MoDULA: Mixture of Domain-Specific and Universal LoRA for Multi-Task Learning

Yufei Ma[1,*] **Zihan Liang**[3,*] **Huangyu Dai**[3,*] **Ben Chen**[3,†]
Dehong Gao[1,4,†] **Zhuoran Ran**[2,3] **Zihan Wang**[3] **Linbo Jin**[3]
Wen Jiang[3] **Guannan Zhang**[3] **Xiaoyan Cai**[2] **Libin Yang**[1]
[1]Northwestern Polytechnical University, School of Cybersecurity, Xi'an, China
[2]Northwestern Polytechnical University, School of Automation, Xi'an, China
[3]Alibaba Group, Hangzhou, China
[4]Binjiang Institute of Artificial Intelligence, ZJUT, Hangzhou, China

Abstract

The growing demand for larger-scale models in the development of **L**arge **L**anguage **M**odels (LLMs) poses challenges for efficient training within limited computational resources. Traditional fine-tuning methods often exhibit instability in multi-task learning and rely heavily on extensive training resources. Here, we propose *MoDULA* (**M**ixture **o**f **D**omain-Specific and **U**niversal **L**o**RA**), a novel **P**arameter **E**fficient **F**ine-**T**uning (PEFT) **M**ixture-**o**f-**E**xpert (MoE) paradigm for improved fine-tuning and parameter efficiency in multi-task learning. The paradigm effectively improves the multi-task capability of the model by training universal experts, domain-specific experts, and routers separately. *MoDULA-Res* is a new method within the *MoDULA* paradigm, which maintains the model's general capability by connecting universal and task-specific experts through residual connections. The experimental results demonstrate that the overall performance of the *MoDULA-Flan* and *MoDULA-Res* methods surpasses that of existing fine-tuning methods on various LLMs. Notably, *MoDULA-Res* achieves more significant performance improvements in multiple tasks while reducing training costs by over 80% without losing general capability. Moreover, *MoDULA* displays flexible pluggability, allowing for the efficient addition of new tasks without retraining existing experts from scratch. This progressive training paradigm circumvents data balancing issues, enhancing training efficiency and model stability. Overall, *MoDULA* provides a scalable, cost-effective solution for fine-tuning LLMs with enhanced parameter efficiency and generalization capability.

1 Introduction

Recent advancements in open-source Large Language Models (LLMs), such as LLaMA (Touvron et al., 2023a), Qwen (Bai et al., 2023), and Yi (Young et al., 2024), have achieved notable successes in natural language processing. However, the increasing complexity and growing size of these models make efficient training within limited computational resources challenging. Researchers tried to address this with Parameter Efficient Fine-Tuning (PEFT), such as LoRA (Hu et al., 2021), Prefix Tuning (Liu et al., 2023), and $(IA)^3$ (Liu et al., 2022). LoRA has gained prominence for its high performance using low-rank matrices, but it often encounters instability when trained on large, mixed datasets. To mitigate this issue, MoLoRA (Zadouri et al., 2024) has been introduced by extending LoRA and integrating the Mixture-of-Expert (MoE) architecture as shown in Figure 1(a). This approach trains multiple LoRA-adapters concurrently, each serving as an expert, to enhance the base LLMs' generalization ability across diverse tasks. The integration of MoE into LoRA aims to improve training efficiency and stability, facilitating more effective fine-tuning of large-scale language models for a wide range of natural language processing applications.

Despite its advantages, MoLoRA has some limitations. One limitation is **the absence of domain-specific LoRA adapters**, as the same experts are employed universally across all tasks. This uniformity may limit the performance ceiling, especially for significantly distinct tasks like math and code, where the inclusion of domain-specific experts could potentially enhance performance (Zeng et al., 2021). Another challenge is **the limited pluggability** of MoLoRA; adding new task capabilities necessitates retraining all parameters from all experts, which can be inefficient and time-consuming.

To address the challenges, we propose a three-stage training paradigm called *MoDULA*, where different domain-specific experts can be trained separately. Moreover, we introduce a more advanced method *MoDULA-Res* (**M**ixture **of** **D**omain-

[*] Equal Contribution.
[†] Corresponding Author.

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 2758–2770
November 12-16, 2024 ©2024 Association for Computational Linguistics

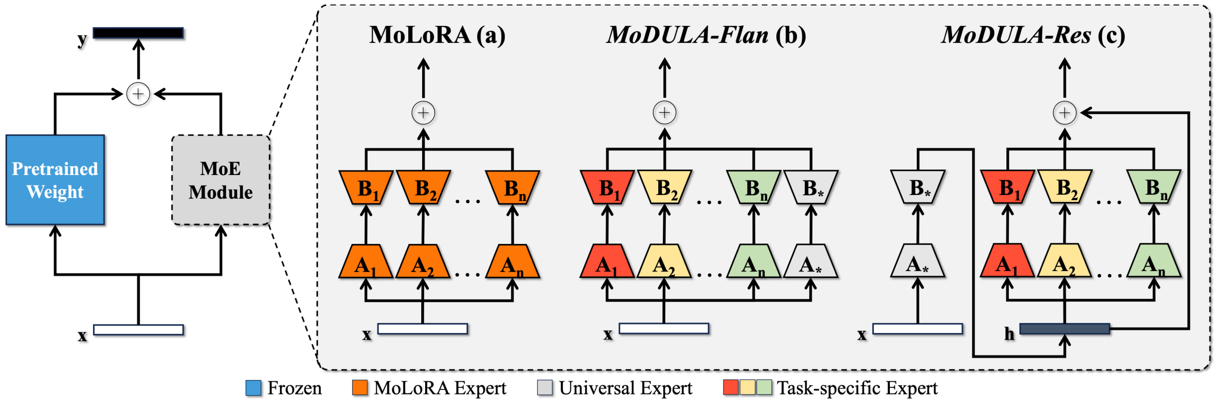


Figure 1: Illustrations of MoLoRA(a), *MoDULA-Flan*(b), and *MoDULA-Res*(c) with router omitted.

Specific and **U**niversal **L**o**RA** with **Res**idual Connection), which incorporates a residual structure to make the training more stable, as seen in Figure 1(c). Unlike MoLoRA, which employs multiple identical LoRA adapters as experts, our paradigm incorporates a universal expert alongside multiple domain-specific experts. The universal expert learns task-agnostic representations, while each domain-specific expert operates as a bias adapter, focusing on domain-specific knowledge. Intuitively, arranging these adapters in parallel and allocating weights to each adapter via a router constitutes the *MoDULA-Flan* (**M**ixture **of** **D**omain-Specific and **U**niversal **L**o**RA** with **Flan** Routing) method as seen in Figure 1(b). However, this method may potentially compromise universal capabilities. To address this, *MoDULA-Res* introduces a refined method that enables domain-specific experts to receive input from the output of the universal expert. This design ensures a coherent flow of information and facilitates the optimal integration of both universal and domain-specific expert functionalities through a residual connection. By dynamically adjusting the contributions of domain-specific experts, *MoDULA-Res* adapts to individual tasks while preserving broad generalization capabilities. This flexibility allows the model to leverage its general competencies for task understanding and summarization when encountering new tasks, thereby achieving a more balanced and effective adaptation in multi-task scenarios.

