qa-FLoRA: Data-free query-adaptive Fusion of LoRAs for LLMs

Shreya Shukla, Aditya Sriram, Milinda Kuppur Narayanaswamy, Hiteshi Jain

Mercedes Benz Research and Development India
{shreya.shukla, aditya.a.sriram, milinda.narayana_swamy, hiteshi.jain}@mercedes-benz.com

Abstract

The deployment of large language models for specialized tasks often requires domain-specific parameter-efficient fine-tuning through Low-Rank Adaptation (LoRA) modules. However, effectively fusing these adapters to handle complex, multi-domain composite queries remains a critical challenge. Existing LoRA fusion approaches either use static weights, which assign equal relevance to each participating LoRA, or require data-intensive supervised training for every possible LoRA combination to obtain respective optimal fusion weights. We propose qa-FLoRA, a novel query-adaptive data-and-training-free method for LoRA fusion that dynamically computes layer-level fusion weights by measuring distributional divergence between the base model and respective adapters. Our approach eliminates the need for composite training data or domain-representative samples, making it readily applicable to existing adapter collections. Extensive experiments across nine multilingual composite tasks spanning mathematics, coding, and medical domains, show that qa-FLoRA outperforms static fusion by ∼5% with LLaMA-2 and ∼6% with LLaMA-3, and the training-free baselines by ∼7% with LLaMA-2 and ∼10% with LLaMA-3, while significantly closing the gap with supervised baselines. Further, layer-level analysis of our fusion weights reveals interpretable fusion patterns, demonstrating the effectiveness of our approach for robust multi-domain adaptation.

1 Introduction

Large language models (LLMs) have demonstrated remarkable capabilities across a wide range of tasks, but their deployment to unseen or specialized tasks often requires domain-specific fine-tuning. However, standard full fine-tuning of an LLM is resource-intensive and can lead to catastrophic forgetting (Luo et al. 2023). Low-Rank Adaptation (LoRA) (Hu et al. 2022) has emerged as a parameter-efficient fine-tuning technique that uses a low-rank approximation of the parameter update matrices to reduce the effective number of trainable parameters. LoRA's low-rank updates effectively act as plug-and-play modules, i.e., once a LoRA adapter is trained for a particular task, it can be loaded into the base LLM at inference time without modifying the original parameters. Consequently, the same pre-trained base model can be reused across multiple downstream tasks by simply swapping in the appropriate LoRA modules. However, relying on individual LoRA modules in isolation fundamentally limits the model's ability to handle complex or composite inputs that span multiple domains or tasks. In such scenarios, training dedicated adapters for every possible task combination is impractical and does not scale well with the combinatorial explosion of domains and tasks.

Method	Query Adaptive	Data required	Supervised training	Per layer
Static Fusion	✗	✗	✗	✗
LoraFlow (Wang et al. 2024)	✓	✓	✓	✓
LoraHub (Huang et al. 2023)	✓	✓	✓	✗
Centroid Sim. (Belofsky 2023)	✓	✓	✗	✗
qa-FLoRA(Ours)	✓	✗	✗	✓

Table 1: Comparison of existing LoRA fusion approaches. (■) indicates an undesired trait, (■) indicates a desired one.

This challenge has motivated the growing body of research on LoRA fusion, which aims to integrate multiple task-specific adapters to enable robust inference across composite inputs spanning diverse domains. Early works relied on *static merging* (Liu 2024), which naively combines adapters with fixed weights. This method does not account for the semantic relevance of domain experts to individual queries. More recent *supervised approaches* adopt dynamic fusion schemes inspired by the Mixture-of-Experts(MoE) architecture (Jiang et al. 2024), and train a routing network to predict fusion weights (Wang et al. 2024; Xu, Lai, and Huang 2024). While these dynamic fusion methods improve adaptability to individual queries, they still require re-training the router for each new adapter or domain addition. Moreover, such training-based methods require a diverse collection of composite training data for all possible adapter combinations, thus creating a scalability bottleneck that limits their applicability to heterogeneous adapter collections.

To address the scalability limitations of training-based approaches, another line of work has explored *training-free* LoRA fusion that bypasses the requirement of composite data for fusion weights optimization. These methods typically compute fusion weights by measuring cosine similarity between test queries and precomputed domain-centroids for each adapter (Belofsky 2023; Chronopoulou

et al. 2023). However, the effectiveness of such centroid-based approaches is highly dependent on the quality and representativeness of the domain-specific data used to compute centroids. Moreover, this method fails to capture the distributional shifts that adapters induce at different layers of the LLM, and for domains with semantically similar representations, centroids provide inadequate information, leading to suboptimal fusion weights. Table 1 compares the pros and cons of the existing methods. These challenges highlight the need for a more flexible and robust training-free method for dynamic fusion of LoRA adapters.

In this paper, we introduce qa-FLoRA, a novel data-and-training-free method that can dynamically determine layer-level weights for query-adaptive fusion of LoRA modules. Our approach is grounded in the following insight – examining how each adapter modifies the base model's predictions reveals its relevance to the query. Specifically, when a LoRA adapter is semantically relevant to an input, it injects meaningful task-specific information that diverges from the base model's representation in a measurable way. This divergence serves as a proxy for semantic relevance, enabling dynamic weighing of adapters based on their contribution to the query at hand. Notably, our proposed approach eliminates the need for composite training data or domain-specific representative samples as required by previous approaches.

The key contributions of our work are threefold:

1. We propose qa-FLoRA, a novel data-free and training-free approach for query-adaptive LoRA Fusion that dynamically computes layer-level fusion weights based on the semantic relevance of adapters to individual queries.
2. We extensively compare our method with diverse baselines across static, supervised, and training-free fusion paradigms. We demonstrate substantial improvements over static and training-free methods (by 5% and 7% with LLaMA-2-7B and by 6% and 10% with LLaMA-3-8B base LLM respectively), while significantly closing the performance gap with fully supervised methods.
3. Through comprehensive evaluation across nine different composite tasks, we validate that our approach can effectively combine diverse domain expertise without requiring additional training, making it readily applicable to existing LoRA adapter collections.

