SAMoRA: Semantic-Aware Mixture of LoRA Experts for Task-Adaptive Learning

Boyan Shi[1,3], Wei Chen[2,*], Shuyuan Zhao[1,3], Junfeng Shen[1,3], Shengnan Guo[1,3], Shaojiang Wang[4,5,*], Huaiyu Wan[1,3]

[1]School of Computer Science and Technology, Beijing Jiaotong University, China
[2]Guangxi Key Lab of Trusted Software, Guilin University of Electronic Technology, China
[3]Beijing Key Lab of Traffic Data Mining and Embodied Intelligence, China
[4]Institute of AI for Industries, Chinese Academy of Sciences, China
[5]Nanjing Institute of Software Technology, China

boyan118@bjtu.edu.cn *Correspondence: w_chen@guet.edu.cn, wangshaojiang@iaii.ac.cn

Abstract

The combination of Mixture-of-Experts (MoE) and Low-Rank Adaptation (LoRA) has shown significant potential for enhancing the multi-task learning capabilities of Large Language Models. However, existing methods face two primary challenges: (1)Imprecise Routing in the current MoE-LoRA method fails to explicitly match input semantics with expert capabilities, leading to weak expert specialization. (2)Uniform weight fusion strategies struggle to provide adaptive update strengths, overlooking the varying complexity of different tasks. To address these limitations, we propose **SAMoRA** (**S**emantic-**A**ware **M**ixture **o**f Lo**RA** Experts), a novel parameter-efficient fine-tuning framework tailored for task-adaptive learning. Specifically, A **Semantic-Aware Router** is proposed to explicitly align textual semantics with the most suitable experts for precise routing. A **Task-Adaptive Scaling** mechanism is designed to regulate expert contributions based on specific task requirements dynamically. In addition, a novel regularization objective is proposed to jointly promote expert specialization and effective scaling. Extensive experiments on multiple multi-task benchmarks demonstrate that SAMoRA significantly outperforms the state-of-the-art methods and holds excellent task generalization capabilities. Code is available at https://github.com/boyan-code/SAMoRA

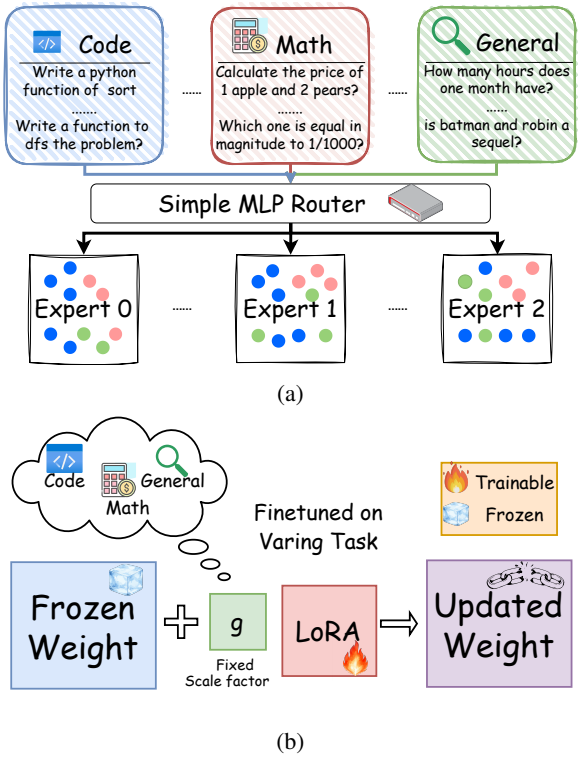

Figure 1: **Illustration of limitations in existing mechanisms.** (a) MLP-based Routing: Fails to explicitly match tasks with expert capabilities, resulting in expert homogenization. (b) Uniform Weight Fusion: Applies a uniform update strength across diverse tasks, ignoring specific requirements and limiting multi-task generalization.

1 Introduction

Large Language Models (LLMs) have achieved impressive performance across a wide range of domains, particularly in natural language processing (NLP) tasks such as content generation and question answering (Hong et al., 2025; Xu et al., 2023; Chen et al., 2025a). This success largely stems from their massive parameter counts and pre-training on large-scale, diverse corpora, which endow LLMs with strong generalization capabilities and robust performance across diverse and complex tasks (Qin et al., 2023; Raffel et al., 2020; Chen et al., 2025b), yet inevitably imposes a substantial parameter burden during fine-tuning.

To mitigate the computational burden of full fine-tuning, Low-Rank Adaptation (LoRA) has emerged as a leading Parameter-Efficient Fine-Tuning (PEFT) strategy (Hu et al., 2022). LoRA injects trainable low-rank matrices into the frozen backbone and merges the updates via a uniform scaling factor. However, while effective for single tasks, this fixed structure limits performance

Findings of the Association for Computational Linguistics: ACL 2026, pages 28173–28188

in complex multi-task scenarios, as a single set of parameters cannot adequately handle diverse task requirements. To address this, recent studies have integrated Mixture-of-Experts (MoE) architectures with LoRA (MoE-LoRA) (Liu et al., 2024a). These methods treat multiple LoRA modules as experts and employ a Multi-Layer Perceptron (MLP) based router to selectively activate them. While these approaches have demonstrated notable success in enhancing model capacity, they still face two critical challenges:

(1) Current routing mechanisms fail to explicitly associate tasks with expert capabilities, leading to imprecise routing. Existing MoE-LoRA methods rely on MLP routers that prioritize learned data distributions over actual expert proficiencies (Tian et al., 2024). As illustrated in Figure 1(a), these strategies fail to explicitly match input semantics with expert expertise, often resulting in homogenized experts that lack distinct roles. Consequently, this inability to specialize prevents the model from handling diverse requirements effectively, leading to suboptimal capabilities in multi-task scenarios. **(2) Uniform weight fusion strategies fail to provide adaptive adjustments for diverse tasks, limiting multi-task generalization.** As shown in Figure 1(b), standard approaches employ a globally fixed scale factor that applies a uniform update strength across all inputs. However, multi-task scenarios involve tasks with varying complexity, where some require significant parameter shifts while others need only minor adjustments. Applying a uniform strategy ignores these distinct requirements, forcing a rigid "one-size-fits-all" adaptation. This lack of flexibility prevents the model from effectively adapting to specific task needs, thereby constraining its overall generalization capability in complex multi-task environments.

To address these challenges, we propose **SAMoRA** (**S**emantic-**A**ware **M**ixture **o**f Lo**RA** Experts), a novel framework tailored for task-adaptive learning. Specifically, SAMoRA consists of a Semantic-Aware Router to explicitly align input semantics with expert capabilities, a Task-Adaptive Scaling mechanism to dynamically regulate expert contributions based on specific task demands, and specialized loss constraints to enforce expert distinctiveness and ensure robust multi-task performance.

The contributions of this work are as follows:

- We propose **SAMoRA**, a novel MoE-LoRA framework enabling precise semantic-aware expert routing and significantly enhancing multi-task generalization capabilities.
- We introduce a **Semantic-Aware Router** to enforce explicit alignment between input semantics and expert capabilities, coupled with a **Task-Adaptive Scaling** mechanism that dynamically regulates parameter updates to effectively adapt to diverse task requirements.
- We design specialized loss constraints to enforce expert distinctiveness and regularize scaling factors, ensuring specialized expert roles and robust performance.
- Extensive experiments across diverse multi-task benchmarks demonstrate that SAMoRA consistently outperforms existing baselines, achieving State-of-the-Art performance.

