HiLoRA: Adaptive Hierarchical LoRA Routing for Training-Free Domain Generalization

Ziyi Han[1], Huanyu Wang[2], Zeyu Zhang[1], Xiangxiang Dai[1], Xutong Liu[3], John C.S. Lui[1]
[1]The Chinese University of Hong Kong
[2]Zhejiang University [3]University of Washington
{zyhan24,zyzhang,xxdai23,cslui}@cse.cuhk.edu.hk
huanyuhello@zju.edu.cn, xutongl@uw.edu

Abstract

Low-Rank Adaptation (LoRA) has emerged as a widely used technique for adapting large language models (LLMs) to new domains, due to its modular design and broad availability on platforms such as HuggingFace. This availability has motivated efforts to reuse existing LoRAs for domain generalization. However, existing methods often rely on explicit task labels or additional training, which are impractical for deployment. Moreover, they typically activate a fixed number of entire LoRA modules, leading to parameter redundancy or insufficiency that degrade performance. In this paper, we propose HiLoRA, a training-free framework that performs adaptive hierarchical routing over LoRA pools. Drawing on structural properties of LoRA, we define rank-one components (ROCs), where each LoRA rank is treated as an independent unit. For a given input, HiLoRA first adaptively selects a subset of LoRAs and determines their ROC allocation based on Gaussian likelihoods at the sequence level. At the token level, it refines routing by activating only the most informative ROCs. We further provide theoretical guarantees that HiLoRA selects the most relevant LoRAs with high probability. Extensive experiments show that HiLoRA achieves substantial improvements in domain generalization, with accuracy gains of up to 70% over state-of-the-art baselines, while maintaining comparable inference throughput.

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable capabilities across a wide variety of tasks [41, 24, 11]. However, adapting LLMs to specialized domains or tasks requires computationally expensive full fine-tuning [12]. To mitigate this cost, parameter-efficient fine-tuning (PEFT) techniques have been developed [7]. Among them, Low-Rank Adaptation (LoRA) [12, 32] has become one of the most effective and widely adopted methods. LoRA introduces lightweight low-rank matrices into selected layers of an LLM, thereby substantially reducing the number of trainable parameters while preserving strong downstream task performance. Building on this success, community platforms such as HuggingFace [14] and ModelScope [21] now host thousands of task-specific LoRA modules trained across diverse domains. This rapidly expanding repository creates a unique opportunity: for tasks without specialized LoRAs, one can directly leverage existing LoRAs instead of training new ones, enabling scalable multi-domain adaptation.

However, realizing this potential is highly non-trivial, as effectively utilizing community-shared LoRAs introduces several challenges. *First*, explicit task labels of inputs are typically unavailable in practice. We refer to tasks with corresponding specialized LoRAs as *seen tasks*, and those without

Corresponding author: Xiangxiang Dai.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

as *unseen tasks*. If labels were available, inputs from seen tasks could be directly routed to their LoRAs, while unseen tasks could be aligned with related ones based on task similarity. Without labels, identifying and routing such inputs becomes highly challenging. *Second*, different inputs may require different levels of LoRA knowledge. Activating too many LoRAs (or entire modules) may introduce redundancy and interference, while activating too few may miss relevant knowledge, leading to reduced accuracy [2]. *Third*, as the number of task-specific LoRAs continues to grow, evaluating and computing many candidate LoRAs becomes increasingly expensive, resulting in substantial inference overhead and scalability challenges [26].

Recent work has attempted to address the above challenges by integrating Mixture-of-Experts (MoE) mechanisms with LoRAs [10], where gating functions route inputs to subsets of LoRAs at sequence-level [8, 9], token-level [26], hybrid hierarchical routing [17], and rank-level routing [40]. However, these gating functions often rely on explicit task labels [18] or require gradient-based training of gating parameters [23], which restricts their applicability in practice. Moreover, many methods adopt fixed top-k selection strategies [26, 38, 9], which can result in either excessive or insufficient activation, thereby reducing adaptability. In parallel, some studies focus on LoRA merging, which combines multiple task-specific LoRAs into a unified module [4, 6, 39]. These methods improve cross-domain generalization by leveraging knowledge across tasks. However, they typically construct a static merged model shared across all inputs, limiting adaptability and potentially introducing parameter interference in heterogeneous settings. A more detailed discussion of related work is provided in Sec. 5. This motivates the following research question:

Can we adaptively leverage a large collection of specialized LoRA modules to support both seen and unseen tasks without gradient-based training or explicit task labels?

In this paper, we highlight **three key observations** about the structure of LoRA, derived from empirical analysis [42, 32] and experimental evidence. (i) Each LoRA rank is formed by a pair of down- and up-projection vectors, which can be treated as an independent *rank-one component (ROC)* and serve as the basic unit of LoRA adaptation. (ii) Within a LoRA, the down-projection vectors across ROCs exhibit strong randomness and primarily serve as scaling factors that modulate the contributions of the corresponding up-projection vectors. (iii) In contrast, the up-projection vectors exhibit clear clustering patterns, often forming multiple groups within a single LoRA, reflecting distinct semantic aspects of the adaptation.

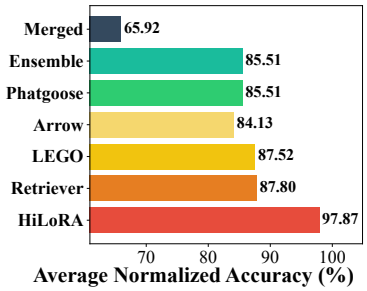


Figure 1: Average normalized accuracy over ten NLI tasks, with five seen tasks and five unseen tasks. `HiLoRA` achieves the best performance and approaches the accuracy of task-specific LoRAs. Detailed results are shown in Tab. 1.

Building on these insights, we propose `HiLoRA`, a hierarchical LoRA routing framework for robust domain generalization. To the best of our knowledge, `HiLoRA` is the *first method* to introduce hierarchical routing at the granularity of ROCs, while providing theoretical guarantees for LoRA identification. At the sequence level, `HiLoRA` narrows the candidate space and improves robustness by activating only an adaptively sized subset of LoRAs based on input-LoRA similarity. To enable comparison across different parameter spaces, each LoRA is represented as a Gaussian distribution fitted to a small set of sampled embeddings, and similarity is measured using Gaussian likelihoods. This probabilistic formulation enables reliable identification of seen and unseen tasks while providing confidence-aware LoRA selection and ROC allocation within the selected LoRAs. At the token level, routing is further refined through ROC selection based on down-projection responses, without introducing additional parameters or training. We summarize our contributions as follows.

- **Key Observation.** We identify the ROC as the basic routing unit of LoRA and show the necessity of performing routing at this fine-grained granularity.
- **Hierarchical LoRA Routing Framework.** `HiLoRA` constructs a dynamic LoRA pool, where each LoRA is represented as a Gaussian distribution fitted from sampled data embeddings. At the sequence level, Gaussian likelihood scores are calculated for adaptive LoRA selection and ROC budget decision, while normalized scores guide probabilistic sampling for ROC allocation. At the token level, routing is further refined by selecting ROCs with stronger down-projection responses.

- **Theoretical Guarantee.** We derive error bounds for LoRA identification, providing formal guarantees that HiLoRA preserves the corresponding LoRAs for seen tasks and the closest LoRAs for unseen tasks with high probability, thereby ensuring robust routing across domains.
- **Experimental Performance.** As shown in Fig. 1 for a representative case, HiLoRA consistently outperforms state-of-the-art baselines in both within-cluster and cross-cluster evaluations, achieving normalized performance gains of up to 70% on LLaMA2-7B and 20% on FLAN-T5-Large (normalized by oracle task-specific LoRAs), while maintaining practical inference throughput.[1]

2 Preliminaries

Basic Formulation of LoRA. LoRA [12] achieves performance comparable to full fine-tuning by freezing the pretrained weights $\boldsymbol{W}_0$ and inserting trainable low-rank matrices $\Delta\boldsymbol{W}$ into selected layers, yielding $\boldsymbol{W}' = \boldsymbol{W}_0 + \Delta\boldsymbol{W}$. The update matrix is factorized as $\Delta\boldsymbol{W} = \boldsymbol{B}\boldsymbol{A}$, where $\boldsymbol{A} \in \mathbb{R}^{r\times d}$ is the down-projection matrix and $\boldsymbol{B} \in \mathbb{R}^{d\times r}$ is the up-projection matrix, with rank $r \ll d$. This reduces the number of trainable parameters from d^2 to $2rd$ while retaining strong adaptability.

Dyadic Product Representation. Let $\{\boldsymbol{a}_i^\top\}_{i=1}^r$ denote the set of row vectors of $\boldsymbol{A}$ and $\{\boldsymbol{b}_i\}_{i=1}^r$ denote the set of column vectors of $\boldsymbol{B}$, where $\boldsymbol{a}_i, \boldsymbol{b}_i \in \mathbb{R}^d$, $r \in \mathbb{R}$ denotes the rank of matrices $\boldsymbol{A}$ and $\boldsymbol{B}$. Under this notation, the low-rank update can be written as $\Delta\boldsymbol{W} = \boldsymbol{B}\boldsymbol{A} = \sum_{i=1}^r \left(\boldsymbol{b}_i\boldsymbol{a}_i^\top\right)$, which expresses $\Delta\boldsymbol{W}$ as a sum of r dyadic products, each formed by the outer product of two vectors $(\boldsymbol{a}_i, \boldsymbol{b}_i)$. Given an input $\boldsymbol{x} \in \mathbb{R}^d$, the sub-module output $\boldsymbol{y} \in \mathbb{R}^d$ is reformulated under LoRA adaptation as:

$$\boldsymbol{y} = \boldsymbol{W}_0\boldsymbol{x} + \Delta\boldsymbol{W}\boldsymbol{x} = \boldsymbol{W}_0\boldsymbol{x} + \boldsymbol{B}\boldsymbol{A}\boldsymbol{x} = \boldsymbol{W}_0\boldsymbol{x} + \sum_{i=1}^r \left(\boldsymbol{b}_i\boldsymbol{a}_i^\top\right)\boldsymbol{x} = \boldsymbol{W}_0\boldsymbol{x} + \sum_{i=1}^r \boldsymbol{b}_i\left(\boldsymbol{a}_i^\top\boldsymbol{x}\right). \quad (1)$$

In this decomposition, each row of the down-projection matrix $\boldsymbol{A}$ is paired with the corresponding column of the up-projection matrix $\boldsymbol{B}$. From a computational perspective, each pair $(\boldsymbol{a}_i, \boldsymbol{b}_i)$ forms the smallest indivisible unit in LoRA, which we define as a rank-one component (ROC). A ROC corresponds to one LoRA rank and serves as the fundamental unit of LoRA adaptation. This formulation naturally motivates routing at the ROC level, where fine-grained component selection enables more flexible composition of LoRA knowledge and may improve domain generalization.

3 Methodology

3.1 HiLoRA Framework

Problem Formulation. Consider a pre-trained LLM $\boldsymbol{L}$ and an existing pool of I task-specific LoRAs, denoted as $\Phi = \{\phi_1, \phi_2, \ldots, \phi_I\}$, where each LoRA ϕ_i is individually trained for task i. It is implemented by inserting low-rank matrices into selected layers of $\boldsymbol{L}$. For clarity, the low-rank matrices of ϕ_i at a given layer are denoted as $\boldsymbol{A}_i$ and $\boldsymbol{B}_i$, with rank r_i. Each LoRA is accompanied by a small set of representative inputs for offline profiling. Here, training-free means that LoRA reuse requires no additional gradient-based optimization, while task-label-free means that incoming test inputs do not require task identities. Our objective is to design a routing mechanism that performs competitively on tasks whose task-specific LoRAs are available in the LoRA pool (*seen tasks*), while also generalizing to inputs from domains without specialized LoRAs (*unseen tasks*).

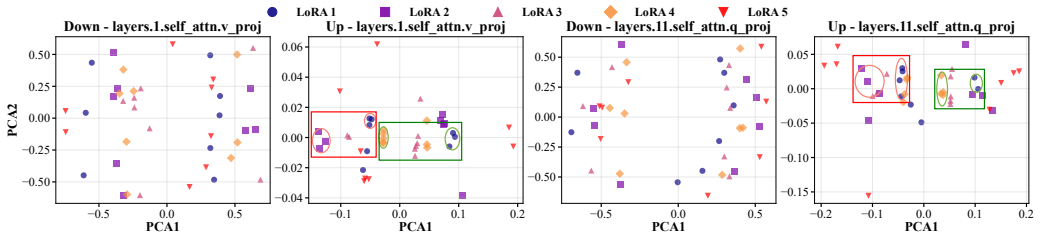


Figure 2: Scatter plots of the first two principal components of vectors in LoRA projection matrices specialized for five NLI tasks. The highlighted regions show that optimal routing for an unseen task (pink triangles) selects only the vectors aligned with relevant semantics.

Motivating Observations. The functional distinction between the down- and up-projection vectors follows naturally from the LoRA update structure. Each ROC operates as $(\boldsymbol{b}_i\boldsymbol{a}_i^\top)\boldsymbol{x} = \boldsymbol{b}_i(\boldsymbol{a}_i^\top\boldsymbol{x})$,

[1] Code link: https://github.com/ziyihanzyh/HiLoRA

where the down-projection vector $\boldsymbol{a}_i$ controls the activation strength, while the up-projection vector $\boldsymbol{b}_i$ determines the semantic direction of the update. This interpretation is consistent with prior observations [42, 32]. To further validate this distinction and analyze properties of ROCs, we visualize LoRA parameters using Principal Component Analysis (PCA) [1]. Specifically, the ROCs within a LoRA $\{\boldsymbol{a}_i, \boldsymbol{b}_i\}_{i=1}^{r}$ are projected into a two-dimensional space. We analyze five LoRAs fine-tuned on different NLI tasks, with the resulting scatter plots shown in Fig. 2, where vectors sharing the same color and shape originate from the same LoRA.

Three key observations arise from these visualizations. (i) The down-projection vectors of ROCs exhibit highly dispersed distributions and show little semantic structure. This confirms that the down-projection vector $\boldsymbol{a}$ mainly acts as a scaling factor rather than encoding task-specific information. (ii) In contrast, the up-projection vectors exhibit clear task-dependent patterns and form multiple clusters within a LoRA, where different clusters capture different semantic aspects of the adaptation. (iii) Unseen tasks tend to align with only a subset of semantic clusters rather than the entire LoRA. As illustrated in Fig. 2, when the pink LoRA corresponds to an unseen task, the optimal routing selectively activates only specific clusters from related LoRAs (*e.g.*, the red box selects the purple and blue clusters, while the green box selects the orange and blue clusters). Similar selective activation patterns are also observed in the fourth subfigure. Taken together, these observations suggest that uniformly activating an entire LoRA may introduce parameter redundancy and semantic interference. Instead, effective routing should selectively activate the ROCs aligned with the relevant semantics, motivating routing at the ROC level. To demonstrate that these observations generalize beyond this example, additional visualizations are provided in Figs. 7 and 8 of Appendix B.2.

Workflow of `HiLoRA`. Motivated by these observations, routing at the granularity of ROCs is highly desirable. However, directly selecting ROCs from the entire LoRA pool faces two main challenges. First, the candidate space is excessively large, which makes exhaustive selection computationally infeasible. Second, the space is noisy, as ROCs from different LoRAs vary in relevance and quality, making it difficult to evaluate them under a unified criterion. To address these issues, we introduce `HiLoRA`, an adaptive hierarchical routing framework over a pool of task-specific LoRAs designed to achieve training-free domain generalization. The overview of our framework `HiLoRA` is illustrated in Fig. 3. Given an input sequence $\boldsymbol{x}$, `HiLoRA` operates in two stages. (i) *Input-Aware ROC Allocation*: At the sequence level, the framework measures the similarity between $\boldsymbol{x}$ and each LoRA ϕ_i using Gaussian likelihoods. Based on these probabilistic similarities, it adaptively selects a subset of LoRAs and assigns an appropriate number of ROCs to each. (ii) *Token-Level ROC Routing*: At the token level, the framework further refines adaptation by dynamically routing each token in $\boldsymbol{x}$ to the most relevant ROCs within the subset of LoRAs selected in stage (i). In both stages, comparisons are performed under a unified criterion, which ensures fair evaluation across LoRAs and their ROCs. Gaussian profiles are constructed offline from representative inputs and reused during inference. For each incoming sequence, `HiLoRA` determines the candidate LoRAs and their ROC budgets. These decisions are shared across the sequence, while the identities of the selected ROCs can vary across tokens and layers.