During model training, our *MoDULA* employs a three-stage optimization process, with detailed illustrations displayed in Figure 2: 1) Initially, only the universal expert is trained to adapt to general tasks quickly; 2) Subsequently, each domain-specific expert is trained individually, focusing on its corresponding task; 3) Finally, the parameters of all experts are frozen, and only router is trained to learn the optimal combination strategy for different tasks. This progressive training paradigm allows our methods to avoid retraining from scratch, distinguishing it from MoLoRA, which trains only a new expert for a new specific task and retraining the router. This paradigm significantly reduces computational costs, mitigates data balancing challenges, and enhances the model's pluggability.

To evaluate the effectiveness of our proposed methods, we conduct extensive experiments on a diverse set of open-source LLMs, including LLaMA-2 (Touvron et al., 2023b), Qwen (Bai et al., 2023), and Yi (Young et al., 2024), across various tasks. The results consistently demonstrate that *MoDULA* exhibits a significant performance, achieving 4.5% improvements compared to MoLoRA. By introducing residual connections, *MoDULA-Res* achieves even greater improvements without compromising the general capabilities. Additionally, our approach showcases superior adaptability to new tasks, outperforming MoLoRA in finance and e-commerce domain with less training data and parameters, highlighting the enhanced task pluggability of our approach, making it an efficient and general solution for multi-task learning in LLMs.

2 Related Works

2.1 Large Language Model

Recently, the field of natural language processing has witnessed a paradigm shift with the advent of LLMs (Anil et al., 2023b; Almazrouei et al., 2023; Xu et al., 2023; Scao et al., 2022; Brown et al., 2020; Achiam et al., 2023; Zhang et al., 2023; Du et al., 2022). These state-of-the-art models have departed from traditional approaches that relied on convolutional or recurrent architectures for feature extraction, instead embracing novel techniques such as BERT (Devlin et al., 2019), which leverages the power of Transformers trained on extensive datasets, yielding bidirectional encoder representations. Similarly, Generative Pretrained Transformer (GPT) (Brown et al., 2020) employs decoder layers from Transformer architecture (Vaswani et al., 2017) as feature extractors and utilizes autoregressive training on vast texts.

Guided by the principles of scaling laws (Kaplan et al., 2020), the development of LLMs has led to the emergence of colossal models boasting over 100 billion parameters, with prominent examples including GPT-4 (Achiam et al., 2023) and Gemini (Anil et al., 2023a). Interestingly, open-source models such as OPT (Zhang et al., 2022), Falcon (Almazrouei et al., 2023), and Gemma (Mesnard et al., 2024) have demonstrated competitive performance compared to their closed-source counterparts, despite possessing a more modest parameter count. The training process of LLMs typically involves leveraging immense amounts of textual data to enable the prediction of subsequent tokens, empowering these models to generate coherent and comprehensible responses to a wide range of prompts. This training method has proven to be highly effective in capturing the intricacies of language and paved the way for LLMs to achieve SOTA performance across various NLP tasks.

2.2 MoE for PEFT

Our research closely aligns with the work done by MoLoRA (Zadouri et al., 2024), LoraHub (Huang et al., 2023a), MoELoRA (Liu et al., 2024), SiRA (Zhu et al., 2023), and C-Poly (Wang et al., 2023), which explore the intersection of PEFT and MoE. MoLoRA employs a full soft MoE on top of LoRA, utilizing a learned gating mechanism to average all experts, and trains the experts in a single stage. LoraHub investigates LoRA composability for cross-task generalization and introduces a simple framework for the purposive assembly of LoRA modules trained on diverse given tasks, aiming to achieve adaptable performance on unseen tasks. It can fluidly combine multiple LoRA modules with just a few examples from a new task, without requiring additional model parameters or human expertise. MoELoRA devises multiple experts as the trainable parameters and proposes a task-motivated gate function for all MOELoRA layers to regulate the contributions of each expert and generate distinct parameters for various tasks. SiRA proposes a sparse mixture of low rank adaption that enforces the top k experts' routing with a capacity limit. It uses expert dropout to reduce over-fitting. C-Poly combines task-common skills and task-specific skills and jointly learns a skill assignment matrix.

While these methods have significantly contributed to the field, they face particular challenges and limitations. Training experts on mixed datasets as in MoLoRA may lead to performance degradation due to data inconsistency and interference (Dong et al., 2024). LoraHub relies on few-shot examples in inference stage, and MoELoRA requires task-id to determine which experts should be activated, which weaken the flexibility of both methods. Sparse routing, as used by SiRA, requires careful tuning of the top-k and capacity hyperparameters for each dataset. C-Poly's joint learning of task-common and task-specific skills can make balancing general and specialized abilities difficult. Additionally, incorporating new experts or skills in these methods may require retraining or modifying existing components, potentially impacting system stability and training complexity. Training new experts often demands substantial data, resulting in high training costs and sub-optimal performance in specific domains. Maintaining optimal performance on domain-specific benchmarks after adding new capabilities can be challenging, and newly added modules may not consistently achieve top performance in their respective benchmarks. These factors can affect the adaptability and efficiency of MoLoRA, SiRA, and C-Poly in meeting expanding task demands.

In contrast, *MoDULA* method trains universal and domain-specific experts separately, mitigating performance degradation from mixed datasets. Designed with "pluggability" in mind, the *MoDULA* method allows new experts to be added without changing existing ones, ensuring system stability and low training costs. After adding a new expert, only the router requires retraining to maintain near-optimal performance. This staged training balances general and domain-specific capabilities, making our method adaptable and efficient for growing task requirements.