2 Related Work

2.1 Mixture of Experts.

MoE was initially proposed to decompose complex tasks into simpler subtasks, where a router dynamically assigns different inputs to specialized expert subnetworks (Jacobs et al., 1991). A key later advancement was the sparsely-gated MoE, which activates only a small subset of experts per forward pass to significantly improve computational efficiency (Shazeer et al., 2017). This sparse-gating mechanism was subsequently extended to Transformer architectures, further enhancing training efficiency and model scalability (Lepikhin et al., 2021). Subsequent strategies have further optimized routing mechanisms, such as simplified top-1 routing for stability (Fedus et al., 2022) and differentiable soft routing for effective expert combination (Muqeeth et al., 2024).

Despite these architectural improvements, current methods share a fundamental limitation: they rely on implicit routing strategies that lack explicit semantic guidance. These approaches typically map inputs to experts based on learned statistical distributions rather than establishing an explicit association between input semantics and expert capabilities. Consequently, the routing decision remains decoupled from actual expert specialization, hindering the model's ability to precisely match diverse inputs to the most suitable experts based on their semantic features.

2.2 LoRA for Multi-Task Learning

LoRA has attracted widespread attention due to its ability to achieve performance comparable to full fine-tuning under limited computational resources. However, its performance in complex multi-task scenarios remains suboptimal. To address this, several extensions have been proposed to enhance adaptability. MultiLoRA introduces a parallelized design with learnable scaling factors to decouple task-specific features (Wang et al., 2023), while MTL-LoRA employs task-specific transformation matrices to capture distinct information (Yang et al., 2025). Furthermore, methods like MoELoRA and HydraLoRA integrate MoE architectures, treating LoRA modules as experts to improve generalization and parameter efficiency (Liao et al., 2025; Tian et al., 2024).

Despite these architectural advancements, these methods share a fundamental limitation in their weight fusion mechanism. Most approaches rely on uniform scaling strategies to merge LoRA updates with the pre-trained model. This fixed approach ignores the varying complexity of different tasks, where some require significant parameter shifts while others need only minor adjustments. Consequently, applying the same update strength to all tasks fails to meet specific requirements, thereby limiting the model's overall multi-task adaptation performance.

3 Preliminary

3.1 PEFT for LLMs

PEFT for LLMs involves adapting pretrained models to downstream tasks by introducing a small set of trainable parameters ΔW, while keeping the original model weights W frozen. The model is jointly trained on multiple tasks in multi-task scenarios to learn shared and task-specific representations (Wei et al., 2022). The training objective is to fine-tune ΔW such that the conditional probability P of autoregressively generating target sequences across all tasks is maximized. Formally, the training loss can be written as:

$$\mathcal{L}_{\text{task}}(\Delta W) = \sum_{(s_{\text{in}}, s_{\text{out}}) \in \mathcal{D}} \sum_{i=1}^{|s_{\text{out}}|} \log P_{W+\Delta W}\left(s_{\text{out}}^{(i)} \mid s_{\text{in}}, s_{\text{out}}^{(<i)}\right), \tag{1}$$

where $\mathcal{D}$ denotes the training dataset containing input-output sentence pairs $(s_{\text{in}}, s_{\text{out}})$ from multiple tasks. This objective formalizes the autoregressive training process, where the model predicts target sentence incrementally by adapting only the incremental parameters ΔW.

3.2 Mixture of LoRA Experts

LoRA implements PEFT by freezing the original pretrained weights W and introducing two trainable low-rank matrices. For a weight matrix $W \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$, these matrices are specifically defined as $A \in \mathbb{R}^{r \times d_{\text{in}}}$ and $B \in \mathbb{R}^{d_{\text{out}} \times r}$, where the rank r is significantly smaller than the original dimensions. The resulting product BA provides a low-rank update ΔW to W, enabling effective adaptation with minimal additional parameters (Hu et al., 2022). The LoRA update process is illustrated in Figure 1(b).

To leverage the parameter-efficiency of LoRA for complex multi-task scenarios, a promising direction in recent work has been to integrate it with MoE (Yang et al., 2025; Liu et al., 2024b; Feng et al., 2024). By structuring multiple LoRAs as lightweight experts within the attention and feed-forward layers of an LLM, the forward pass in such a layer is formalized as:

$$Y = WX + \sum_{i=1}^{N} g_i B_i A_i X, \tag{2}$$

where $X \in \text{Emb}(s_{\text{in}})$ is a hidden representation derived from the input sentence s_{in}, and Y is the corresponding output. The set $\{A_i, B_i\}_{i=1}^{N}$ represents N distinct LoRA experts. The gating weights g_i are dynamically generated by a router conditioned on input X, determining which expert to activate.

4 Methodology

As illustrated in Figure 2, SAMoRA integrates two core components: a **Semantic-Aware Router** designed to explicitly match input semantics with expert expertise, and a **Task-Adaptive Scaling** mechanism that dynamically regulates update strengths to meet specific task requirements. In the following, we introduce these components in detail.

4.1 Semantic-Aware Router

Most existing MoE approaches rely on MLP-based routing strategies that often fail to associate input contents with expert capabilities. To address this, we introduce a **Semantic-Aware Router** designed

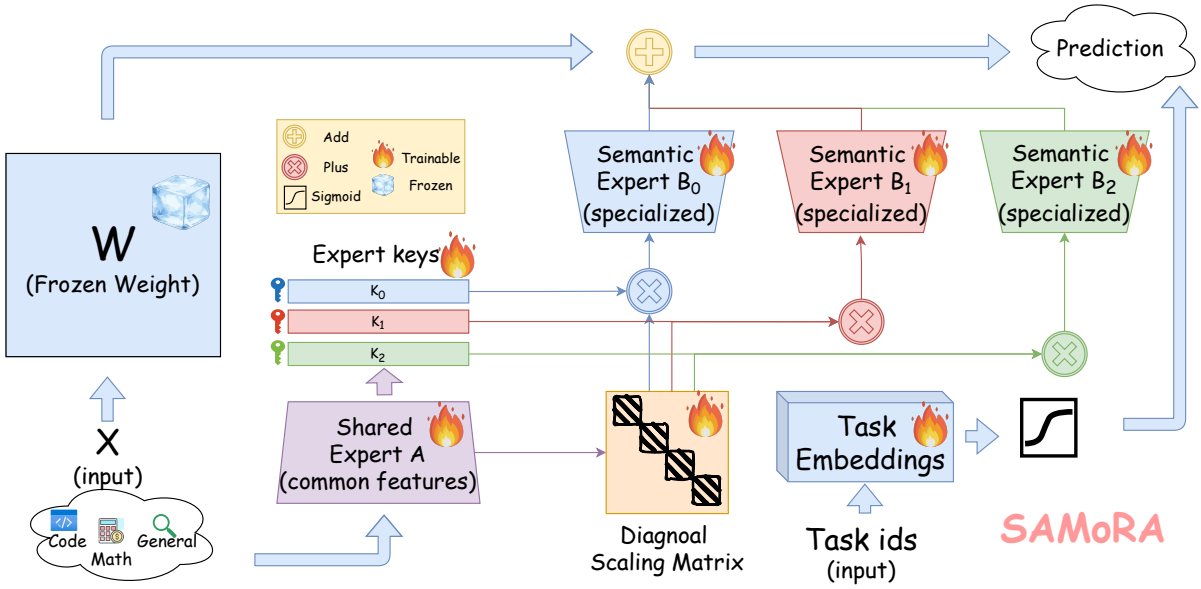

Figure 2: Overview of our SAMoRA. We design a Semantic-Aware Router and a Task-Adaptive Scaling mechanism, integrated within an asymmetric MoE-LoRA architecture consisting of a shared Expert A and multiple Semantic Experts B.

to explicitly match input semantics with expert expertise.