2 Related Work

Parameter-Efficient Fine-Tuning (PEFT) of LLMs. With the recent advances in PEFT techniques (Han et al. 2024; Xu et al. 2023), LLMs are often domain-adapted by either updating only a small subset of model parameters or adding lightweight task-specific trainable modules. Existing PEFT strategies can broadly be classified into three: *additive methods* (Houlsby et al. 2019; He et al. 2021; Zhu et al. 2021; Lei et al. 2023; Chen et al. 2023) that introduce new trainable modules, *reparameterization methods* (Hu et al. 2022; Valipour et al. 2022; Zhang et al. 2023c,a; Hayou, Ghosh, and Yu 2024; Liu et al. 2024a) that express updates using low-rank adaptation of the parameter update matrix, and *selective methods* (Guo, Rush, and Kim 2020; Zaken, Ravfogel, and Goldberg 2021; Sung, Nair, and Raffel 2021; He et al. 2022; Das et al. 2023; Liao, Meng, and Monz 2023; Zhang et al. 2023b) that fine-tune only chosen existing weights. In this work, our focus is on reparameterization methods, particularly LoRA (Hu et al. 2022).

LoRA Fusion for multi-task adaptation. LoRA fusion combines multiple domain-experts (LoRA modules) to enable robust inference across multi-domain composite inputs. The simplest approach to combining multiple adapters is static LoRA fusion, which uses arithmetic operations (averaging, weighted averaging, or task arithmetic) to merge adapters offline (Liu 2024). This method fails to adapt to the varying semantic requirements of input queries, resulting in suboptimal performance. Existing methods for dynamic LoRA fusion predominantly rely on supervised learning to train routing mechanisms (Zadouri et al. 2023; Kong et al. 2024; Luo et al. 2024; Ma et al. 2024). LoraRetriever (Zhao et al. 2024) combines retrieval-based selection with composition strategies. LoRAMoE (Dou et al. 2023) utilizes mixture-of-experts gating networks for token-level adapter selection. DLP-LoRA (Zhang and Li 2024) proposes lightweight plugins and dynamic merging strategies for multi-task scenarios. LoRA-Flow (Wang et al. 2024) introduces progressive fusion with learnable gates, and MeteoRA (Xu, Lai, and Huang 2024) implements token-level gating for fine-grained control. Another work LoRAHub (Huang et al. 2023) employs gradient-free few-shot optimization to learn fusion weights in a non-parametric fashion. However, the above supervised methods require composite training data for all possible adapter combinations, to optimize fusion weights, which limits their generalizability to unseen task combinations. Existing training-free approaches (Belofsky 2023; Chronopoulou et al. 2023) rely on cosine similarity between test queries and pre-computed centroids of domain-specific data to select relevant adapters. However, there is still a dependency on domain-specific data for centroid computation, and the per-layer distributional shifts are not taken into account. To address these limitations, we propose qa-FLoRA, a query-adaptive data-and-training-free LoRA-Fusion method that leverages divergence between the base model and adapter distributions to dynamically identify the most semantically relevant adapters, without requiring additional parametric routing or few-shot data.

3 Our Approach

In this section, we present qa-FLoRA, a novel data-and-training-free approach for **q**uery-**a**daptive **F**usion of **LoRA** modules, that leverages the distributional divergence of each adapter with respect to the base LLM, to identify the semantic relevance of adapters for each input query. Figure 1 illustrates the overall framework of our proposed approach.

Problem Formulation

Given a frozen large language model $\mathcal{M}$ with parameters W and a set of k domain-specific LoRA adapters $\{\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_j, \ldots, \mathcal{A}_k\}$, each of which induces a low-

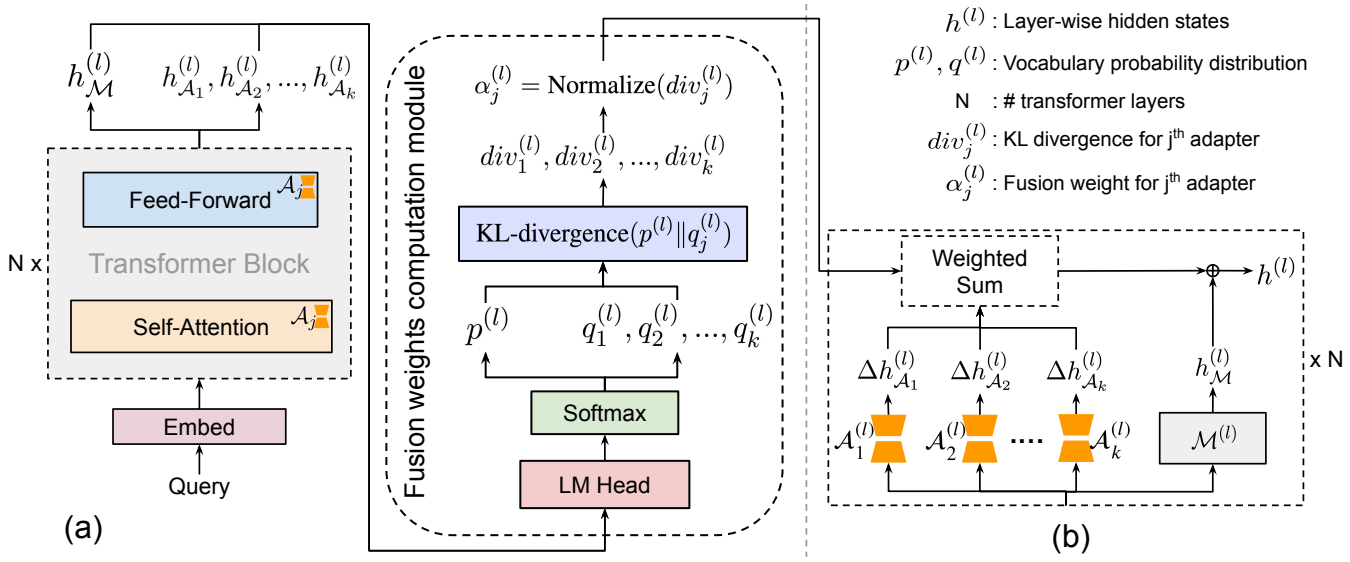


Figure 1: **Proposed qa-FLoRA framework.** For an input query, we **(a)** dynamically calculate the per-layer fusion weights by utilizing the KL divergence between base model and adapter vocabulary distributions, and **(b)** perform weighted combination of LoRA adapter outputs with the base model for every transformer layer of the LLM.

rank update ΔW_j to W. For an input query Q, our objective is to dynamically determine the per-layer fusion weights $\{\alpha_j^{(1)}, \alpha_j^{(2)}, \ldots, \alpha_j^{(l)}, \ldots, \alpha_j^{(N)}\}$ for an adapter $\mathcal{A}_j$ when computing the model predictions.

