsymbol{A}$ and $\boldsymbol{B}$ are low-rank matrices. LoRA offers performance comparable to full fine-tuning when focusing on the one-dimensional domain or task with less computational effort. Nonetheless, qualitative research

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 13622–13637
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

highlights limitations of LoRA in handling multidimensional task scenarios, mainly due to the catastrophic forgetting and interference (Kalajdzievski, 2024; Zhang et al., 2024b) between tasks in the training stage.

One straightforward solution is to adaptively integrate the knowledge diversity of multiple LoRA experts to handle different task characteristics, a method known as *multi-LoRA architecture* (MoELoRA). Specifically, this method adds multiple LoRA experts to the pre-trained linear layer (Gao et al., 2024) and dynamically allocates expert weights through a gating mechanism, thereby enhancing the performance of multi-task learning. Currently, multi-LoRA architecture (Dou et al., 2023; Luo et al., 2024; Li et al., 2024) effectively captures and integrates multi-domain knowledge from multidimensional task scenarios, leading to performance improvements in complex downstream applications.

Despite its promise, MoELoRA may not effectively adapt the multi-task scenario, which can be distilled into two principal aspects: **(i) Training and Inference Efficiency.** Our observations show that MoELoRA fails to effectively balance performance against computational costs, contradicting the efficient characterization of PEFT, as illustrated in Figure 1 (training time is nearly 62% slower compared to LoRA). Additionally, multiplying the number of LoRA experts means introducing a proportional increase in matrix operations, which escalates training costs and inference latency. **(ii) Effectiveness of Expert Combination.** While advanced multi-LoRA architecture-based PEFT methods focus on adaptively selecting a subset of experts for updating, qualitative analysis (Zuo et al., 2021) reveals that commonly-adopted mechanisms suffer from the notorious *load imbalance* and *overconfidence*. Gating mechanisms may not effectively learn task patterns and could lead to weight collapse, causing some experts to consistently dominate. In addition, the uniformity of the structure raises concerns about the homogenization of LoRA experts' knowledge, which may lead to diminishing marginal returns in expert collaboration, hindering the ability to differentiate specific tasks and make precise decisions (Jiang et al., 2024). Summing up, these limitations necessitate a reevaluation of MoELoRA and its solutions for handling multidimensional tasks, with the objective of achieving the right balance between effectiveness and efficiency.

To address the aforementioned limitations, we propose **TeamLoRA**, treating multiple LoRA experts as a "*Team*" that enhances efficiency and effectiveness through internal collaboration and competition. TeamLoRA comprises two key components: **(i) Efficient Collaboration Module:** We design an asymmetric architecture for knowledge sharing among experts, leveraging the hierarchical relationship between matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ (Hayou et al., 2024) to capture diverse features. Matrix $\boldsymbol{A}$ is a domain-agnostic network that encodes general knowledge to capture cross-task shared features; matrix $\boldsymbol{B}$, on the other hand, focuses on task-specific features, enhancing performance through enriched domain knowledge transfer. This structure allows different $\boldsymbol{B}$ matrices to complement $\boldsymbol{A}$, forming a *plug-in* method for knowledge organization, and improves computational efficiency by reducing matrix operations, making it more advantageous compared to MoELoRA. **(ii) Effective Competition Module:** We propose that the knowledge transfer among experts involves a game-theoretic relationship, leading to the introduction of a competitive interaction mechanism. This mechanism adaptively coordinates the participation confidence of experts based on the task, influencing the distribution of expert weights within the gating mechanism. Its aim is to address the problem of overconfident routing in mixture of experts (MoE) (Fedus et al., 2022). We apply the concept of *fuzzy shapley values* to promote finer-grained interactive competition among experts, encouraging the effective transfer of domain-specific knowledge to the corresponding tasks. In summary, TeamLoRA aims to enhance the overall performance of multi-task learning through the integration of collaboration and competition.

To evaluate TeamLoRA in multi-task learning, we introduce a **comprehensive multi-task evaluation (CME)** benchmark, which comprises 2.5 million samples spanning diverse domains and task types. Beyond single-modal fine-tuning, we also investigate the application of our approach for visual instruction tuning in multi-modal architectures (Liu et al., 2024b). The experimental results demonstrate that TeamLoRA achieves an optimal balance between effectiveness and efficiency. Our contributions are as follows:

(i) A collaborative mechanism is designed to facilitate *plug-in* knowledge sharing, significantly reducing computational costs.

(ii) A competition module is proposed that adaptively adjusts expert participation, enhancing

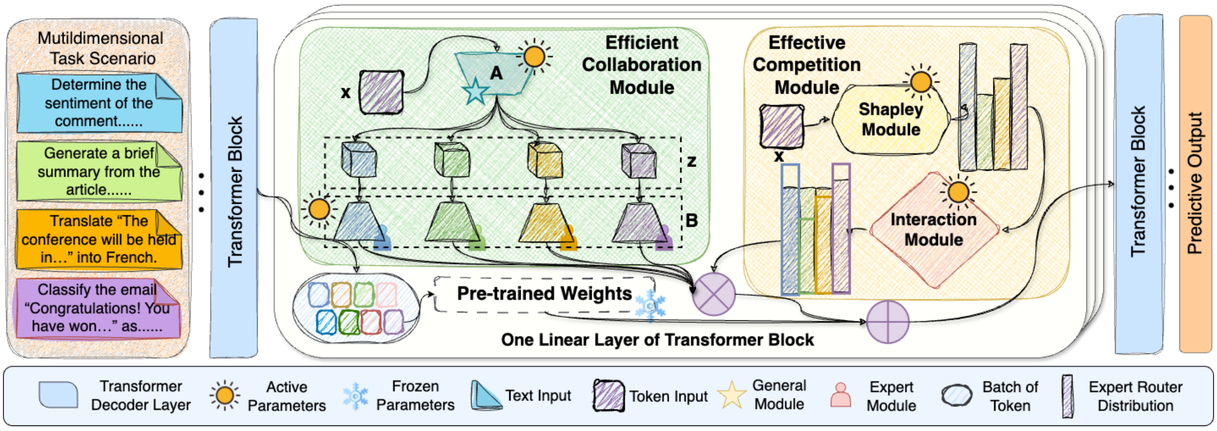
Mutildimensional Task Scenario
Determine the sentiment of the comment......
Generate a brief summary from the article......
Translate "The conference will be held in..." into French.
Classify the email "Congratulations! You have won..." as......
Transformer Block
x
A
Efficient Collaboration Module
z
B
Pre-trained Weights
One Linear Layer of Transformer Block
Effective Competition Module
x
Shapley Module
Interaction Module
Transformer Block
Predictive Output
Transformer Decoder Layer
Active Parameters
Frozen Parameters
Text Input
Token Input
General Module
Expert Module
Batch of Token
Expert Router Distribution

Figure 2: TeamLoRA employs an asymmetric structure consisting of a general module and multiple expert modules, which enhances interactions between experts using a collaboration module and competition module.

domain-specific knowledge transfer.

(iii) The integration of a CME benchmark provides a comprehensive framework for evaluating PEFT methods across a diverse set of tasks and domains, ensuring a thorough assessment of their performance in real-world applications.

2 Related Work

Mixture of Experts. MoE integrates the outputs of multiple experts through a dynamic routing mechanism (Shazeer et al., 2017). Fedus et al. (Fedus et al., 2022) introduced a sparse gating top-k mechanism that activates a subset of experts by the router for each input token, significantly enhancing training and inference speed. To balance the load among experts, GShard (Lepikhin et al., 2020) and OpenMoE (Xue et al., 2024) implemented importance and load losses to ensure a fair distribution among experts, alleviating issues related to tail loss and early routing learning. Additionally, the z-loss from the router improves training stability (Zoph et al., 2022) and addresses the expert balancing problem in multi-task models by maximizing the mutual information between tasks and experts (Chen et al., 2023). Furthermore, certain approaches distinguish between shared and task-specific experts to meet the supplementary knowledge requirements of different tasks, yielding significant results (Liu et al., 2024a).