Semantic Extraction via Shared Expert. To realize explicit routing, the model must first effectively grasp the semantic intent of the input. Inspired by HydraLoRA (Tian et al., 2024), we establish an asymmetric architecture by utilizing a single shared expert $A \in \mathbb{R}^{r \times d_{\text{in}}}$, while maintaining multiple experts $\{B_i\}_{i=1}^{N}$ to capture distinct semantic capabilities. This shared component naturally functions as a semantic encoder, eliminating the need for a separate, decoupled routing network. This shared module extracts a compact, unified semantic representation $\mathbf{h} = AX$ from the input X. By using the shared expert A, we ensure that the routing decision is grounded in the same feature space used for expert computation, facilitating consistent semantics aggregation. Building upon Eq. (2), the core forward process is reformulated as:

$$Y = WX + \sum_{i=1}^{N} g_i B_i \mathbf{h} = WX + \sum_{i=1}^{N} g_i B_i (AX). \tag{3}$$

Explicit Matching with Expert Keys. With the extracted semantic features $\mathbf{h}$, the next step is to align them with the specific capabilities of the semantic experts $\{B_i\}_{i=1}^{N}$. To this end, we assign a trainable **Expert Key** $k_i \in \mathbb{R}^r$ to each expert B_i. These keys function as semantic anchors, explicitly representing the unique specialization learned by each expert. During training, the keys are optimized alongside the experts, ensuring that k_i moves closer to the semantic clusters that expert B_i is best at handling. The routing score g_i is then derived by measuring the Cosine Similarity between the input's semantic representation $\mathbf{h}$ and each expert key k_i:

$$g_i = \frac{\exp\left(\cos\left(\mathbf{h}, k_i\right)/\tau\right)}{\sum_{j=1}^{N} \exp\left(\cos\left(\mathbf{h}, k_j\right)/\tau\right)}, \tag{4}$$

where τ is a temperature coefficient that regulates the strictness of the matching. A smaller τ sharpens the distribution, forcing the router to strictly select only the expert with the highest semantic alignment, while a larger τ softens this constraint, allowing for broader expert collaboration. This mechanism ensures that inputs are routed based on explicit semantic similarity rather than implicit statistical bias.

4.2 Task-Adaptive Scaling

As illustrated in Figure 1(b), standard LoRA employs a uniform scaling factor to merge the updates. However, this fixed approach is problematic in multi-task scenarios as it ignores the varying complexity of different tasks. Some tasks require significant parameter shifts while others need only

minor adjustments. Consequently, applying the same update strength to all tasks fails to meet specific requirements, limiting the model's adaptability. To address this, we propose a **Task-Adaptive Scaling** mechanism that dynamically regulates the update magnitude based on specific task demands.

Spectral Initialization via SVD. First, inspired by recent work (Yuan et al., 2025; Zhao et al., 2025), we aim to ensure our asymmetric structure starts with a theoretically grounded scale alignment. We introduce a trainable Diagonal Scaling Matrix $S \in \mathbb{R}^{r\times r}$ positioned between the shared expert A and the semantic experts B_i. By performing Singular Value Decomposition (SVD) on the pre-trained weight $W = U\Sigma V^\top$, we initialize S using the top-r singular values:

$$S = \Sigma_{1:r,1:r} = \mathrm{diag}(\sigma_1, \ldots, \sigma_r). \tag{5}$$

This design aligns our components with the dominant semantic directions of the original weights, providing a stable structural basis for subsequent adaptation.

Task-Dependent Dynamic Regulation. Building upon this aligned basis, we introduce a task-driven mechanism to dynamically control the fusion ratio. We assign a learnable **Task Embedding** $e_{\text{task}} \in \mathbb{R}^{d_g}$ to each task, which captures latent task characteristics such as complexity and domain divergence. To determine the optimal update strength for a given task, we project this embedding into a scalar gating factor $g_{\text{task}} \in (0, 1)$ via a non-linear mapping:

$$g_{\text{task}} = \sigma(W_{\text{gate}} e_{\text{task}} + b_{\text{gate}}), \tag{6}$$

where $\sigma(\cdot)$ is the sigmoid function. This mechanism allows the model to dynamically adjust the update strength based on input features. It assigns larger scales for tasks needing significant adaptation and smaller scales for those requiring only minor adjustments, effectively meeting diverse task requirements.

By integrating the SVD-based alignment and task-dependent regulation into the formulations of Eq. (2) and Eq. (3), the final output Y is derived as:

$$Y = WX + g_{\text{task}} \sum_{i=1}^{N} g_i B_i(SAX). \tag{7}$$

4.3 Training Objective

To ensure the effective implementation of our proposed mechanisms, we incorporate two specialized regularization terms alongside the standard LLM generation loss. Specifically, these terms are designed to align Expert Keys with their corresponding experts and impose the necessary SVD constraints for the Task-Adaptive Scaling mechanism. The total training objective is formulated as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}} + \lambda_{\text{orth}} \cdot \mathcal{L}_{\text{orth}} + \lambda_{\text{match}} \cdot \mathcal{L}_{\text{match}}, \tag{8}$$

where $\mathcal{L}_{\text{task}}$ denotes the multi-task language modeling loss Ep. (1), and λ_{orth}, λ_{match} are scalar hyperparameters weighting the auxiliary constraints.

Orthogonality Regularization for Scale Decoupling. We introduce an orthogonality regularization term $\mathcal{L}_{\text{orth}}$ to strictly decouple *directional semantics* from *magnitude scaling*. In our SVD-based design, the diagonal matrix S and the gating factor g_{task} are intended to handle all "scaling" effects. Specifically, we force the rows of the shared encoder A and the columns of each semantic expert B_i to be orthonormal:

$$\mathcal{L}_{\text{orth}} = \|AA^\top - I\|_F^2 + \sum_{i=1}^{N} \|B_i^\top B_i - I\|_F^2, \tag{9}$$

where $I \in \mathbb{R}^{r\times r}$ is the identity matrix. By enforcing this constraint, A and B_i focus purely on learning distinct semantic directions, ensuring that the control of adaptation strength remains exclusively within the purview of our Task-Adaptive Scaling mechanism.

Semantic Match Regularization via KL Divergence. The effectiveness of our Semantic-Aware Router hinges on the semantic consistency between the learnable key k_i and the functional specialization of the expert B_i. Any misalignment between k_i and B_i inevitably leads to erroneous expert selection. To mitigate this, we introduce a regularization loss that explicitly minimizes the divergence between k_i and the semantic representation derived from B_i. We detail the specific implementation steps as follows.