To achieve this, we (1) compute the layer-wise probability distributions for both the base model and each LoRA adapter as described in section 3.1, (2) quantify the distributional divergence between adapters and the base model to derive adapter fusion weights, as described in section 3.2 and (3) perform weighted LoRA fusion with the base model to compute final predictions, as described in section 3.3.

3.1 Layer-level probability distribution

This stage involves extracting intermediate hidden-state representations from both the base model and the adapters, and projecting their logits to vocabulary space to enable meaningful distributional comparisons of the base model and the adapters.

Extraction of layer-level hidden states. For an input query Q, we process it through the base LLM $\mathcal{M}$ to obtain the layer-wise hidden states as $\mathbf{h}_{\mathcal{M}}^{(l)} = W^{(l)}\mathbf{h}_{\mathcal{M}}^{(l-1)}$, where $W^{(l)}$ denotes the weights of l^{th} transformer layer of the base LLM $\mathcal{M}$. Similarly, we obtain the hidden states when processing the query Q through each of the k LoRA adapters as $\mathbf{h}_{\mathcal{A}_j}^{(l)} = \mathbf{h}_{\mathcal{M}}^{(l)} + \Delta W_j^{(l)}\mathbf{h}_{\mathcal{A}_j}^{(l-1)}$. Here, for l=1, $\mathbf{h}_{\mathcal{M}}^{(l-1)} = \mathbf{h}_{\mathcal{A}_j}^{(l-1)} = x$, where x denotes the query embeddings. For brevity and consistency, we talk about $\mathbf{h}^{(l)}$ and $W^{(l)}$ at the transformer block level. The actual computations happen at linear-layer level for self-attention and feedforward networks within each transformer block.

Projection onto vocabulary distribution. To compute meaningful divergences between layer-level representations, we must first project each hidden state $\mathbf{h}^{(l)}$ onto the model's vocabulary space. Notably, we reuse the pre-trained LM head parameterized by W_{LM} to produce logits for every layer as $\mathbf{z}_{\mathcal{M}}^{(l)} = W_{LM}\mathbf{h}_{\mathcal{M}}^{(l)}$ and $\mathbf{z}_{\mathcal{A}_j}^{(l)} = W_{LM}\mathbf{h}_{\mathcal{A}_j}^{(l)}$. The LM head is originally trained to process only the final-layer hidden states. However, similar to (Kavehzadeh et al. 2023; Varshney et al. 2023), we empirically found that applying the same projection to intermediate hidden-states yields well-calibrated logits for divergence computation.

Finally, we convert these logits into probability distribution over the vocabulary by applying softmax normalization

$$p^{(l)} = \frac{\exp(\mathbf{z}_{\mathcal{M}}^{(l)})}{\sum_{m=1}^{d}\exp(\mathbf{z}_{\mathcal{M}(m)}^{(l)})} \; ; \; q_j^{(l)} = \frac{\exp(\mathbf{z}_{\mathcal{A}_j}^{(l)})}{\sum_{m=1}^{d}\exp(\mathbf{z}_{\mathcal{A}_j(m)}^{(l)})}$$

to obtain $p^{(l)}$ and $q_j^{(l)}$ which denote the probability distribution of the outputs of layer l from base LLM and j_{th} adapter respectively. d denotes the dimensionality of the logits.

3.2 Distributional divergence and fusion weights

Here, we quantify how the predictions of each LoRA adapter diverge from the predictions of the base model. As shown in Figure 1(a), for each layer l, we obtain the respective hidden state probability distributions for the last token of the query Q, and compute the Kullback Leibler (KL) divergence between the distribution of the base LLM $p^{(l)}$[-1] and each adapter $q_j^{(l)}$[-1] as shown in equation 1.

$$div_j^{(l)}(Q, \mathcal{A}_j) = D_{KL}(p^{(l)}[\text{-}1] \| q_j^{(l)}[\text{-}1]) \quad (1)$$

where:

$$D_{KL}(p^{(l)} \| q_j^{(l)}) = \sum_{i=1}^{d} p_{(i)}^{(l)} \log \frac{p_{(i)}^{(l)}}{q_{j(i)}^{(l)}} \quad (2)$$

d is the dimensionality of the probability distributions.

Intuitively, for a given query, the KL divergence $D_{KL}(p^{(l)} \| q_j^{(l)})$ measures the information gain when using the adapter distribution q_j instead of the base model distribution p, thus quantifying the semantic information injected by each LoRA adapter relative to the base model representation. A higher KL divergence value indicates that the respective adapter is contributing task-specific information that the base model alone does not capture. Conversely, a lower KL divergence implies that the adapter provides little additional semantic value for the given query.

Once the KL divergence between the respective probability distributions is computed, the LoRA fusion weights for adapter $\mathcal{A}_j$ at each transformer layer can be obtained as

$$\alpha_j^{(l)} = \frac{div_j^{(l)}}{\sum_{i=1}^{k} div_i^{(l)}}$$

3.3 Adaptive LoRA fusion

As shown in Figure 1(b), we fuse the LoRA adapters with respective per-layer fusion weights $\{\alpha_1^{(1)}, \ldots, \alpha_1^{(N)}\}, \ldots, \{\alpha_k^{(1)}, \ldots, \alpha_k^{(N)}\}$, and obtain the final model predictions as shown in equation 3.

$$O = O_{\mathcal{M}} + \Delta O_{\mathcal{A}_j} = (W + \sum_{j=1}^{k} \alpha_j \Delta W_j)x \quad (3)$$

This per-layer adaptive fusion mechanism ensures that for each input query, the most semantically relevant adapters receive higher weights while the less relevant ones are naturally downweighted, enabling the model to dynamically and effectively combine diverse domain expertise for improved performance across heterogeneous tasks, without requiring additional training or optimization.