Parameter-Efficient Fine-Tuning. PEFT (He et al., 2021) reduces the computational cost dependence on fine-tuning large language models (LLMs) by introducing additional modules, thereby replacing the need to update large-scale pre-trained weights. Adapters (Houlsby et al., 2019) introduce extra feature transformations between modules, while prefix tuning (Li and Liang, 2021; Liu et al., 2021) updates parameters through pre-inserted learnable embeddings, and operations on pre-trained weights (Liu et al., 2022) offer a viable solution. Prompt Tuning (Lester et al., 2021; Li et al., 2023b) provides a soft prompt mechanism to condition pre-trained LLMs, enabling efficient adaptation with only a small number of learnable parameters. Low-Rank Adaptation (LoRA) (Hu et al., 2022) and its variants (Yeh et al., 2024; Wu et al., 2024) deliver exceptional performance through low-rank matrix factorization, while AdaLoRA (Zhang et al., 2023) seeks to further optimize the embedding dimensions. Additionally, DoRA (Liu et al., 2024d) decomposes matrices into direction and magnitude to refine learnable parameters, whereas LoHa (Hyeon-Woo et al., 2021) enhances the model's representational capacity using low-rank Hadamard products. In parallel, reasoning-oriented techniques such as Chain-of-Thought prompting (Wei et al., 2022; Yao et al., 2023; Besta et al., 2024; Liu et al., 2024c) provide complementary improvements by guiding multi-step inference, and have been combined with PEFT in recent efforts to boost complex task performance with minimal overhead.

Multi-LoRA Architectures. Multi-LoRA architectures garners widespread attention. Methods based on categorical assignments (Zhao et al., 2024; Feng et al., 2024; Wu et al., 2024) train multiple dedicated LoRAs that dynamically combine when handling complex tasks, providing robust performance. For general scenarios, researchers aim to introduce the dynamic capabilities of MoE, adaptively learning and combining multiple domain experts (Luo et al., 2024; Tian et al., 2024; Gao et al., 2024; Li et al., 2024; Lin et al., 2025). In this work, we propose **TeamLoRA**, designed to mitigate the efficiency limitations of multi-LoRA architectures, offering enhanced performance and faster response times.

3 Methods

3.1 Problem Formulation

In multi-task learning scenarios, PEFT methods adapt to various applications through a lightweight auxiliary module shared among tasks. This multi-task PEFT approach enables the model to remain compact while fulfilling the requirements of multiple tasks. Specifically, PEFT organizes the shared auxiliary module C_{aux} into a pre-trained layer C_{pre} to handle input token sequences $\boldsymbol{x} = (\boldsymbol{x}_i)_{i=1}^{N}$ of different task types, as illustrated below:

$$C_{\text{mix}}(\boldsymbol{x}; \theta_{\text{pre}}, \theta_{\text{aux}}) = C_{\text{pre}}(\boldsymbol{x}; \theta_{\text{pre}}) \oplus C_{\text{aux}}(\boldsymbol{x}; \theta_{\text{aux}}) \,,$$

where θ_{pre} and θ_{aux} denote the parameters of the pre-trained layer and the auxiliary module, respectively. $\oplus$ represents combination strategies based on the method being used.

During training, only the parameters of the auxiliary module are updated, which maintains knowledge stability and reduces computational overhead:

$$\theta_{\text{pre}} \leftarrow \theta_{\text{pre}}, \; \theta_{\text{aux}} \leftarrow \theta_{\text{aux}} - \eta \nabla_{\theta_{\text{aux}}} \mathcal{L}(\boldsymbol{y}, \boldsymbol{y}_{\text{gt}}) \,,$$

where η represents the learning rate and target optimization function $\mathcal{L}$ assesses the deviation between the predicted output $\boldsymbol{y}$ and the ground truth $\boldsymbol{y}_{\text{gt}}$.

3.2 Preliminaries

Low-Rank Adaptation. LoRA (Hu et al., 2022) captures downstream data features by introducing a pair of low-rank matrices as auxiliary modules. The core idea of LoRA is to decompose the auxiliary weight matrix $\Delta\boldsymbol{W} \in \mathbb{R}^{d_{\text{in}} \times d_{\text{out}}}$ of the linear layer into two matrices, $\boldsymbol{A} \in \mathbb{R}^{d_{\text{in}} \times r}$ and $\boldsymbol{B} \in \mathbb{R}^{r \times d_{\text{out}}}$ with r $\ll \min\{d_{\textbf{in}}, d_{\textbf{out}}\}$, reducing the number of learnable parameters. Assuming the origin input to pre-trained weights is $\boldsymbol{x} \in \mathbb{R}^{N \times d_{\text{in}}}$ and the output $\boldsymbol{h} \in \mathbb{R}^{N \times d_{\text{out}}}$ with LoRA can be represented as:

$$\boldsymbol{h} = \boldsymbol{x}\boldsymbol{W}_0 + \boldsymbol{x}\Delta\boldsymbol{W} = \boldsymbol{x}\boldsymbol{W}_0 + \boldsymbol{x}\boldsymbol{A}\boldsymbol{B} \,,$$

where matrix $\boldsymbol{A}$ is initialized with a random Gaussian distribution and matrix $\boldsymbol{B}$ as a zero matrix to ensure that LoRA does not affect the original output at the start of training. Typically, $\Delta\boldsymbol{W}$ is scaled by α/r, using a scaling factor α to adjust the impact of the LoRA module.

Mixture of Experts. MoE (Fedus et al., 2022) greatly expands the model scale while activating only a small number of parameters. In LLMs, MoE duplicates the Feed-Forward Network (FFN) to create a collection of experts, facilitating the transfer of specific knowledge to downstream tasks, thereby enhancing model performance without significantly increasing training time and inference latency. Specifically, MoE constructs a set of k experts, $\{E_i\}_{i=1}^{k}$, and utilizes a linear router R to dynamically allocate a set of weights $\boldsymbol{\omega} = \textit{Softmax}(R(\boldsymbol{x}; \theta_R))$ for token participation. The output of the FFN layer can be represented as $\boldsymbol{y} = C_{\text{ffn}}(\boldsymbol{x}; \theta_{\text{ffn}})$. Correspondingly, the output with MoE is as follows:

$$\boldsymbol{y} = C_{\text{MoE}}(\boldsymbol{x}; \theta_R, \{\theta_{\text{ffn}}^{i}\}_{i=1}^{k}) = \sum_{i=1}^{k} \omega_i E_i(\boldsymbol{x}; \theta_{\text{ffn}}^{i}) \,,$$

where E_i represents i-th extended FFN expert, and θ_{ffn}^{i} denotes the parameters of the corresponding expert.

3.3 TeamLoRA

TeamLoRA facilitates efficient collaboration and effective competition among experts (See Figure 2), optimizing the mechanisms for knowledge sharing and transfer to boost performance:

$$C_{\text{mix}}(\boldsymbol{x}; \boldsymbol{W}_0, \theta_{\text{col}}, \theta_{\text{cop}}) = \boldsymbol{x}\boldsymbol{W}_0 + C_{\text{aux}}(\boldsymbol{x}; \theta_{\text{col}}, \theta_{\text{cop}}) \,,$$

where θ_{col} represents parameters of efficient collaboration module $\mathcal{M}_{\text{col}}$ and θ_{cop} represents parameters of effective competition module $\mathcal{M}_{\text{cop}}$.

Efficient Collaboration among Experts.

We first analyze the multi-LoRA architecture, which employs gate mechanism to dynamically combine the knowledge of LoRA experts $\{E\}_{i=1}^{k}$. Specifically, MoELoRA constructs multiple identical experts to output features for $\{\boldsymbol{A}_i, \boldsymbol{B}_i\}_{i=1}^{k}$:

$$C_{\text{aux}}(\boldsymbol{x}; \theta_R, \{\boldsymbol{A}_i, \boldsymbol{B}_i\}_{i=1}^{k}) = \sum_{i=1}^{k} \omega_i \boldsymbol{x}\boldsymbol{A}_i\boldsymbol{B}_i \,,$$

where $\boldsymbol{\omega}$ is the same as MoE.