(1) Extracting Representative Vectors. Since the expert $B_i \in \mathbb{R}^{d_{\text{out}}\times r}$ is a matrix while the key $k_i \in \mathbb{R}^r$ is a vector, we obtain a representative vector b_i from each expert. This is achieved by mean-pooling the row vectors of B_i, which aggregates

Method	TP (%)	BoolQ	PIQA	SIQA	WinoG	ARC-C	ARC-E	OBQA	HellaS	CSQA	Avg.
Backbone: Llama3.1-8B											
LoRA†	2.09	70.43	82.97	76.00	71.11	77.56	85.77	81.60	93.00	77.40	79.54
MultiLoRA	0.26	70.95	80.81	80.91	82.15	71.70	86.12	80.60	94.01	80.34	80.84
MoORE	0.77	74.49	88.63	82.99	87.74	79.95	88.80	86.20	95.48	84.60	85.43
HydraLoRA	0.17	74.31	90.15	82.49	88.47	84.06	92.18	87.80	93.18	83.81	86.27
MTL-LoRA	0.16	74.34	89.90	82.95	88.08	84.55	93.81	88.20	95.15	83.94	86.77
SAMoRA (Ours)	0.15	**74.89**	**90.37**	**83.32**	**88.95**	**86.35**	**94.70**	**89.80**	**95.97**	**84.85**	**87.64**
Backbone: Qwen3-8B											
LoRA	0.74	73.80	91.45	83.00	88.39	92.40	97.60	90.20	94.60	86.32	88.64
MultiLoRA	0.29	71.89	89.88	81.83	83.89	92.15	97.60	90.60	93.07	85.74	87.64
MoELoRA	0.56	73.90	91.18	81.47	83.10	92.49	97.34	89.60	92.30	84.43	87.31
HydraLoRA	0.16	73.14	90.69	83.21	87.92	**92.90**	97.47	89.40	94.60	87.01	90.33
MoORE	0.84	73.60	91.26	80.80	86.55	90.10	93.30	90.20	94.09	86.56	90.28
MTL-LoRA	0.16	73.51	91.13	82.08	88.87	92.15	97.55	91.40	95.47	86.08	90.98
SAMoRA (Ours)	0.15	**74.68**	**92.00**	**83.78**	**88.95**	92.58	**97.94**	**91.80**	**96.01**	**87.31**	**91.71**

Table 1: Results of comparison experiments across Commonsense Reasoning benchmarks. TP indicates Trainable Parameters (%). †means the results from MoORE (Yuan et al., 2025). **Bold**: Best results; Underline: Second-best results.

the features learned by that expert:

$$b_i = \frac{1}{d_{\text{out}}} \sum_{j=1}^{d_{\text{out}}} B_i^{(j)} \in \mathbb{R}^r. \tag{10}$$

(2) Alignment via Distribution Matching. To align the routing key with the expert's actual capability, we map both the key k_i and the semantic centroid b_i into probability distributions ($P_k^{(i)}$ and $P_b^{(i)}$) via Softmax. We then minimize the Kullback-Leibler (KL) divergence between them:

$$\mathcal{L}_{\text{match}} = \frac{1}{N} \sum_{i=1}^{N} D_{\text{KL}} \left(P_b^{(i)} \parallel P_k^{(i)} \right). \tag{11}$$

Crucially, we employ the direction $D_{\text{KL}}(P_{\text{Expert}} \parallel P_{\text{Key}})$. This effectively treats the expert's functional distribution $P_b^{(i)}$ as the target, compelling the key $P_k^{(i)}$ to shift towards and accurately represent the expert's specialization. This ensures consistency between the routing keys and the actual expert characteristics.

4.4 Complexity Analysis

To demonstrate the computational efficiency and parameter economy of our framework, we compare the complexity of SAMoRA with the standard MoE-LoRA paradigm. For a comprehensive breakdown of all baselines and the detailed analysis process, please refer to Appendix A.

Standard MoE-LoRA architectures typically assign independent down-projection and up-projection matrices to each expert. This results in a parameter complexity of $\mathcal{O}(N(d_{\text{in}} + d_{\text{out}})r)$ and necessitates high-dimensional computations for routing, incurring a cost of $\mathcal{O}(Nd_{\text{in}})$.

In contrast, SAMoRA optimizes both storage and inference efficiency through its asymmetric design and low-rank routing mechanism. Specifically, by using a shared expert A, SAMoRA eliminates the redundancy of learning separate input projections, reducing the parameter complexity to $\mathcal{O}((d_{\text{in}} + Nd_{\text{out}})r)$. Furthermore, unlike standard methods that calculate routing scores in the high-dimensional input space (d_{in}), SAMoRA performs routing in the low-rank latent space (r). Given that $r \ll d_{\text{in}}$, this design significantly reduces the routing FLOPs from $\mathcal{O}(Nd_{\text{in}})$ to $\mathcal{O}(Nr)$, ensuring minimal latency overhead during inference.

Overall, SAMoRA achieves a substantial reduction in both parameter count and computational cost compared to other MoE-LoRA baselines, offering a superior trade-off between model capacity and efficiency.

5 Experiments

5.1 Experiment Setting

Dataset We evaluate SAMoRA on two challenging multi-task benchmarks that target different capabilities of LLMs: **(1) Commonsense Reasoning**: A curated benchmark comprising nine representative commonsense reasoning tasks: ARC-Challenge (ARC-C), ARC-Easy (ARC-E) (Clark et al., 2018), OpenBookQA (OBQA) (Mihaylov et al., 2018), PIQA (Bisk et al., 2020), SocialIQA

Model	TP (%)	CoLA (Mcc.)	MNLI (Acc.)	MRPC (Acc.)	QNLI (Acc.)	QQP (Acc.)	RTE (Acc.)	SST2 (Acc.)	Avg.
Backbone: Qwen3-8B									
LoRA	0.21	64.06	91.84	88.20	96.01	91.12	91.16	96.50	88.41
MultiLoRA	0.67	58.50	90.83	80.88	94.12	89.57	85.56	97.02	85.21
MoeLoRA	0.60	67.01	91.71	83.82	95.88	90.07	91.69	96.67	88.12
HydraLoRA	0.20	67.04	91.90	85.04	90.51	90.68	75.45	96.55	85.31
MTL-LoRA	0.20	66.32	91.93	**89.46**	95.77	91.39	92.77	96.67	89.18
SAMoRA (Ours)	0.18	**69.75**	**91.96**	89.21	**96.17**	**91.41**	**94.22**	**97.13**	**89.98**
w/o Router	0.20	68.19	92.08	89.46	95.91	91.41	91.33	97.13	89.36
w/o Scaling	0.18	66.43	91.93	88.97	95.93	90.84	91.33	96.90	88.90
w/o $\mathcal{L}_{\text{orth}}$	0.15	68.32	91.99	87.99	96.11	90.89	90.61	97.01	88.99
w/o $\mathcal{L}_{\text{match}}$	0.15	68.73	91.88	87.25	95.97	90.63	91.69	97.01	89.02

Table 2: Results of comparison experiments across GLUE benchmark. The upper block presents the baselines, while the lower block reports the performance of SAMoRA and its ablation variants. **Bold**: Best results; Underline: Second-best results.

(SIQA) (Sap et al., 2019), BoolQ (Wang et al., 2019a), HellaSwag (HellaS) (Zellers et al., 2019), Winogrande (WinoG) (Sakaguchi et al., 2021) and CommonsenseQA(CSQA) (Talmor et al., 2019). These datasets cover diverse commonsense challenges, including science QA, physical and social reasoning, and everyday inference, and are widely used to evaluate the multi-task capabilities of LLMs. **(2) Natural Language Understanding**: We use widely used subset of seven tasks from the GLUE benchmark (Wang et al., 2019b), including CoLA, SST-2, MRPC, QQP, MNLI, QNLI, and RTE. These tasks assess linguistic phenomena such as grammaticality, sentiment analysis, paraphrase detection, and textual entailment, thus comprehensively evaluating general language understanding capabilities.

Following the same train-test split protocol and instruction prompts as in prior works (Yang et al., 2025; Liu et al., 2024b), we conduct our evaluation. Detailed descriptions of the data splits and prompt formats are provided in Appendix B.1.

Implementation Details. We conduct experiments using Qwen3-8B (Team, 2025) and LLaMA3.1-8B (Team, 2024) as the backbone architectures. We compare SAMoRA against a comprehensive set of competitive baselines, including LoRA (Hu et al., 2022), MultiLoRA (Wang et al., 2023), MoELoRA (Liu et al., 2024a), HydraLoRA (Tian et al., 2024), MTL-LoRA (Yang et al., 2025), and MoORE (Yuan et al., 2025). To ensure a fair comparison, we modify the hyperparameters of the baselines to make the number of trainable parameters comparable for each method. We report detailed training settings for all baselines in Appendix B.2.