3 Method

In this section, we present *MoDULA* for LLM fine-tuning. Within this paradigm, we propose

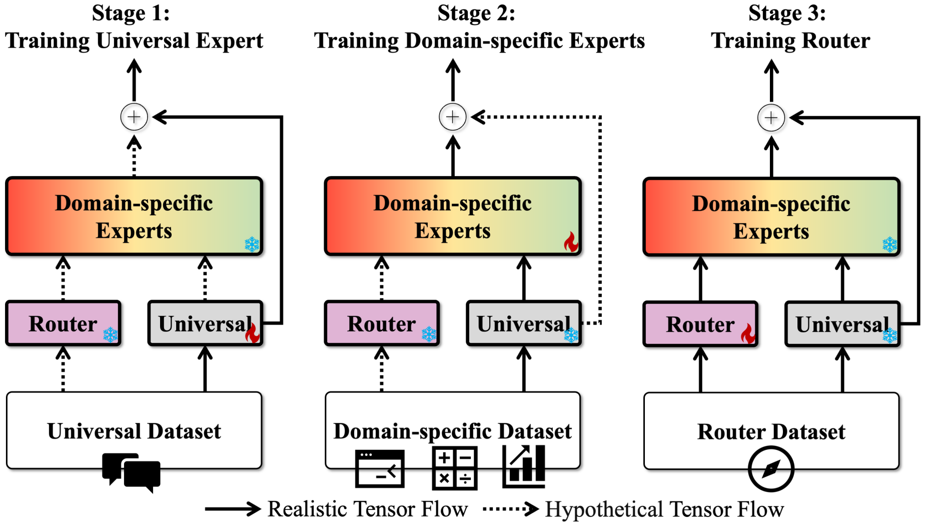


Figure 2: Illustrations of the three-stage training paradigm for *MoDULA-Res*.

two methods: *MoDULA-Flan* and *MoDULA-Res*. *MoDULA-Flan* consists of a universal expert and an array of domain-specific experts, while *MoDULA-Res* further incorporates residual connections between the universal and domain-specific experts to enhance performance and stability. Figure 1 illustrates the differences between MoLoRA, our proposed *MoDULA-Flan* and *MoDULA-Res*. In all of these, the base LLMs retain a frozen weight configuration, denoted as W_0, corresponding to the fixed linear layers within the architecture.

MoLoRA. The MoLoRA method serves as the foundation of our *MoDULA*. As shown in Figure 1(a), the MoLoRA consists of a router θ_R^M and a set of LoRA experts $E_1, E_2, \ldots, E_n$. Each expert E_i includes two key components: B_i^M and A_i^M. The dynamics of the MoLoRA method can be summarized by the following equations:

$$s_i^M = \theta_R^M(x_m)_i = softmax(W_R^M x_m)_i \quad (1)$$

$$y_m^M = E^M(x_m) + W_0 x_m \quad (2)$$

$$E^M(x_m) = \sum_{i=1}^{n} s_i^M B_i^M A_i^M x_m \quad (3)$$

In these equations, x_m represents the hidden vector of the m-th token in the input sequence, s_i^M denotes the routing coefficient for expert E_i, W_R^M is the weight matrix of the router, and $E^M(\cdot)$ expresses the collective function of the experts in the MoLoRA module.

MoDULA. Based on MoLoRA, we propose a three-stage training paradigm called *MoDULA*, as illustrated in Figure 2. In the first stage, only the universal expert is trained, while the domain-specific experts and router are deactivated. In the second stage, the domain-specific experts are trained individually for each corresponding task, while the parameters of the universal expert are kept frozen. In the third stage, all the experts' parameters are fixed, and only the router is trained. With the *MoDULA* paradigm, we propose two methods: *MoDULA-Flan* and *MoDULA-Res*.

MoDULA-Flan. *MoDULA-Flan* maintains the same architecture as MoLoRA, as illustrated in Figure 1(b). However, it implements the *MoDULA* paradigm to separate the experts in MoLoRA into universal expert and domain-specific experts. The specific training details are as follows. In the first stage, the universal expert E_*^{flan} is trained on universal datasets. In the second stage, the domain-specific experts $E_1^{flan}, E_2^{flan}, \ldots, E_n^{flan}$ are trained on their respective domain-specific datasets. The forward process in this stage is formally articulated through Equations (4) and (8).

$$y_m^{flan} = E_i^{flan}(x_m) + W_0 x_m \quad (4)$$

where $i \in \{1, 2, \ldots, n\}$. In the third stage, the parameters of all experts are kept frozen, and only the router θ_R^{flan} is trained. The calculation involved in this routing determination is formally illuminated through the following equations:

$$s_i^{flan} = \theta_R^{flan}(x_m)_i = softmax(W_R^{flan} x_m)_i \quad (5)$$

$$y_m^{flan} = E^{flan}(x_m) + W_0 x_m \quad (6)$$

$$E^{flan}(x_m) = \sum_i s_i^{flan} E_i^{flan}(x_m) \quad (7)$$

$$E_i^{flan}(x_m) = B_i^{flan} A_i^{flan} x_m \quad (8)$$

Base Model	Method	Avg.	GSM8K	Arithmetic	MathQA	HumanEval	MBPP	Medical	MedQA
Qwen-7B	Not fine-tuned	44.65	46.63	56.65	35.48	21.95	32.00	76.00	43.83
	LoRA	25.93	7.21	49.61	26.40	9.15	17.20	42.80	29.14
	LoraHub	49.37	44.81	86.33	37.09	22.40	29.60	81.00	44.38
	MoLoRA	48.94	48.21	78.49	37.42	23.78	32.78	79.20	42.73
	MoDULA-Flan	50.32	**48.67**	87.06	36.98	23.17	**33.60**	78.40	44.38
	MoDULA-Res	**51.36**	46.63	**90.37**	**37.98**	**25.00**	33.00	**82.00**	**44.55**
LLaMA-2-7B	Not fine-tuned	27.45	13.72	6.89	29.41	14.63	18.00	77.60	31.89
	LoRA	15.40	1.29	2.69	22.48	0.00	0.00	53.40	27.97
	LoraHub	38.69	22.03	63.47	31.17	13.80	24.00	83.60	32.79
	MoLoRA	38.53	**23.12**	60.87	30.48	15.24	21.40	83.60	**35.03**
	MoDULA-Flan	38.67	20.39	61.40	31.35	15.24	**24.40**	84.20	33.69
	MoDULA-Res	**39.62**	22.37	**70.66**	**31.73**	**15.24**	22.80	**85.20**	29.31
Yi-6B	Not fine-tuned	38.04	33.81	39.92	35.41	14.63	23.00	70.00	49.49
	LoRA	16.07	2.51	0.88	20.41	0.00	0.00	61.20	27.49
	LoraHub	46.77	**35.97**	82.03	35.50	14.24	**24.80**	84.60	50.28
	MoLoRA	41.49	34.87	46.50	34.50	16.46	23.20	82.80	**52.08**
	MoDULA-Flan	45.09	35.25	73.85	35.88	12.20	23.00	84.80	50.66
	MoDULA-Res	**48.61**	34.50	**92.72**	**36.29**	**16.46**	24.40	**85.80**	50.12
Qwen-14B	Not fine-tuned	54.55	61.87	69.32	44.42	24.39	43.80	85.60	52.47
	LoRA	55.58	56.86	**92.58**	39.23	**26.83**	37.60	82.80	53.18
	LoraHub	57.47	66.74	88.91	43.91	24.32	38.10	86.20	**54.12**
	MoLoRA	56.79	63.38	83.56	44.48	26.22	41.40	85.80	52.71
	MoDULA-Flan	56.95	63.53	83.19	**45.25**	25.61	42.40	85.60	53.10
	MoDULA-Res	**58.42**	**67.78**	91.45	45.13	18.90	**44.80**	**88.00**	52.87
LLaMA-2-13B	Not fine-tuned	41.71	23.28	80.28	32.53	15.24	27.20	70.60	**42.81**
	LoRA	16.33	1.18	4.28	25.27	0.00	0.00	55.00	28.59
	LoraHub	44.01	34.21	72.15	**36.17**	14.23	26.20	84.20	40.92
	MoLoRA	45.62	33.51	74.57	34.21	19.51	30.40	85.80	41.32
	MoDULA-Flan	44.70	35.48	67.31	34.53	20.73	28.60	83.80	42.46
	MoDULA-Res	**47.93**	**36.47**	**84.26**	35.18	**20.73**	**31.20**	**86.40**	41.24
Yi-9B	Not fine-tuned	56.45	51.33	93.27	39.97	25.61	49.20	82.60	53.18
	LoRA	16.23	0.69	0.82	22.95	0.00	0.00	61.40	27.73
	LoraHub	58.54	54.13	89.47	**42.21**	33.13	53.10	85.20	52.56
	MoLoRA	56.97	57.99	68.89	41.86	32.32	54.20	86.80	**56.72**
	MoDULA-Flan	60.54	**60.04**	96.36	41.47	29.88	**54.80**	86.80	54.43
	MoDULA-Res	**60.55**	59.06	**96.86**	41.51	**34.15**	51.20	**87.20**	53.86