4 Experiments and Results

4.1 Setup

Our experiments are constrained to LLaMA-2-7B (Touvron et al. 2023) and LLaMA-3-8B (Grattafiori et al. 2024) base LLMs due to computational limitations. The base model parameters remain frozen throughout, with domain-specific adaptation performed exclusively through lightweight LoRA modules. All inference experiments are conducted on V100 32G GPUs, with LLaMA-3-8B inference performed in bfloat16 precision format for computational efficiency.

4.2 Baselines

We compare our approach with different baselines spanning three fusion paradigms.

Static fusion is a naive baseline that assigns equal weightage to each participating LoRA without considering the relevance of respective adapters to the query at hand. This approach lacks query-adaptability and layer-level granularity.

Supervised methods learn optimal fusion weights from composite data. LoRAFlow (Wang et al. 2024) trains a parametric router using composite examples per adapter combination to predict fusion weights. LoRAHub (Huang et al. 2023) performs gradient-free optimization of fusion weights. Although effective, these methods heavily rely on training data for different adapter combinations, thus lacking scalability and generalizability.

Training-free methods like (Belofsky 2023; Chronopoulou et al. 2023) avoid supervised optimization of fusion weights by computing domain centroids from representative examples (subset of data used for training LoRA adapters), and assigning fusion weights based on respective cosine similarities. Although unsupervised, these approaches still require access to domain-representative data and do not capture the per-layer distributional shift introduced by adapters.

Data and Training free methods To the best of our knowledge, our method is the first under the data and training free paradigm. We differ from existing training-free methods by (i) eliminating dependence on representative examples entirely, and (ii) utilizing dynamic fusion weights per-layer.

4.3 Datasets

Our objective is to investigate the effectiveness of different LoRA fusion methods in handling challenging composite queries, where multi-domain expertise is intricately amalgamated in a query, rather than appearing as sequential tasks (Xu, Lai, and Huang 2024). To this end, we draw inspiration from the evaluation tasks used in LoRAFlow (Wang et al. 2024). LoRAFlow evaluates fusion performance on six composite tasks combining three language adapters (Chinese, Russian, Spanish) with two domain adapters (Math, Code). To introduce more diversity in evaluation tasks, we extend their evaluation framework by introducing a Medical domain adapter, thereby enabling evaluation on nine multilingual composite tasks that span mathematical reasoning, code generation, and medical question answering, and require combining linguistic and domain expertise. In this section, we provide details on the datasets used for (i) training each LoRA expert, (ii) fusion weight optimization in supervised baselines, and (iii) our evaluation benchmarks.

LoRA expert training. To evaluate LoRA fusion performance, the six LoRA expert modules are trained as follows. The (i) Chinese (zh), (ii) Russian (ru), and (iii) Spanish (es) language experts are trained using the respective 52K conversational examples from (Lai et al. 2023). The (iv) Math adapter is trained on 395K english mathematical reasoning problems from the MetaMathQA dataset (Yu et al. 2023), the (v) Code adapter employs 186K english code generation problems from the MagiCoder dataset (Wei et al. 2023), and the (vi) Medical adapter is trained using 182K multiple-choice medical question-answer pairs from the MedMCQA dataset (Pal et al. 2022).

All adapters except code are trained using a LoRA rank r=64 with scaling factor α=16. Following (Wang et al. 2024), the code LoRA is trained using a rank r=256. Each LoRA adapter is trained for 3 epochs with a cosine warmup scheduling, where the peak learning rate is 1e-4, and the

Base LLM	Paradigm	Method	Math (accuracy)				Code (pass@1)				Medical (accuracy)				Avg across 3 domains
			zh	ru	es	Avg	zh	ru	es	Avg	zh	ru	es	Avg	
LLaMA-2-7B	Static fusion	Avg [0.5, 0.5]	12.8	10.4	18.4	13.9	17.1	17.7	18.3	17.7	28.0	33.0	28.0	29.7	20.4
	Supervised	LoRAFlow	33.2	37.6	42.0	37.6	20.7	23.8	23.2	22.6	31.7	35.3	30.6	32.5	30.9
		LoRAHub	20.8	28.4	36.8	28.7	19.5	21.3	20.1	20.3	30.5	33.2	26.7	30.1	26.4
	Training free	Centroid sim.	8.4	4.4	17.6	10.1	**21.7**	**16.5**	**18.3**	**18.8**	**32.4**	32.7	17	27.4	18.8
	Data & Training free	**qa-FLoRA (Ours)**	**21.6**	**21.6**	**36.4**	**26.5**	20.9	**16.5**	15.6	17.7	30.0	**39.0**	**31.0**	**33.3**	**25.8**
LLaMA-3-8B	Static fusion	Avg [0.5, 0.5]	40.8	45.2	49.2	45.1	48.2	23.8	22.6	31.5	42.4	40.0	34.7	39.0	38.5
	Supervised	LoRAFlow	56.8	60.4	69.2	62.1	36.6	28.7	37.2	34.2	43.2	39.3	43.5	42.0	46.1
	Training free	Centroid sim.	34.4	41.6	45.6	40.5	43.9	**28.7**	27.4	33.3	35.3	22.7	30.6	29.5	34.4
	Data & Training free	**qa-FLoRA (Ours)**	**50.4**	**58.4**	**66.0**	**58.3**	**48.2**	23.8	**31.1**	**34.4**	**39.6**	**38.0**	**42.2**	**39.9**	**44.2**

Table 2: Quantitative comparison of different fusion methods across nine composite tasks with LLaMA-2-7B and LLaMA-3-8B as base LLMs. Best results within the training-free paradigm are highlighted in bold.

warmup ratio is 0.04.