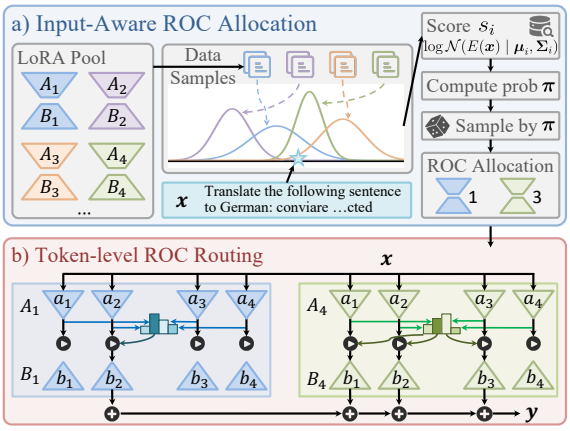


Figure 3: Overview of `HiLoRA` architecture.

3.2 Input-Aware ROC Allocation

At the sequence level, the goal is to identify candidate LoRAs from the pool and allocate a suitable number of ROCs to each, according to their relevance to the input. A key challenge arises because the input representations and LoRA parameters reside in distinct spaces, which prevents direct comparison. To address this issue, inspired by retrieval-based methods, each LoRA can be represented by a small set of samples drawn from its training dataset [38]. Instead of embedding LoRAs and inputs into a shared space and computing cosine similarity, we approximate each LoRA with a Gaussian

distribution fitted to the sampled embeddings [9, 4]. This yields a probabilistic representation that enables more robust matching. This probabilistic representation provides an information-theoretic characterization: inputs from seen tasks attain high likelihood under their corresponding LoRA distributions, while inputs from unseen tasks can still be aligned by evaluating their likelihood across all source distributions. Moreover, the resulting probabilities guide stochastic allocation of ROCs, which not only reduces over-reliance on a single LoRA but also encourages exploration across multiple relevant candidates.

Formally, let E denote a sentence embedding model, and $\boldsymbol{c}$ denote an instruction. The instructed embedding of an input $\boldsymbol{x}$ is given by $\boldsymbol{z} = E(\boldsymbol{c} \oplus \boldsymbol{x})$, where $\oplus$ denotes concatenation. Following [38], we set the instruction to "Represent the sentence for similar task retrieval" to encourage sequence-level similarity. For LoRA module ϕ_i, we randomly sample m domain-specific examples, obtain instructed embeddings $\{\boldsymbol{z}_1^{(i)}, \ldots, \boldsymbol{z}_m^{(i)}\}$, and fit a Gaussian distribution:

$$p_i(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z} \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i), \tag{2}$$

where $\boldsymbol{\mu}_i = \frac{1}{m}\sum_{j=1}^{m} \boldsymbol{z}_j^{(i)}$, $\boldsymbol{\Sigma}_i = \frac{1}{m-1}\sum_{j=1}^{m} \big(\boldsymbol{z}_j^{(i)} - \boldsymbol{\mu}_i\big)\big(\boldsymbol{z}_j^{(i)} - \boldsymbol{\mu}_i\big)^\top + \varepsilon \boldsymbol{I}$. Here, $\boldsymbol{I}$ is the identity matrix and ε is a small constant. The term $\varepsilon \boldsymbol{I}$ prevents degeneracy by ensuring that $\boldsymbol{\Sigma}_i$ remains full-rank and well-conditioned, which is essential when the number of samples is small or when the empirical covariance is nearly singular. For a given input $\boldsymbol{x}$, we then compute its log-likelihood under each LoRA distribution as the similarity score: $s_i(\boldsymbol{x}) = (1/\tilde{d}) \log p_i(\boldsymbol{z}), \forall i \in \{1, \ldots, I\}$, where $\tilde{d}$ denotes the embedding dimension. The normalization by $\tilde{d}$ reduces the scale sensitivity of log-likelihoods in high-dimensional embedding spaces. Since inputs may come from either seen or unseen tasks, scores typically exhibit two regimes: seen-task inputs often produce at least one high-scoring LoRA, whereas unseen-task inputs tend to yield uniformly low scores across the LoRA pool. Accordingly, we define the candidate LoRA set as

$$\mathbb{C}(\boldsymbol{x}) = \begin{cases} \{i \mid s_i(\boldsymbol{x}) > 0\}, & \text{if } \max_i s_i(\boldsymbol{x}) > 0, \\ \arg\operatorname{top}_i^n \, s_i(\boldsymbol{x}), & \text{if } \max_i s_i(\boldsymbol{x}) \le 0, \end{cases} \tag{3}$$

where $n = \lceil |\max_i s_i(\boldsymbol{x})| \rceil$. When at least one positive score exists, we retain only positively scored LoRAs. Otherwise, all LoRAs receive low scores for the current input, and the candidate set is adaptively expanded according to the degree of mismatch. More negative maximum scores indicate weaker alignment between the input and the current LoRA pool, motivating a larger candidate set to improve robustness under unseen-task scenarios.

Because the set of activated LoRAs $\mathbb{C}(\boldsymbol{x})$ varies across inputs, the ROC budget also needs to adapt dynamically. Using a fixed number of ROCs can easily become suboptimal: when only a few LoRAs are selected, a static budget may introduce redundancy, whereas selecting many LoRAs may lead to insufficient capacity. Therefore, the total ROC budget is defined as $O(\boldsymbol{x}) = \gamma \cdot \sum_{i \in \mathbb{C}(\boldsymbol{x})} r_i$, where $\gamma \in (0, 1)$ is a scaling factor. A large value of γ may introduce redundancy and interference, whereas a small value may exclude essential information. Thus, γ is set to balance accuracy and efficiency by activating a compact yet sufficient set of ROCs. To allocate ROCs, we use Gaussian-likelihood scores as they offer a principled measure of input–LoRA alignment, enabling proportionally allocating more ROCs to better-matched LoRAs. Concretely, the scores of selected LoRAs are normalized into probabilities: $\pi_i(\boldsymbol{x}) = \frac{\exp(s_i(\boldsymbol{x}))}{\sum_{j \in \mathbb{C}(\boldsymbol{x})} \exp(s_j(\boldsymbol{x}))}, \forall i \in \mathbb{C}(\boldsymbol{x})$. Using these probabilities, first draw an initial ROC allocation $\{o_i\}_{i \in \mathbb{C}(\boldsymbol{x})}$ from a multinomial distribution with parameters $O(\boldsymbol{x})$ and $\boldsymbol{\pi}(\boldsymbol{x})$. To enforce the per-LoRA capacity constraint $o_i \le r_i$, any allocation exceeding the LoRA rank is clipped to its maximum capacity, and the excess budget is redistributed among LoRAs with remaining capacity according to their residual capacities.

3.3 Token-level ROC Routing

ROC Routing within Chosen LoRAs. At the token level, routing is refined by operating on the granularity of ROCs. As discussed in Sec. 3.1, the down-projection vectors mainly act as scaling factors. Therefore, the projection value $\boldsymbol{a}^\top \boldsymbol{x}$ provides a natural criterion for ROC selection, with larger values indicating stronger relevance between the token and the corresponding ROC. This criterion helps reduce redundancy by prioritizing the most informative ROCs while filtering out those with limited contribution or potential interference. Formally, for each layer and each token, and for every LoRA $i \in \mathbb{C}(\boldsymbol{x})$ selected at the sequence level, we compute the projection values $\boldsymbol{A}_i \boldsymbol{x}$. The most informative ROCs are then identified by selecting the indices of the top-o_i components ranked by the normalized projection magnitude: $\mathbb{J}_i = \arg\operatorname{top}_j^{o_i} \big(|\boldsymbol{a}_{ij}^\top \boldsymbol{x}| / \|\boldsymbol{a}_{ij}\|\big)$. Here, the projection

is taken in absolute value to capture both positive and negative activations [26]. The normalization is applied only during ROC selection to emphasize directional alignment and avoid domination by large-norm components, while the original magnitudes are preserved in the subsequent LoRA forward computation. The LoRA output for this layer is then obtained by aggregating the contributions of all activated ROCs: $\boldsymbol{y}' = \sum_{i\in\mathbb{C}(\boldsymbol{x})} \sum_{j\in\mathbb{J}_i} \boldsymbol{b}_{ij}\,(\boldsymbol{a}_{ij}^\top \boldsymbol{x})$.

It is important to emphasize that this routing introduces no additional parameters or retraining. Since projection values $\boldsymbol{a}^\top \boldsymbol{x}$ are required for all activated ROCs, the only extra computation arises from evaluating projections of ROCs that are ultimately not selected. This overhead is minimal compared to the overall forward pass, ensuring efficiency while preserving robust adaptation.

Variance Normalization for Adaptive ROCs. In `HiLoRA`, the number of activated ROCs is adaptive and may range from 1 to $\sum_{i=1}^{I} r_i$, where r_i is the rank of LoRA ϕ_i. This variability can cause fluctuations in the scale of the aggregated LoRA output, which in turn may reduce the stability of model performance. Empirical findings in [39] show that LoRA outputs are approximately distributed as zero-mean Gaussians, with variance that grows with the number of activated ROCs. To mitigate this effect, we normalize the aggregated output by a scaling factor $\sqrt{\overline{r}(\boldsymbol{x})/O(\boldsymbol{x})}$, where $\overline{r}(\boldsymbol{x}) = \frac{1}{|\mathbb{C}(\boldsymbol{x})|}\sum_{i\in\mathbb{C}(\boldsymbol{x})} r_i$ is the average rank of the selected LoRAs [34]. Therefore, the output of a given layer for input $\boldsymbol{x}$ becomes: $\boldsymbol{y} = \boldsymbol{W}_0\boldsymbol{x} + \sqrt{\overline{r}/O(\boldsymbol{x})}\sum_{i\in\mathbb{C}(\boldsymbol{x})}\sum_{j\in\mathbb{J}_i} \boldsymbol{b}_{ij}(\boldsymbol{a}_{ij}^\top\boldsymbol{x})$. This variance normalization property has been formally established in Theorem 3.1 of [39]. For clarity and completeness, we restate it as a Lemma A.1 in Appendix A.

3.4 Theoretical Analysis

We present the error bounds of LoRA identification in `HiLoRA` under two scenarios: (i) *in-distribution (ID)* inputs from seen tasks, and (ii) *out-of-distribution (OOD)* inputs from unseen tasks.

We assume that the embedding representations associated with each LoRA can be approximated by Gaussian distributions, and provide empirical validation of this assumption in Tab. 13 of Appendix C. For any candidate size $\kappa \in \{1,\dots,I\}$, let $\mathbb{C}_\kappa(\boldsymbol{x})$ contain the indices of the κ LoRAs with the highest Gaussian likelihoods. We analyze how the exclusion probability depends on the candidate size and the relationships among task distributions.

Theorem 3.1. *For an input $\boldsymbol{x}$ with true label i, the probability that the LoRA ϕ_i is excluded from the candidate set $\mathbb{C}_\kappa(\boldsymbol{x})$ is bounded as:*

$$\Pr\big(i \notin \mathbb{C}_\kappa(\boldsymbol{x})\big) \;\le\; \tfrac{1}{\kappa}\textstyle\sum_{j\neq i}\exp\big(-B_{ij}\big). \tag{4}$$

where $B_{ij} = \frac{1}{8}(\boldsymbol{\mu}_i-\boldsymbol{\mu}_j)^\top\Big(\frac{\boldsymbol{\Sigma}_i+\boldsymbol{\Sigma}_j}{2}\Big)^{-1}(\boldsymbol{\mu}_i-\boldsymbol{\mu}_j) \;+\; \frac{1}{2}\log\frac{\left|(\boldsymbol{\Sigma}_i+\boldsymbol{\Sigma}_j)/2\right|}{\sqrt{|\boldsymbol{\Sigma}_i||\boldsymbol{\Sigma}_j|}}$ is the Bhattacharyya distance.

For an input $\boldsymbol{x}$ from unseen tasks, no exact task-specific LoRA exists in the pool, suppose it comes from an unknown target distribution q. Define the information-theoretically closest source domain as $i^\star := \arg\min_{i\in\{1,\cdots,I\}} D_{\mathrm{KL}}(q\,\|\,p_i)$.

Theorem 3.2. *Let $\phi_{i^\star}$ denote the closest LoRA for unseen task input $\boldsymbol{x}$. For any $\alpha\in(0,1]$ and $M_\alpha^j = \boldsymbol{\Sigma}_q^{-1} + \alpha\boldsymbol{\Sigma}_j^{-1} - \alpha\boldsymbol{\Sigma}_{i^\star}^{-1} \succ 0$, the probability that the LoRA $\phi_{i^\star}$ is excluded from $\mathbb{C}_\kappa(\boldsymbol{x})$ satisfies:*

$$\Pr\big(i^\star \notin \mathbb{C}_\kappa(\boldsymbol{x})\big) \;\le\; \tfrac{1}{\kappa}\sum_{j\neq i^\star} C_\alpha^j |M_\alpha^j|^{-1/2}\cdot\exp\Big(\tfrac{1}{2}(h_\alpha^j)^\top(M_\alpha^j)^{-1}h_\alpha^j - K_\alpha^j\Big). \tag{5}$$

where $h_\alpha^j = \boldsymbol{\Sigma}_q^{-1}\boldsymbol{\mu}_q + \alpha\boldsymbol{\Sigma}_j^{-1}\boldsymbol{\mu}_j - \alpha\boldsymbol{\Sigma}_{i^\star}^{-1}\boldsymbol{\mu}_{i^\star}$, $K_\alpha^j = \frac{1}{2}\boldsymbol{\mu}_q^\top\boldsymbol{\Sigma}_q^{-1}\boldsymbol{\mu}_q + \frac{\alpha}{2}(\boldsymbol{\mu}_j^\top\boldsymbol{\Sigma}_j^{-1}\boldsymbol{\mu}_j - \boldsymbol{\mu}_{i^\star}^\top\boldsymbol{\Sigma}_{i^\star}^{-1}\boldsymbol{\mu}_{i^\star})$, $C_\alpha^j = \exp(-\frac{\alpha}{2}\log|\boldsymbol{\Sigma}_j| + \frac{\alpha}{2}\log|\boldsymbol{\Sigma}_{i^\star}| - \frac{1}{2}\log|\boldsymbol{\Sigma}_q|)$.

The detailed proofs of Theorem 3.1 and Theorem 3.2, together with interpretations of the parameters used in the theoretical analysis, are provided in Appendix A.

Remark 3.3. Theorem 3.1 and Theorem 3.2 characterize how candidate-set size and distributional relationships affect LoRA exclusion. Increasing the candidate size improves retrieval coverage, but may also increase computation and introduce irrelevant updates. This coverage–cost trade-off motivates the adaptive candidate-selection strategy in `HiLoRA`. Empirical analyses of domain separability, divergence behavior, and OOD routing conditions are provided in Appendix C.

4 Experiments

4.1 Experimental Setup

Datasets and Models. We use a subset of tasks from FLAN-v2 [36], and organize them into ten clusters: Natural Language Inference (NLI), Question Answering (QA), Sentiment Analysis, Translation, Commonsense Reasoning, Paraphrase, Struct-to-Text, Coreference Resolution, Text Correction, and Word-level tasks. We construct the LoRA pool by downloading task-specific LoRAs for the selected tasks from HuggingFace. Since evaluation metrics vary across tasks, we adopt task-dependent measures including accuracy, F1 score, BLEU, and ROUGE-1, 2, L. Details of the selected tasks, their grouping, and metrics are provided in Appendix B.1. As backbone models, we use LLaMA2-7B and LLaMA2-13B [33], as well as FLAN-T5-large [5], covering two model scales within the LLaMA2 family and both decoder-only and encoder–decoder architectures. All experiments are conducted using PyTorch on NVIDIA H100 GPUs (80GB) running Ubuntu 22.04. Other implementation details are provided in Appendix B.1.

Baselines. We compare `HiLoRA` with the following state-of-the-art methods. (i) `Retriever` [38]: a sequence-level method that retrieves the top-k LoRAs based on cosine similarity between input and LoRA embedding. (ii) `LEGO` [39]: a ROC-level merging method that clusters all ROCs into k groups, merges each cluster into a new ROC, and applies the merged clusters to all tasks. (iii) `Arrow` [26]: a token-level routing approach that builds gating vectors from the first right singular vector of the LoRA update $\boldsymbol{BA}$. (iv) `Phatgoose` [23]: a token-level routing method where gating vectors are trained separately for each LoRA. (v) `Ensemble` [22]: an ensemble method that combines all LoRAs by averaging their outputs. (vi) `Merged` [25]: a method where all LoRAs are merged into a single module shared across tasks. For these methods, we follow the original implementations and standard hyperparameter settings reported in prior work. We further evaluate two variants of `HiLoRA`: `HiLoRA-GS`, which performs only sequence-level routing, and `HiLoRA-ROC`, which performs only token-level ROC routing by ranking all ROCs across LoRAs and selecting the top-k. These variants are introduced to evaluate the effectiveness of each routing level and the necessity of hierarchical routing. We also report `LoRA` as an oracle reference. `LoRA` denotes the oracle setting, where each input is routed to its corresponding task-specific LoRA.