5.2 Overall Performance

As shown in Table 1 and Table 2, SAMoRA consistently outperforms existing baselines on both Llama3.1-8b and Qwen3-8b across Commonsense Reasoning and GLUE benchmarks, while maintaining strong parameter efficiency. Compared to the single-adapter method LoRA, SAMoRA demonstrates clear advantages in handling diverse tasks, underscoring the importance of multi-expert architectures in multi-task adaptation.

Compared with MTL-LoRA and HydraLoRA, which rely on conventional MLP-based routers, SAMoRA enables more accurate and flexible expert selection through its semantic-aware routing mechanism. Regarding MoORE, it attempts to leverage the original LLM weights by exclusively training the router. However, this approach performs poorly due to the limited number of trainable parameters, which proves insufficient for effective adaptation on downstream tasks. Furthermore, while MoELoRA introduces task-specific experts and MultiLoRA assigns a separate trainable scale factor to each LoRA module, they fail to account for task-specific characteristics and varying task complexity simultaneously. In contrast, SAMoRA introduces a task-adaptive scaling mechanism that

dynamically modulates this balance, enabling more precise and efficient adaptation across diverse tasks with fewer trainable parameters.

5.3 Ablation Study

To better understand the effectiveness of each component in SAMoRA, we conduct a comprehensive ablation study. We evaluate the impact of the proposed semantic-aware router by comparing it against a conventional MLP-based router (w/o Router). We assess the contribution of the task-adaptive scaling mechanism by removing it across all tasks(w/o Scaling). In addition, we examine the influence of the auxiliary losses by removing the orthogonality loss (w/o $\mathcal{L}_{\text{orth}}$) and semantic match loss (w/o $\mathcal{L}_{\text{match}}$), respectively. The results are summarized in Table 2, and further implementation details are provided in Appendix C.1.

As presented in Table 2, SAMoRA consistently achieves the best performance across all tasks, validating the synergy of its components. Notably, removing the task-adaptive scaling mechanism (w/o Scaling) leads to the most significant performance degradation (a sharp drop from 69.75% to 66.43% on CoLA), underscoring its critical role in resolving task conflicts and mitigating negative transfer. Similarly, replacing the semantic-aware router with a standard MLP (w/o Router) results in a clear decline, confirming the necessity of explicit semantic alignment for precise expert allocation. Furthermore, excluding the auxiliary regularization terms (w/o $\mathcal{L}_{\text{orth}}$ and w/o $\mathcal{L}_{\text{match}}$) also impairs overall results, demonstrating their importance in maintaining expert distinctiveness and stabilizing training.

Analysis of Semantic-Aware Router. To investigate expert specialization, we visualize the PCA projections of the latent representations derived from the Semantic Expert B matrices. As illustrated in Figure 3, the standard MLP router results in entangled clusters with blurred boundaries. In contrast, our Semantic-Aware Router yields distinct and well-separated clusters for the Semantic Expert B modules, explicitly confirming that each expert has specialized in a specific semantic subspace. Detailed experimental settings are provided in Appendix C.2.

Analysis of Task-Adaptive Scaling. To validate the effectiveness of our mechanism, we visualize the learned scaling factors of SAMoRA trained on Qwen3-8B. Figure 4 displays the factors for the query (q), key (k), value (v), and output (o) projections within the final attention layer. The observed variations across different tasks validate the effectiveness of our proposed mechanism.

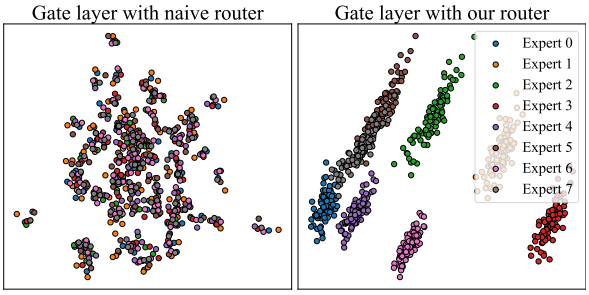

Figure 3: PCA visualization of expert features extracted from the final gate layer trained on Commonsense Reasoning dataset.

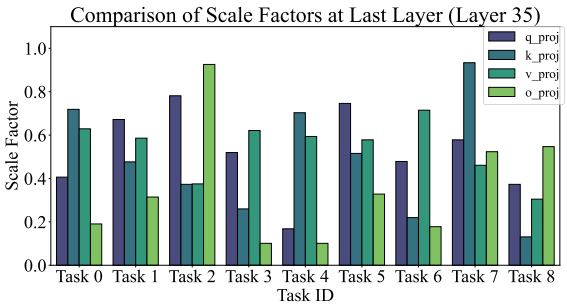

Figure 4: Visualization of task scaling factors across tasks trained on the Commonsense Reasoning dataset.

5.4 Sensitivity Analysis

We conduct a comprehensive sensitivity analysis on key hyperparameters, including model architecture (N, r, d_g) and training objectives ($\lambda_{\text{orth}}, \lambda_{\text{KL}}, \tau$), with detailed results provided in Appendix D. Overall, the model exhibits strong robustness across varying configurations. Notably, regarding the task embedding dimension d_g, we observe that compact embeddings are sufficient for effective routing; increasing d_g to excessive levels introduces unnecessary complexity that hinders convergence.

6 Conclusion

In this paper, we propose SAMoRA, a novel PEFT framework significantly enhancing multi-task generalization. By ensuring precise expert routing and dynamic task adaptation, our approach effectively secures robust and superior performance across diverse multi-task scenarios. Extensive experiments demonstrate that SAMoRA consistently outperforms existing baselines, achieving a favorable

trade-off between performance and parameter efficiency.

Limitations

In this paper, we conduct experiments on Commonsense Reasoning and GLUE benchmarks by fine-tuning models at the 8B parameter scale. Due to limited computational resources, the scalability of our framework to significantly larger foundation models (e.g., 70B scale or above) has not yet been empirically verified. Furthermore, there is a broader range of application scenarios unexplored, particularly in the multimodal domain, such as visual instruction tuning and visual question answering tasks. We plan to extend our method to these large-scale and multimodal settings in future work to further explore its generalization capabilities.

Acknowledgments

This work was suported by Frontier Technologies R&D Program of Jiangsu (Grant No. BF2024052), Nanjing Municipal Science and Technology Bureau (Grant No.202512136), and Chengdu Science and Technology Program (Grant No.2025-YF08-00097-GX).

References

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. PIQA: reasoning about physical commonsense in natural language. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7432–7439. AAAI Press.

Jiali Chen, Xusen Hei, Yuqi Xue, Zihan Wu, Jiayuan Xie, and Yi Cai. 2025a. Classic4children: Adapting chinese literary classics for children with large language model. In *Findings of the Association for Computational Linguistics: NAACL 2025, Albuquerque, New Mexico, USA, April 29 - May 4, 2025*, Findings of ACL, pages 2473–2488. Association for Computational Linguistics.

Jiali Chen, Yujie Jia, Zihan Wu, Jinyu Yang, Jianpeng Chen, Xusen Hei, Jiayuan Xie, Yi Cai, and Qing Li. 2025b. Expstar: Towards automatic commentary generation for multi-discipline scientific experiments. In *Proceedings of the 33rd ACM International Conference on Multimedia, MM 2025, Dublin, Ireland, October 27-31, 2025*, pages 6576–6585. ACM.