Table 1: Main experimental results of baseline methods, *MoDULA-Flan*, and *MoDULA-Res* on domain-specific benchmarks.

MoDULA-Res. In order to further improve the general ability of the model, we propose *MoDULA-Res*, a more advanced method that leverages the strengths of both universal and domain-specific experts. The architecture of *MoDULA-Res* is shown in Figure 1(c). *MoDULA-Res* integrates both the universal expert E_*^{res} and the domain-specific experts $E_1^{res}, E_2^{res}, \ldots, E_n^{res}$, tuned in a balanced way to cater to both general and domain-specific tasks. *MoDULA-Res* introduces a residual connection that allows the model to incorporate the output of universal expert directly into the final result, ensuring that critical information is preserved and enhancing model robustness.

The forward process in *MoDULA-Res* module involves two stages. Initially, a hidden vector h_m is computed using the universal expert:

$$h_m = B_*^{res} A_*^{res} x_m \tag{9}$$

where x_m is the hidden vector for the m-th token, and B_*^{res} and A_*^{res} correspond to the universal expert matrices. Subsequently, the hidden vector h_m is refined by the domain-specific experts with residual connection to produce the final output y_m^{res}:

$$y_m^{res} = E^{res}(h_m) + W_0 x_m + h_m \tag{10}$$

where the function $E^{res}(\cdot)$ represents the operation of the domain-specific experts:

$$E^{res}(h_m) = \sum_{i=1}^{n} s_i^{res} B_i^{res} LeakyReLU(A_i^{res} h_m) \tag{11}$$

s_i^{res} is the weight for each expert, computed as:

$$s_i^{res} = \theta_R^{res}(x_m)_i = softmax(W_R^{res} x_m)_i \tag{12}$$

This integration of a three-stage training paradigm and residual connection ensures that the *MoDULA-Res* module effectively generalizes and specializes simultaneously, thereby enhancing performance across both broad and focused applications.

4 Experiments

4.1 Expert Configurations

A detailed comparison is conducted among the standard LoRA (Hu et al., 2021), MoLoRA (Zadouri et al., 2024), and our newly proposed *MoDULA-Flan* and *MoDULA-Res*. The base models selected for this study include LLaMA-2 (Touvron et al., 2023b), Qwen (Bai et al., 2023), and Yi (Young et al., 2024). In the training of *MoDULA*, a batch size of 128 is utilized, encompassing 1 epoch with a learning rate of 2e-4. The maximum input sequence length is defined as 4096 tokens for both LLaMA-2 and Yi. In contrast, Qwen series has 8192 tokens due to variations in maximum positional embeddings among different model zoos. The intrinsic rank is configured to 16 for universal and 8 for domain-specific experts. For the multi-task results, the checkpoint selection is based on the average metrics across all tasks. To enhance fine-tuning efficiency, we leverage libraries like HuggingFace's Transformers (Wolf et al., 2020) and PEFT (Mangrulkar et al., 2022), based on which we design *MoDULA*.

Benchmark	Few-Shot	Metric
GSM8K	5	acc
Arithmetic	0	acc
MathQA	5	acc
HumanEval	0	pass@1
MBPP	0	pass@1
Medical	5	acc
MedQA	0	acc
MMLU	5	acc
C-Eval	5	acc
FinGPT-headline	0	acc
Title-Optimization	0	GPT-4 Judge
Keyword-Recommendation	0	GPT-4 Judge

Table 2: Few-shot example numbers and evaluation metrics for benchmarks.

4.2 Training Datasets

To equip our *MoDULA-Flan* and *MoDULA-Res* with comprehensive capabilities across universal, mathematical, coding, and medical domains, the datasets **airoboros-3.2** [1], **orca-math-word-problems-200k** [2], **CodeAlpaca-20k** [3], and **MedQA** (Jin et al., 2019) are integrated.

In order to evaluate the pluggability of our methods, we fine-tune the baselines, *MoDULA-Flan*, and *MoDULA-Res* on three datasets from different domains: **FinGPT-headline** [4] from the finance domain, and **Title-Optimization** and **Keyword-Recommendation** from the e-commerce domain. The Title-Optimization and Keyword-Recommendation datasets are sourced from real-world requirements on alibaba.com [5], a leading e-commerce platform. By fine-tuning on these diverse datasets, we aim to demonstrate the adaptability and effectiveness of *MoDULA-Res* in various domain-specific applications, showcasing its modular design and ability to capture both general and domain-specific knowledge.

Base Model	Method	MMLU	C-Eval
Qwen-7B	Not fine-tuned	**58.21**	62.1
	MoLoRA	55.77	61.44
	MoDULA-Flan	56.16	62.29
	MoDULA-Res	57.65	**62.34**
LLaMA-2-7B	Not fine-tuned	45.91	34.02
	MoLoRA	47.45	35.95
	MoDULA-Flan	45.65	34.22
	MoDULA-Res	**48.23**	**36.18**
Yi-6B	Not fine-tuned	63.30	73.63
	MoLoRA	63.11	73.17
	MoDULA-Flan	62.17	72.33
	MoDULA-Res	**63.41**	**74.15**
Qwen-14B	Not fine-tuned	66.89	**70.87**
	MoLoRA	**67.21**	70.35
	MoDULA-Flan	65.98	69.82
	MoDULA-Res	66.58	70.13
LLaMA-2-13B	Not fine-tuned	54.92	38.11
	MoLoRA	56.08	40.34
	MoDULA-Flan	**57.01**	39.22
	MoDULA-Res	56.23	**40.94**
Yi-9B	Not fine-tuned	68.10	**70.57**
	MoLoRA	67.70	69.83
	MoDULA-Flan	66.07	68.41
	MoDULA-Res	**68.13**	69.83

Table 3: Experimental results of different methods on universal benchmarks.