Metric Presentation. For tasks with multiple evaluation metrics, we first compute the average score for each task. The cluster-level score is obtained by averaging across all tasks within the cluster. Since tasks may involve heterogeneous metrics and scales, we adopt a *normalization strategy* inspired by standard multi-task evaluation practices [35]. Specifically, each method is normalized by the corresponding oracle task-specific LoRA performance before averaging. For completeness, we additionally report the raw averages for `LoRA` as a reference baseline.

4.2 Main Results

Experimental results are reported under two evaluation settings: (i) the *within-cluster* setting, where unseen tasks originate from the same cluster as the seen tasks, and (ii) the *cross-cluster* setting, where unseen tasks come from different clusters. During evaluation, the LoRAs of seen tasks remain in the pool, while the LoRAs associated with unseen tasks are removed, ensuring a strict zero-shot setting.

Within-cluster Setting. In this setting, experiments are conducted on ten NLI tasks, with half designated as seen tasks and the other half as unseen tasks. Results are summarized in Tab. 1; per-task results for LLaMA2-13B and FLAN-T5-large are provided in Tabs. 3 and 4 of Appendix B.3, respectively. `HiLoRA` achieves the highest mean normalized accuracy among reuse methods on all three backbones. More specifically: (i) On *seen* tasks, `HiLoRA` achieves performance comparable to the oracle `LoRA` setting, and occasionally surpasses it, *e.g.*, on ANLI-r3 and QNLI. This suggests that `HiLoRA` can not only identify the most relevant LoRA, but also benefit from selectively activating complementary ROCs from selected LoRAs. (ii) On *unseen* tasks, `HiLoRA` also maintains strong performance, demonstrating its ability to generalize by leveraging semantically related LoRAs through selective ROC activation. Methods based on fixed LoRA activation, such as `Retriever`, `Arrow`, `Phatgoose`, and `Ensemble`, may introduce parameter redundancy or interference, limiting performance. Merge-based methods such as `LEGO` and `Merged` remain input-agnostic and cannot effectively eliminate redundant or irrelevant components. In contrast, `HiLoRA` combines hierarchical routing with fine-grained ROC selection, yielding stronger aggregate performance across seen

Table 1: Performance on the NLI cluster using LLaMA2-7B, LLaMA2-13B, and FLAN-T5-large. Tasks with white and gray backgrounds correspond to *seen* and *unseen* tasks, respectively. For each task, the best accuracy is shown in **bold** and the second best is underlined.

Methods	LoRA	HiLoRA	HiLoRA-GS	HiLoRA-ROC	Retriever	LEGO	Arrow	Phatgoose	Ensemble	Merged
LLaMA2-7B										
ANLI-r1	46.40	**45.00**	42.10	38.90	36.10	37.00	38.90	37.00	35.80	31.70
ANLI-r2	40.10	**40.60**	38.70	36.20	36.40	37.70	36.40	36.40	36.80	32.60
ANLI-r3	36.92	**37.67**	36.17	35.92	35.25	34.75	36.25	35.42	34.50	31.08
CB	80.00	68.00	70.00	64.00	66.00	66.00	64.00	**74.00**	68.00	56.00
MNLI	77.66	**76.33**	74.06	70.78	74.22	71.91	60.51	62.58	67.66	39.92
MNLI-mis	79.69	**78.59**	74.69	69.38	75.78	71.80	60.82	62.34	68.75	40.59
QNLI	77.27	**78.28**	77.23	59.02	62.19	58.71	59.02	59.80	57.89	45.23
RTE	72.96	74.44	**75.56**	65.93	65.93	71.11	**75.56**	74.07	71.48	53.70
SNLI	67.42	69.45	68.13	69.34	**70.94**	67.46	59.06	57.89	62.58	35.27
WNLI	72.86	**65.71**	62.29	48.57	47.14	50.00	48.57	52.86	50.00	50.00
Avg (Norm.)	65.13	**97.87**	95.19	86.59	87.80	87.52	84.12	85.51	85.51	65.92
LLaMA2-13B										
Avg (Norm.)	74.00	**98.93**	97.66	91.82	95.37	93.34	85.48	88.78	89.24	49.04
FLAN-T5-large										
Avg (Norm.)	67.81	**100.31**	96.30	98.10	98.86	83.88	86.55	82.32	84.10	79.50

and unseen tasks. Moreover, the degraded performance of `HiLoRA-GS` and `HiLoRA-ROC` further demonstrates that relying on only sequence-level or token-level routing is insufficient, validating the necessity of hierarchical routing.

Table 2: Performance of LLaMA2-7B and FLAN-T5-large under the cross-cluster setting. For each cluster, the best and second-best reuse-method results are **bold** and underlined, respectively.

Methods	LoRA	HiLoRA	HiLoRA-GS	HiLoRA-ROC	Retriever	LEGO	Arrow	Phatgoose	Ensemble	Merged
LLaMA2-7B										
NLI	65.13	**72.56**	68.90	69.58	68.49	66.76	66.62	68.26	68.12	22.21
QA	59.66	**79.91**	73.60	72.12	73.33	79.61	67.16	77.20	76.44	14.68
Senti.	59.87	91.46	85.58	**92.79**	85.50	90.52	69.32	90.50	85.82	9.50
Trans.	21.98	94.56	**99.70**	67.67	41.00	73.73	95.60	93.42	94.98	56.17
Common.	67.11	**79.59**	75.86	77.27	67.22	75.47	76.79	76.98	78.59	23.22
Paraph.	66.88	76.27	74.17	62.59	**78.30**	57.86	65.90	68.43	70.56	10.38
StT	44.51	64.46	**64.66**	55.50	61.69	38.23	62.80	63.95	62.25	56.39
Corefe.	47.95	128.57	**129.54**	123.68	123.24	122.82	127.54	121.79	126.62	14.58
Text-Corr.	54.73	58.91	**62.22**	48.73	50.86	45.38	55.69	56.99	56.56	12.71
Word	67.02	71.10	70.09	66.90	**71.89**	60.98	70.76	70.13	69.39	15.07
FLAN-T5-Large										
NLI	67.81	**93.78**	85.55	92.68	91.11	78.09	76.23	91.67	78.99	74.15
QA	67.40	**94.22**	91.73	93.30	89.43	88.95	88.13	93.56	89.50	87.02
Senti.	59.18	99.18	98.50	**100.31**	97.96	98.45	98.27	98.37	98.30	98.22
Trans.	18.97	99.03	98.72	98.10	**99.46**	98.81	98.51	97.24	98.73	97.90
Paraph.	78.33	**95.80**	95.27	84.97	91.66	93.57	92.39	95.19	94.21	92.94
StT	60.18	99.53	99.48	98.90	**99.57**	99.43	99.40	99.38	99.40	99.34
Corefe.	63.13	108.27	102.95	**116.59**	104.10	101.49	101.49	103.47	104.10	101.19
Text-Corr.	54.91	99.69	98.90	97.17	98.44	99.55	99.41	**99.72**	99.70	99.00
Word	71.55	100.94	100.74	88.42	100.58	102.95	101.09	102.55	**103.03**	102.30

Cross-cluster Setting. In this setting, each cluster is treated as unseen in turn, while the remaining clusters serve as seen. The full collections contain 50 and 42 task-specific LoRAs for LLaMA2-7B and LLaMA2-13B, respectively, while FLAN-T5-large includes 33 LoRAs due to limited community availability. Performance is evaluated on all tasks within the unseen cluster. Tab. 2 summarizes the LLaMA2-7B and FLAN-T5-large results; detailed results for all three backbones are provided in Tabs. 5–7 of Appendix B.3. Compared with the within-cluster setting, this setting is substantially more challenging because unseen tasks may differ significantly from the seen clusters. Nevertheless, `HiLoRA` achieves strong cross-domain generalization, with relative improvements over `LEGO` in cluster-average normalized performance of up to approximately 70%, 75%, and 20% on LLaMA2-7B, LLaMA2-13B, and FLAN-T5-large, respectively. As shown in Tabs. 2 and 5–7, `HiLoRA` achieves competitive cross-cluster performance across the evaluated backbones. Interestingly, `Ensemble` performs relatively better in the cross-cluster setting, suggesting that activating more LoRAs helps capture broader knowledge for unseen clusters. However, activating all LoRAs still introduces redundancy and interference. In contrast, `HiLoRA` adaptively selects relevant LoRAs according to input-LoRA similarity, achieving a better balance between information coverage and parameter efficiency.

4.3 Further Analysis

Performance of Input Mapping. To evaluate the input routing capability of `HiLoRA`, we analyze the similarity structure of task embeddings across different tasks. The detailed heatmap visualizations are

shown in Fig. 4, where tasks from the same cluster are enclosed within green boxes for clarity. Three observations can be made: (i) Task embeddings within the same domain exhibit higher similarity, indicating that HiLoRA effectively captures semantic relationships across related tasks. (ii) Compared with Retriever (see Fig. 9 in Appendix B.3), which uses cosine similarity scores in the bounded range $[-1, 1]$, the Gaussian likelihood formulation in HiLoRA produces clearer relative separation between highly matched and weakly matched LoRAs. (iii) HiLoRA adaptively adjusts the number of activated LoRAs according to the input-LoRA similarity. For seen tasks, only 1-2 LoRAs are typically activated, while for cross-cluster inputs, the candidate set can expand to 11 LoRAs to improve coverage. This adaptive mechanism reduces redundancy while maintaining robustness for unseen tasks.

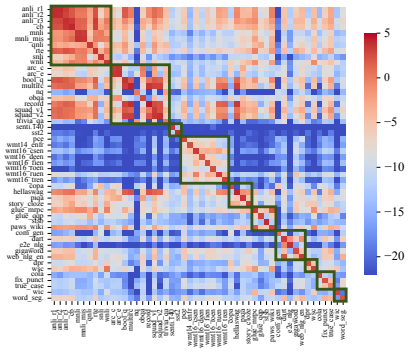


Figure 4: Input-LoRA similarity heatmap produced by HiLoRA.

Effect of Candidate-Selection Strategy. For candidate selection, we compare HiLoRA with: (i) Pos_topk, which preserves positive-score filtering but applies fixed top-k selection when all scores are non-positive; and (ii) Topk, which always uses fixed top-k selection. Detailed results are reported in Tab. 10 of Appendix B.3. HiLoRA achieves the highest mean normalized score in both within-cluster and cross-cluster settings. Since HiLoRA and Pos_topk share the same positive-score filtering rule, their comparison isolates the effect of adaptive candidate selection when all scores are non-positive.

Sensitivity of Input Mapping. We further evaluate the robustness of the proposed input mapping mechanism under different covariance regularization strengths, numbers of fitting samples, and embedding models. The detailed results are reported in Tabs. 11 and 12 of Appendix B.3. The results show that HiLoRA performs well with moderate ε, including 10^{-5}–10^{-4}. Too little regularization can leave covariance estimates ill-conditioned, whereas too much can obscure task-specific structure and weaken routing discrimination. The results also show that HiLoRA remains stable even with limited samples, with only minor performance degradation when reducing the number of fitting samples. In addition, replacing the instructor-tuned embedding model with a standard model (MPNet-base-v2 [30]) causes only a modest accuracy drop, indicating that the proposed routing mechanism is not overly sensitive to the embedding backbone. Together, these results demonstrate the robustness and generalizability of the proposed sequence-level routing framework across diverse settings.

Effect of ROC Budget. Here, we conduct an ablation study on the scaling factor γ, which controls the number of total ROCs activated. Experiments are performed under both within- and cross-cluster settings. While $\gamma = 40\%$ performs best on NLI tasks (as shown in Fig. 5), additional experiments on cross-cluster Translation and Struct-to-Text tasks (see Tab. 8 in Appendix B.3) show that the optimal value of γ is task-dependent, reflecting differences in task complexity and semantic alignment. Specifically, inputs with weaker alignment to existing LoRAs may benefit from larger γ, since activating more ROCs enables broader knowledge composition. However, excessively large γ may introduce redundancy and interference, while overly small values may omit informative ROCs and reduce representation capacity.

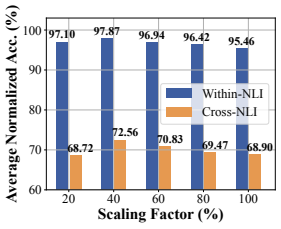


Figure 5: Performance of HiLoRA with different values of scaling factor γ.

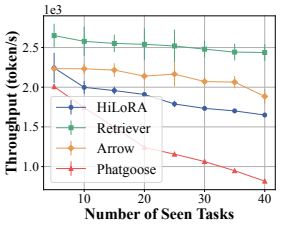


Figure 6: Inference throughput with different numbers of seen tasks.

Inference Throughput. We further evaluate inference throughput by comparing HiLoRA with dynamic routing baselines under different numbers of seen tasks, ranging from 5 to 40. For each configuration, inference throughput is measured on a test set containing 5 seen tasks and 5 unseen tasks. As shown in Fig. 6, throughput decreases gradually as the number of seen tasks increases. Compared with some single-level routing methods, HiLoRA incurs a throughput reduction of about 7-30%, but still achieves up to 90% higher throughput compared to Phatgoose. Considering the

substantial performance gains observed in both within- and cross-cluster settings, this moderate reduction in throughput is acceptable. Detailed latency breakdowns are provided in Appendix B.3 (Tab. 9), showing that HiLoRA maintains relatively stable throughput as the LoRA pool grows.

5 Related Work

Recent advances in extending LoRA for cross-domain adaptation fall into two primary directions: MoE-style routing and LoRA merging.

MoE-style Routing. These methods extend LoRA adaptation by dynamically activating subsets of LoRAs through gating functions [19]. At the sequence level, routing is performed using task-level similarity or global gating scores to select LoRA experts for the entire input, as in MoA [8] and MoLE [37]. Related approaches also perform adapter selection based on probabilistic similarity. For example, AdapterSwap [9] fits a Gaussian mixture model over embedding representations and selects top-k adapters according to likelihood scores during inference. At the token level, methods such as LoRA-Switch [15] and Arrow [26] introduce token-wise gating to activate different LoRAs for different positions. Hybrid strategies such as HMoRA [17] combine these two levels, aiming to balance efficiency and flexibility. Beyond entire LoRA routing, rank-level routing has also been explored, where each rank is treated as a micro-expert and subsets are activated, as in SMoRA [40] and LoRACoE [16]. Although these methods demonstrate the benefits of dynamic expert selection, they exhibit two key limitations: (i) they typically require training additional gating parameters, which undermines scalability and hinders deployment in training-free scenarios, and (ii) they impose a fixed activation budget, which reduces adaptability when handling diverse or unseen tasks. In contrast, our training-free framework combines sequence-level LoRA selection with ROC-level routing, reducing redundancy and improving robustness across both seen and unseen domains.

LoRA Merging. These methods aim to combine multiple task-specific LoRAs into a single unified module to support domain generalization [13, 6, 27]. AdapterSoup merges multiple task-specific adapters via weight averaging to improve generalization [4]. ZipLoRA merges independently trained subject and style LoRAs for personalized text-to-image generation [29]. LoRA-LEGO introduces rank-wise clustering and re-assembly of LoRA ranks to construct merged adapters with adjustable capacity [39]. Beyond heuristic merging, recent works explore more principled strategies: Closed-Form Merging (LoRM) derives analytical solutions for merging parameter-efficient modules in federated continual learning settings [28], while Adaptive LoRA Merge with Parameter Pruning further enhances robustness in low-resource domains by combining merging with pruning and lightweight fine-tuning [20]. While these approaches enhance cross-domain generalization by leveraging knowledge across tasks, they enforce a one-size-fits-all merged model. This limits flexibility and often degrades performance in scenarios involving diverse or unseen tasks. Our adaptive framework combines sequence-level LoRA selection with ROC-level routing for task-aware composition, reducing redundancy and interference.

6 Conclusion

In this paper, we present HiLoRA, a training-free framework for adaptive hierarchical routing over a task-specific LoRA pool to support robust domain generalization. HiLoRA builds on structural insights into LoRA by treating each ROC as the minimal routing unit. At the sequence level, it adaptively selects candidate LoRAs and allocates ROCs using Gaussian likelihoods, narrowing the search space and improving robustness. At the token level, routing is further refined by selecting the most informative ROCs, which reduces redundancy and interference. Theoretical analysis and extensive experiments show that HiLoRA reliably identifies relevant LoRAs, substantially improves domain generalization, and maintains efficiency with only a moderate reduction in inference throughput.

Despite its strengths, HiLoRA has several limitations. It relies on a small number of task-specific samples to construct Gaussian representations, which may not always be accessible and could raise privacy concerns. Moreover, the token-level routing mechanism is empirically validated but lacks formal theoretical guarantees. Future research could focus on addressing these limitations.