In fact, we have two key observations: (i) The stacking of multiple LoRA experts introduces approximately $O(2k)$ multiplications and other operations, significantly impairing the parallel processing capability of GPUs. For instance, in the CME benchmark, when k is 2, 4, and 8, MoELoRA incurs an additional training time of

$\mathbf{19\%}$, $\mathbf{62\%}$, and $\mathbf{138\%}$, respectively, compared to LoRA. (ii) In experiments, MoELoRA employs a homogeneous LoRA stacking design that can somewhat cover the complete features of diverse data (performing better than LoRA in the CME benchmark), but it also exhibits issues of overconfidence; when dynamic routing retains only the best-performing expert, the overall model ensemble performance in the CME benchmark reaches $\mathbf{98.0\%}$. This design simultaneously exposes inherent redundancy among the experts (as the performance of a single static expert is comparable), as they often capture similar knowledge rather than unique expertise, which may limit the overall expressive capability of the expert ensemble (See Appendix D.1 for details). To address this issue, our goal is to optimize the diversity of the knowledge captured by different experts, enabling the model to make more accurate and specialized decisions in a multi-task environment.

In light of the above challenges, TeamLoRA is designed with a hierarchical collaboration module that employs an *expert post-assignment* approach to guide a single matrix $\boldsymbol{A}$ and multiple matrices $\boldsymbol{B}$ towards differentiated feature learning, thereby enhancing overall expressive capability. The general module (matrix $\boldsymbol{A}$) is utilized to capture general features across tasks and subdivides them into fine-grained features across multiple dimensions, while the expert module (matrix $\boldsymbol{B}$) acts as a domain-specific plug-in, facilitating the effective supplementation of specialized knowledge to general knowledge. Together, they collaborate to facilitate the transfer of domain knowledge to specific downstream tasks .

TeamLoRA defines matrix $\boldsymbol{A} \in \mathbb{R}^{d_{\text{in}} \times r_A}$ and k matrices $\boldsymbol{B}_i \in \mathbb{R}^{r_B \times d_{\text{out}}}$, where $r_A = kr_B$. The input $\boldsymbol{x}$ is processed through matrix $\boldsymbol{A}$ to compute an intermediate state $\boldsymbol{z} = \boldsymbol{x}\boldsymbol{A} \in \mathbb{R}^{N \times r_A}$. Then $\boldsymbol{z}$ is evenly split into k segments along its last dimension, a process we refer to as *split*:

$$\boldsymbol{z}_i = \textit{split}(\boldsymbol{z})_i = \boldsymbol{z}[(i-1)_{[r_B+1:ir_B]} \,.$$

Subsequently, each segment $\boldsymbol{z}_i$ undergoes a linear transformation through its corresponding matrix $\boldsymbol{B}_i$. The partial output $\boldsymbol{h}_i \in \mathbb{R}^{N \times d_{\text{out}}}$ as below:

$$\boldsymbol{h}_i = \mathcal{M}_{\text{col}}(\boldsymbol{x}; \boldsymbol{A}, \boldsymbol{B}_i) = \textit{split}(\boldsymbol{x}\boldsymbol{A})_i \boldsymbol{B}_i \,.$$

Unlike the homogeneous multi-LoRA structure of MoELoRA, the direct use of expert weights may implicitly weaken the expectations of TeamLoRA regarding the scaling factor (from α/r to $\alpha/(rk)$), consequently diminishing the influence of LoRA experts on the frozen weights. Moreover, prior studies (Kalajdzievski, 2023a) show that the performance of LoRA is sensitive to the choice of scaling factor. A reduced scaling factor may lead to slower convergence or degraded optimization efficiency. To address this, preserving the expected contribution of each expert becomes crucial for maintaining training stability and ensuring effective integration. Therefore, we ensure invariance of expectation to maintain this stability property. Let the expert weights be represented as $\boldsymbol{\omega}$, the final output of the collaboration module added to LLMs can then be expressed as

$$\boldsymbol{h} = \sum_{i=1}^{k} k\omega_i h_i \,.$$

This operation is regarded as an organization and effective transfer of knowledge conducted by a "*Team*" of LoRA experts.

Through the aforementioned design, the Efficient Collaboration Module enables the general module and expert module to adaptively organize team knowledge, effectively addressing multi-task scenarios. Furthermore, the collaboration module significantly reduces computational costs by minimizing matrix operations. When k is set to 2, 4, 8, and 16, the required training times are only $\mathbf{87\%}$, $\mathbf{70\%}$, $\mathbf{63\%}$, and $\mathbf{58\%}$ of that required by MoELoRA using the same number of experts, thereby successfully achieving the efficiency objectives.

Effective Competition among Experts. Common routing mechanisms have key flaws such as inefficiency in allocation and knowledge silos (Zuo et al., 2021), which contradict the design philosophy. To address this, we introduce a shapley-based mechanism (Shapley et al., 1953) that actively shapes expert competition based on adaptive interactions. This approach prevents centralized decision-making and promotes the effective transfer of expertise to specific downstream tasks. By dynamically adjusting input distribution and expert responsibilities, the competition module ensures more effective and equitable knowledge transfer across tasks.

We first introduce the concept of *fuzzy Shapley values* to offer a perspective on how routers assess the marginal contributions of experts. Unlike

Method	MoE	Rank	Time	Params%	OAI-Sum	IMDB	ANLI	QQP	RTE	WinG	ARC	WQA	NQ	TQA	MMLU	Avg.
Prompt-Tuning	✗	-	23h	0.02	25.3	91.1	44.2	77.0	65.4	59.7	54.8	38.7	16.2	19.4	31.2	47.55
IA3	✗	-	24h	0.03	26.4	92.0	48.7	78.3	68.1	61.5	55.1	37.7	18.8	19.5	34.9	49.18
LoRA	✗	128	26h	2.68	27.3	95.6	56.8	87.4	85.7	71.6	70.8	47.2	25.2	36.8	42.5	58.81
DoRA*	✗	64	46h	2.73	27.4	95.7	57.4	86.9	86.3	73.5	70.6	49.1	25.4	37.9	42.9	59.37
AdaLoRA	✗	128	30h	2.56	27.4	95.5	57.2	87.0	86.3	72.1	71.1	46.8	25.5	35.2	42.9	58.82
MoSLoRA	✗	128	28h	2.70	27.3	95.6	58.3	86.8	86.6	73.2	71.9	47.4	25.8	38.4	41.4	59.34
HydraLoRA	✓	32	34h	1.84	27.6	95.9	57.8	86.5	87.2	70.1	70.2	50.6	24.6	37.0	42.2	59.06
MoELoRA	✓	32	42h	2.71	27.4	95.5	59.3	87.2	86.1	72.9	71.8	50.1	25.1	38.4	42.8	59.69
DSLoRA	✓	32	44h	3.38	27.6	95.6	59.2	87.6	86.9	72.7	72.2	51.1	25.6	38.7	43.0	<u>60.02</u>
TeamLoRA(Ours)	✓	16	28h	1.35	27.4	95.9	59.2	86.6	87.0	73.1	73.1	51.3	25.9	37.1	42.8	59.95
TeamLoRA(Ours)	✓	32	29h	2.71	27.6	95.7	58.9	87.5	87.1	73.8	72.3	51.8	26.4	38.8	43.3	**60.29**

Table 1: Performance comparison of TeamLoRA and other PEFT methods on the CME benchmark. *MoE* indicates whether the MoE architecture is used, *Rank* represents the dimension of the expert modules (r_B for TeamLoRA and r for other methods), *Time* denotes the training time of the model on 8×A800 GPUs, and *Params%* represents the number of learnable parameters. The best results are marked in bold, while the second-best results are underlined.