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. *CoRR*, abs/1803.05457.

William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *J. Mach. Learn. Res.*, 23:120:1–120:39.

Wenfeng Feng, Chuzhan Hao, Yuewei Zhang, Yu Han, and Hao Wang. 2024. Mixture-of-loras: An efficient multitask tuning method for large language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC/COLING 2024, 20-25 May, 2024, Torino, Italy*, pages 11371–11380. ELRA and ICCL.

Wenyi Hong, Wenmeng Yu, Xiaotao Gu, Guo Wang, Guobing Gan, Haomiao Tang, Jiale Cheng, Ji Qi, Junhui Ji, Lihang Pan, Shuaiqi Duan, Weihan Wang, Yan Wang, Yean Cheng, Zehai He, Zhe Su, Zhen Yang, Ziyang Pan, Aohan Zeng, and 58 others. 2025. Glm-4.1v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning. *CoRR*, abs/2507.01006.

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

Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. 1991. Adaptive mixtures of local experts. *Neural Comput.*, 3(1):79–87.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. 2021. Gshard: Scaling giant models with conditional computation and automatic sharding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Mengqi Liao, Wei Chen, Junfeng Shen, Shengnan Guo, and Huaiyu Wan. 2025. Hmora: Making llms more effective with hierarchical mixture of lora experts. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Qidong Liu, Xian Wu, Xiangyu Zhao, Yuanshao Zhu, Derong Xu, Feng Tian, and Yefeng Zheng. 2024a. 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, SIGIR 2024, Washington DC, USA, July 14-18, 2024*, pages 1104–1114. ACM.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. 2024b. Dora: Weight-decomposed low-rank adaptation. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? A new dataset for open book question answering. *CoRR*, abs/1809.02789.

Mohammed Muqeeth, Haokun Liu, and Colin Raffel. 2024. Soft merging of experts with adaptive routing. *Trans. Mach. Learn. Res.*, 2024.

Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi Yang. 2023. Is chatgpt a general-purpose natural language processing task solver? In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 1339–1384. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

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

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019. Socialiqa: Commonsense reasoning about social interactions. *CoRR*, abs/1904.09728.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. 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 4149–4158.

Llama Team. 2024. The llama 3 herd of models. *CoRR*, abs/2407.21783.

Qwen Team. 2025. Qwen3 technical report. *CoRR*, abs/2505.09388.

Chunlin Tian, Zhan Shi, Zhijiang Guo, Li Li, and Cheng-Zhong Xu. 2024. Hydralora: An asymmetric lora architecture for efficient fine-tuning. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019a. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 3261–3275.

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

Yiming Wang, Yu Lin, Xiaodong Zeng, and Guannan Zhang. 2023. Multilora: Democratizing lora for better multi-task learning. *CoRR*, abs/2311.11501.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. 2024. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned language models are zero-shot learners. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

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

Yaming Yang, Dilxat Muhtar, Yelong Shen, Yuefeng Zhan, Jianfeng Liu, Yujing Wang, Hao Sun, Weiwei Deng, Feng Sun, Qi Zhang, Weizhu Chen, and Yunhai Tong. 2025. Mtl-lora: Low-rank adaptation for multi-task learning. In *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA*, pages 22010–22018. AAAI Press.

Shen Yuan, Yin Zheng, Taifeng Wang, Binbin Liu, and Hongteng Xu. 2025. Moore: Svd-based model moeization for conflict-and oblivion-resistant multi-task adaptation. *arXiv preprint arXiv:2506.14436*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4791–4800. Association for Computational Linguistics.

Shuyuan Zhao, Wei Chen, Boyan Shi, Liyong Zhou, Shuohao Lin, and Huaiyu Wan. 2025. Spatial-temporal knowledge distillation for takeaway recommendation. In *Thirty-Ninth AAAI Conference on Artificial Intelligence, Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence, Fifteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2025, Philadelphia, PA, USA, February 25 - March 4, 2025*, pages 13365–13373. AAAI Press.

A Complexity Analysis

A.1 Theoretical Analysis

In this section, we analyze the theoretical complexity of SAMoRA in terms of trainable parameters and computational overhead. We compare our method against standard LoRA (Hu et al., 2022) and representative MoE-based PEFT frameworks, including MoeLoRA (Liu et al., 2024a), HydraLoRA (Tian et al., 2024) and MTL-LoRA (Yang et al., 2025).

For clarity, we define the following notations: d_{in} and d_{out} denote the input and output dimensions of the adapter layer, respectively. r represents the low-rank dimension, N is the number of experts, and K is the number of tasks.

Parameter Efficiency. The comparison of learnable parameters is summarized in Table 3.

- **Standard LoRA** employs a single pair of low-rank matrices per layer, resulting in $(d_{\text{in}} + d_{\text{out}})r$ parameters. It serves as the most parameter-efficient baseline but lacks multi-task flexibility.

- **MoELoRA** adopts a task-conditioned routing mechanism. It introduces a task embedding layer (Kd_g) and a router projection matrix (d_gN) to generate routing probabilities based on task IDs. Unlike the asymmetric design in SAMoRA, MoELoRA maintains *fully independent* low-rank experts. Consequently, its parameter complexity for the adapters is $N(d_{\text{in}} + d_{\text{out}})r$, which is significantly higher than shared-weight approaches. The total parameter count is given by $N(d_{\text{in}} + d_{\text{out}})r + Kd_g + Nd_g$, where d_g is the task embedding dimension.

- **HydraLoRA**, **MTL-LoRA** and **SAMoRA** adopt an *asymmetric expert architecture*. To optimize parameter efficiency, we share the projection matrix on the input side ($A \in \mathbb{R}^{d_{\text{in}} \times r}$), while maintaining N expert-specific matrices on the output side ($B \in \mathbb{R}^{r \times d_{\text{out}}}$). This design reduces the complexity from the standard MoE's $N(d_{\text{in}} + d_{\text{out}})r$ to $(d_{\text{in}} + Nd_{\text{out}})r$.

- **MTL-LoRA** creates task-specific experts, scaling the number of parameters linearly with the number of tasks K. This results in a significantly higher parameter count of approximately $KN(d_{\text{in}} + d_{\text{out}})r$, making it less scalable for scenarios with many tasks.

Computational Overhead. Our SAMoRA framework introduces minimal computational overhead. The **Semantic-Aware Router** requires a lightweight projection from d_{in} to the rank space r (where $r \ll min(d_{\text{in}}, d_{\text{out}})$), adding only $\mathcal{O}(Nr)$ operations. The **Task-Adaptive Scaling** mechanism introduces a lightweight parameter set of size Kd_g to capture task-specific characteristics. Since the scaling process primarily involves element-wise multiplications, the resulting computational overhead is negligible compared to the matrix multiplications in the backbone model.

B Experimental Setup

B.1 Datasets and Prompts

Following the experimental setup in (Yang et al., 2025), we summarize the statistics for the Commonsense Reasoning and GLUE benchmarks in Table 4 and 5, respectively. The corresponding prompt templates used are detailed in Table 7.

B.2 Implementation Details

We implement all methods using the PyTorch framework. Detailed hyperparameter configurations for our proposed SAMoRA and all baseline methods are summarized in Table 8.