Acknowledgements and Disclosures

The work of Xiangxiang Dai was supported in part by the National Natural Science Foundation of China (625B2163). The work of John C. S. Lui was supported in part by the Research Grants Council (RGC) General Research Fund (GRF) under Grant 14202923.

References

[1] Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.

[2] Ming Cheng, Jiaying Gong, and Hoda Eldardiry. Sci-LoRA: Mixture of scientific LoRAs for cross-domain lay paraphrasing. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 18524–18541, July 2025. URL https://aclanthology.org/2025.findings-acl.953/.

[3] Herman Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *The Annals of Mathematical Statistics*, pages 493–507, 1952.

[4] Alexandra Chronopoulou, Matthew E Peters, Alexander Fraser, and Jesse Dodge. Adaptersoup: Weight averaging to improve generalization of pretrained language models. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 2054–2063, 2023.

[5] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, 2024.

[6] Eric Nuertey Coleman, Luigi Quarantiello, Julio Hurtado, and Vincenzo Lomonaco. Adaptive lora merging for efficient domain incremental learning. In *Adaptive Foundation Models: Evolving AI for Personalized and Efficient Learning*, 2024.

[7] Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, et al. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature machine intelligence*, 5(3):220–235, 2023.

[8] Wenfeng Feng, Chuzhan Hao, Yuewei Zhang, Yu Han, and Hao Wang. 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)*, pages 11371–11380, May 2024. URL https://aclanthology.org/2024.lrec-main.994/.

[9] William Fleshman, Aleem Khan, Marc Marone, and Benjamin Van Durme. Adapterswap: Continuous training of llms with data removal and access-control guarantees. *arXiv preprint arXiv:2404.08417*, 2024.

[10] Chendi Ge, Xin Wang, Zeyang Zhang, Hong Chen, Jiapei Fan, Longtao Huang, Hui Xue, and Wenwu Zhu. Dynamic mixture of curriculum loRA experts for continual multimodal instruction tuning. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=zpGK1bOlHt.

[11] Ziyi Han, Xutong Liu, Ruiting Zhou, Xiangxiang Dai, and John Lui. Faster, smaller, and smarter: Task-aware expert merging for online moe inference. *arXiv preprint arXiv:2509.19781*, 2025.

[12] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

[13] Chengsong Huang, Qian Liu, Bill Yuchen Lin, Tianyu Pang, Chao Du, and Min Lin. Lorahub: Efficient cross-task generalization via dynamic loRA composition. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=TrloAXEJ2B.

[14] Hugging Face. Hugging Face. Official website. https://huggingface.co/. Accessed: 2026-10-22.

[15] Rui Kong, Qiyang Li, Xinyu Fang, Qingtian Feng, Qingfeng He, Yazhu Dong, Weijun Wang, Yuanchun Li, Linghe Kong, and Yunxin Liu. Lora-switch: Boosting the efficiency of dynamic llm adapters via system-algorithm co-design. *arXiv preprint arXiv:2405.17741*, 2024.

[16] Guanyu Li, Zhiheng Xi, Zhihao Zhang, Boyang Hong, Tao Gui, Qi Zhang, and Xuan-Jing Huang. Loracoe: Improving large language model via composition-based lora expert. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 31290–31304, 2025.

[17] Mengqi Liao, Wei Chen, Junfeng Shen, Shengnan Guo, and Huaiyu Wan. HMoRA: Making LLMs more effective with hierarchical mixture of loRA experts. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=lTkHiXeuDl.

[18] Yufei Ma, Zihan Liang, Huangyu Dai, Ben Chen, Dehong Gao, Zhuoran Ran, Wang Zihan, Linbo Jin, Wen Jiang, Guannan Zhang, Xiaoyan Cai, and Libin Yang. MoDULA: Mixture of domain-specific and universal LoRA for multi-task learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 2758–2770, November 2024. URL https://aclanthology.org/2024.emnlp-main.161/.

[19] Yuren Mao, Yuhang Ge, Yijiang Fan, Wenyi Xu, Yu Mi, Zhonghao Hu, and Yunjun Gao. A survey on lora of large language models. *Frontiers of Computer Science*, 19(7):197605, 2025.

[20] Ryota Miyano and Yuki Arase. Adaptive LoRA merge with parameter pruning for low-resource generation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 19353–19366, July 2025. URL https://aclanthology.org/2025.findings-acl.990/.

[21] ModelScope. Modelscope: Model-as-a-service platform. https://modelscope.cn, 2025. Accessed: 2025-09-1.

[22] Dominik J Mühlematter, Michelle Halbheer, Alexander Becker, Dominik Narnhofer, Helge Aasen, Konrad Schindler, and Mehmet Ozgur Turkoglu. Lora-ensemble: Efficient uncertainty modelling for self-attention networks. *arXiv preprint arXiv:2405.14438*, 2024.

[23] Mohammed Muqeeth, Haokun Liu, Yufan Liu, and Colin Raffel. Learning to route among specialized experts for zero-shot generalization. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=r0qcGcFL4U.

[24] Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. A comprehensive overview of large language models. *ACM Transactions on Intelligent Systems and Technology*, 16(5):1–72, 2025.

[25] Oleksiy Ostapenko, Lucas Caccia, Zhan Su, Nicolas Le Roux, Laurent Charlin, and Alessandro Sordoni. A case study of instruction tuning with mixture of parameter-efficient experts. In *NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following*, 2023.

[26] Oleksiy Ostapenko, Zhan Su, Edoardo Ponti, Laurent Charlin, Nicolas Le Roux, Lucas Caccia, and Alessandro Sordoni. Towards modular LLMs by building and reusing a library of loRAs. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=0ZFWfeVsaD.

[27] Reza Qorbani, Gianluca Villani, Theodoros Panagiotakopoulos, Marc Botet Colomer, Linus Härenstam-Nielsen, Mattia Segu, Pier Luigi Dovesi, Jussi Karlgren, Daniel Cremers, Federico Tombari, and Matteo Poggi. Semantic library adaptation: Lora retrieval and fusion for open-vocabulary semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9804–9815, June 2025.

[28] Riccardo Salami, Pietro Buzzega, Matteo Mosconi, Jacopo Bonato, Luigi Sabetta, and Simone Calderara. Closed-form merging of parameter-efficient modules for federated continual learning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=R0pY0qRUXL.

[29] Viraj Shah, Nataniel Ruiz, Forrester Cole, Erika Lu, Svetlana Lazebnik, Yuanzhen Li, and Varun Jampani. Ziplora: Any subject in any style by effectively merging loras. In *European Conference on Computer Vision*, pages 422–438. Springer, 2024.

[30] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mpnet: Masked and permuted pre-training for language understanding. *Advances in neural information processing systems*, 33:16857–16867, 2020.

[31] Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. One embedder, any task: Instruction-finetuned text embeddings. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1102–1121, Toronto, Canada, 2023. doi: 10.18653/v1/2023.findings-acl.71.

[32] Chunlin Tian, Zhan Shi, Zhijiang Guo, Li Li, and Cheng zhong Xu. HydraloRA: An asymmetric loRA architecture for efficient fine-tuning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=qEpi8uWX3N.

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

[34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[35] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32, 2019.

[36] Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=gEZrGCozdqR.

[37] Xun Wu, Shaohan Huang, and Furu Wei. Mixture of loRA experts. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=uWvKBCYh4S.

[38] Ziyu Zhao, Leilei Gan, Guoyin Wang, Wangchunshu Zhou, Hongxia Yang, Kun Kuang, and Fei Wu. Loraretriever: Input-aware lora retrieval and composition for mixed tasks in the wild. In *ACL (Findings)*, pages 4447–4462, 2024. URL https://doi.org/10.18653/v1/2024.findings-acl.263.

[39] Ziyu Zhao, Tao Shen, Didi Zhu, Zexi Li, Jing Su, Xuwu Wang, and Fei Wu. Merging loRAs like playing LEGO: Pushing the modularity of loRA to extremes through rank-wise clustering. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=j6fsbpAllN.

[40] Ziyu Zhao, Yixiao Zhou, Zhi Zhang, Didi Zhu, Tao Shen, Zexi Li, Jinluan Yang, Xuwu Wang, Jing Su, Kun Kuang, et al. Each rank could be an expert: Single-ranked mixture of experts lora for multi-task learning. *arXiv preprint arXiv:2501.15103*, 2025.

[41] Zixuan Zhou, Xuefei Ning, Ke Hong, Tianyu Fu, Jiaming Xu, Shiyao Li, Yuming Lou, Luning Wang, Zhihang Yuan, Xiuhong Li, et al. A survey on efficient inference for large language models. *arXiv preprint arXiv:2404.14294*, 2024.

[42] Jiacheng Zhu, Kristjan Greenewald, Kimia Nadjahi, Haitz Sáez De Ocáriz Borde, Rickard Brüel Gabrielsson, Leshem Choshen, Marzyeh Ghassemi, Mikhail Yurochkin, and Justin Solomon. Asymmetry in low-rank adapters of foundation models. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

A Theoretical Demonstration

A.1 Variance Normalization Property

For completeness, we restate the variance normalization property, originally established as Theorem 3.1 in [39]. As the full proof is already provided in the cited work, we omit the derivation here and present the result in the form of a lemma below.

Lemma A.1 (Theorem 3.1 in [39]). *Let $\boldsymbol{A}_1 \in \mathbb{R}^{d\times r}$, $\boldsymbol{B}_1 \in \mathbb{R}^{r\times d}$, and $\boldsymbol{A}_2 \in \mathbb{R}^{d\times k}$, $\boldsymbol{B}_2 \in \mathbb{R}^{k\times d}$, where all entries are independently sampled from the standard normal distribution $\mathcal{N}(0,1)$. If the product $\boldsymbol{A}_2\boldsymbol{B}_2$ is rescaled by the factor $\sqrt{r/k}$, then the variance of the entries in $\boldsymbol{A}_1\boldsymbol{B}_1$ coincides with that of the normalized product:* $\mathrm{Var}(\boldsymbol{A}_1\boldsymbol{B}_1) \;=\; \mathrm{Var}\big(\sqrt{\frac{r}{k}}\,\boldsymbol{A}_2\boldsymbol{B}_2\big)$.

A.2 Proof of Theorem 3.1

For inputs from seen tasks, we derive a top-κ error bound that measures the probability of excluding the correct LoRA from the selected set.

First, we show the Bayes error of two distributions as follows:

Lemma A.2. *For any two distributions i, j with class-conditional Gaussians $\mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ and $\mathcal{N}(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$ and prior probabilities π_i, π_j, the Bayes error rate satisfies: $P_{\mathrm{err}}^{(2)}(i,j) \;\le\; \sqrt{\pi_i\pi_j}\;\exp\big(-B_{ij}\big)$, where $B_{ij} = \frac{1}{8}(\boldsymbol{\mu}_i-\boldsymbol{\mu}_j)^\top\Big(\frac{\boldsymbol{\Sigma}_i+\boldsymbol{\Sigma}_j}{2}\Big)^{-1}(\boldsymbol{\mu}_i-\boldsymbol{\mu}_j) \;+\; \frac{1}{2}\log\frac{\big|(\boldsymbol{\Sigma}_i+\boldsymbol{\Sigma}_j)/2\big|}{\sqrt{|\boldsymbol{\Sigma}_i||\boldsymbol{\Sigma}_j|}}$.*

Proof. The Bayes error for the optimal (MAP) decision rule is:

$$\begin{aligned} P_{\mathrm{err}}^{(2)}(i,j) &= \int \min\{\pi_i p_i(\boldsymbol{z}),\; \pi_j p_j(\boldsymbol{z})\}\,d\boldsymbol{z} \;\le\; \int \sqrt{\pi_i p_i(\boldsymbol{z})\;\pi_j p_j(\boldsymbol{z})}\,d\boldsymbol{z} \\ &= \sqrt{\pi_i\pi_j}\int \sqrt{p_i(\boldsymbol{z})\,p_j(\boldsymbol{z})}\,d\boldsymbol{z} \;=\; \sqrt{\pi_i\pi_j}\,\rho(p_i,p_j), \end{aligned}$$

where the first inequality follows from the elementary bound $\min\{a,b\} \le \sqrt{ab}$ for $a,b \ge 0$, and $\rho(p_i,p_j) := \int\sqrt{p_i p_j}d\boldsymbol{z}$ denotes the Bhattacharyya coefficient (affinity) between p_i and p_j.

For $k \in \{i,j\}$, the Gaussian densities are:

$$p_k(\boldsymbol{z}) = (2\pi)^{-d/2}|\boldsymbol{\Sigma}_k|^{-1/2}\exp\Big(-\tfrac{1}{2}(\boldsymbol{z}-\boldsymbol{\mu}_k)^\top\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{z}-\boldsymbol{\mu}_k)\Big).$$

Thus, we have:

$$\sqrt{p_i(\boldsymbol{z})p_j(\boldsymbol{z})} = (2\pi)^{-d/2}|\boldsymbol{\Sigma}_i|^{-1/4}|\boldsymbol{\Sigma}_j|^{-1/4}\exp\Big(-\tfrac{1}{2}(\boldsymbol{z}-\tilde{\boldsymbol{\mu}})^\top\tilde{\boldsymbol{\Sigma}}^{-1}(\boldsymbol{z}-\tilde{\boldsymbol{\mu}})\Big)\,e^{-C},$$

where $\tilde{\boldsymbol{\Sigma}}^{-1} = \frac{1}{2}(\boldsymbol{\Sigma}_i^{-1}+\boldsymbol{\Sigma}_j^{-1}), \tilde{\boldsymbol{\mu}} = \tilde{\boldsymbol{\Sigma}}\cdot\frac{1}{2}(\boldsymbol{\Sigma}_i^{-1}\boldsymbol{\mu}_i+\boldsymbol{\Sigma}_j^{-1}\boldsymbol{\mu}_j)$, with $C = \frac{1}{4}\Big(\boldsymbol{\mu}_i^\top\boldsymbol{\Sigma}_i^{-1}\boldsymbol{\mu}_i+\boldsymbol{\mu}_j^\top\boldsymbol{\Sigma}_j^{-1}\boldsymbol{\mu}_j-2\tilde{\boldsymbol{\mu}}^\top\tilde{\boldsymbol{\Sigma}}^{-1}\tilde{\boldsymbol{\mu}}\Big)$.

Integration yields:

$$\rho(p_i,p_j) = |\boldsymbol{\Sigma}_i|^{-1/4}|\boldsymbol{\Sigma}_j|^{-1/4}\,e^{-C}\,|\tilde{\boldsymbol{\Sigma}}|^{1/2}.$$

Using standard matrix identities, we have:

$$C = \tfrac{1}{8}(\boldsymbol{\mu}_i-\boldsymbol{\mu}_j)^\top\Big(\tfrac{\boldsymbol{\Sigma}_i+\boldsymbol{\Sigma}_j}{2}\Big)^{-1}(\boldsymbol{\mu}_i-\boldsymbol{\mu}_j), \quad |\tilde{\boldsymbol{\Sigma}}|^{1/2}|\boldsymbol{\Sigma}_i|^{-1/4}|\boldsymbol{\Sigma}_j|^{-1/4} = exp(-\tfrac{1}{2}\log\frac{\big|\frac{\boldsymbol{\Sigma}_i+\boldsymbol{\Sigma}_j}{2}\big|}{\sqrt{|\boldsymbol{\Sigma}_i||\boldsymbol{\Sigma}_j|}}).$$

Substituting them into $\rho(p_i,p_j)$ gives:

$$\rho(p_i,p_j) = \exp\left(-\tfrac{1}{8}(\boldsymbol{\mu}_i-\boldsymbol{\mu}_j)^\top\Big(\tfrac{\boldsymbol{\Sigma}_i+\boldsymbol{\Sigma}_j}{2}\Big)^{-1}(\boldsymbol{\mu}_i-\boldsymbol{\mu}_j) - \tfrac{1}{2}\log\frac{\big|\frac{\boldsymbol{\Sigma}_i+\boldsymbol{\Sigma}_j}{2}\big|}{\sqrt{|\boldsymbol{\Sigma}_i||\boldsymbol{\Sigma}_j|}}\right) = \exp(-B_{ij}).$$

Therefore, we proved:

$$P_{\mathrm{err}}^{(2)}(i,j) \;\le\; \sqrt{\pi_i\pi_j}\;\rho(p_i,p_j) = \sqrt{\pi_i\pi_j}\;\exp\big(-B_{ij}\big).$$

□

In this paper, priors are not incorporated in the score. The same derivation yields the simplified form $P_{\text{err}}^{(2)}(i,j) \le \exp(-B_{ij})$. Then, we have the following error bound.