Rank	Method	OAI-Sum	IMDB	QQP	WinG	NQ	TQA	Rank	Method	OAI-Sum	IMDB	QQP	WinG	NQ	TQA
32	LoRA	27.2	95.6	84.9	65.8	23.3	34.7	64	LoRA	**27.4**	95.7	86.4	70.2	25.6	35.5
8	MoELoRA	27.3	95.5	**86.3**	67.8	21.9	33.7	16	MoELoRA	27.7	95.6	86.3	69.5	24.3	36.4
	TeamLoRA	**27.9**	**96.1**	**86.3**	**68.7**	**24.0**	**35.7**		TeamLoRA	**27.4**	**95.9**	**86.6**	**73.1**	**25.9**	**37.1**
128	LoRA	27.3	95.6	87.4	71.6	25.2	36.8	256	LoRA	26.3	96.0	87.8	71.7	17.5	23.8
32	MoELoRA	27.4	95.5	87.2	72.9	25.1	38.4	64	MoELoRA	**26.9**	**96.2**	87.3	71.8	21.8	35.1
	TeamLoRA	**27.6**	**95.7**	**87.5**	**73.8**	**26.4**	**38.8**		TeamLoRA	**26.9**	95.4	**88.1**	**71.9**	**21.9**	**35.5**

Table 2: Performance of different methods across various tasks with different ranks.

the traditional binary participation (participation or absence), fuzzy Shapley values permit participation degrees to range from 0 to 1. The following equation is the marginal contribution of experts:

$$\phi_i(\boldsymbol{x};\omega_i) = \int_s \left(v_i(\boldsymbol{x}, w_i, s) - v_i(\boldsymbol{x}, 0, s)\right)\, ds\,,$$

where $\phi_i(\boldsymbol{x};\omega_i)$ represents the marginal contribution of expert i with participation degree ω_i, and s denotes the space of possible participation degrees for the remaining experts, satisfying $\sum_j s_j = 1 - \omega_i$ and $j \neq i$. $v_i(\boldsymbol{x}, \omega_i, s)$ represents the total payoff from the combined participation $\{\omega_i\} + s$.

From the perspective of shapley values, the mechanism of the router can be understood as assessing the average marginal contributions of each expert across all possible combinations of experts. This provides a theoretical basis for the allocation of activation weights and highlights the importance of considering synergistic effects among experts. Although calculating shapley values is an NP-hard problem in practical applications, we can use an MLP as an approximation module for fuzzy Shapley values, estimating the marginal contributions:

$$\phi_i(\boldsymbol{x};\theta_S) \leftarrow \mathit{Softmax}(S(\boldsymbol{x};\theta_S))_i\,,$$

where ϕ_i represents the fuzzy Shapley value of the i-th expert and S calculates Shapley value.

To fully capture the competitive dynamics among experts, we introduce an interaction matrix that evaluates and adjusts their interactions. This matrix captures the mutual influences among experts and adjusts their participation based on Shapley interactions. Specifically, the interaction matrix $\boldsymbol{M}$ is designed to adaptively adjust each expert's participation based on their competitive relationships, as detailed below:

$$\omega_i = \mathcal{M}_{\text{cop}}(\boldsymbol{x};\theta_S,\boldsymbol{M}) = \sum_{j=1}^{k} k\boldsymbol{M}_{ij}\phi_j(\boldsymbol{x};\theta_S)\,,$$

where ω_i represents the adjusted optimal degree of participation, and $\boldsymbol{M}_{ij}$ denotes the element in the interaction matrix reflecting the influence of expert j on expert i. The interaction matrix $\boldsymbol{M}$ is initialized as a learnable unit diagonal matrix to ensure self-influence during the initial stages, taking into full account the synergistic effects among experts while adequately capturing the competitive relationships. And k is utilized to correct the scaling factor degradation introduced by multi-LoRA architecture. Ultimately, the output of TeamLoRA is represented as:

$$\boldsymbol{h} = \boldsymbol{x}\boldsymbol{W}_0 + \mathcal{M}_{\text{col}}(\boldsymbol{x};\boldsymbol{A},\{\boldsymbol{B}_i\}_{i=1}^{k}) \odot \mathcal{M}_{\text{cop}}(\boldsymbol{x};\theta_S,\boldsymbol{M}),$$

where $\odot$ represents the element-wise product.

Cop	Col	Avg$_{r=8}$	Avg$_{r=16}$	Avg$_{r=32}$	Avg$_{r=64}$
-	-	57.65	59.08	59.69	58.88
✓	-	58.18	59.25	59.77	**59.07**
-	✓	58.27	59.77	60.24	58.87
✓	✓	**58.31**	**59.95**	**60.29**	58.94

Table 3: Ablation analysis for collaboration and competition modules.

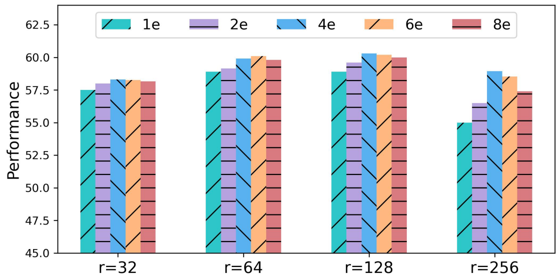

Figure 3: Comparison of different rank setting.

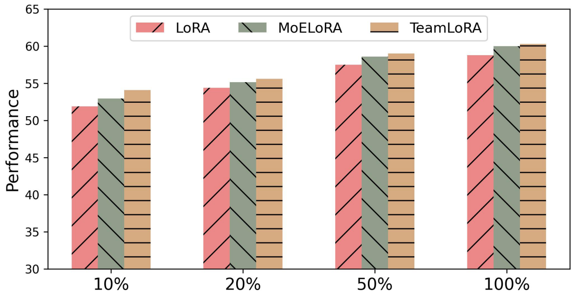

Figure 4: Comparison of different data scales.

4 Experiments

4.1 Benchmark and Setting

Benchmark. All PEFT methods use the 2.5M training set from 22 datasets effectively organized by CME (refer to Appendix A) and are comprehensively evaluated on tasks across 11 different tasks: OpenAI-Summarize-TLDR (Stiennon et al., 2020), IMDB (Maas et al., 2011), ANLI (Nie et al., 2020), QQP (Wang et al., 2017), RTE (Wang et al., 2019), WinoGrande (Sakaguchi et al., 2021), ARC (Clark et al., 2018), WebQA (Li et al., 2016), NQ (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), and MMLU (Hendrycks et al., 2021).

Training Details. We select the LLaMA-2 7B (Touvron et al., 2023) as the base model and continue pre-training it on the expanded Chinese LLaMA-2-7B corpus (Cui et al., 2023a) to enhances the model's knowledge capacity and multilingual capability by expanding the vocabulary and incorporating general corpora. To ensure fairness, for all LoRA-based PEFT methods, we add parameters only to the FFN module and maintain nearly identical parameter increments within the same experimental setup to minimize the potential impact of parameter size on performance. All experiments are conducted on 8×A800 GPUs, using the same hyperparameter settings listed in Appendix B.

Comparison of Methods. To evaluate the superiority of TeamLoRA, we select several prominent PEFT methods, including Prompt-Tuning (Lester et al., 2021), IA3 (Liu et al., 2022), LoRA (Hu et al., 2022), DoRA (Liu et al., 2024d), AdaLoRA (Zhang et al., 2023) and MoSLoRA (Wu et al., 2024). We also compare methods based on MoE, including MoELoRA (Luo et al., 2024) and HydraLoRA (Tian et al., 2024), which encompass different structural and routing mechanisms. Additionally, we analyze the state-of-the-art MoE architectures employed in LLMs, where the design of DeepSeek-V3(Liu et al., 2024a) integrates a shared expert with multiple specialized experts, referred to as DSLoRA.

We also conduct evaluations on Llama-3 8B (Dubey et al., 2024) and LLaVA-1.5 7B (Liu et al., 2024b) to further explore the generalization ability of the proposed method across different base models and modality settings.