4.3 Evaluation Benchmarks and Metrics

To comprehensively assess the performance of various methods, we conduct evaluations across a diverse set of benchmarks. Domain-specific performance is evaluated by testing mathematical abilities on **GSM8K** (Cobbe et al., 2021), **Arithmetic** (Brown et al., 2020), and **MathQA** (Amini et al., 2019), coding skills on **HumanEval** (Chen et al., 2021) and **MBPP** (Austin et al., 2021), and medical knowledge on **MedQA** (Jin et al., 2020)

[1] https://huggingface.co/datasets/jondurbin/airoboros-3.2

[2] https://huggingface.co/datasets/microsoft/orca-math-word-problems-200k

[3] https://huggingface.co/datasets/sahil2801/CodeAlpaca-20k

[4] https://huggingface.co/datasets/FinGPT/fingpt-headline

[5] https://www.alibaba.com/

Base Model	Method	Avg.	GSM8K	Arithmetic	MathQA	HumanEval	MBPP	Medical	MedQA	FinGPT headline
Qwen-7B	Not fine-tuned	44.65	46.63	56.65	35.48	21.95	32.00	76.00	43.83	74.91
	MoLoRA	49.92	47.38	84.05	36.88	22.56	32.00	**80.20**	**46.34**	75.41
	MoDULA-Flan	50.66	**48.36**	88.12	36.41	26.22	**32.60**	79.00	43.93	76.61
	MoDULA-Res	**50.85**	45.87	**89.37**	**37.99**	**28.05**	31.02	79.60	44.06	**80.61**
LLaMA-2-7B	Not fine-tuned	27.45	13.72	6.89	29.41	14.63	18.00	77.60	31.89	22.39
	MoLoRA	37.05	16.75	57.20	30.51	**16.46**	20.40	**82.20**	**35.82**	32.38
	MoDULA-Flan	37.37	17.43	59.38	30.61	15.85	23.20	81.80	33.30	24.89
	MoDULA-Res	**37.86**	**21.61**	**67.59**	**31.26**	12.80	**24.00**	81.20	26.56	**33.83**
Yi-6B	Not fine-tuned	38.04	33.81	39.92	35.41	14.63	23.00	70.00	49.49	64.92
	MoLoRA	48.58	**36.42**	93.50	**36.78**	16.46	23.80	81.20	**51.92**	65.96
	MoDULA-Flan	47.98	35.17	92.57	36.18	16.46	**24.80**	80.20	50.50	61.77
	MoDULA-Res	**48.70**	34.57	**93.63**	36.15	**17.07**	23.40	**85.60**	50.51	**73.26**

Table 4: Experimental results of MoLoRA, *MoDULA-Flan*, and *MoDULA-Res* on domain-specific and FinGPT-headline (finance) benchmarks.

Base Model	Method	Avg.	T.O.	Avg.	K.R.
Qwen-7B	Not fine-tuned	44.65	6.23	44.65	5.28
	MoLoRA	49.89	6.94	48.64	5.92
	MoDULA-Flan	50.19	5.44	49.59	6.78
	MoDULA-Res	**51.17**	**7.28**	**50.29**	**7.02**
LLaMA-2-7B	Not fine-tuned	27.45	2.76	27.45	4.25
	MoLoRA	35.35	3.54	36.23	5.98
	MoDULA-Flan	37.21	6.48	37.33	6.52
	MoDULA-Res	**38.80**	**6.62**	**38.12**	**7.37**
Yi-6B	Not fine-tuned	38.04	3.01	38.04	5.45
	MoLoRA	45.91	3.92	44.37	5.78
	MoDULA-Flan	47.93	5.92	46.59	6.38
	MoDULA-Res	**48.28**	**6.94**	**47.88**	**7.58**

Table 5: Experimental results of methods on T.O. and K.R. (e-commerce) benchmarks. Avg. denotes the average performance of different methods on domain-specific benchmarks. T.O. denotes the Title Optimization task and K.R. the Keyword Recommendation task.

and the **Medical** (Jin et al., 2019) dataset. General capabilities are measured via **MMLU** (Hendrycks et al., 2021) and **C-Eval** (Huang et al., 2023b) benchmarks, which both cover a wide range of tasks. To evaluate the pluggability and adaptability of different methods on new domain-specific tasks, we test their performance on the **FinGPT-headline** (Yang et al., 2023) dataset from the finance domain, as well as the **Title-Optimization** and **Keyword-Recommendation** datasets from the e-commerce domain.

Title optimization and keyword recommendation are critical tasks in e-commerce that aim to enhance product visibility and market responsiveness. These tasks involve integrating high-exposure queries from specific leaf categories into product titles to refine original titles and generate new keywords, ultimately achieving a higher Click-Through Rate (CTR). By evaluating the methods of these real-world e-commerce tasks, we can assess their effectiveness in capturing domain-specific knowledge and potential for practical application in industry settings. The specific evaluation metrics used for each benchmark are summarized in Table 2, providing a clear overview of the performance measures employed in our experiments.

4.4 Main Experimental Results

Our experimental results yield several significant observations that demonstrate the robustness and effectiveness of the proposed approach, providing valuable insights into its performance across various benchmarks and real-world applications.

Superior Advancement over Baselines: Table 1 highlights the significant performance improvements achieved by our proposed paradigm across Qwen, LLaMA-2, and Yi. Models that are fine-tuned with our paradigm outperform the base models by an average of 16.6% and surpass the performance of MoLoRA by 6.3% on average. Notably, Yi demonstrates the most substantial improvement, with an impressive average increase of 10.9% over MoLoRA.

Further analysis reveals that performance advancements are more pronounced in smaller-scale models than in their larger counterparts, e.g., 4.9% for Qwen-7B while 2.9% for Qwen-14B. This indicates that small-scale models with fewer parameters and inadequate training are more prone to losing general capability when learning multiple tasks, while residual connections can effectively mitigate this problem.

Moreover, *MoDULA-Flan* does not consistently outperform MoLoRA, suggesting that it has the issue of decreased general capabilities (for example, the arithmetic benchmark of LLaMA-2-13B dropped sharply due to the decline in text understanding ability). In contrast, *MoDULA-Res* addresses this issue by introducing residual connections for general and expert modules, leading to more stable performance and significant improvements over MoLoRA and *MoDULA-Flan*.