Pairwise Error. For a given input $\boldsymbol{x}$ and $label(x) = i$, define the pairwise overtake events as $A_{ij} = \{p_j(\boldsymbol{z}) \ge p_i(\boldsymbol{z})\}, \quad j \ne i$. For A_{ij}, we have:

$$\begin{aligned}\Pr(A_{ij} \mid label(x) = i) &= \int_{\{p_j(\boldsymbol{z}) \ge p_i(\boldsymbol{z})\}} p_i(\boldsymbol{z})\, d\boldsymbol{z} = \int_{\{p_j(\boldsymbol{z}) \ge p_i(\boldsymbol{z})\}} \min\{p_i(\boldsymbol{z}), p_j(\boldsymbol{z})\} d\boldsymbol{z} \quad (6)\\ &\le \int \min\{p_i, p_j\}\, d\boldsymbol{z} \le \int \sqrt{p_i p_j}\, d\boldsymbol{z} = \rho(p_i, p_j) = \exp(-B_{ij}),\end{aligned}$$

where the first equality follows from the definition of the error event: under class i, misclassification occurs precisely when $p_j(\boldsymbol{z}) \ge p_i(\boldsymbol{z})$; the second equality holds because, on the region $\{p_j(\boldsymbol{z}) \ge p_i(\boldsymbol{z})\}$, we have $\min\{p_i(\boldsymbol{z}), p_j(\boldsymbol{z})\} = p_i(\boldsymbol{z})$; the inequality is obtained by extending the domain of integration; and the last two equalities use the Bhattacharyya coefficient, as established in Lemma A.2.

Top-κ Error. Let $N_1 = \sum_{j \ne i} \mathbf{1}_{A_{ij}}$ denote the number of rivals that beat i. Then the Top-κ error event under $label(\boldsymbol{x}) = i$ is $\{N_1 \ge \kappa\}$. Now, we have the following analysis:

$$\begin{aligned}\Pr(N_1 \ge \kappa \mid label(x) = i) &\le \frac{\mathbb{E}[N_1 \mid label(x) = i]}{\kappa}\\ &= \frac{1}{\kappa} \sum_{j \ne i} \Pr(A_{ij} \mid label(x) = i)\\ &\le \frac{1}{\kappa} \sum_{j \ne i} exp(-B_{i,j}),\end{aligned}$$

where the first inequality applies the Markov's inequality; the equality follows from computing $\mathbb{E}[N_1]$ and substituting the pairwise error terms; and the final inequality then uses the bound in Eq. (6).

A.3 Proof of Theorem 3.2

Before presenting the proof, we first briefly interpret the main terms used in the bound. The matrix M_α^j is a weighted precision matrix combining the covariance information of q, j, and $i^\star$, while the condition $M_\alpha^j \succ 0$ ensures that the corresponding quadratic form is well defined and finite. The vector h_α^j captures the covariance-aware relative displacement between q, j, and $i^\star$. The term K_α^j represents a second-order quadratic correction, and C_α^j is a determinant-based scaling factor reflecting covariance volume mismatch across distributions. These terms together characterize the geometric relationship among the Gaussian distributions involved in the routing analysis.

Based on these definitions, we now present the proof of Theorem 3.2. For any competitor $j \ne i^\star$, consider the event $p_j(\boldsymbol{z}) \ge p_{i^\star}(\boldsymbol{z})$. For any $\alpha \in (0, 1]$, the Markov-Chernoff technique gives:

$$\Pr_{\boldsymbol{z} \sim q}\big(p_j(\boldsymbol{z}) \ge p_{i^\star}(\boldsymbol{z})\big) = \Pr\left(\left(\tfrac{p_j(\boldsymbol{z})}{p_{i^\star}(\boldsymbol{z})}\right)^\alpha \ge 1\right) \le \mathbb{E}_q\left[\left(\tfrac{p_j(\boldsymbol{z})}{p_{i^\star}(\boldsymbol{z})}\right)^\alpha\right],$$

where the first equality holds because, for any $\alpha > 0$, the event $\{p_j \ge p_{i^\star}\}$ is equivalent to $\left\{\left(\frac{p_j}{p_{i^\star}}\right)^\alpha \ge 1\right\}$; and the first inequality then follows from Markov's inequality: if a random variable $X \ge 0$, then for any $t > 0$, $\Pr(X \ge t) \le \frac{\mathbb{E}[X]}{t}$.

Lemma A.3. *Let $q = \mathcal{N}(\boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q)$, $p_j = \mathcal{N}(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$, and $p_{i^\star} = \mathcal{N}(\boldsymbol{\mu}_{i^\star}, \boldsymbol{\Sigma}_{i^\star})$ be full-rank d-variate Gaussians. For any $\alpha \in (0, 1]$, assume $M_\alpha^j := \boldsymbol{\Sigma}_q^{-1} + \alpha\, \boldsymbol{\Sigma}_j^{-1} - \alpha\, \boldsymbol{\Sigma}_{i^\star}^{-1} \succ 0$. Then the α-moment of the likelihood ratio admits the closed form as follows:*

$$\mathbb{E}_{\boldsymbol{z} \sim q}\left[\left(\tfrac{p_j(\boldsymbol{z})}{p_{i^\star}(\boldsymbol{z})}\right)^\alpha\right] = C_\alpha^j\ |M_\alpha^j|^{-1/2} \exp\big(\tfrac{1}{2}\,(h_\alpha^j)^\top (M_\alpha^j)^{-1} h_\alpha^j - K_\alpha^j\big),$$

where $h_\alpha^j = \boldsymbol{\Sigma}_q^{-1}\boldsymbol{\mu}_q + \alpha\, \boldsymbol{\Sigma}_j^{-1}\boldsymbol{\mu}_j - \alpha\, \boldsymbol{\Sigma}_{i^\star}^{-1}\boldsymbol{\mu}_{i^\star}$, $K_\alpha^j = \frac{1}{2}\boldsymbol{\mu}_q^\top \boldsymbol{\Sigma}_q^{-1}\boldsymbol{\mu}_q + \frac{\alpha}{2}\big(\boldsymbol{\mu}_j^\top \boldsymbol{\Sigma}_j^{-1}\boldsymbol{\mu}_j - \boldsymbol{\mu}_{i^\star}^\top \boldsymbol{\Sigma}_{i^\star}^{-1}\boldsymbol{\mu}_{i^\star}\big)$, $C_\alpha^j = \exp\Big(-\frac{\alpha}{2}\log|\boldsymbol{\Sigma}_j| + \frac{\alpha}{2}\log|\boldsymbol{\Sigma}_{i^\star}| - \frac{1}{2}\log|\boldsymbol{\Sigma}_q|\Big)$.

Proof. By Chernoff/Markov's trick (see [3]), we have:

$$\mathbb{E}_q\Big[\Big(\tfrac{p_j(\boldsymbol{z})}{p_{i^\star}(\boldsymbol{z})}\Big)^\alpha\Big] = \int_{\mathbb{R}^d} q(\boldsymbol{z}) \exp\Big(\alpha\big(\log p_j(\boldsymbol{z}) - \log p_{i^\star}(\boldsymbol{z})\big)\Big)\, d\boldsymbol{z}.$$

Write each log-density in quadratic form: $\log p(\boldsymbol{z}) = -\frac{d}{2}\log(2\pi) - \frac{1}{2}\log|\boldsymbol{\Sigma}| - \frac{1}{2}(\boldsymbol{z}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{z}-\boldsymbol{\mu})$. Collecting the constant (determinant) terms yields the prefactor $C_\alpha^{(q)}$. Collecting the quadratic and linear terms in $\boldsymbol{z}$ gives:

$$-\tfrac{1}{2}\boldsymbol{z}^\top M_\alpha \boldsymbol{z} + h_\alpha^\top \boldsymbol{z} - K_\alpha, \quad M_\alpha = \boldsymbol{\Sigma}_q^{-1} + \alpha\,\boldsymbol{\Sigma}_j^{-1} - \alpha\,\boldsymbol{\Sigma}_{i^\star}^{-1},$$

with h_α, K_α as stated. Completing the square and using the multivariate Gaussian integral $\int \exp(-\frac{1}{2}\boldsymbol{z}^\top A\boldsymbol{z} + b^\top \boldsymbol{z})\, d\boldsymbol{z} = (2\pi)^{d/2}|A|^{-1/2}\exp(\frac{1}{2}b^\top A^{-1}b)$ (valid for $A \succ 0$), and noticing that $(2\pi)^{d/2}$ cancels with the corresponding factor in q, we obtain the claimed closed form. □

If $q = p_{i^\star}$, i.e., $\boldsymbol{\mu}_q = \boldsymbol{\mu}_{i^\star}$ and $\boldsymbol{\Sigma}_q = \boldsymbol{\Sigma}_{i^\star}$, then we have:

$$\mathbb{E}_{\boldsymbol{z}\sim p_{i^\star}}\left[\left(\frac{p_j(\boldsymbol{z})}{p_{i^\star}(\boldsymbol{z})}\right)^\alpha\right] = \int p_{i^\star}(\boldsymbol{z})^{1-\alpha} p_j(\boldsymbol{z})^\alpha\, d\boldsymbol{z} = \rho_\alpha(p_{i^\star}, p_j),$$

where the right-hand side is the multivariate Gaussian Chernoff coefficient with weight α on p_j and $1-\alpha$ on $p_{i^\star}$:

$$\rho_\alpha(p_{i^\star}, p_j) = \frac{|\boldsymbol{\Sigma}_{i^\star}|^{\alpha/2}|\boldsymbol{\Sigma}_j|^{(1-\alpha)/2}}{|\alpha\boldsymbol{\Sigma}_{i^\star} + (1-\alpha)\boldsymbol{\Sigma}_j|^{1/2}} \cdot \exp\left(-\frac{\alpha(1-\alpha)}{2}(\boldsymbol{\mu}_j - \boldsymbol{\mu}_{i^\star})^\top (\alpha\boldsymbol{\Sigma}_{i^\star} + (1-\alpha)\boldsymbol{\Sigma}_j)^{-1}(\boldsymbol{\mu}_j - \boldsymbol{\mu}_{i^\star})\right).$$

This is equivalent to the standard Chernoff coefficient up to the convention of whether α is assigned to the first or second distribution.

Let $N_2 = \sum_{j\neq i^\star} \mathbf{1}_{p_j \geq p_{i^\star}}$ denote the number of rivals that beat $i^\star$. Then the Top-κ error event under $\boldsymbol{z} \sim q$ is $\{N_2 \geq \kappa\}$. Similar to the proof of Theorem 3.1, we have:

$$\begin{aligned}
\Pr(N_2 \geq \kappa \mid \boldsymbol{z} \sim q) &\leq \frac{\mathbb{E}[N_2 \mid \boldsymbol{z} \sim q]}{\kappa} \\
&= \frac{1}{\kappa}\sum_{j\neq i^\star} \Pr(p_j(\boldsymbol{z}) \geq p_{i^\star}(\boldsymbol{z}) \mid \boldsymbol{z} \sim q) \\
&\leq \frac{1}{\kappa}\sum_{j\neq i^\star} \mathbb{E}_q\Big[\Big(\tfrac{p_j(\boldsymbol{z})}{p_{i^\star}(\boldsymbol{z})}\Big)^\alpha\Big], \\
&= \frac{1}{\kappa}\sum_{j\neq i^\star} C_\alpha^j\ |M_\alpha^j|^{-1/2}\ \exp\big(\tfrac{1}{2}\,(h_\alpha^j)^\top (M_\alpha^j)^{-1} h_\alpha^j\ -\ K_\alpha^j\big)
\end{aligned}$$

B Experimental Supplementary Material

B.1 Details of experiment setting

Details of Evaluation Datasets and Metrics. We employ a subset of the FLAN-v2 datasets [36] for domain generalization. FLAN-v2 dataset is a large-scale instruction-tuning corpus that integrates diverse Natural Language Understanding (NLU) and Natural Language Generation (NLG) tasks into an instruction–response format. The full collections contain 50, 42, and 33 task-specific LoRAs for LLaMA2-7B, LLaMA2-13B, and FLAN-T5-large, respectively. In the cross-cluster setting, each cluster is held out in turn, and only LoRAs from the remaining clusters are used for routing. A detailed summary of the selected datasets, together with their associated evaluation metrics, is provided below.

Natural Language Inference. Natural language inference tasks require models to determine logical relations (entailment, contradiction, or neutrality) between pairs of sentences. We use the following datasets: (1) ANLI (v1-v3); (2) CB; (3) MNLI (matched, mismatched); (4) QNLI; (5) RTE; (6) SNLI; (7) WNLI. All datasets in this cluster are evaluated using accuracy as the metric.

Question Answering. Question answering tasks evaluate the ability to retrieve or generate correct answers from passages or knowledge bases. We use the following datasets: (1) ARC (Challenge, Easy); (2) BoolQ; (3) MultiRC; (4) NaturalQuestions; (5) OpenBookQA; (6) ReCoRD; (7) SQuAD (v1-v2); (8) TriviaQA. For ARC, BoolQ, OpenBookQA, and ReCoRD, accuracy is used as the evaluation metric. For the remaining datasets, both accuracy and F1 score are reported.

Sentiment Analysis. Sentiment analysis tasks involve classifying the polarity or emotional tone of text, such as positive or negative sentiment. We use the following datasets: (1) Sentiment140; (2) SST2. For Sentiment140, both accuracy and F1 score are reported, while SST2 is evaluated using accuracy only.

Translation. Translation tasks test the capacity to generate fluent and semantically correct text across different languages. We use the following datasets: (1) ParaCrawl_EnEs; (2) WMT14_EnFr; (3) WMT16_CsEn; (4) WMT16_DeEn; (5) WMT16_FiEn; (6) WMT16_RoEn; (7) WMT16_RuEn; (8) WMT16_TrEn. All translation tasks are evaluated using BLEU, which measures n-gram overlap between system outputs and reference translations.

Commonsense Reasoning. Commonsense reasoning tasks require leveraging everyday knowledge and logical inference to choose or generate plausible answers. We use the following datasets: (1) COPA; (2) HellaSwag; (3) PIQA; (4) StoryCloze. All datasets in this cluster are evaluated using accuracy as the metric.

Paraphrase. Paraphrase tasks assess whether two sentences express the same underlying meaning, despite differences in wording. We use the following datasets: (1) GLUE_MRPC; (2) GLUE_QQP; (3) STS-B; (4) PAWS_Wiki. All datasets in this cluster are evaluated using accuracy as the metric.

Struct-to-Text Generation. These tasks focus on converting structured data, such as triples or tables, into coherent natural language text. We use the following datasets: (1) CommonGen; (2) DART; (3) E2E_NLG; (4) Gigaword; (5) WebNLG_En. All datasets in this cluster are evaluated using ROUGE (ROUGE-1,2,L) and BLEU, since n-gram overlap captures the informativeness and fluency of generated text.

Coreference Resolution. Coreference tasks require identifying expressions in text that refer to the same entity. We use the following datasets: (1) Definite Pronoun Resolution; (2) WSC. Both datasets are evaluated using accuracy.

Text Correction. Text correction tasks involve detecting and fixing grammatical errors or inconsistencies in sentences. We use the following datasets: (1) CoLA; (2) FixPunct; (3) TrueCase. All datasets in this cluster are evaluated using accuracy.

Word-level Tasks. Word-level tasks examine lexical semantics and basic text processing, such as contextual meaning and segmentation. We use the following datasets: (1) WiC; (2) Word_Segment. WiC is evaluated using accuracy, while Word_Segment is evaluated using both accuracy and F1.

Accuracy is sufficient when tasks have clear-cut, single-label predictions, such as classification or multiple-choice settings, where each prediction is either entirely correct or incorrect. In contrast, tasks with span-based, multi-label, or imbalanced data distributions may yield partially correct predictions. In these cases, the F1 score is reported alongside accuracy, as it balances precision and recall and provides a more sensitive evaluation of partial correctness.

Implementation Details. We set the inference batch size to 32. For each seen task, $m = 20$ domain-specific samples from the corresponding dataset are used to fit a Gaussian distribution. The sentence embedding model E is implemented with instructor-base [31], an instruction-tuned encoder that produces task-aware representations. The regularization constant ε is set to 10^{-5}. The scaling factor γ is fixed at 40%. Following [38], we set the parameter $k = 3$ for all LoRA-level routing methods, and correspondingly $k = 24$ for all ROC-level routing methods.

B.2 Extended Visualization of LoRA Projections

In the main text, we reported scatter plots of the first two principal components obtained from vectors in LoRA projection matrices fine-tuned on five NLI tasks, focusing on two representative layers. To further validate these findings, Fig. 7 presents results from additional layers, which reveal consistent structural patterns across model depth. We additionally extend the analysis to tasks drawn from different clusters (As shown in Fig. 8), where similar trends are observed. Taken together, these results provide stronger empirical support for the key observations discussed in Sec. 3.1.