4.2 Overall Performance

We evaluate the performance of TeamLoRA in a multi-task learning scenarios using the CME benchmark, compared to other PEFT methods as shown in Table 1. Our observations are summarized as follows: (i) TeamLoRA (Rank=32) shows the best or second-best performance across multiple tasks, **with an average score of 60.29, significantly higher than other PEFT methods**. Particularly, it achieves the best performance on MMLU, demonstrating TeamLoRA's strong capability in handling multi-domain tasks. (ii) Despite a training time of 28 hours for TeamLoRA (Rank=16), slightly longer than baseline methods like LoRA, Prompt-Tuning, and IA3, it achieves competitive average scores of **59.95** with half the parameter count, highlighting its efficient parameter utilization. (iii) Compared to other multi-LoRA architectures, our approach shows significant performance improvements with less training costs significantly, approximately 70% of MoELoRA, 76% of MixLoRA, and 66% of DSLoRA. This results demonstrate effective balance of TeamLoRA between efficiency and

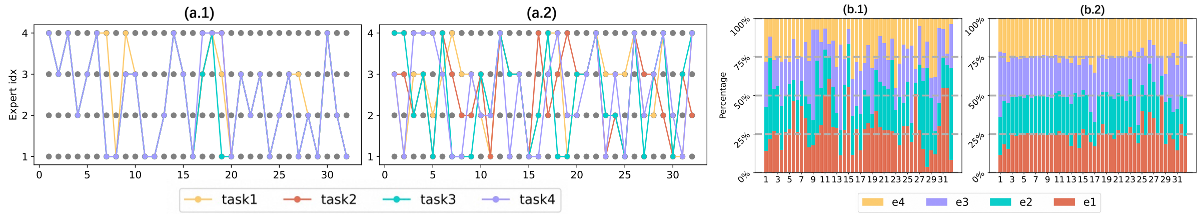

Figure 5: Deep analysis of router. (a) Forward path of expert. (b) Router load visualization.

Method		OAI-Sum	IMDB	ANLI	QQP	RTE	WinG	ARC	WQA	NQ	TQA	MMLU	Avg.
Llama-3-8B +	LoRA$_{r=32}$	24.3	**95.1**	47.2	78.9	80.1	**58.3**	**70.6**	34.6	19.3	37.1	**52.2**	54.34
	MoELoRA$_{r=8}$	24.8	94.9	47.6	**79.0**	81.0	58.2	69.8	35.1	20.2	40.4	49.2	54.56
	TeamLoRA$_{r=8}$	**25.2**	94.2	**49.7**	**79.0**	**81.4**	**58.3**	70.1	**36.1**	**22.2**	**41.3**	52.1	**55.42**

Table 4: Performance analysis based on different LLM Model.

Method		MME	MMB	MMB-CN	SEED	POPE	SQA-I	VQA-T	MM-Vet	VizWiz	Avg.
LLaVA-1.5-7B +	LoRA$_{r=32}$	1505.2	**62.8**	53.7	**60.2**	84.8	67.8	56.9	30.2	48.4	60.01
	MoELoRA$_{r=8}$	1472.7	62.3	53.8	59.5	84.4	**68.7**	**57.1**	30.1	48.7	59.80
	TeamLoRA$_{r=8}$	**1513.5**	62.6	**54.0**	60.0	**85.3**	**68.7**	**57.1**	**31.2**	**49.4**	**60.44**

Table 5: Performance analysis of MLLM on diverse multimodal benchmarks.

effectiveness.

4.3 Quantitative Analysis

Analysis of Parameter Scales. We explore performance of our approach in multi-task learning across different parameter scales as shown in Table 2. Experiments demonstrate that TeamLoRA performs exceptionally well across various parameter configurations, indicating that our approach consistently exhibits superior performance compared to MoELoRA. Notably, with an increase in parameter size, LoRA encounters catastrophic forgetting, as evidenced by a sharp decline in scores for TQA (close book QA). In contrast, both MoELoRA and TeamLoRA alleviate this knowledge collapse, reflecting the stability of their adaptive mechanisms.

Ablation Analysis. We conduct a study on the collaborative and competitive modules. As shown in Table 3, the individual modules and their combinations significantly enhance TeamLoRA's adaptability in multi-task scenarios. The collaborative module is based on the "*Team*" architecture of knowledge sharing, which facilitates overall expression among experts, thereby achieving *plug-in* knowledge integration. In contrast, the competitive module focuses on the interactions among experts, adaptively adjusting the model's preference for specific knowledge transfer to downstream tasks. It is noteworthy that when the number of experts is 64, the collaborative module exhibited a slight negative optimization compared to the multi-LoRA architecture (from 58.88 to 58.87), which we attribute to the extreme rank settings leading to gradient norm vanishing (Kalajdzievski, 2023b), while MoELoRA did not reach this bottleneck due to its split specialists. This limitation can be further mitigated by adjusting the scaling factor — for example, rsLoRA adopts a scaling strategy of $\alpha/\sqrt{r}$ to maintain consistency between parameter efficiency and performance, which can be seamlessly integrated with TeamLoRA. Under standard settings, the collaborative and competitive modules are generally capable of finding near-optimal configurations, as extreme parameter choices (e.g., rank $= 64 \times 4$) are rarely selected in practice.

4.4 In-Depth Analysis

Stability Analysis. We explore performance across varying configurations of expert numbers, as illustrated in Figure 3. Results show a progressive improvement when increasing the number of experts from 1 to 4; however, performance declines when expanding to 8 experts. This indicates that performance does not always scale positively with the number of experts — an appropriate number of experts is essential for balancing efficiency and effectiveness. In addition, as shown in Figure 4, TeamLoRA consistently outperforms across different data scales, ranging from as little as 10% to the full dataset. This consistent advantage highlights the broad adaptability of our proposed method in multi-task learning scenarios.

Expert Load Analysis. We observe the expert paths of MoELoRA across four tasks. The features exhibit overconfidence (Figure 5a.1) in the model's forward path. In contrast, TeamLoRA effectively learns task-specific models by assigning different expert modules as *plug-in* for knowledge combinations (Figure 5a.2). Furthermore, we conduct balanced load testing on 57 tasks in MMLU showed in Figure 5b.1 (MoELoRA) and Figure 5b.2 (TeamLoRA). Our approach demonstrate better load balancing, ensuring greater model stability.

Different Base Models Comparison. To explore the performance of TeamLoRA on other models, we replace the base model with the more powerful Llama-3 8B (Dubey et al., 2024) and conduct a comprehensive comparison of the CME benchmark. Table 4 shows the results of this experiment, where TeamLoRA consistently displays the best performance. This indicates that our approach maintains its advantages across different base models.

Performance Analysis of MLLM. We further expand the applicability of our approach by extending the model from single-modal to multimodal. We fine-tune the LLaVA-1.5 7B (Liu et al., 2024b) model and evaluated it on nine benchmark tests, including MME (Fu et al., 2023), MMB and MMB-CN (Liu et al., 2023), SEED (Li et al., 2023a), POPE (Li et al., 2023d), SQA-I (Lu et al., 2022), VQA-T (Singh et al., 2019), MM-Vet (Yu et al., 2023), and VizWiz (Gurari et al., 2018). As seen, TeamLoRA achieves the best performance on the majority of benchmarks (see Table 5), indicating that our approach demonstrates strong generalizability in multimodal scenarios.

5 Conclusion

TeamLoRA introduces an innovative PEFT framework that integrates collaborative and competitive modules to address the limitations of existing methods in multi-task learning. While traditional PEFT approaches like LoRA are resource-efficient, they often underperform in complex task scenarios. By treating task-specific LoRA modules as domain experts, TeamLoRA enables structured collaboration for efficient knowledge sharing and competition to encourage specialization and generalization.

This design achieves a balance between effectiveness and efficiency. On the proposed CME benchmark and other standard tasks, TeamLoRA demonstrates faster inference and superior performance compared to existing PEFT methods. These results underscore its potential as a generalizable and scalable solution for multi-task adaptation. Future work will further explore game-theoretic strategies to deepen the collaborative–competitive dynamic.

Limitations

Our limitations and potential risks are as follows:

An excessive number of expert modules. Although TeamLoRA significantly reduces the computational costs associated with multiple LoRA architectures, having too many expert modules still leads to an unacceptably high training complexity.