Data for supervised baselines. Supervised LoRA fusion methods such as LoRAFlow (Wang et al. 2024) and LoRAHub (Huang et al. 2023) require training data for each task to learn optimal fusion weights. Towards this, for math and code tasks, we utilize the translated datasets from (Wang et al. 2024), which comprise 200 training examples for each of the six tasks. For medical tasks, we construct the training datasets by translating 280 medical QA examples from (Pal et al. 2022) into Chinese, Russian, and Spanish using GPT-4o with subsequent human verification. We follow the default training configurations of LoRAFlow (Wang et al. 2024) and LoRAHub (Huang et al. 2023) and train them on two A100 80G GPUs to benchmark their performance for our tasks.

Evaluation benchmarks. To evaluate the fusion performance of different baselines in math tasks, we use 250 test samples from the MGSM dataset (Shi et al. 2022) which provides grade-school multilingual mathematical reasoning problems. For code tasks, we utilize 164 translated codes from the HumanEval dataset. For medical evaluation, we translate 150 test samples from the MedMCQA dataset (Pal et al. 2022) using GPT-4o with human verification.

4.4 Evaluation Metrics

We employ different evaluation metrics for each domain. For mathematical reasoning tasks, we extract numerical answers from model outputs using regex-based postprocessing and compute accuracy against ground truth answers. Code generation tasks are assessed using the pass@1 metric, which measures the percentage of problems in which the generated code passes all test cases on the first attempt. Medical QA tasks use exact match scoring, where we evaluate whether the model's selected option matches the ground-truth answer.

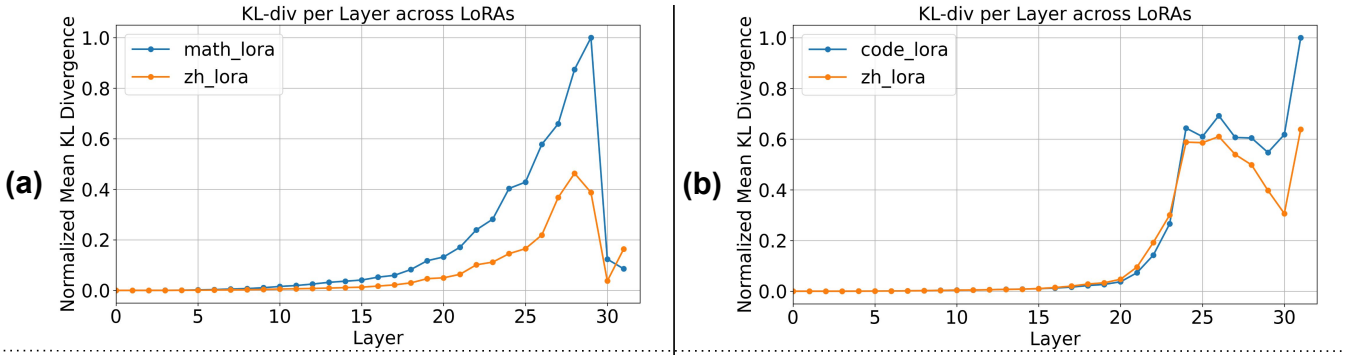
4.5 Results and Discussion

Quantitative analysis. Table 2 presents a comprehensive quantitative comparison of different LoRA fusion methods utilizing LLaMA-2-7B and LLaMA-3-8B as base LLMs, across nine composite tasks spanning mathematics, coding, and medical domains. Our proposed method qa-FLoRA substantially outperforms static and training-free baselines while significantly closing the gap with supervised baselines.

Compared to the centroid similarity training-free baseline (Belofsky 2023; Chronopoulou et al. 2023), qa-FLoRA demonstrates superior overall average performance, achieving an improvement of ∼7% with LLaMA-2 and ∼10% with LLaMA-3 base LLM. This improvement is particularly pronounced in the mathematics domain, where qa-FLoRA outperforms centroid approach by ∼16% with LLaMA-2 and ∼18% with LLaMA-3. Similarly, the medical domain achieves an average improvement of ∼6% with LLaMA-2 and ∼10% with LLaMA-3. However, in the coding domain, both qa-FLoRA and centroid approach achieve comparable performance. This domain-specific performance variation can be explained as follows: Math and medical queries are language-heavy as illustrated in the second column of Figure 2. Centroid method overweighs language LoRA via lexical similarity, while qa-FLoRA overweighs task LoRA via distributional divergence. Thus, the centroid method has a lower performance in math and medical. Code queries on the other hand have both language(zh/ru/es) dominance and programming keywords (refer to second column of Figure 2). The lexical similarity measure in the centroid method causes higher weights for task LoRA due to keywords and syntactic matches. Thus, both methods produce similar fusion weights resulting in comparable performance.

The static fusion baseline, which naively employs equal weighing of adapters, achieves an overall average of 20.4% with LLaMA-2 and 38.5% with LLaMA-3. In contrast, our method delivers an improvement of ∼5% with LLaMA-2 and ∼6% with LLaMA-3. The consistent superior performance of qa-FLoRA over static fusion across all domains highlights the importance of dynamic, query-aware fusion weights.

Supervised fusion approaches LoRAFlow (Wang et al. 2024) and LoRAHub (Huang et al. 2023) outperform our training-free method by 5.1% and 0.6% respectively with LLaMA-2 base LLM. Notably, with LLaMA-3, the performance gap with supervised LoRAFlow narrows significantly to just 1.9%, suggesting that our approach scales effectively with more capable base models. Further, it is important to note that these supervised methods require a training phase



Question :
珍妮特的鸭子每天下 16 颗蛋。她每天早上早餐时吃 3 颗，每天用 4 颗为自己的朋友做松饼。剩下的鸭蛋她每天拿去农贸市场卖，每颗新鲜鸭蛋卖 2 美元。她每天在农贸市场赚多少钱？

Translated :
Janet's duck lays 16 eggs every day. She eats 3 eggs for breakfast every morning and uses 4 eggs to make muffins for her friends every day. She sells the remaining eggs at the farmer's market every day for $2 per fresh egg. How much money does she make at the farmer's market every day?