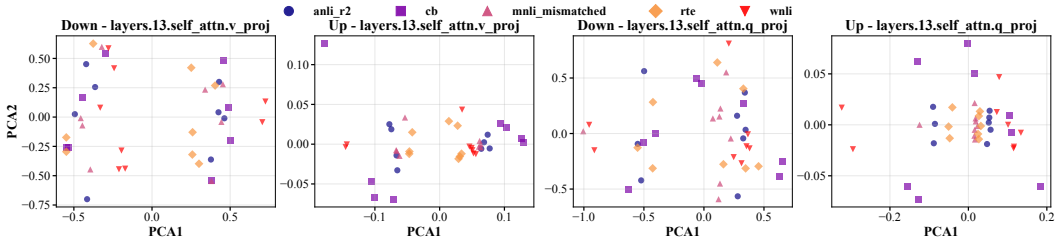


Figure 7: Scatter plots of the first two principal components derived from vectors in LoRA query and value projection matrices in layer 13 across five NLI tasks.

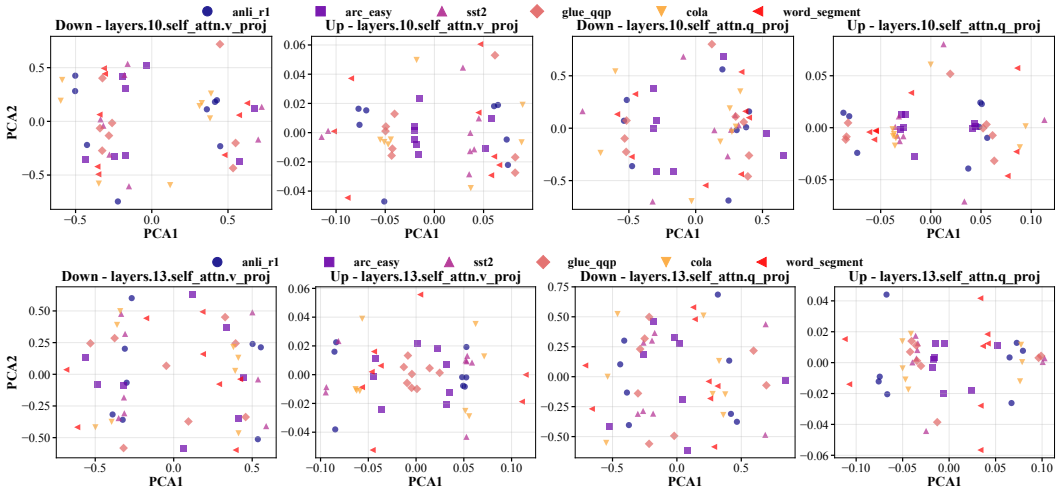


Figure 8: Scatter plots of the first two principal components derived from vectors in LoRA query and value projection matrices in layers 10 and 13 across six tasks from different clusters.

B.3 Full Experimental Results

Tabs. 3 and 4 report per-task within-cluster NLI results for LLaMA2-13B and FLAN-T5-large, respectively. Detailed cross-cluster results are provided for LLaMA2-7B in Tab. 5, LLaMA2-13B in Tab. 6, and FLAN-T5-large in Tab. 7. The LLaMA2-13B cross-cluster results cover 42 tasks across nine clusters, excluding Translation. For consistency with the main tables, per-task entries retain the raw metrics, while averages for reuse methods are computed after task-wise normalization by oracle LoRA performance. On cross-cluster NLI with LLaMA2-13B, `HiLoRA` achieves a mean normalized score of 78.50, compared with 78.19 for `Arrow`, 73.96 for `HiLoRA-GS`, and 56.87 for `HiLoRA-ROC`. Its largest relative improvement over `LEGO` in cluster-average normalized performance is approximately 74.6% on Struct-to-Text (65.69 versus 37.62). Performance varies across clusters: `HiLoRA` obtains the highest cluster average on NLI, QA, sentiment, and commonsense, but not on every cluster. Figs. 4 and 9 show a heatmap of input–LoRA similarities produced by `HiLoRA` and `Retriever` across tasks, respectively, with tasks from the same cluster grouped by green boxes. Tab. 8 summarizes the performance of `HiLoRA` under different settings of the scaling factor γ across various task types.

Beyond the throughput analysis in Sec. 4, Tab. 9 reports the per-batch input-mapping latency and inference latency across the ten evaluated tasks under varying LoRA-pool sizes. We can further observe that the throughput of HiLoRA decreases more slowly as the number of seen tasks grows. This behavior arises from two factors. First, although the input–LoRA mapping latency increases with larger LoRA pools, this component constitutes only a small fraction of the total inference time. Second, the number of selected LoRAs can change in two characteristic ways: (i) If at least one LoRA yields a positive similarity score, the selected set expands only when additional positive-score LoRAs appear as more seen tasks are added. (ii) If all similarity scores are negative, the maximum negative score tends to decrease as more LoRAs are included, leading to a smaller selected set. Once a positive-score LoRA appears, the system transitions back to case (i). These two mechanisms together explain the gradual and occasionally non-monotonic latency patterns observed at the per-task level. However, because throughput is computed as an average over all tasks in the test set, the aggregated trend becomes smoother and exhibits a modest decline followed by slow stabilization as the LoRA pool continues to grow.

Tab. 10 presents performance comparison under different candidate-selection rules with LLaMA2-7B on NLI tasks under both within- and cross-cluster settings. Tab. 11 reports sensitivity to the covariance regularization constant on within-cluster NLI with LLaMA2-7B. Here, Active LoRAs denotes the number of LoRAs selected at the sequence level for ROC allocation, and Total ROC Budget denotes the actual total number of ROCs allocated. Tab. 12 presents performance sensitivity to sample size and embedding models with LLaMA2-7B under within- and cross-cluster settings on the NLI cluster.

Table 3: Performance on the NLI cluster using LLaMA2-13B. Tasks with white and gray backgrounds correspond to *seen* and *unseen* tasks, respectively. For each task, the best accuracy is shown in **bold** and the second best is underlined.

Task	LoRA	HiLoRA	HiLoRA-GS	HiLoRA-ROC	Retriever	LEGO	Arrow	Phatgoose	Ensemble	Merged
ANLI-r1	60.30	**62.70**	60.60	58.80	61.80	58.30	47.40	55.60	52.00	24.70
ANLI-r2	47.30	**48.60**	47.00	46.20	48.50	44.30	41.00	43.70	41.40	24.30
ANLI-r3	49.92	48.92	47.75	49.08	**49.75**	47.17	44.92	47.33	46.42	26.42
CB	88.00	**86.00**	84.00	80.00	82.00	84.00	80.00	76.00	84.00	38.00
MNLI	87.97	**87.58**	87.34	82.58	86.56	85.47	70.00	76.45	75.35	37.30
MNLI-mis	89.80	**89.49**	88.63	83.79	88.87	85.70	71.02	75.94	76.41	36.64
QNLI	82.66	**83.13**	82.66	56.29	77.89	68.95	67.97	70.98	69.96	44.73
RTE	80.74	75.19	75.56	73.33	74.81	72.59	**78.89**	73.33	76.30	52.59
SNLI	81.91	81.84	**81.99**	76.72	81.60	79.65	67.19	75.23	72.38	32.46
WNLI	71.43	**67.14**	**67.14**	**67.14**	51.43	64.29	62.86	58.57	65.71	42.80
Avg. (Norm.)	74.00	**98.93**	97.66	91.82	95.37	93.34	85.48	88.78	89.24	49.04

Table 4: Performance on the NLI cluster using FLAN-T5-large. Tasks with white and gray backgrounds correspond to *seen* and *unseen* tasks, respectively. For each task, the best accuracy is shown in **bold** and the second best is underlined.

Methods	LoRA	HiLoRA	HiLoRA-GS	HiLoRA-ROC	Retriever	LEGO	Arrow	Phatgoose	Ensemble	Merged
ANLI-r1	60.20	60.40	61.20	57.90	60.70	60.60	60.80	**61.30**	60.70	60.80
ANLI-r2	43.30	42.20	42.70	41.00	42.80	42.50	**43.70**	42.90	43.50	43.40
ANLI-r3	44.50	43.08	44.42	42.42	44.25	44.17	**45.25**	44.67	44.33	44.33
CB	78.00	78.00	78.00	78.00	78.00	78.00	**80.00**	78.00	78.00	78.00
MNLI	88.59	**89.12**	84.77	89.00	88.16	64.69	61.76	66.52	63.52	58.87
MNLI-mis	89.14	**88.91**	82.07	88.52	88.67	62.93	68.79	64.10	61.05	56.76
QNLI	82.54	**82.70**	82.54	**82.70**	**82.70**	82.38	81.95	81.02	82.38	80.66
RTE	78.89	61.15	61.11	61.48	62.59	62.96	**63.70**	60.00	**63.70**	57.04
SNLI	60.08	**80.00**	60.31	**80.00**	68.44	13.75	17.85	10.14	13.28	6.13
WNLI	52.86	51.43	51.43	44.29	51.29	50.00	**54.29**	44.29	51.43	44.29
Avg (Norm.)	67.81	**100.31**	96.30	98.10	98.86	83.88	86.55	82.32	84.10	79.50

Table 5: Performance of LLaMA2-7B under the cross-cluster setting. For each cluster, the best and second-best reuse-method results are **bold** and underlined, respectively.

Tasks/Methods	Metric	LoRA	HiLoRA	HiLoRA-GS	HiLoRA-ROC	Retriever	LEGO	Arrow	Phatgoose	Ensemble	Merged
ANLI_r1	ACC	46.40	30.70	27.30	27.70	30.00	28.20	**31.30**	29.40	29.90	21.40
ANLI_r2	ACC	40.10	**34.50**	32.50	30.20	34.30	30.30	32.20	31.50	33.00	22.00
ANLI_r3	ACC	36.92	**31.67**	30.42	30.67	31.33	29.42	30.58	30.08	29.75	16.17
CB	ACC	80.00	70.00	62.00	**72.00**	66.00	60.00	52.00	64.00	64.00	4.00
MNLI	ACC	77.66	**50.74**	49.65	49.92	48.95	49.34	45.12	47.85	45.35	0.94
MNLI_mis	ACC	79.69	**51.29**	49.45	50.59	49.45	50.82	45.66	48.05	46.37	1.21
QNLI	ACC	77.27	46.84	42.62	46.56	42.23	41.41	**47.97**	44.80	47.19	8.59
RTE	ACC	72.96	62.22	57.41	52.96	**72.59**	50.74	58.15	55.93	56.30	39.26
SNLI	ACC	67.42	40.31	**41.09**	40.23	15.82	40.08	34.18	39.06	36.68	0.51
WNLI	ACC	72.86	47.14	**49.86**	49.14	47.14	48.57	45.71	47.14	47.14	2.86
Avg._NLI (Norm.)		65.13	**72.56**	68.90	69.58	68.49	66.76	66.62	68.26	68.12	22.21
ARC_C	ACC	40.43	**34.74**	30.78	30.43	34.31	32.33	30.00	32.76	31.90	0.43
ARC_E	ACC	40.17	**48.64**	44.28	45.89	47.50	46.74	41.57	46.06	45.42	0.55
Bool_Q	ACC	85.23	**78.36**	75.86	64.34	77.27	75.43	64.45	76.41	76.09	21.21
MultiRC	ACC	62.15	36.48	34.49	**40.82**	33.36	39.57	32.58	34.41	35.08	3.52
	F1	65.86	38.21	36.10	**42.99**	34.93	41.28	34.94	36.58	36.84	13.15
NaturalQuestions	ACC	18.95	12.03	9.38	5.90	8.59	13.40	9.88	**13.48**	11.80	0.35
	F1	29.82	20.39	18.45	11.33	17.13	**21.42**	17.82	21.37	19.65	9.13
OpenBookQA	ACC	58.20	45.60	44.60	45.40	38.60	**45.80**	43.00	42.80	45.20	0.20
RecoRD	ACC	92.87	69.81	63.51	66.45	**72.30**	68.19	66.26	65.18	62.78	35.42
SQuAD_v1	ACC	55.20	44.65	40.90	39.80	40.35	**45.78**	25.51	43.40	42.97	2.42
	F1	74.91	**64.25**	59.47	58.99	60.01	63.75	41.89	61.98	62.36	21.43
SQuAD_v2	ACC	64.34	26.45	24.57	24.41	19.26	**26.48**	14.77	24.06	25.78	0.35
	F1	73.80	36.03	34.01	35.56	30.42	**36.51**	23.35	33.99	35.16	9.06
TriviaQA	ACC	54.02	47.19	41.13	43.95	37.27	**49.18**	42.77	47.46	46.99	3.91
	F1	60.27	58.96	54.73	54.95	49.81	**59.14**	53.35	58.79	58.33	22.88
Avg._QA (Norm.)		59.66	**79.91**	73.60	72.12	73.33	79.61	67.16	77.20	76.44	14.68
Sentiment140	ACC	43.06	43.27	40.61	**44.49**	39.59	42.73	31.84	42.22	40.41	4.49
	F1	44.70	44.26	41.91	**45.51**	41.34	43.68	33.05	43.67	41.08	12.05
SST2	ACC	75.86	**63.10**	58.51	62.99	59.77	62.64	49.08	63.06	59.77	0.11
Avg._Senti. (Norm.)		59.87	91.46	85.58	**92.79**	85.50	90.52	69.32	90.50	85.82	9.50
ParaCrawl_EnEs	BLEU	29.05	27.18	**28.61**	18.15	15.25	19.31	27.01	26.11	25.34	11.12
WMT14_EnFr	BLEU	30.49	30.23	**30.95**	23.20	12.57	25.79	29.67	29.83	29.91	15.64
WMT16_CsEn	BLEU	19.67	18.56	**19.37**	12.55	7.52	14.10	18.26	18.04	18.29	11.76
WMT16_DeEn	BLEU	26.72	27.26	**27.40**	20.06	11.22	21.67	26.57	26.19	26.61	16.12
WMT16_FiEn	BLEU	14.58	14.63	**15.36**	10.31	5.40	11.20	14.58	14.23	14.50	9.04
WMT16_RoEn	BLEU	24.91	22.80	**22.87**	16.19	12.01	17.57	22.47	22.13	22.33	13.72
WMT16_RuEn	BLEU	22.27	17.84	**21.54**	13.33	10.08	16.82	20.99	19.46	21.31	12.50
WMT16_TrEn	BLEU	8.11	7.74	**8.33**	5.54	1.91	5.12	7.93	7.79	7.88	5.39
Avg._Trans. (Norm.)		21.98	94.56	**99.70**	67.67	41.00	73.73	95.60	93.42	94.98	56.17
COPA	ACC	72.00	65.00	66.00	66.00	**71.00**	68.00	59.00	62.00	63.00	20.00
HellaSwag	ACC	71.76	28.87	22.42	26.41	22.85	23.55	**28.98**	23.91	26.99	0.00
PIQA	ACC	61.75	**53.72**	50.98	48.58	53.22	47.70	51.75	52.90	51.64	1.80
StoryCloze	ACC	62.94	63.48	61.66	64.17	32.89	61.28	63.58	64.71	**66.47**	39.14
Avg._Common. (Norm.)		67.11	**79.59**	75.86	77.27	67.22	75.47	76.79	76.98	78.59	23.22
GLUE_MRPC	ACC	68.00	65.25	**65.75**	53.50	58.50	36.75	53.25	59.50	58.75	18.50
GLUE_QQP	ACC	76.13	64.80	67.66	53.44	**68.12**	58.71	63.09	56.80	64.73	2.62
STSB	ACC	34.82	17.83	**20.33**	16.78	19.78	15.95	17.27	16.78	16.43	0.21
PAWS_Wiki	ACC	88.55	64.45	46.68	47.19	**71.64**	48.24	46.76	56.17	56.33	9.10
Avg._Paraph. (Norm.)		66.88	76.27	74.17	62.59	**78.30**	57.86	65.90	68.43	70.56	10.38
CommonGen	ROUGE-1	54.60	43.27	**44.89**	43.08	43.03	24.14	39.70	38.95	37.43	35.44
	ROUGE-2	23.18	**14.05**	13.88	2.21	8.65	1.47	10.48	12.26	11.26	9.68
	ROUGE-L	47.82	**36.84**	36.30	28.31	32.51	21.70	32.57	33.79	32.47	27.93
	BLEU	11.95	6.74	**7.21**	0.27	3.90	0.08	4.20	5.60	5.08	3.71
DART	ROUGE-1	72.16	52.74	52.89	51.78	**53.03**	35.26	49.97	50.33	50.61	42.92
	ROUGE-2	47.78	25.66	25.48	23.79	24.49	16.03	26.05	26.08	**26.19**	21.06
	ROUGE-L	56.07	**39.76**	39.68	38.74	39.44	28.32	39.27	39.19	39.45	33.72
	BLEU	36.44	14.99	14.60	13.19	13.20	6.39	14.55	**15.74**	15.26	10.43
E2E_NLG	ROUGE-1	73.04	56.60	58.55	47.01	51.66	18.73	**59.00**	57.95	56.15	54.62
	ROUGE-2	44.84	29.79	29.29	21.20	27.65	9.24	**33.25**	32.76	31.78	29.99
	ROUGE-L	52.67	41.28	41.45	33.49	39.45	16.72	43.66	**43.81**	42.31	40.50
	BLEU	31.79	17.99	16.56	7.75	13.89	0.28	**21.17**	18.94	18.24	20.17
Gigaword	ROUGE-1	36.28	25.28	25.24	22.26	25.18	24.64	25.43	**26.50**	26.31	22.92
	ROUGE-2	16.07	8.58	8.57	7.46	8.31	8.27	8.64	9.50	**9.55**	7.76
	ROUGE-L	32.90	21.67	21.70	19.11	21.63	21.21	21.90	**22.90**	**22.90**	19.54
	BLEU	10.32	3.30	3.37	2.82	3.08	3.42	3.37	**4.09**	4.03	3.19
WebNLG_En	ROUGE-1	78.37	48.01	48.29	52.02	**53.28**	32.22	45.27	45.01	42.99	43.94
	ROUGE-2	56.19	24.95	25.53	25.93	**26.66**	15.57	24.09	24.60	22.90	23.76
	ROUGE-L	62.52	39.14	39.16	42.23	**42.80**	28.15	37.65	37.52	35.71	35.89
	BLEU	45.18	**15.60**	15.02	14.60	14.62	5.92	13.94	14.73	13.65	11.58
Avg._StT. (Norm.)		44.51	64.46	**64.66**	55.50	61.69	38.23	62.80	63.95	62.25	56.39
DPR	ACC	45.89	60.18	**61.07**	56.61	58.04	58.57	**61.07**	59.46	58.39	6.96
WSC	ACC	50.00	**63.00**	**63.00**	62.00	60.00	59.00	61.00	57.00	**63.00**	7.00
Avg._Corefe. (Norm.)		47.95	128.57	**129.54**	123.68	123.24	122.82	127.54	121.79	126.62	14.58
CoLA	ACC	62.24	55.49	55.30	53.18	54.82	**55.78**	**55.78**	55.20	55.30	5.20
FixPunct	ACC	34.69	**22.85**	**22.85**	17.93	21.09	15.94	21.13	21.09	21.17	6.60
TrueCase	ACC	67.27	14.61	**21.48**	6.09	2.50	0.39	11.13	14.45	13.32	7.23
Avg._Text_Corr. (Norm.)		54.73	58.91	**62.22**	48.73	50.86	45.38	55.69	56.99	56.56	12.71
WIC	ACC	57.78	49.68	49.68	49.68	50.63	49.37	**51.49**	50.32	48.95	0.16
Word_Segment	ACC	62.46	**25.70**	23.44	18.87	19.57	14.57	24.07	24.06	23.78	10.74
	F1	90.07	60.04	59.24	54.07	**66.07**	41.13	55.88	57.03	58.67	34.80
Avg._Word (Norm.)		67.02	71.10	70.09	66.90	**71.89**	60.98	70.76	70.13	69.39	15.07