More model Architecture. We have validated the superiority of TeamLoRA with Transformer-based autoregressive models, and this advantage can be further verified with other network architectures.

Acknowledgement

This work has been supported in part by the NSFC (No. 62402426, 62272411), the Key Research and Development Projects in Zhejiang Province (No. 2025C01128, 2024C01106, 2025C01030, 2025C02156), Ningbo Yongjiang Talent Introduction Programme (2023A-400-G), Zhejiang University Education Foundation Qizhen Scholar Foundation.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023a. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023b. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. 2025. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*.

BELLEGroup. 2023. Belle: Be everyone's large language model engine. https://github.com/LianjiaTech/BELLE.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, et al. 2024. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17682–17690.

Zheng Cai, Maosong Cao, Haojiong Chen, Kai Chen, Keyu Chen, Xin Chen, Xun Chen, Zehui Chen, Zhi Chen, Pei Chu, et al. 2024. Internlm2 technical report. *arXiv preprint arXiv:2403.17297*.

Yingshan Chang, Mridu Narang, Hisami Suzuki, Guihong Cao, Jianfeng Gao, and Yonatan Bisk. 2022. Webqa: Multihop and multimodal qa. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16495–16504.

Sahil Chaudhary. 2023. Code alpaca: An instruction-following llama model for code generation. https://github.com/sahil280114/codealpaca.

Zitian Chen, Yikang Shen, Mingyu Ding, Zhenfang Chen, Hengshuang Zhao, Erik G Learned-Miller, and Chuang Gan. 2023. Mod-squad: Designing mixtures of experts as modular multi-task learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11828–11837.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv:1803.05457v1*.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. Free dolly: Introducing the world's first truly open instruction-tuned llm.

Yiming Cui, Ziqing Yang, and Xin Yao. 2023a. Efficient and effective text encoding for chinese llama and alpaca. *arXiv preprint arXiv:2304.08177*.

Yiming Cui, Ziqing Yang, and Xin Yao. 2023b. Efficient and effective text encoding for chinese llama and alpaca. *arXiv preprint arXiv:2304.08177*.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *Machine learning challenges workshop*, pages 177–190. Springer.

Shihan Dou, Enyu Zhou, Yan Liu, Songyang Gao, Jun Zhao, Wei Shen, Yuhao Zhou, Zhiheng Xi, Xiao Wang, Xiaoran Fan, et al. 2023. Loramoe: Revolutionizing mixture of experts for maintaining world knowledge in language model alignment. *arXiv preprint arXiv:2312.09979*, 4(7).

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39.

Wenfeng Feng, Chuzhan Hao, Yuewei Zhang, Yu Han, and Hao Wang. 2024. Mixture-of-loras: An efficient multitask tuning for large language models. *arXiv preprint arXiv:2403.03432*.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, et al. 2023. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*.

Chongyang Gao, Kezhen Chen, Jinmeng Rao, Baochen Sun, Ruibo Liu, Daiyi Peng, Yawen Zhang, Xiaoyuan Guo, Jie Yang, and VS Subrahmanian. 2024. Higher layers need more lora experts. *Preprint*, arXiv:2402.08562.

Danna Gurari, Qing Li, Abigale J Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P Bigham. 2018. Vizwiz grand challenge: Answering visual questions from blind people. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3608–3617.

Zeyu Han, Chao Gao, Jinyang Liu, Sai Qian Zhang, et al. 2024. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*.

Soufiane Hayou, Nikhil Ghosh, and Bin Yu. 2024. Lora+: Efficient low rank adaptation of large models. *arXiv preprint arXiv:2402.12354*.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2021. Towards a unified view of parameter-efficient transfer learning. *arXiv preprint arXiv:2110.04366*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

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 *International conference on machine learning*, 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 *International Conference on Learning Representations*.

Shaohan Huang, Li Dong, Wenhui Wang, Yaru Hao, Saksham Singhal, Shuming Ma, Tengchao Lv, Lei Cui, Owais Khan Mohammed, Qiang Liu, et al. 2023. Language is not all you need: Aligning perception with language models. *arXiv preprint arXiv:2302.14045*.

Nam Hyeon-Woo, Moon Ye-Bin, and Tae-Hyun Oh. 2021. Fedpara: Low-rank hadamard product for communication-efficient federated learning. *arXiv preprint arXiv:2108.06098*.

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. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. triviaqa: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. *arXiv e-prints*, arXiv:1705.03551.

Damjan Kalajdzievski. 2023a. A rank stabilization scaling factor for fine-tuning with lora. *arXiv preprint arXiv:2312.03732*.

Damjan Kalajdzievski. 2023b. A rank stabilization scaling factor for fine-tuning with lora. *Preprint*, arXiv:2312.03732.

Damjan Kalajdzievski. 2024. Scaling laws for forgetting when fine-tuning large language models. *Preprint*, arXiv:2401.05605.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Ariel N Lee, Cole J Hunter, and Nataniel Ruiz. 2023. Platypus: Quick, cheap, and powerful refinement of llms. *arXiv preprint arXiv:2308.07317*.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. 2020. Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668*.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *Preprint*, arXiv:2104.08691.

Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. 2023a. Seed-bench: Benchmarking multimodal llms with generative comprehension. *arXiv preprint arXiv:2307.16125*.

Dengchun Li, Yingzi Ma, Naizheng Wang, Zhengmao Ye, Zhiyuan Cheng, Yinghao Tang, Yan Zhang, Lei Duan, Jie Zuo, Cal Yang, and Mingjie Tang. 2024. Mixlora: Enhancing large language models fine-tuning with lora-based mixture of experts. *Preprint*, arXiv:2404.15159.

Juncheng Li, Minghe Gao, Longhui Wei, Siliang Tang, Wenqiao Zhang, Mengze Li, Wei Ji, Qi Tian, Tat-Seng Chua, and Yueting Zhuang. 2023b. Gradient-regulated meta-prompt learning for generalizable vision-language models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2551–2562.

Juncheng Li, Kaihang Pan, Zhiqi Ge, Minghe Gao, Wei Ji, Wenqiao Zhang, Tat-Seng Chua, Siliang Tang, Hanwang Zhang, and Yueting Zhuang. 2023c. Fine-tuning multimodal llms to follow zero-shot demonstrative instructions. *arXiv preprint arXiv:2308.04152*.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, pages 12888–12900. PMLR.

Peng Li, Wei Li, Zhengyan He, Xuguang Wang, Ying Cao, Jie Zhou, and Wei Xu. 2016. Dataset and neural recurrent sequence labeling model for open-domain factoid question answering. *Preprint*, arXiv:1607.06275.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. 2023d. Evaluating object hallucination in large vision-language models. *arXiv preprint arXiv:2305.10355*.

Wing Lian, Guan Wang, Bleys Goodson, Eugene Pentland, Austin Cook, Chanvichet Vong, and "Teknium". 2023. Slimorca: An open dataset of gpt-4 augmented flan reasoning traces, with verification.

Tianwei Lin, Wenqiao Zhang, Sijing Li, Yuqian Yuan, Binhe Yu, Haoyuan Li, Wanggui He, Hao Jiang, Mengze Li, Xiaohui Song, et al. 2025. Healthgpt: A medical large vision-language model for unifying comprehension and generation via heterogeneous knowledge adaptation. *arXiv preprint arXiv:2502.09838*.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. 2024a. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin A Raffel. 2022. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems*, 35:1950–1965.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024b. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306.

Jiang Liu, Bolin Li, Haoyuan Li, Tianwei Lin, Wenqiao Zhang, Tao Zhong, Zhelun Yu, Jinghao Wei, Hao Cheng, Wanggui He, et al. 2024c. Boosting private domain understanding of efficient mllms: A tuning-free, adaptive, universal prompt optimization framework. *arXiv preprint arXiv:2412.19684*.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. 2024d. Dora: Weight-decomposed low-rank adaptation. *arXiv preprint arXiv:2402.09353*.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Lam Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2021. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv preprint arXiv:2110.07602*.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. 2023. Mmbench: Is your multi-modal model an all-around player? *arXiv preprint arXiv:2307.06281*.