Despite *MoDULA-Res* demonstrates overall strong performance, it faces challenges with GSM8K and MedQA tasks, likely due to the mismatch between pre-training data and task-specific requirements. We recognize these limitations and leave them for further research.

Excellent Robustness on Comprehensive Benchmarks: In order to determine whether the general capability of *MoDULA-Res* trained on multiple tasks will decline, we conduct experiments using the base, MoLoRA, and the *MoDULA-Res* model on the comprehensive benchmarks MMLU and C-Eval.

The results in Table 3 indicate that the average performance of *MoDULA-Res* across multiple models is about 1% higher than that of MoLoRA and the base model, suggesting that the model's general capability is maintained and even partially improved through residual connection.

Flexible Pluggability over Baselines: To showcase *MoDULA-Res*'s pluggability, we introduce the finance domain (FinGPT-headline) in addition to the initial domains of mathematics, coding, and medical care. Then, we retrained MoLoRA, *MoDULA-Flan*, and *MoDULA-Res*, respectively. MoLoRA is trained from scratch on the combined dataset, while *MoDULA-Flan* and *MoDULA-Res* only require training a new financial expert and the router. This results in *MoDULA-Flan* and *MoDULA-Res* using only 19.8% and 37.3% of the training parameters and data compared to MoLoRA, respectively.

The results in Table 4 indicate that *MoDULA-Res* achieves the best average multi-task performance among the three models, with an average improvement of 8.0% in the financial task. Notably, the overall improvement of Yi-6B is more significant, exceeding 11.0%, due to the fewer parameters and relatively balanced pre-training data. MoLoRA encounters issues with data balance, requiring numerous experiments to adjust the data ratio for each task to achieve the best overall performance when new domain-specific tasks are introduced, which is time-consuming and labor-intensive.

Outstanding Performance in E-Commerce: To assess *MoDULA*'s practical applicability in e-commerce, we introduce title optimization and keyword recommendation tasks, which involve refining titles and generating keywords using high-exposure queries to enhance readability and include more key points. We employ GPT-4 to evaluate the optimized titles and keywords across five dimensions: helpfulness, relevance, accuracy, readability, and fluency. Each dimension is scored 0, 1, or 2, with a maximum total score of 10.

Table 5 demonstrates that *MoDULA-Res* significantly improves performance on title optimization and keyword recommendation benchmarks, with gains of 44.7% and 24.3% over MoLoRA, respectively. Moreover, *MoDULA-Res* maintains superior performance on the original multi-task benchmarks. These results highlight *MoDULA-Res*'s potential for e-commerce applications and adaptability to new tasks under resource constraints.

4.5 Analysis on Domain-specific Experts Allocation

To further analyze *MoDULA-Res*, router distributions for domain-specific experts based on Yi-6B and Qwen-14B are visualized in Figure 3. Models in Table 1 are reused, and we select layer 0-10-20-30 and 10-20-30-40 for Yi-6B and Qwen-14B, respectively.

The results indicate that for both the Yi and Qwen models, the router within the *MoDULA* paradigm allows various experts to concentrate on their own domain. However, the interpretation of expert assignments varies across different layers in different models due to the model's training data and method. For instance, Yi's deeper layers focus more on separating experts, while Qwen in the shallower layers.

5 Conclusion

In this paper, we introduce *MoDULA*, a novel multi-stage training PEFT MoE paradigm that enhances efficiency and domain-specific adaptation for LLMs. By integrating universal and domain-specific experts through a three-stage training methodology, *MoDULA* optimizes both generalization and specialized performance. Experiments on various open-source LLMs, such as LLaMA-2,

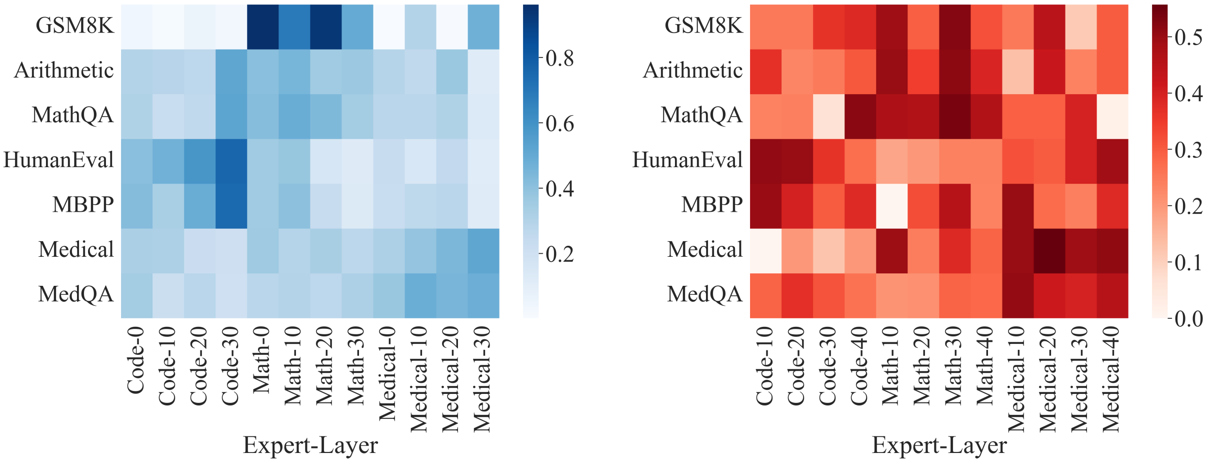


Figure 3: Router distributions of *MoDULA-Res* based on Yi-6B (left) and Qwen-14B (right) on domain-specific tasks.

Qwen, and Yi, demonstrate that *MoDULA* outperforms existing methods, achieving over 80% reduction in training costs and a 5% performance improvement. These results highlight *MoDULA*'s potential as a scalable and efficient solution for fine-tuning LLMs, paving the way for future advancements in NLP.

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China under Grants U20B2065, U22B2036, 62372380, and 62103374, National Key Research and Development Project under Grant 2022YFB3104005, and the Natural Science Basic Research Program of Shaanxi (Program No.2024JC-YBMS-513), and Key Research and Development Program of Zhejiang Province under Grants 2024C01025.

Limitations

While our proposed *MoDULA* paradigm shows significant advancements in parameter efficiency and multi-task adaptability for LLMs, there are still some limitations that need to be addressed. Despite the overall strong performance of *MoDULA-Res*, it shows sub-optimal results on certain benchmarks like GSM8K and MedQA. This may be due to discrepancies between the model's pre-training data and the specific task datasets, requiring further investigation to identify the root causes and develop targeted solutions. Our experiments also focus on a limited set of language models (LLaMA-2, Qwen, Yi) and domain-specific tasks (mathematics, coding, medical, finance, e-commerce). To establish stronger generalizability, it would be valuable to extend our evaluations to a broader range of base models and diverse task domains. Furthermore, the current study primarily emphasizes the pluggability and training efficiency of *MoDULA* when incorporating new domain experts. However, the scalability and robustness of this approach when integrating a larger number of experts require further exploration and stress testing.