Table 6: Performance of LLaMA2-13B under the cross-cluster setting. For each cluster, the best and second-best reuse-method results are **bold** and underlined, respectively.

Task	Metric	LoRA	HiLoRA	HiLoRA-GS	HiLoRA-ROC	Retriever	LEGO	Arrow	Phatgoose	Ensemble	Merged
ANLI-r1	ACC	60.30	41.00	38.50	30.80	37.10	32.30	**44.20**	36.80	42.80	0.00
ANLI-r2	ACC	47.30	**41.00**	36.80	29.80	35.80	31.00	40.80	38.20	39.70	0.10
ANLI-r3	ACC	49.92	40.17	37.50	32.42	37.25	31.92	**40.58**	37.42	40.33	0.00
CB	ACC	88.00	**82.00**	70.00	74.00	74.00	68.00	74.00	74.00	72.00	0.00
MNLI	ACC	87.97	**62.42**	60.94	40.74	60.98	45.66	61.80	55.82	62.19	0.04
MNLI-mis	ACC	89.80	62.93	61.37	39.92	61.91	46.02	63.01	56.33	**63.09**	0.00
QNLI	ACC	82.66	61.84	57.66	43.55	52.03	42.19	61.68	55.27	**63.95**	0.00
RTE	ACC	80.74	72.22	64.07	50.37	72.22	53.33	**73.70**	65.93	72.96	0.00
SNLI	ACC	81.91	53.59	51.21	30.94	35.82	39.30	**56.21**	53.75	56.09	0.00
WNLI	ACC	71.43	61.43	**67.14**	44.29	52.86	47.14	58.57	45.71	57.14	24.29
Avg._NLI (Norm.)		74.00	**78.50**	73.96	56.87	70.43	59.45	78.19	70.52	77.47	3.43
ARC-C	ACC	48.88	**49.48**	48.79	47.76	47.41	41.03	46.98	48.10	48.53	0.00
ARC-E	ACC	58.98	58.77	**58.90**	58.26	58.22	54.92	57.20	58.22	58.69	0.00
Bool-Q	ACC	89.34	**85.27**	84.84	84.80	82.34	84.45	83.71	83.09	83.36	28.09
MultiRC	ACC	67.93	43.44	43.71	43.75	**48.75**	40.62	40.66	35.55	39.88	0.00
	F1	71.58	45.19	45.17	45.22	**51.04**	42.47	43.33	38.51	42.64	12.25
NaturalQuestions	ACC	19.92	14.96	14.30	2.97	12.38	**17.11**	16.80	16.25	16.80	0.55
	F1	30.87	26.00	25.15	15.53	23.83	**27.78**	26.86	25.64	26.84	9.93
OpenBookQA	ACC	63.80	53.20	52.20	51.60	49.60	51.00	53.20	**54.20**	53.60	0.20
Record	ACC	95.19	80.21	80.58	80.60	**81.35**	77.41	76.40	75.90	78.17	36.88
SQuAD-v1	ACC	57.34	**49.49**	45.04	45.27	45.31	45.27	48.20	47.73	48.12	0.04
	F1	75.76	68.52	64.90	65.44	66.08	64.59	68.19	68.32	**68.56**	15.29
SQuAD-v2	ACC	70.78	**34.87**	31.99	31.95	29.61	28.20	32.11	31.52	31.56	0.00
	F1	80.71	**45.38**	42.82	42.81	41.31	38.89	42.63	41.95	42.36	5.66
TriviaQA	ACC	60.78	44.84	54.73	17.07	50.86	55.20	57.27	57.15	**57.38**	2.03
	F1	67.42	**70.04**	68.11	35.71	64.72	67.11	69.61	68.41	69.47	23.68
Avg._QA (Norm.)		65.77	**83.94**	83.03	73.09	81.46	80.30	83.24	82.40	83.87	13.52
Sentiment140	ACC	42.04	44.69	44.30	35.10	41.22	**44.90**	43.06	43.06	43.47	0.20
	F1	43.59	**46.09**	45.63	39.48	43.56	45.92	45.01	44.84	45.38	9.47
SST2	ACC	76.32	74.94	74.83	**75.52**	61.84	74.60	74.71	74.37	74.48	0.00
Avg._Senti. (Norm.)		59.57	**102.10**	101.53	93.02	90.02	101.90	100.37	100.05	100.67	5.65
COPA	ACC	72.00	**76.00**	75.00	67.00	72.00	68.00	74.00	72.00	74.00	3.00
HellaSwag	ACC	90.78	**45.90**	39.45	45.59	44.41	27.19	34.10	35.94	39.26	0.00
PIQA	ACC	66.72	55.52	56.34	56.07	57.60	51.75	55.68	54.92	**57.81**	0.77
StoryCloze	ACC	77.70	70.37	72.41	70.11	39.36	66.79	73.80	73.16	**74.01**	1.60
Avg._Common. (Norm.)		76.80	**82.47**	81.31	79.39	71.48	71.98	79.69	79.02	81.98	1.84
GLUE-MRPC	ACC	89.25	**71.75**	70.00	63.75	69.25	68.50	70.75	68.25	69.50	18.00
GLUE-QQP	ACC	84.88	75.86	**76.17**	45.47	69.02	55.62	57.89	43.95	51.33	6.41
STSB	ACC	44.99	16.09	18.38	8.70	**18.94**	14.62	17.06	16.71	17.62	0.00
PAWS-Wiki	ACC	93.79	**80.78**	79.18	62.42	68.67	53.95	59.45	51.41	57.15	10.08
Avg._Paraph. (Norm.)		78.23	72.91	**73.36**	52.72	68.56	58.07	62.19	55.05	59.61	9.62
CommonGen	ROUGE-1	55.16	44.13	46.71	34.99	**47.15**	34.02	43.99	41.94	44.24	33.74
	ROUGE-2	25.11	14.14	**15.38**	1.56	2.69	5.04	14.79	13.99	14.70	9.96
	ROUGE-L	48.78	36.58	**39.41**	23.50	31.35	26.87	37.47	36.53	37.77	27.09
	BLEU	13.05	7.82	**8.36**	0.51	0.24	1.78	7.24	6.52	7.11	2.68
DART	ROUGE-1	74.34	53.06	53.37	49.68	52.10	31.15	53.66	49.81	**54.22**	46.32
	ROUGE-2	50.87	26.48	26.46	22.47	23.73	14.91	29.11	26.69	**29.33**	24.21
	ROUGE-L	58.92	40.34	40.36	37.25	38.43	26.31	41.99	40.12	**42.14**	36.00
	BLEU	41.53	15.66	16.08	12.18	12.31	3.32	16.90	13.33	**17.37**	9.61
E2E-NLG	ROUGE-1	72.92	63.45	60.32	54.20	62.77	13.31	65.19	62.38	**65.40**	54.60
	ROUGE-2	45.55	34.42	31.70	24.28	33.41	5.38	**36.35**	34.37	**36.35**	29.90
	ROUGE-L	53.34	46.14	44.15	37.42	44.67	11.99	47.01	45.57	**47.06**	39.75
	BLEU	32.50	20.73	19.46	8.06	20.85	0.04	**21.26**	17.51	20.93	18.72
Gigaword	ROUGE-1	36.99	25.80	25.35	22.46	24.88	25.15	26.79	**28.27**	27.00	21.03
	ROUGE-2	16.38	8.82	8.50	7.29	8.17	8.40	9.67	**10.47**	9.61	6.85
	ROUGE-L	33.25	22.13	21.82	19.26	21.30	21.49	23.17	**24.77**	23.32	18.03
	BLEU	10.38	3.63	3.32	2.61	3.05	3.21	4.27	**4.99**	4.25	2.58
WebNLG-En	ROUGE-1	81.11	51.50	**54.92**	50.51	53.78	32.95	50.09	48.44	50.83	44.55
	ROUGE-2	60.23	27.70	**30.25**	25.08	26.62	16.71	28.79	28.38	28.99	24.45
	ROUGE-L	65.37	42.50	**44.49**	41.10	43.08	30.17	42.05	41.25	42.31	36.40
	BLEU	52.29	16.26	**18.92**	14.21	14.21	2.63	15.36	12.04	16.48	9.53
Avg._StT. (Norm.)		46.40	65.69	66.41	52.18	60.98	37.62	67.44	65.27	**67.84**	53.50
DPR	ACC	90.54	64.11	**64.46**	61.61	63.39	61.61	**64.46**	63.21	64.11	0.18
WSC	ACC	67.00	59.00	55.00	52.00	**60.00**	45.00	51.00	43.00	47.00	2.00
Avg._Corefe. (Norm.)		78.77	79.43	76.64	72.83	**79.78**	67.61	73.66	67.00	70.48	1.59
CoLA	ACC	69.08	**63.78**	62.52	53.76	58.19	55.78	63.49	63.01	63.20	0.00
FixPunct	ACC	45.20	22.66	22.70	12.23	**23.32**	16.21	22.27	22.11	23.16	1.37
TrueCase	ACC	72.85	34.34	29.41	6.76	10.74	5.16	31.99	32.89	**35.411**	1.21
Avg._Text_Corr. (Norm.)		62.38	63.20	60.37	38.05	50.19	41.23	61.70	61.76	**63.78**	1.56
WIC	ACC	73.65	**58.57**	55.87	54.60	52.86	51.43	55.00	57.14	58.41	0.16
Word-Segment	ACC	74.06	30.35	31.02	23.87	22.38	13.55	28.41	26.46	**32.55**	3.20
	F1	93.34	60.61	62.37	54.18	**65.42**	35.96	62.52	63.54	62.71	39.59
Avg._Word (Norm.)		78.68	66.93	65.82	60.38	62.11	49.70	64.50	65.67	**68.11**	12.89

Table 7: Performance of FLAN-T5-large under the cross-cluster setting. For each cluster, the best and second-best reuse-method results are **bold** and underlined, respectively.

Tasks/Methods	Metric	LoRA	HiLoRA	HiLoRA-GS	HiLoRA-ROC	Retriever	LEGO	Arrow	Phatgoose	Ensemble	Merged
ANLI_r1	ACC	60.20	57.90	59.20	54.00	60.60	**60.70**	60.50	60.00	60.60	60.60
ANLI_r2	ACC	43.30	42.70	42.20	41.40	42.00	42.80	**43.10**	42.90	43.00	42.70
ANLI_r3	ACC	44.50	43.42	**44.50**	41.83	43.83	44.00	44.00	44.25	43.92	44.00
CB	ACC	78.00	78.00	78.00	80.00	80.00	78.00	78.00	**82.00**	78.00	78.00
MNLI	ACC	88.59	81.76	82.50	**83.48**	83.40	58.24	56.52	79.49	59.10	50.78
MNLI_mis	ACC	89.14	82.66	83.40	83.87	**84.77**	55.62	50.20	79.10	56.37	48.20
QNLI	ACC	82.54	81.09	80.78	**82.38**	82.23	77.93	75.78	80.98	78.44	74.10
RTE	ACC	78.89	74.07	67.04	**74.81**	73.33	59.26	53.33	72.96	60.00	52.59
SNLI	ACC	60.08	36.13	10.35	**38.87**	10.27	3.79	1.64	23.40	3.83	1.52
WNLI	ACC	52.86	57.14	38.57	51.43	**60.00**	41.43	42.86	55.71	44.29	38.57
Avg._NLI (Norm.)		67.81	**93.78**	85.55	92.68	91.11	78.09	76.23	91.67	78.99	74.15
Bool_Q	ACC	87.27	79.22	76.09	**81.60**	80.70	74.73	73.05	80.04	75.23	70.39
MultiRC	ACC	55.31	**54.02**	53.48	52.42	53.20	52.97	53.59	**54.02**	53.24	52.77
	F1	58.71	**57.19**	56.76	55.64	56.53	56.30	56.94	57.12	56.53	56.11
SQuAD_v1	ACC	42.66	36.21	34.30	**36.33**	27.38	30.27	29.45	35.00	30.94	29.92
	F1	64.25	**59.92**	57.50	57.56	51.23	53.92	53.01	58.28	54.31	53.29
SQuAD_v2	ACC	66.56	**65.39**	64.26	64.34	62.97	63.09	62.46	64.77	63.09	61.91
	F1	77.13	**76.39**	75.36	75.16	74.24	74.27	73.67	75.78	74.53	73.30
Avg._QA (Norm.)		67.40	**94.22**	91.73	93.30	89.43	88.95	88.13	93.56	89.50	87.02
Sentiment140	ACC	42.04	**41.84**	41.63	41.63	41.63	41.63	41.63	41.43	41.63	41.63
	F1	43.18	**43.41**	43.10	43.21	43.09	43.01	43.10	42.82	43.02	43.02
SST2	ACC	75.75	74.48	73.91	**76.55**	73.10	73.91	73.56	74.14	73.68	73.56
Avg._Senti. (Norm.)		59.18	99.18	98.50	**100.31**	97.96	98.45	98.27	98.37	98.30	98.22
ParaCrawl_EnEs	BLEU	27.15	26.78	26.80	26.30	**26.93**	26.71	26.61	26.66	26.72	26.53
WMT16_RoEn	BLEU	20.81	20.79	20.93	20.51	20.83	20.87	**20.97**	20.79	20.88	20.88
WMT16_TrEn	BLEU	8.94	8.81	8.66	8.84	**8.86**	8.74	8.65	8.37	8.71	8.55
Avg._Trans. (Norm.)		18.97	99.03	98.72	98.10	**99.46**	98.81	98.51	97.24	98.73	97.90
GLUE_MRPC	ACC	89.00	81.75	**82.00**	76.50	77.00	80.75	80.25	81.50	81.00	81.25
GLUE_QQP	ACC	85.43	**83.38**	82.66	79.45	82.19	79.91	78.09	82.19	79.96	75.65
STSB	ACC	44.29	**41.99**	41.23	31.89	37.95	40.39	39.42	41.57	41.30	41.02
PAWS_Wiki	ACC	94.61	93.59	93.75	84.14	92.93	93.48	93.63	93.79	93.63	**93.95**
Avg._Paraph. (Norm.)		78.33	**95.80**	95.27	84.97	91.66	93.57	92.39	95.19	94.21	92.94
DART	ROUGE-1	76.03	75.82	75.75	**75.95**	75.85	75.77	75.79	75.66	75.72	75.60
	ROUGE-2	54.63	**54.39**	54.22	54.27	54.38	54.21	54.32	54.17	54.19	54.11
	ROUGE-L	61.99	61.76	61.52	61.61	**61.78**	61.62	61.66	61.50	61.48	61.43
	BLEU	46.56	46.66	46.73	45.78	46.57	46.65	**46.76**	46.53	46.68	46.72
E2E_NLG	ROUGE-1	73.08	73.48	73.27	**73.95**	73.55	73.23	73.09	73.29	73.19	73.14
	ROUGE-2	46.57	46.73	46.62	**46.92**	46.77	46.63	46.47	46.66	46.57	46.53
	ROUGE-L	54.16	54.21	54.04	**54.40**	54.25	54.15	53.96	54.12	54.04	53.97
	BLEU	35.27	35.43	35.37	35.02	**35.44**	35.33	35.12	35.38	35.28	35.23
WebNLG_En	ROUGE-1	83.34	82.50	82.71	81.77	82.61	82.70	**82.78**	82.49	82.75	82.71
	ROUGE-2	64.51	63.24	**63.39**	62.52	63.35	63.29	63.35	63.26	63.36	63.27
	ROUGE-L	69.18	68.12	68.19	67.77	**68.25**	68.04	67.98	68.02	68.03	67.96
	BLEU	56.89	55.83	56.18	53.07	55.72	55.98	56.26	56.11	56.16	**56.36**
Avg._StT. (Norm.)		60.18	99.53	99.48	98.90	**99.57**	99.43	99.40	99.38	99.40	99.34
DPR	ACC	86.25	76.79	76.25	**78.21**	76.07	75.89	75.89	77.14	76.07	75.36
WSC	ACC	40.00	51.00	47.00	**57.00**	48.00	46.00	46.00	47.00	48.00	46.00
Avg._Corefe. (Norm.)		63.13	108.27	102.95	**116.59**	104.10	101.49	101.49	103.47	104.10	101.19
CoLA	ACC	64.26	58.00	56.55	**59.25**	56.17	56.94	56.36	56.84	56.94	56.45
FixPunct	ACC	41.25	39.45	40.16	37.27	39.61	40.47	40.55	40.39	**40.62**	**40.62**
TrueCase	ACC	59.22	**67.03**	65.94	64.53	66.25	66.29	66.45	66.80	66.33	65.55
Avg._Text_Corr. (Norm.)		54.91	99.69	98.90	97.17	98.44	99.55	99.41	**99.72**	99.70	99.00
WIC	ACC	66.98	65.71	65.40	48.25	65.40	66.03	63.81	64.92	**66.19**	65.40
Word_Segment	ACC	63.91	67.15	67.34	66.72	67.07	70.62	70.20	**71.53**	70.55	70.35
	F1	88.33	90.83	90.74	92.82	90.52	92.76	92.56	**93.15**	92.72	92.47
Avg._Word (Norm.)		71.55	100.94	100.74	88.42	100.58	102.95	101.09	102.55	**103.03**	102.30