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in Neural Information Processing Systems*, 35:2507–2521.

Tongxu Luo, Jiahe Lei, Fangyu Lei, Weihao Liu, Shizhu He, Jun Zhao, and Kang Liu. 2024. Moelora: Contrastive learning guided mixture of experts on parameter-efficient fine-tuning for large language models. *arXiv preprint arXiv:2402.12851*.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. 2023. Orca: Progressive learning from complex explanation traces of gpt-4. *arXiv preprint arXiv:2306.02707*.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial nli: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*.

Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.

Elvis Saravia, Hsien-Chi Toby Liu, Yen-Hao Huang, Junlin Wu, and Yi-Shin Chen. 2018. CARER: Contextualized affect representations for emotion recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3687–3697, Brussels, Belgium. Association for Computational Linguistics.

Lloyd S Shapley et al. 1953. A value for n-person games.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. 2019. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8317–8326.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021.

Chunlin Tian, Zhan Shi, Zhijiang Guo, Li Li, and Chengzhong Xu. 2024. Hydralora: An asymmetric lora architecture for efficient fine-tuning. *arXiv preprint arXiv:2404.19245*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

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. *Preprint*, arXiv:1804.07461.

Zhiguo Wang, Wael Hamza, and Radu Florian. 2017. Bilateral multi-perspective matching for natural language sentences. *arXiv preprint arXiv:1702.03814*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Taiqiang Wu, Jiahao Wang, Zhe Zhao, and Ngai Wong. 2024. Mixture-of-subspaces in low-rank adaptation. *arXiv preprint arXiv:2406.11909*.

Ge Zhang Yao Fu Wenhao Huang Huan Sun Yu Su Wenhu Chen Xiang Yue, Xingwei Qu. 2023. Mammoth: Building math generalist models through hybrid instruction tuning. *arXiv preprint arXiv:2309.05653*.

Fuzhao Xue, Zian Zheng, Yao Fu, Jinjie Ni, Zangwei Zheng, Wangchunshu Zhou, and Yang You. 2024. Openmoe: An early effort on open mixture-of-experts language models. *arXiv preprint arXiv:2402.01739*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822.

Shih-Ying Yeh, Yu-Guan Hsieh, Zhidong Gao, Bernard B W Yang, Giyeong Oh, and Yanmin Gong. 2024. Navigating text-to-image customization: From lycoris fine-tuning to model evaluation. *Preprint*, arXiv:2309.14859.

Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. 2023. Mm-vet: Evaluating large multimodal models for integrated capabilities. *arXiv preprint arXiv:2308.02490*.

Yuqian Yuan, Hang Zhang, Wentong Li, Zesen Cheng, Boqiang Zhang, Long Li, Xin Li, Deli Zhao, Wenqiao Zhang, Yueting Zhuang, et al. 2024. Videorefer suite: Advancing spatial-temporal object understanding with video llm. *arXiv preprint arXiv:2501.00599*.

Boqiang Zhang, Kehan Li, Zesen Cheng, Zhiqiang Hu, Yuqian Yuan, Guanzheng Chen, Sicong Leng, Yuming Jiang, Hang Zhang, Xin Li, et al. 2025. Videollama 3: Frontier multimodal foundation models for image and video understanding. *arXiv preprint arXiv:2501.13106*.

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

Wenqiao Zhang, Tianwei Lin, Jiang Liu, Fangxun Shu, Haoyuan Li, Lei Zhang, He Wanggui, Hao Zhou, Zheqi Lv, Hao Jiang, et al. 2024a. Hyperllava: Dynamic visual and language expert tuning for multimodal large language models. *arXiv preprint arXiv:2403.13447*.

Wenqiao Zhang, Zheqi Lv, Hao Zhou, Jia-Wei Liu, Juncheng Li, Mengze Li, Yunfei Li, Dongping Zhang, Yueting Zhuang, and Siliang Tang. 2024b. Revisiting the domain shift and sample uncertainty in multi-source active domain transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16751–16761.

Wenqiao Zhang, Haochen Shi, Siliang Tang, Jun Xiao, Qiang Yu, and Yueting Zhuang. 2021. Consensus graph representation learning for better grounded image captioning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 3394–3402.

Ziyu Zhao, Leilei Gan, Guoyin Wang, Wangchunshu Zhou, Hongxia Yang, Kun Kuang, and Fei Wu. 2024. Loraretriever: Input-aware lora retrieval and composition for mixed tasks in the wild. *arXiv preprint arXiv:2402.09997*.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2024. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36.

Barret Zoph, Irwan Bello, Sameer Kumar, Nan Du, Yanping Huang, Jeff Dean, Noam Shazeer, and William Fedus. 2022. St-moe: Designing stable and transferable sparse expert models. *arXiv preprint arXiv:2202.08906*.

Simiao Zuo, Xiaodong Liu, Jian Jiao, Young Jin Kim, Hany Hassan, Ruofei Zhang, Tuo Zhao, and Jianfeng Gao. 2021. Taming sparsely activated transformer with stochastic experts. *arXiv preprint arXiv:2110.04260*.

Appendix

This section is the appendix for "TeamLoRA: Boosting Low-Rank Adaptation with Expert Collaboration and Competition", providing additional technical details.

A CME Benchmark

A.1 Data Organization and Categorization

To evaluate the performance of PEFT in multidimensional task understanding and sensitivity to domain conflicts, we constructe a large and diverse composite dataset. The dataset encompasses various domains and scenarios and has undergone detailed formatting and content alignment. As shown in Table 7, the training set organized by CME includes the following task scenarios: **(1) Formalized Closed-Book QA. (2) Closed-Book QA. (3) Coreference Resolution. (4) Text Generation. (5) Code Generation. (6) Mathematical Generation. (7) Question Answering. (8) Complex Language Modelling. (9) Natural Language Inference. (10) Sentiment Analysis. (11) Summarization.**

Given the significant impact of data mixing ratios on model generation quality, we conducte experiments with small sample data to assess how different ratios affect generation quality. These findings are then extrapolated to a training set with 2.5M samples to ensure diversity.

A.2 Data Format

Our dataset is organized in a triplet format, comprising the following three components:

- **Task Instruction:** Instructions for the task, including objectives and format requirements. Note that some task scenarios may not have explicit Task Instructions; in such cases, the Task Question encompasses all necessary information.
- **Task Question:** A complete and standardized description of the question.
- **Response:** The standard answer based on the Task Instruction and Task Question.

For data originally not formatted as QA, **manual rewriting** is performed to meet the triplet structure requirements.

A.3 Evaluation

We extracte 11 different task types from the dataset as benchmark for evaluation. Detailed descriptions and classification information for each task are provided in Table 8. These tasks cover a variety of response formats and scenarios to thoroughly test the performance and adaptability of models.

B Training Settings

For all LLaMA series fine-tuning, TeamLoRA uses a unified set of hyperparameters:

Hyperparameter	Value
Learning Rate	0.0001
Batch Size	64
Number of Epochs	1
Optimizer	AdamW
Weight Decay	0.0
Dropout Rate	0.05
Warm-up Ratio	0.03
Max Sequence Length	2048
α	$4 * r_B$
Target Module	up, gate, down

Table 6: Basic Training Hyperparameters of LLaMA

Multimodal Large Language Models (Huang et al., 2023; Achiam et al., 2023b; Zhang et al., 2024a, 2025; Bai et al., 2025) have demonstrated remarkable potential in tasks such as visual grounding, image understanding, and caption generation (Zhang et al., 2021; Li et al., 2022, 2023c; Yuan et al., 2024). In LLaVA-1.5, we skipp the feature alignment stage and directly tested the performance of TeamLoRA during the visual instruction tuning stage. The only difference from LLaMA is that we assigned $Lr_1 = 0.00002$ to the projector, allocated $Lr_2 = 0.0002$ to the LoRA part of TeamLoRA, and increased $B = 128$ to maintain consistency with the official implementation.