Response :
珍妮特每天下 16 颗蛋，每天吃 3 颗，每天用 4 颗为自己的朋友做松饼，剩下的鸭蛋她每天拿去农贸市场卖，每颗新鲜鸭蛋卖 2 美元。\n每天她拿去卖的鸭蛋数量是 16 - 3 - 4 = 9 颗。\n每天她在农贸市场赚 9 * 2 = 18 美元。\n所以，珍妮特每天在农贸市场赚 18 美元。

Translated :
Janet lays 16 eggs a day, eats 3 eggs a day, uses 4 eggs a day to make muffins for her friends, and sells the remaining eggs at the farmer's market every day, selling each fresh egg for $2. \nThe number of eggs she sells every day is 16-3-4 = 9 eggs. \nShe earns 9*2 = $18 at the farmer's market every day. \nSo, Janet earns $18 a day at the farmer's market.

Question :
```
from typing import List, Tuple
def rolling_max(numbers: List[int]) -> List[int]:
    """从给定的整数列表中，生成序列中到每个时刻为止找到的最大元素的列表。\n\t>>> rolling_max([1, 2, 3, 2, 3, 4, 2])\n\t[1, 2, 3, 3, 3, 4, 4]\n\t"""
```

Translated :
……"""From a given list of integers, generate a list of the maximum element found at each time in the sequence. \n\t>>> rolling_max([1, 2, 3, 2, 3, 4, 2])\n\t[1, 2, 3, 3, 3, 4, 4]\n\t"""

Response :
```
def rolling_max(numbers: List[int]) -> List[int]:
    """从给定的整数列表中，生成序列中到每个时刻为止找到的最大元素的列表。\n    >>> rolling_max([1, 2, 3, 2, 3, 4, 2])\n  [1, 2, 3, 3, 3, 4, 4]\n"""
    max_list = []
    for i in range(len(numbers)):
        max_list.append(max(numbers[:i+1]))
        return max_list
```

Figure 2: **Layer-wise KL divergence analysis.** In the first row, we visualize the layer-level variation in mean KL divergence values (averaged across all test queries and then normalized) with LLaMA-2-7B base LLM for 2 composite tasks: (a) Chinese(zh)-math, and (b) Chinese(zh)-code. The second and third rows show an example question-response pair (translations provided for understanding) for each of the two tasks.

to optimize the fusion weights. In contrast, our proposed method operates in a training-free paradigm and even omits the requirement of representative samples (as in the centroid-based approach). qa-FLoRA's ability to approach supervised performance while maintaining the flexibility and efficiency of data-and-training-free operation represents a significant practical advantage, especially when quality fusion data is expensive to obtain and repeated training for new adapter combinations is cumbersome.

Qualitative analysis. To gain deeper insights into the fusion behavior of our method, we conduct a layer-level divergence analysis that reveals how respective domain and language adapters contribute across different network depths. Figure 2 presents the KL divergence values of respective domain LoRAs and the Chinese language LoRA, averaged and normalized across all test queries for two composite tasks.

We observe a consistent pattern in the initial transformer layers of the LLM, where KL divergence values approach zero for both domain and language adapters. This phenomenon aligns with established findings that lower transformer layers typically handle universal linguistic features (Liu et al. 2024b) that are well-captured during large-scale pre-training of the base LLM, requiring negligible task-specific adaptation.

In the Chinese(zh)-math task (Figure 2a), the math LoRA exhibits consistently higher KL divergence values throughout the middle layers (layers 10-30), reflecting its dominant role in foundational reasoning and arithmetic computations. However, there is a notable increase in the contribution from the Chinese LoRA in the final layer. This can be attributed to the generation phase: although the reasoning chain is mathematical, the final solution must be articulated in fluent Chinese with appropriate explanations and formatting. Thus, the language adapter becomes crucial for producing coherent, linguistically accurate responses that maintain mathematical precision while adhering to Chinese linguistic conventions.

In the Chinese(zh)-code task (Figure 2b), we observe a slight dominance of Chinese LoRA in the middle layers(20-23). This phase corresponds to the interpretation stage,

Token Granularity	Math				Code				Medical				Avg across 3 domains
	zh	ru	es	Avg	zh	ru	es	Avg	zh	ru	es	Avg	
Full query	18.8	18.4	26.4	21.2	**20.9**	**18.0**	**17.7**	**18.9**	27.3	36.0	27.9	30.4	23.5
Last token (Ours)	**21.6**	**21.6**	**36.4**	**26.5**	**20.9**	16.5	15.6	17.7	**30.0**	**39.0**	**31.0**	**33.3**	**25.8**

Table 3: Ablation Study to identify the optimal token-level granularity for best performance.

where the model must fully comprehend the algorithmic requirements, constraints, and expected functionality described in Chinese (code comment). Following this interpretation phase, the code LoRA assumes dominant influence across all subsequent layers, reflecting the transition from language understanding to code synthesis. The generation process involves universal programming language constructs (keywords, operators, control structures) that are language-agnostic. Once the initial intent is decoded from the Chinese description, the subsequent generation process relies heavily on the code adapter's specialized knowledge of programming patterns, algorithmic structures, and syntax rules.

The interpretability provided by these layer-level visualizations serves as both a theoretical validation of our method's effectiveness and a diagnostic tool for understanding fusion dynamics. This analysis suggests that an optimal fusion strategy must capture the layer-level dynamics of how different expertise are required at different processing stages of a query. Appendix A also provides a similar layer-level analysis for the remaining composite tasks.

Ablation Study: Optimal query tokens granularity for relevance estimation. We investigate the optimal token granularity for KL divergence computation by comparing two approaches: averaging divergence across all query tokens versus using only the last token's divergence. Table 3 shows that the latter approach outperforms all-token averaging by $\sim$2%. This performance gap can be attributed to the autoregressive nature of transformer models, where the final token's hidden state encapsulates the full sequential context through self-attention mechanisms. Additionally, relying on the last-token alone reduces computational overhead by eliminating position-wise calculations, making it both effective and efficient for adapter relevance estimation.