Method	# Learnable Parameters	Computational Complexity
LoRA	$(d_{\text{in}} + d_{\text{out}})r$	$\mathcal{O}((d_{\text{in}} + d_{\text{out}})r)$
MoELoRA	$N(d_{in} + d_{out})r + Kd_g + Nd_g$	$\mathcal{O}(N(d_{in} + d_{out})r + d_g)$
MTL-LoRA	$(d_{\text{in}} + Nd_{\text{out}})r + Kr$	$\mathcal{O}(r(1 + d_{\text{in}} + Nd_{\text{out}}))$
HydraLoRA	$(d_{\text{in}} + Nd_{\text{out}})r + Nd_{\text{in}}$	$\mathcal{O}(d_{\text{in}} + (d_{\text{in}} + Nd_{\text{out}})r)$
SAMoRA (Ours)	$(d_{\text{in}} + N + Nd_{\text{out}})r + Kd_g + d_g$	$\mathcal{O}((d_{\text{in}} + N + Nd_{\text{out}})r + d_g)$

Table 3: Comparison of learnable parameters and computational complexity. Notations: $d_{\text{in}}/d_{\text{out}}$ are input/output dimensions, r is the rank, N is the expert number, K is the task number. d_g denote task embedding sizes for MoELoRA and SAMoRA. SAMoRA achieves a superior trade-off by combining asymmetric experts with efficient routing.

Corpus	#Train	#Val.	Metrics
BoolQ	9427	3270	Accuarcy
PIQA	16100	1840	Accuarcy
SocialIQA	33410	1954	Accuarcy
WinoGrande	9248	1267	Accuarcy
ARC-Challenge	1119	1172	Accuarcy
ARC-Easy	2250	2380	Accuarcy
OpenBookQA	4957	500	Accuarcy
HellaSwag	39905	10042	Accuarcy
CommonsenseQA	9741	1140	Accuarcy

Table 4: The basic information of Commonsense Reasoning Dataset

Corpus	#Train	#Validation	Metrics
CoLA	8,551	1,043	Matthew's Corr.
MNLI	392,702	9,815	Accuracy
MRPC	3,668	408	Accuracy
QNLI	104,743	5,463	Accuracy
QQP	363,846	40,430	Accuracy
RTE	2,490	277	Accuracy
SST-2	67,349	872	Accuracy

Table 5: The basic information of GLUE Benchmark

C Extended Analyses and Ablation Studies

C.1 Setup of Ablation Variants

To rigorously evaluate the contribution of each component in SAMoRA, we conduct ablation studies using the Qwen3-8B model on the GLUE benchmark. The specific configurations of the ablated variants are defined as follows:

- **w/o Router**: We replace our proposed Semantic-Aware Router with a standard MLP-based gating network. As analyzed in Section A, this substitution leads to an increase in trainable parameters due to the dense connections in the MLP layers.
- **w/o Scaling**: We disable the dynamic scaling mechanisms to verify their impact on task adaptation. Specifically, we fix all elements of the Diagonal Scaling Matrix S to 1.0 and set the task-dependent scalar g_{task} to 1.0 throughout the training process. Under this setting, the scaling strategy effectively reverts to the standard LoRA formulation.

C.2 Analysis of Semantic-Aware Router

To strictly isolate the efficacy of our routing mechanism and eliminate interference from other components, we conduct a controlled experiment based on the asymmetric MoE-LoRA architecture (featuring one shared matrix A and multiple semantic experts B). In this setup, we vary only the routing module (comparing our Semantic-Aware Router against a standard MLP router) while keeping all other structures identical. To make the expert specialization patterns more observable, we scale the number of experts to $N = 8$ and employ Llama-3.1-8B as the backbone, training on the Commonsense Reasoning benchmark.

Expert Representation Analysis. As illustrated in Figure 3, we visualize the Principal Component Analysis (PCA) projection of the learned expert features. The visualization reveals a stark contrast in the latent structure of the experts. With the MLP-based router, the expert representations tend to cluster closely together with ambiguous boundaries, indicating a high degree of functional

overlap. In contrast, our SAMoRA framework produces highly distinct and separated expert clusters. This explicitly demonstrates that our approach successfully enforces expert distinctiveness, allowing each expert to specialize in different semantic subspaces.

Routing Behavior on Unseen Tasks. To further evaluate the generalization capability of the router, we extend our analysis to the MMLU benchmark (Wang et al., 2024), which serves as an unseen task during training. We visualize the proportion of activations for each expert in Figure 5. Here, we display a subset of six randomly selected subjects characterized by diverse semantic distributions. The figure is organized by subject columns, with the top row representing the MLP router and the bottom row representing ours.

A critical observation from the top row is that the MLP router suffers from severe representation collapse: regardless of the input subject, it predominantly selects **Expert 5**, with other experts being rarely activated. This behavior suggests that the MLP router fails to align expert specialization with input semantics, causing the dynamic MoE architecture to effectively degrade into a static, non-MoE model. Conversely, our method (bottom row) exhibits diverse and balanced activation patterns adaptive to different subjects, validating its ability to maintain precise routing even on out-of-distribution data.

Theoretical Rationale for Semantic Match Regularization. In our architecture, learnable Expert Keys represent the intended specialties. However, during unconstrained joint optimization, these keys risk diverging from the actual parameters the experts learn. We employ KL Divergence to penalize this structural misalignment. KL Divergence is theoretically optimal here because it rigorously measures the relative entropy between two probability distributions. By forcing the expert key's assignment distribution to closely track the intrinsic capability distribution (derived from expert weights), we steer the optimization trajectory toward a state where routing decisions are strictly anchored in the experts' genuine functional capabilities, rather than arbitrary local minima.

Theoretical Basis for Matrix B Row Averaging.

- **Isolating Expert Knowledge:** In our asymmetric LoRA structure ($\Delta W = \sum_{i=1}^{N} g_i B_i A$), the shared down-projection A acts as a universal feature extractor, leaving matrix B_i strictly responsible for the specialized mapping back to the output space. Consequently, B_i inherently encodes the unique capabilities of that specific expert.
- **Geometric Centroid as Capability Anchor:** The rows of B_i are transformation vectors residing in the r-dimensional latent space—the exact space where our routing occurs. By computing the average of these rows, we calculate the geometric centroid of the expert's parameter subspace. Theoretically, this centroid represents the dominant, macro-level semantic direction of the expert's transformations.
- **Robust and Dimensional Alignment:** This aggregation smooths out localized parameter noise, yielding a highly stable global representation in $\mathbb{R}^r$. This perfectly aligns with the dimensionality of the Expert Keys, allowing for a mathematically sound distance computation and ensuring the alignment loss is both meaningful and computationally efficient.

D Hyperparameter Sensitivity

We conduct a comprehensive sensitivity analysis to evaluate the robustness of our proposed framework under various hyperparameter configurations. All experiments in this section are performed on the Commonsense Reasoning benchmark using Llama-3.1-8B as the backbone model, trained for 1 epoch.

D.1 Hyperparameter Sensitivity Analysis

We conduct a comprehensive sensitivity analysis to investigate how different hyperparameter configurations affect SAMoRA's performance.

Impact of Model Architecture (N, r, d_g). We first evaluate the impact of model capacity (N, r) and the task embedding dimension (d_g) in Figure 6(b).

- **Robustness to Capacity (N, r):** The performance remains relatively stable across a broad range of expert counts N and LoRA ranks r. Specifically, increasing r from 8 to 64 yields marginal gains, confirming that our method is parameter-efficient and does not rely on high-rank adapters. Similarly, the robustness against N indicates that our routing mechanism effectively utilizes available experts without suffering from redundancy.