C Compared Methods

To evaluate the superiority of TeamLoRA, we select several prominent PEFT methods, including Prompt-Tuning (Lester et al., 2021), IA3 (Liu et al., 2022), LoRA (Hu et al., 2022), DoRA (Liu et al., 2024d), AdaLoRA (Zhang et al., 2023) and MoSLoRA (Wu et al., 2024). We also compare methods based on MoE, including MoELoRA (Luo et al., 2024) and HydraLoRA (Tian et al., 2024), which encompass different structural and routing mechanisms. Additionally, we analyze the state-of-the-art MoE architectures employed in LLMs, where the design of DeepSeek-V3(Liu et al., 2024a) integrates a shared expert with multiple specialized experts, referred to as DSLoRA.

Task Name	#Train	Task Type
ARC (Clark et al., 2018)	4239	Formalized Closed-Book QA
Natural Questions (Kwiatkowski et al., 2019)	79168	Closed-Book QA
TriviaQA (Joshi et al., 2017)	61688	Closed-Book QA
WebQA (Chang et al., 2022)	36174	Formalized Closed-Book QA
WinoGrande (Sakaguchi et al., 2021)	40398	Coreference Resolution
Alpaca-GPT4-data (Peng et al., 2023)	90000	Text Generation
BELLE (BELLEGroup, 2023)	514062	Text Generation
Code-Alpaca-20K (Chaudhary, 2023)	18019	Code Generation
LIMA-sft (Zhou et al., 2024)	1330	Question Answering
MathInstruct (Xiang Yue, 2023)	259418	Mathematical Generation
ruozhiba (Cui et al., 2023b)	4408	Question Answering
stem-zh-instruction (Cui et al., 2023b)	230567	Question Answering
databricks-dolly-15K (Conover et al., 2023)	15011	Complex Language Modelling
Open-Platypus (Lee et al., 2023)	24926	Complex Language Modelling
OpenOrca (Mukherjee et al., 2023)	331632	Complex Language Modelling
SlimOrca (Lian et al., 2023)	484065	Complex Language Modelling
ANLI (Nie et al., 2020)	81433	Natural Language Inference
QQP (Wang et al., 2018)	72770	Natural Language Inference
RTE (Dagan et al., 2005)	2490	Natural Language Inference
Emotion (Saravia et al., 2018)	18000	Sentiment Analysis
IMDB (Maas et al., 2011)	45000	Sentiment Analysis
OpenAI-Summarize-TLDR (Stiennon et al., 2020)	116722	Summarization

Table 7: Detailed description of the training set in the CME benchmark

Prompt-Tuning introduces a novel method for improving the performance and stability of pre-trained LLM adaptation. It combines continuous prompt embeddings to the input with standard discrete prompts. These continuous prompts are all learned through backpropagation, allowing them to dynamically adjust and compensate for minor variations in the discrete prompts, thus enhancing training stability.

IA3 is a PEFT method for pre-trained LLMs. By injecting learned vectors into the internal activation of the network's layers,it enables the model with minimal additional parameters. The method is accomplished by applying these vectors to the keys and values in the self-attention and encoder-decoder attention mechanism, as well as the internal activation to the position-wise feedforward network.

AdaLoRA expands on the idea behind LoRA by introducing an element: adaptive budget allocation. It adapts the rank of every incremental matrix by modulating its importance, instead of sharing parameter budget evenly among all weight matrices. This is achieved through Singular Value Decomposition (SVD) parameterization and an importance score that considers the contribution of each triplet to the model performance. AdaLoRA is able to learn from data of low-budget and it enhances the performance effectively by proposing a component scheduling mechanism that prevents overfitting.

LoRA presents a way to fine-tune LLMs efficiently for various tasks. Of updating all the pre-trained parameters, LoRA integrates trainable low rank matrices into each layer of the Transformer architecture , which effectively fine-tunes the model without adding extra inference delay. This method reduces memory requirement significantly than the full fine-tuning, making it suitable for use in real world applications.

DoRA. Based on weight decomposition analysis, DoRA propose Weight-Decomposed Low-Rank Adaptation method that enhances learning efficiency by decomposing pre-trained weights and fine-tuning the magnitude and direction components separately.

MoSLoRA. MoSLoRA draws inspiration from the subspace structure seen in LoRA, introduces a mixer matrix that can be fine-tuned to enhance its effectiveness. By dividing the low rank segment into subspaces and merging them through a

Task Name	#Test	Evaluation Method
ARC (Clark et al., 2018)	887	Accuracy
Natural Questions (Kwiatkowski et al., 2019)	1751	Jaccard
TriviaQA (Joshi et al., 2017)	1557	Jaccard
WebQA (Chang et al., 2022)	1800	Accuracy
WinoGrande (Sakaguchi et al., 2021)	1200	Accuracy
MMLU (Hendrycks et al., 2021)	13985	Accuracy
ANLI (Nie et al., 2020)	1200	Accuracy
QQP (Wang et al., 2018)	1200	Accuracy
RTE (Dagan et al., 2005)	231	Accuracy
IMDB (Maas et al., 2011)	1000	Accuracy
OpenAI-Summarize-TLDR (Stiennon et al., 2020)	215	Rouge-L

Table 8: Detailed description of the evaluation dataset

mixer, MoSLoRA captures more nuanced information, leading to improved performance across various tasks. Its compatibility with quantization methods and minimal additional parameters make it a feasible and versatile approach.

HydraLoRA. HydraLoRA proposes a PEFT method which breaks the symmetry of traditional LoRA. In HydraLoRA, instead of a single A and B matrix for every pair, HydraLoRA share the same A but multiple Bs based on each task or sub-domain. Meanwhile, HydraLoRA exploits MoE framework to route inputs in a dynamic manner into the right B matrix during training and inference. This does not require domain expertise or no-interference between tasks and this results in better performance on complex domains.

MoELoRA. MoELoRA innovates with a new PEFT method for LLMs, relying on the Mixture of Experts (MoE) architecture. It achieves more flexibility in matching the requirements of downstream tasks by viewing different LoRA modules as experts and dynamically combining them using a gating network.

DSLoRA. Represented by DeepSeek-V3 (Liu et al., 2024a), the MoE framework exhibits remarkable performance. We explore this structure, characterized by the coexistence of shared and specialized experts, within the multi-LoRA architecture and named it DSLoRA.

D Additional experiments and analysis

D.1 Knowledge overlap in MoELoRA

We analyze the expert behavior of MoELoRA and make the following observations: (i) Under conditions of load balancing loss and non-sparse gating, inference can still achieve scores close to the performance ceiling through sparse gating. This result is consistent with Table 9, indicating the presence of a certain degree of overconfidence in MoELoRA. (ii) The performance using a single static expert suggests that there may not be significant differences between experts, which contradicts the anticipated behavior of the MoE architecture and could potentially serve as a bottleneck affecting the performance of MoELoRA.

Expert ID	1	2	3	4	Top-1	All
Perf↑	41.69	47.14	44.37	39.83	58.78	**59.96**

Table 9: Expert redundancy analysis of MoELoRA.

D.2 Computational Costs and Loss Convergence

Figure 6 illustrates the advantages of TeamLoRA over MoELoRA in terms of training and inference times. Specifically, TeamLoRA reduces training time by 30% and increases inference speed by 40%, as shown in Figure 6(a). Additionally, the loss convergence curve in Figure 6(b) demonstrates that TeamLoRA achieves lower loss values more quickly, highlighting its optimization in training efficiency.

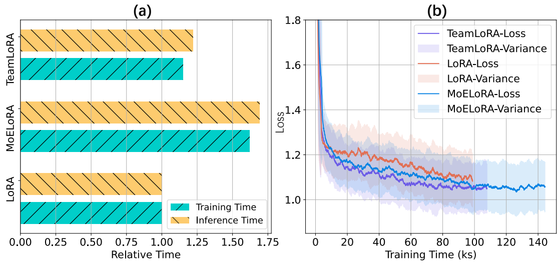

Figure 6: Visualization of Efficiency and Loss. (a) describes the relative training and inference latency of TeamLoRA and MoELoRA compared to LoRA. (b) displays the loss convergence.