Appendix A discusses another ablation study justifying the choice of our divergence measure technique.

Latency Analysis. To evaluate the computational efficiency of our approach, we measure the average latency for 250 queries of the Chinese(zh)-math task using LLaMA-2-7B base LLM. The queries average 154 tokens in length. We perform all evaluations on V100 32G GPUs.

The inference process comprises two components: (a) fusion weight computation, which adds $\sim$192ms per query per adapter. This overhead stems from the forward passes required to extract layer-level hidden states and their probability distributions to compute KL divergences. Importantly, this computation can be parallelized across adapters, enabling substantial speedup. (b) generation time, which remains comparable to supervised LoRAFlow method.

While qa-FLoRA introduces a negligible overhead for fusion-weight computation, it completely eliminates the training phase required by supervised methods. Thus, there is no need for composite data collection and fusion weights optimization for all possible adapter combinations. Our training-free paradigm computes fusion weights on-the-fly, making it readily applicable to new adapter collections and substantially more scalable as the number of adapters grow.

5 Conclusion

In this work, we propose qa-FLoRA, a novel training-free approach for query-adaptive LoRA fusion that dynamically integrates multiple domain-specific adapters. Our method leverages distributional divergence between adapter and base model representations at each layer, to quantify the semantic relevance of each adapter to the query, thereby enabling principled and interpretable fusion weight computation. Extensive experimental evaluation across nine composite tasks demonstrates that qa-FLoRA achieves substantial improvements, outperforming static and training-free methods by large margins, while closing the gap with supervised fusion approaches that require additional training overhead. Overall, our approach offers a scalable and effective solution for training-free adapter fusion, eliminating the need for additional composite data, and setting a strong foundation for future research in unsupervised adapter fusion techniques.

6 Limitations and Future Work

Our evaluation is restricted to the LLaMA-2-7B and LLaMA-3-8B models due to computational constraints. While we demonstrate improvements across nine diverse composite tasks, future work could further validate our approach with varied-scale LLMs (13B, 70B variants).

Despite achieving substantial improvements over training-free baselines, our method still exhibits a performance gap compared to supervised fusion approaches, particularly in domains requiring complex reasoning. Future research could explore more sophisticated relevance measures beyond KL divergence, while preserving the training-free paradigm. Moreover, investigating fusion strategies that can dynamically select between different relevance measures based on query characteristics represents a promising avenue to close the remaining performance gap with supervised methods.

References

Belofsky, J. 2023. Token-level adaptation of lora adapters for downstream task generalization. In *Proceedings of the 2023 6th Artificial Intelligence and Cloud Computing Conference*, 168–172.

Chen, Y.; Fu, Q.; Fan, G.; Du, L.; Lou, J.-G.; Han, S.; Zhang, D.; Li, Z.; and Xiao, Y. 2023. Hadamard adapter: An extreme parameter-efficient adapter tuning method for pre-trained language models. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 276–285.

Chronopoulou, A.; Peters, M. E.; Fraser, A.; and Dodge, J. 2023. Adaptersoup: Weight averaging to improve generalization of pretrained language models. *arXiv preprint arXiv:2302.07027*.

Das, S. S. S.; Zhang, R. H.; Shi, P.; Yin, W.; and Zhang, R. 2023. Unified low-resource sequence labeling by sample-aware dynamic sparse finetuning. *arXiv preprint arXiv:2311.03748*.

Dou, S.; Zhou, E.; Liu, Y.; Gao, S.; Zhao, J.; Shen, W.; Zhou, Y.; Xi, Z.; Wang, X.; Fan, X.; et al. 2023. LoRAMoE: Alleviate world knowledge forgetting in large language models via MoE-style plugin. *arXiv preprint arXiv:2312.09979*.

Grattafiori, A.; Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Vaughan, A.; et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Guo, D.; Rush, A. M.; and Kim, Y. 2020. Parameter-efficient transfer learning with diff pruning. *arXiv preprint arXiv:2012.07463*.

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

Hayou, S.; Ghosh, N.; and Yu, B. 2024. Lora+: Efficient low rank adaptation of large models. *arXiv preprint arXiv:2402.12354*.

He, J.; Zhou, C.; Ma, X.; Berg-Kirkpatrick, T.; and Neubig, G. 2021. Towards a unified view of parameter-efficient transfer learning. *arXiv preprint arXiv:2110.04366*.

He, S.; Ding, L.; Dong, D.; Zhang, M.; and Tao, D. 2022. Sparseadapter: An easy approach for improving the parameter-efficiency of adapters. *arXiv preprint arXiv:2210.04284*.

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

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W.; et al. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2): 3.

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

Jiang, A. Q.; Sablayrolles, A.; Roux, A.; Mensch, A.; Savary, B.; Bamford, C.; Chaplot, D. S.; Casas, D. d. l.; Hanna, E. B.; Bressand, F.; et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Kavehzadeh, P.; Valipour, M.; Tahaei, M.; Ghodsi, A.; Chen, B.; and Rezagholizadeh, M. 2023. Sorted LLaMA: Unlocking the potential of intermediate layers of large language models for dynamic inference. *arXiv preprint arXiv:2309.08968*.

Kong, R.; Li, Q.; Fang, X.; Feng, Q.; He, Q.; Dong, Y.; Wang, W.; Li, Y.; Kong, L.; and Liu, Y. 2024. LoRA-Switch: Boosting the Efficiency of Dynamic LLM Adapters via System-Algorithm Co-design. *arXiv preprint arXiv:2405.17741*.

Lai, V. D.; Van Nguyen, C.; Ngo, N. T.; Nguyen, T.; Dernoncourt, F.; Rossi, R. A.; and Nguyen, T. H. 2023. Okapi: Instruction-tuned large language models in multiple languages with reinforcement learning from human feedback. *arXiv preprint arXiv:2307.16039*.