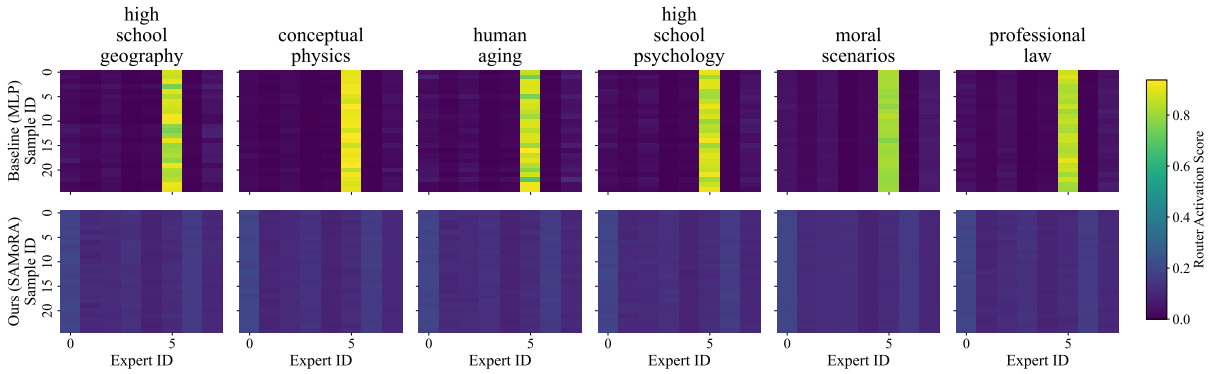

Figure 5: Visualization of expert activation patterns on the unseen MMLU benchmark. The top row (MLP Router) exhibits severe mode collapse, while our SAMoRA (bottom row) maintains diverse and adaptive routing across different subjects.

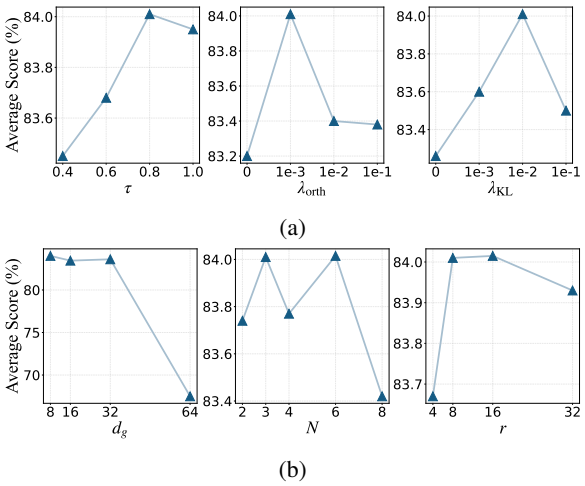

Figure 6: Sensitivity Analysis on hyperparameters evaluated on the Commonsense Reasoning dataset. Subfigure (a) and (b) illustrate different ablation settings.

- **Task Embedding Dimension (d_g):** We observe a distinct behavior regarding d_g. While the model performs well with compact dimensions, there is a sharp accuracy drop when d_g is increased to 64. This suggests that overly large task embeddings may introduce excessive parameters relative to the supervision signal, hindering convergence. Thus, a compact d_g is sufficient for effective semantic encoding.

Impact of Optimization Hyperparameters ($\tau, \lambda_{\text{orth}}, \lambda_{\text{KL}}$). We further analyze the regularization terms and routing temperature. Figure 6(a) illustrates the individual sensitivity trends for the temperature τ, orthogonality loss weight λ_{orth}, and KL divergence weight λ_{KL}. We observe that moderate values generally facilitate better convergence, preventing the router from collapsing or becoming too uniform.

To identify the optimal interaction between these terms, we report the joint ablation results in Table 6.

- **Temperature (τ):** The temperature controls the sharpness of the routing distribution. We find that $\tau = 0.8$ achieves the optimal performance (84.01%). Lower temperatures (e.g., $\tau = 0.4$) lead to premature expert collapse (83.45%), while higher temperatures (e.g., $\tau = 1.0$) result in an overly smooth distribution (83.95%).

- **Regularization Weights:** Combined with $\tau = 0.8$, appropriate regularization weights (λ_{orth} and λ_{KL}) are essential to balance expert specialization and load distribution, securing the best trade-off between plasticity and stability.

D.2 Impact of Loss Weights

Finally, we analyze the sensitivity of the regularization hyperparameters: the orthogonality weight λ_{orth} and the semantic match divergence weight λ_{KL}.

Orthogonality Weight (λ_{orth}). This term encourages diversity among experts. Comparing the rows in Table 6:

- Removing the regularization ($\lambda_{\text{orth}} = 0$) results in a performance drop to 83.20%, confirming the necessity of promoting expert diversity.

- However, setting λ_{orth} too high (1E-2) causes a significant performance degradation to 79.35%. This suggests that excessive constraints on orthogonality might hinder the optimization of the primary task loss.

- A moderate value of **1E-3** proves to be the most effective, striking a balance between expert diversity and task adaptation.

Semantic Match Weight ($\lambda_{\mathbf{KL}}$). This term aligns the routing decisions with semantic information. The results show a positive correlation between λ_{KL} and model performance within the tested range. Increasing λ_{KL} from 0 to 1E-2 consistently improves accuracy (from 83.26% to 84.01%), highlighting the benefit of guiding the router with semantic knowledge derived from task embeddings.

λ_{orth}	λ_{KL}	τ	**Avg.**
1E-3	1E-2	0.4	83.45
1E-3	1E-2	0.6	83.68
1E-3	1E-2	0.8	**84.01**
1E-3	1E-2	1	<u>83.95</u>
1E-3	1E-3	0.8	83.61
1E-3	0	0.8	83.26
1E-2	1E-2	0.8	79.35
0	1E-2	0.8	83.20

Table 6: Sensitivity Analysis (%) of regularization weights and temperature on Commonsense Reasoning dataset (Backbone: Llama-3.1-8B).

Task	Prompt Template
CoLA	Is the following sentence "{sentence}" grammatically acceptable? Answer:
SST-2	Is the following sentence "{sentence}" sentimently positive? Answer:
MRPC	Does the following sentence "{sentence1}" convey the equivalent meaning as "{sentence2}"? Answer:
QQP	Is the following question "{question1}" essentially asking the same thing as "{question2}"? Answer:
MNLI	Does the statement "{premise}" imply that "{hypothesis}"? Answer:
QNLI	Based on the statement: "{question}" does the following sentence "{sentence}" have a definitive answer? Answer:
RTE	Does the text "{sentence1}" entail the statement "{sentence2}"? Answer:

Table 7: Prompt templates used for the Natural Language Understanding benchmark (GLUE). The placeholders (e.g., {sentence}) represent the input fields from the dataset.

Hyperparameter	**LoRA**	**MultiLoRA**	**MoELoRA**	**HydraLoRA**	**MTL-LoRA**	**MoORE**	**SAMoRA**
Global Training Configurations							
Optimizer	AdamW						
Weight Decay	0						
β_1	0.9						
β_2	0.95						
Learning Rate	2×10^{-4}/ 3×10^{-4}						
Batch Size	8 / 64						
Training Epochs	3						
Warmup Ratio	0.01						
Max Sequence Length	512						
Target Modules	Q,K,V,O						
Method-Specific Architectures							
Rank (r)	16	8	8	8	8	8	8
Scale (α)	32	16	16	16	16	16	-
num_experts (N)	-	3	8	3	3	3	3
Task Embedding Size (d_g)	-	-	64	-	-	64	8
Temperature (τ)	-	-	-	-	0.8	-	0.8
λ_{orth}	-	-	-	-	-	-	1e-3
λ_{match}	-	-	-	-	-	-	1e-2

Table 8: Detailed hyperparameter settings for all baseline methods on Commonsense Reasoning and GLUE benchmark. Common settings are listed in the top section, while method-specific parameters are detailed below. "-" indicates the parameter is not applicable.