Future research directions include investigating techniques to mitigate performance degradation on specific benchmarks, conducting comprehensive evaluations on a wider range of models and tasks, exploring the scalability limits of expert integration, streamlining the multi-stage training process, and enhancing the interpretability of the router's decision-making. By acknowledging these limitations and outlining potential avenues for future work, we aim to provide a balanced perspective on the current state of our research and highlight opportunities for further advancements in PEFT for LLMs.

References

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

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Maitha Alhammadi, Mazzotta Daniele, Daniel Heslow, Julien Launay, Quentin Malartic, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo.

2023. tiiuae/falcon-180b. https://huggingface.co/tiiuae/falcon-180B.

Aida Amini, Saadia Gabriel, Peter Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. Mathqa: Towards interpretable math word problem solving with operation-based formalisms. *arXiv preprint arXiv: 1905.13319*.

Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, and et al. 2023a. Gemini: A family of highly capable multi-modal models. *arXiv preprint arXiv: 2312.11805*.

Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, and et al. 2023b. Palm 2 technical report. *arXiv preprint arXiv: 2305.10403*.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. 2021. Program synthesis with large language models. *arXiv preprint arXiv: 2108.07732*.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, and et al. 2023. Qwen technical report. *arXiv preprint arXiv: 2309.16609*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and et al. 2020. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, volume 159 of *NIPS'20*, pages 1877–1901, Vancouver, BC, Canada.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, and Greg Brockman et.al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv: 2107.03374*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv: 2110.14168*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota.

Guanting Dong, Hongyi Yuan, Keming Lu, Chengpeng Li, Mingfeng Xue, Dayiheng Liu, Wei Wang, Zheng Yuan, Chang Zhou, and Jingren Zhou. 2024. How abilities in large language models are affected by supervised fine-tuning data composition. *arXiv preprint arXiv: 2310.05492*.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. Glm: General language model pretraining with autoregressive blank infilling. *arXiv preprint arXiv: 2103.10360*.

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

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

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

Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Jiayi Lei, Yao Fu, Maosong Sun, and Junxian He. 2023b. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. *arXiv preprint arXiv: 2305.08322*.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2020. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *arXiv preprint arXiv: 2009.13081*.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William W. Cohen, and Xinghua Lu. 2019. Pubmedqa: A dataset for biomedical research question answering. *arXiv preprint arXiv: 1909.06146*.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv: 2001.08361*.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. 2022. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *arXiv preprint arXiv: 2205.05638*.

Qidong Liu, Xian Wu, Xiangyu Zhao, Yuanshao Zhu, Derong Xu, Feng Tian, and Yefeng Zheng. 2024. When moe meets llms: Parameter efficient fine-tuning for multi-task medical applications. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1104–1114.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2023. Gpt understands, too. *arXiv preprint arXiv: 2103.10385*.

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

Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, and et. al. Pouya Tafti. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv: 2403.08295*.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, and et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv: 2211.05100*.

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

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NeurIPS'17, pages 6000–6010, Long Beach, California, USA.

Haowen Wang, Tao Sun, Cong Fan, and Jinjie Gu. 2023. Customizable combination of parameter-efficient modules for multi-task learning. *arXiv preprint arXiv: 2312.03248*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45.

Canwen Xu, Daya Guo, Nan Duan, and Julian McAuley. 2023. Baize: An open-source chat model with parameter-efficient tuning on self-chat data. *arXiv preprint arXiv: 2304.01196*.

Hongyang Yang, Xiao-Yang Liu, and Christina Dan Wang. 2023. Fingpt: Open-source financial large language models. *arXiv preprint arXiv: 2306.06031*.

Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, Kaidong Yu, Peng Liu, Qiang Liu, Shawn Yue, Senbin Yang, Shiming Yang, Tao Yu, Wen Xie, Wenhao Huang, Xiaohui Hu, Xiaoyi Ren, Xinyao Niu, Pengcheng Nie, Yuchi Xu, Yudong Liu, Yue Wang, Yuxuan Cai, Zhenyu Gu, Zhiyuan Liu, and Zonghong Dai. 2024. Yi: Open foundation models by 01.ai. *arXiv preprint arXiv: 2403.04652*.

Ted Zadouri, Ahmet Üstün, Arash Ahmadian, Beyza Ermis, Acyr Locatelli, and Sara Hooker. 2024. Pushing mixture of experts to the limit: Extremely parameter efficient moe for instruction tuning. In *The Twelfth International Conference on Learning Representations*.

Wei Zeng, Xiaozhe Ren, Teng Su, Hui Wang, Yi Liao, Zhiwei Wang, Xin Jiang, ZhenZhang Yang, Kaisheng Wang, and Xiaoda Zhang et al. 2021. Pangu-α: Large-scale autoregressive pretrained chinese language models with auto-parallel computation. *arXiv preprint arXiv: 2104.12369*.

Pan Zhang, Xiaoyi Dong, Bin Wang, Yuhang Cao, Chao Xu, Linke Ouyang, Zhiyuan Zhao, Haodong Duan, Songyang Zhang, and Shuangrui Ding et al. 2023. Internlm-xcomposer: A vision-language large model for advanced text-image comprehension and composition. *arXiv preprint arXiv: 2309.15112*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, and Xi Victoria Lin et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv: 2205.01068*.

Yun Zhu, Nevan Wichers, Chu-Cheng Lin, Xinyi Wang, Tianlong Chen, Lei Shu, Han Lu, Canoee Liu, Liangchen Luo, Jindong Chen, and Lei Meng. 2023. Sira: Sparse mixture of low rank adaptation. *arXiv preprint arXiv: 2311.09179*.

A Analysis on the Residual Connection

The results in Table 6 validate the importance of the residual connection in the *MoDULA-Res* method. Comparing *MoDULA-Res* with its non-residual counterpart reveals the residual connection's role in enhancing domain-specific tasks while preserving general language understanding.