Lei, T.; Bai, J.; Brahma, S.; Ainslie, J.; Lee, K.; Zhou, Y.; Du, N.; Zhao, V.; Wu, Y.; Li, B.; et al. 2023. Conditional adapters: Parameter-efficient transfer learning with fast inference. *Advances in Neural Information Processing Systems*, 36: 8152–8172.

Liao, B.; Meng, Y.; and Monz, C. 2023. Parameter-efficient fine-tuning without introducing new latency. *arXiv preprint arXiv:2305.16742*.

Liu, S. 2024. Model merging.

Liu, S.-Y.; Wang, C.-Y.; Yin, H.; Molchanov, P.; Wang, Y.-C. F.; Cheng, K.-T.; and Chen, M.-H. 2024a. Dora: Weight-decomposed low-rank adaptation. In *Forty-first International Conference on Machine Learning*.

Liu, Z.; Kong, C.; Liu, Y.; and Sun, M. 2024b. Fantastic semantics and where to find them: Investigating which layers of generative llms reflect lexical semantics. *arXiv preprint arXiv:2403.01509*.

Luo, T.; Lei, J.; Lei, F.; Liu, W.; He, S.; Zhao, J.; and Liu, K. 2024. Moelora: Contrastive learning guided mixture of experts on parameter-efficient fine-tuning for large language models. *arXiv preprint arXiv:2402.12851*.

Luo, Y.; Yang, Z.; Meng, F.; Li, Y.; Zhou, J.; and Zhang, Y. 2023. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *arXiv preprint arXiv:2308.08747*.

Ma, Y.; Liang, Z.; Dai, H.; Chen, B.; Gao, D.; Ran, Z.; Zihan, W.; Jin, L.; Jiang, W.; Zhang, G.; et al. 2024. MoDULA: Mixture of Domain-Specific and Universal LoRA for Multi-Task Learning. *arXiv preprint arXiv:2412.07405*.

Pal, A.; et al. 2022. MedMCQA: A Large-scale Multi-Subject Multi-Choice Dataset for Medical domain Question Answering. In *Proceedings of the Conference on Health, Inference, and Learning*. PMLR.

Shi, F.; Suzgun, M.; Freitag, M.; Wang, X.; Srivats, S.; Vosoughi, S.; Chung, H. W.; Tay, Y.; Ruder, S.; Zhou, D.; et al. 2022. Language models are multilingual chain-of-thought reasoners. *arXiv preprint arXiv:2210.03057*.

Sung, Y.-L.; Nair, V.; and Raffel, C. A. 2021. Training neural networks with fixed sparse masks. *Advances in Neural Information Processing Systems*, 34: 24193–24205.

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

Valipour, M.; Rezagholizadeh, M.; Kobyzev, I.; and Ghodsi, A. 2022. Dylora: Parameter efficient tuning of pre-trained models using dynamic search-free low-rank adaptation. *arXiv preprint arXiv:2210.07558*.

Varshney, N.; Chatterjee, A.; Parmar, M.; and Baral, C. 2023. Accelerating llama inference by enabling intermediate layer decoding via instruction tuning with lite. *arXiv preprint arXiv:2310.18581*.

Wang, H.; Ping, B.; Wang, S.; Han, X.; Chen, Y.; Liu, Z.; and Sun, M. 2024. Lora-flow: Dynamic lora fusion for large language models in generative tasks. *arXiv preprint arXiv:2402.11455*.

Wei, Y.; Wang, Z.; Liu, J.; Ding, Y.; and Zhang, L. 2023. Magicoder: Empowering code generation with oss-instruct. *arXiv preprint arXiv:2312.02120*.

Xu, J.; Lai, J.; and Huang, Y. 2024. Meteora: Multiple-tasks embedded lora for large language models. *arXiv preprint arXiv:2405.13053*.

Xu, L.; Xie, H.; Qin, S.-Z. J.; Tao, X.; and Wang, F. L. 2023. Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment. *arXiv preprint arXiv:2312.12148*.

Yu, L.; Jiang, W.; Shi, H.; Yu, J.; Liu, Z.; Zhang, Y.; Kwok, J. T.; Li, Z.; Weller, A.; and Liu, W. 2023. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*.

Zadouri, T.; Üstün, A.; Ahmadian, A.; Ermiş, B.; Locatelli, A.; and Hooker, S. 2023. Pushing mixture of experts to the limit: Extremely parameter efficient moe for instruction tuning. *arXiv preprint arXiv:2309.05444*.

Zaken, E. B.; Ravfogel, S.; and Goldberg, Y. 2021. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. *arXiv preprint arXiv:2106.10199*.

Zhang, F.; Li, L.; Chen, J.; Jiang, Z.; Wang, B.; and Qian, Y. 2023a. Increlora: Incremental parameter allocation method for parameter-efficient fine-tuning. *arXiv preprint arXiv:2308.12043*.

Zhang, M.; Chen, H.; Shen, C.; Yang, Z.; Ou, L.; Yu, X.; and Zhuang, B. 2023b. LoRAPrune: Structured pruning meets low-rank parameter-efficient fine-tuning. *arXiv preprint arXiv:2305.18403*.

Zhang, Q.; Chen, M.; Bukharin, A.; Karampatziakis, N.; He, P.; Cheng, Y.; Chen, W.; and Zhao, T. 2023c. Adalora: Adaptive budget allocation for parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.10512*.

Zhang, Y.; and Li, R. 2024. DLP-LoRA: Efficient Task-Specific LoRA Fusion with a Dynamic, Lightweight Plugin for Large Language Models. *arXiv preprint arXiv:2410.01497*.

Zhao, Z.; Gan, L.; Wang, G.; Zhou, W.; Yang, H.; Kuang, K.; and Wu, F. 2024. Loraretriever: Input-aware lora retrieval and composition for mixed tasks in the wild. *arXiv preprint arXiv:2402.09997*.

Zhu, Y.; Feng, J.; Zhao, C.; Wang, M.; and Li, L. 2021. Counter-interference adapter for multilingual machine translation. *arXiv preprint arXiv:2104.08154*.