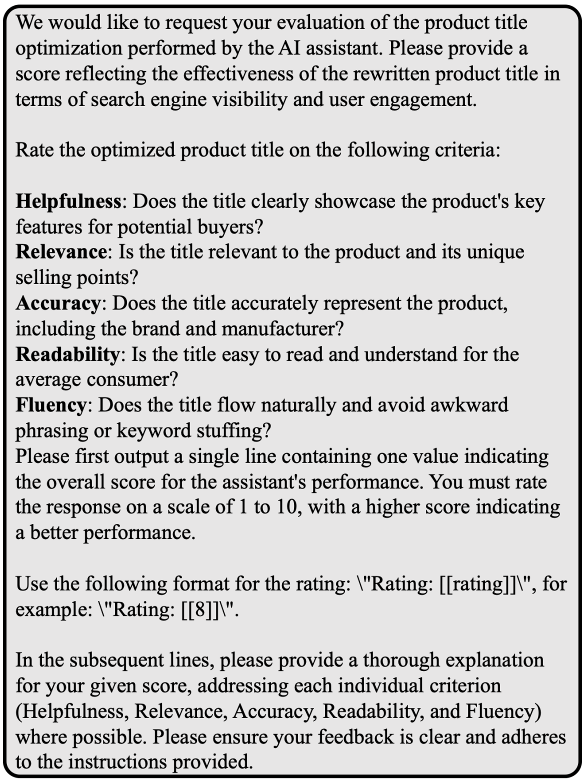
The residual connection's impact varies among models. For instance, Qwen-7B and Yi-6B models show significant score improvements of 1.71 and 3.01 points, respectively, whereas LLaMA-2-7B shows a smaller gain of 1.77 points. This suggests that the benefits may be model-specific, meriting further investigation.

In domain-specific tasks, *MoDULA-Res* excels, particularly in mathematics and medical fields. For example, in Arithmetic and Medical datasets, *MoDULA-Res* exceeds its non-residual variant by over 5 points, signifying the residual connection's role in effective knowledge transfer.

However, in some tasks like MBPP and MedQA, the non-residual model slightly outperforms *MoDULA-Res*. This nuance suggests a need to further analyze the residual connection's mechanism across various tasks to improve the model's robustness.

In conclusion, the findings affirm the *MoDULA-Res* method's efficacy. Residual connections significantly enhance overall performance on domain-specific tasks, offering a promising avenue for future enhancements in the PEFT paradigm. Continued exploration of residual connections in multi-task learning is expected to yield more powerful and versatile language models.

B GPT-4 Judge Prompt for E-commerce Tasks



We would like to request your evaluation of the product title optimization performed by the AI assistant. Please provide a score reflecting the effectiveness of the rewritten product title in terms of search engine visibility and user engagement.

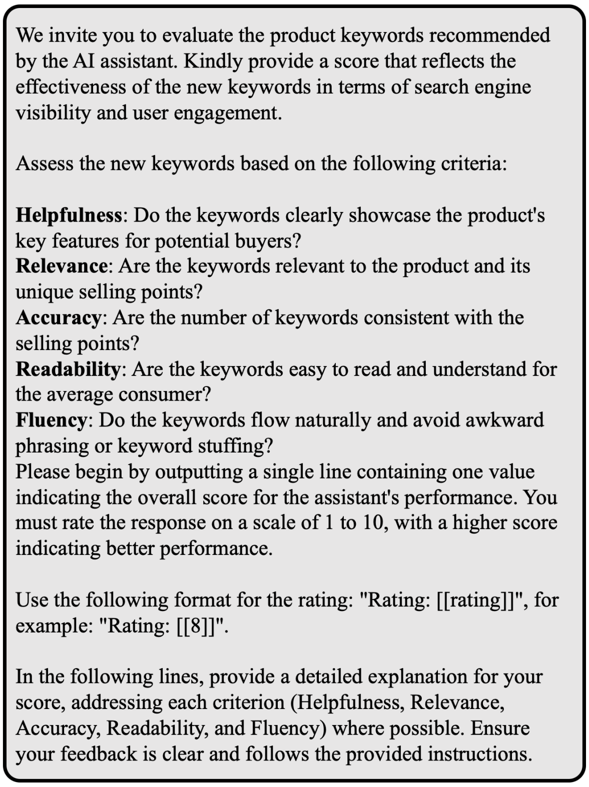
Rate the optimized product title on the following criteria:

Helpfulness: Does the title clearly showcase the product's key features for potential buyers?
Relevance: Is the title relevant to the product and its unique selling points?
Accuracy: Does the title accurately represent the product, including the brand and manufacturer?
Readability: Is the title easy to read and understand for the average consumer?
Fluency: Does the title flow naturally and avoid awkward phrasing or keyword stuffing?
Please first output a single line containing one value indicating the overall score for the assistant's performance. You must rate the response on a scale of 1 to 10, with a higher score indicating a better performance.

Use the following format for the rating: \"Rating: [[rating]]\", for example: \"Rating: [[8]]\".

In the subsequent lines, please provide a thorough explanation for your given score, addressing each individual criterion (Helpfulness, Relevance, Accuracy, Readability, and Fluency) where possible. Please ensure your feedback is clear and adheres to the instructions provided.

Figure 4: The GPT-4 judge prompt for Title-Optimization task.



We invite you to evaluate the product keywords recommended by the AI assistant. Kindly provide a score that reflects the effectiveness of the new keywords in terms of search engine visibility and user engagement.

Assess the new keywords based on the following criteria:

Helpfulness: Do the keywords clearly showcase the product's key features for potential buyers?
Relevance: Are the keywords relevant to the product and its unique selling points?
Accuracy: Are the number of keywords consistent with the selling points?
Readability: Are the keywords easy to read and understand for the average consumer?
Fluency: Do the keywords flow naturally and avoid awkward phrasing or keyword stuffing?
Please begin by outputting a single line containing one value indicating the overall score for the assistant's performance. You must rate the response on a scale of 1 to 10, with a higher score indicating better performance.

Use the following format for the rating: "Rating: [[rating]]", for example: "Rating: [[8]]".

In the following lines, provide a detailed explanation for your score, addressing each criterion (Helpfulness, Relevance, Accuracy, Readability, and Fluency) where possible. Ensure your feedback is clear and follows the provided instructions.

Figure 5: The GPT-4 judge prompt for Keyword-Recommendation task.

Base Model	Method	Avg.	GSM8K	Arithmetic	MathQA	HumanEval	MBPP	Medical	MedQA
Qwen-7B	*MoDULA-Res*	**51.36**	**46.63**	**90.37**	**37.98**	**25.00**	33.00	**82.00**	**44.55**
	MoDULA w/o. Res	49.65	46.47	85.35	35.24	24.39	**33.60**	78.20	44.31
LLaMA-2-7B	*MoDULA-Res*	**39.62**	**22.37**	**70.66**	**31.73**	15.24	**22.80**	**85.20**	29.31
	MoDULA w/o. Res	37.85	20.40	61.82	30.45	**16.46**	22.80	79.00	**34.01**
Yi-6B	*MoDULA-Res*	**48.61**	34.50	**92.72**	**36.29**	**16.46**	**24.40**	**85.80**	50.12
	MoDULA w/o. Res	45.60	**34.87**	90.01	35.94	14.24	15.80	77.60	**50.74**

Table 6: Experimental results of *MoDULA-Res* and *MoDULA* w/o. Res on domain-specific benchmarks.