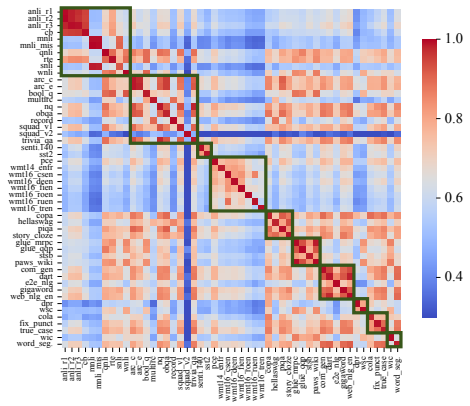


Figure 9: Input-LoRA similarity heatmap produced by `Retriever`, where tasks from the same cluster are enclosed within green boxes for clarity.

Table 8: Performance of HiLoRA under different values of the scaling factor γ across multiple task types.

γ	**20%**	**40%**	**60%**	**80%**	**100%**
Within-NLI	97.1	97.87	96.94	96.42	95.46
Cross-NLI	68.72	72.56	70.83	69.47	68.90
Cross-Trans.	89.02	94.56	96.92	98.67	99.69
Cross-StT	62.89	64.46	65.20	64.84	64.34

Table 9: Per-batch input-mapping and inference latency across tasks under varying pool sizes (different number of seen tasks).

latency (s)	Pool size	**ANLI_r1**	**ARC_C**	**Sentiment140**	**ParaCrawl**	**COPA**	**GLUE_MRPC**	**COMMON_GEN**	**DPR**	**COLA**	**WIC**
Mapping	5	0.0612	0.0596	0.0266	0.0578	0.0259	0.0490	0.0291	0.0286	0.0356	0.0445
	10	0.0597	0.0595	0.0440	0.0617	0.0225	0.0449	0.0304	0.0316	0.0391	0.0493
	15	0.0705	0.0576	0.0340	0.0637	0.0303	0.0465	0.0333	0.0451	0.0423	0.0481
	20	0.0807	0.0630	0.0494	0.0652	0.0396	0.0537	0.0398	0.0389	0.0474	0.0486
	25	0.0821	0.0749	0.0424	0.0664	0.0304	0.0525	0.0423	0.0417	0.0487	0.0683
	30	0.0886	0.0789	0.0490	0.0668	0.0336	0.0599	0.0378	0.0401	0.0431	0.0605
	35	0.0910	0.0759	0.0580	0.0627	0.0343	0.0672	0.0408	0.0466	0.0420	0.0673
	40	0.1026	0.0809	0.0566	0.0862	0.0390	0.0725	0.0453	0.0404	0.0411	0.0615
Inference	5	4.0765	2.2196	1.8765	14.5083	1.0900	0.5494	2.1211	1.2808	0.8343	0.8477
	10	10.0089	1.4993	1.7331	19.8633	1.0506	0.6800	2.1986	1.6111	0.8586	1.4085
	15	10.6356	1.5529	1.9547	21.3176	1.1875	0.4262	2.3208	1.5352	0.9120	1.0494
	20	7.1220	1.5537	1.9695	19.8872	1.1717	0.4234	2.1853	1.4684	0.9324	1.0237
	25	8.6094	1.5760	1.9752	21.2259	1.2098	0.4206	2.4156	1.4883	0.9311	1.1112
	30	9.7096	1.5021	1.9194	20.3829	1.1304	0.4111	2.1553	1.4695	0.9009	0.9758
	35	10.1859	1.7774	1.9127	20.1784	1.1404	0.3770	2.2162	1.4004	0.8978	0.9948
	40	10.8998	1.7849	1.8663	19.9678	1.2508	0.3633	2.1733	1.4737	0.8520	0.9701

Table 10: Performance comparison under different candidate-selection rules with LLaMA2-7B on NLI tasks under both within- and cross-cluster settings.

	Within-cluster			**Cross-cluster**		
Methods	**HiLoRA**	**Pos_topk**	**Topk**	**HiLoRA**	**Pos_topk**	**Topk**
ANLI-r1	45.00	41.70	46.45	30.70	30.20	31.00
ANLI-r2	40.60	38.90	39.80	34.50	34.60	34.30
ANLI-r3	37.67	36.42	38.33	31.67	32.17	31.63
CB	68.00	68.00	62.00	70.00	66.00	66.00
MNLI	76.33	74.41	75.79	50.74	48.05	47.72
MNLI-mis	78.59	75.78	77.99	51.29	49.18	48.97
QNLI	78.28	77.19	72.48	46.84	40.00	38.73
RTE	74.44	76.30	72.96	62.22	58.30	60.26
SNLI	69.45	69.45	69.93	40.31	39.38	38.89
WNLI	65.71	61.43	62.15	47.14	45.86	45.29
Avg (Norm.)	97.87	95.32	95.90	72.56	69.76	69.60

Table 11: Sensitivity to ε on within-cluster NLI with LLaMA2-7B, including seen and unseen tasks. Rank Spearman and allocation cosine are relative to $\varepsilon = 10^{-5}$.

ε	**LoRA Rank Spearman**	**Active LoRAs**	**Total ROC Budget**	**ROC Allocation Cosine**	**Avg. (Norm.)**
10^{-7}	0.974	2.171	9.043	0.843	93.15
10^{-6}	0.975	2.208	7.884	0.929	96.39
10^{-5}	1.000	2.034	7.159	1.000	97.87
10^{-4}	0.968	1.612	6.000	0.943	95.57
10^{-3}	0.938	4.688	16.000	0.709	88.75
10^{-2}	0.906	4.693	16.000	0.711	87.97

Table 12: Performance sensitivity to sample size and embedding models on NLI with LLaMA2-7B. Gray within-cluster rows denote unseen tasks.

	Within-cluster setting				Cross-cluster setting			
Factors	**5-sample**	**10-sample**	**20-sample**	**MPNet**	**5-sample**	**10-sample**	**20-sample**	**MPNet**
ANLI_r1	45.00	45.00	45.00	45.00	30.70	30.50	30.70	31.90
ANLI_r2	39.60	38.70	40.60	39.30	28.70	30.40	34.50	34.90
ANLI_r3	37.83	37.75	37.67	36.75	29.50	28.58	31.67	31.75
CB	66.00	66.00	68.00	68.00	74.00	76.00	70.00	74.00
MNLI	78.36	78.44	76.33	73.24	53.24	53.28	50.74	48.24
MNLI_mis	80.86	81.05	78.59	73.55	53.91	53.36	51.29	49.57
QNLI	69.06	68.20	78.28	65.80	44.38	43.48	46.84	44.77
RTE	71.85	71.85	74.44	73.33	57.78	58.15	62.22	61.85
SNLI	69.77	69.88	69.45	70.20	42.27	42.77	40.31	39.30
WNLI	60.00	64.29	65.71	61.43	48.57	48.57	47.14	45.71
Avg (Norm.)	95.68	95.96	97.87	94.02	71.23	71.56	72.56	72.23

C Empirical Validation of Theoretical Guarantees

To complement the theoretical analysis in Sec. 3, we empirically examine whether the assumptions required by Theorem 3.1 and Theorem 3.2 hold in practice, including both the separability between task domains and the validity of the Gaussian approximation used in the routing formulation. We further evaluate whether the resulting theoretical bounds behave as predicted.

Validation of Gaussian Assumption To examine the Gaussian assumption used in our theoretical analysis, we evaluate the embedding distributions associated with each LoRA. Specifically, we compute the skewness and kurtosis statistics for the embedding representations under two settings: (i) using all samples in each dataset, and (ii) using only the samples employed for Gaussian fitting. The summary statistics are reported in Tab. 13.

Table 13: Statistical properties of LoRA embedding distributions under Gaussianity analysis.

	Fitting Samples		**Full Dataset**	
	Skewness	**Kurtosis**	**Skewness**	**Kurtosis**
Mean	0.496	3.166	0.298	3.199
Std	0.114	0.332	0.158	0.338
Min	0.395	2.669	0.104	2.758
Max	0.970	4.530	0.920	5.052

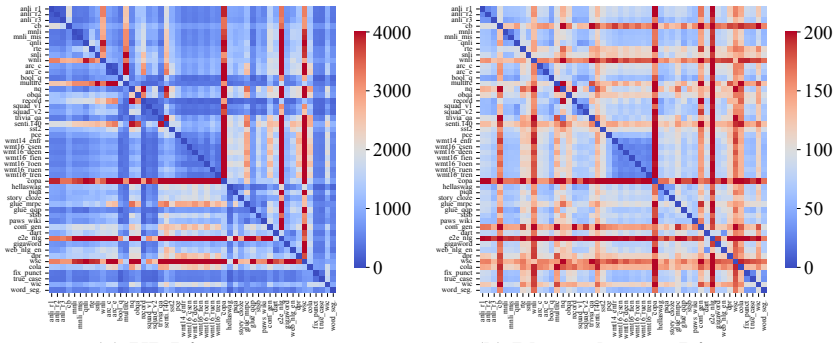
The results show moderate skewness values and kurtosis statistics close to those of a Gaussian distribution. In particular, the average skewness is 0.496 for the fitting samples and 0.298 for the full datasets, indicating relatively symmetric embedding distributions. Meanwhile, the average kurtosis values are 3.166 and 3.199, respectively, which are both close to the Gaussian reference value of 3. Although some tasks exhibit larger deviations, the overall statistics suggest that the embedding distributions are approximately Gaussian in practice. These observations support the use of Gaussian approximation in the proposed routing framework and theoretical analysis.

Domain Separability. We begin by evaluating the separability of task domains by computing the pairwise KL divergence between the Gaussian distributions fitted for each LoRA. As shown in Fig. 10(a), most task pairs exhibit large KL values, with an average divergence of 1432. These results indicate that the task domains are well separated in practice. This observation confirms that the key assumption required by Theorem 3.1 and Theorem 3.2 is satisfied, since greater inter-domain divergence corresponds to a lower probability of routing error.

Verification of Theorem 3.1. For Theorem 3.1, we analyze B_{ij}, the Bhattacharyya distance between Gaussian distributions of task pairs, which determines the exponential decay term in the error bound. As shown in Fig. 10(b), B_{ij} is strictly positive across all pairs and typically large, with an average value of 108.35. These results indicate that task domains are well separated in practice. Such substantial divergence ensures that the bound in Theorem 3.1 is operationally meaningful, as greater domain separability significantly reduces the probability that the correct LoRA is excluded from the top-κ set.

Verification of Theorem 3.2. To validate Theorem 3.2, we examine the feasibility condition $M_\alpha^j = \mathbf{\Sigma}_q^{-1} + \alpha\mathbf{\Sigma}_j^{-1} - \alpha\mathbf{\Sigma}_{i^\star}^{-1} \succ 0$. Since each covariance inverse ($\mathbf{\Sigma}^{-1}$) is positive definite, the matrix M_α^j remains positive definite when α is sufficiently small, because the sum of positive-definite matrices is positive-definite and the subtraction term is scaled down by α. Thus, from a theoretical standpoint, the condition is expected to hold with high probability for small and moderate α. We empirically verify this by evaluating the proportion of task pairs satisfying $M_\alpha^j \succ 0$ across different values of α. As shown in Fig. 11, the condition indeed holds with high probability when α is small or moderate, confirming that the assumptions required by our OOD error bound are realistic in our experimental setting.

Taken together, these results show that the assumptions under which our theoretical bounds become tight are frequently met in practice. The fitted task distributions are well separated, the key ID divergence term $B_{i,j}$ is sufficiently large, and the OOD conditions hold with high probability. These empirical findings validate that `HiLoRA`'s theoretical guarantees are not merely abstract but translate into reliable behavior in real-world applications.



(a) KL Divergence **(b) Bhattacharyya Distance**

Figure 10: (a) KL divergence and (b) Bhattacharyya distance computed across all task pairs used in our experiments.

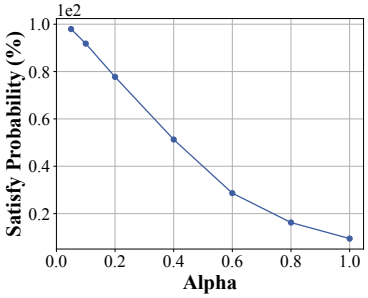


Figure 11: Satisfaction rate of the positive-definiteness condition $M_\alpha^j \succ 0$ in Theorem 3.2 across task pairs under varying α.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We clearly state all the contributions and support them with theoretical analysis and extensive experimental results.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of our work in the conclusion and propose future work directions to handle these limitations.

   Guidelines:

- The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: For all the proofs, we lay out all the assumptions needed and make them clear. We provide the complete proofs in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We disclose all the details to reproduce our results. The details can be found in Sec. 4 and the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We include a code repository link in the main paper and describe the evaluation datasets in Appendix B.1.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We disclose all the hyperparameter choices in Sec. 4 and the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report error bars for the inference throughput measurements in Fig. 6.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide sufficient information on the computer resources used for the experiments, including the type of compute workers (NVIDIA H100 GPUs), memory specifications, and execution time.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We comply with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Our work improves the efficiency and flexibility of LoRA-based adaptation, enabling more scalable deployment of AI systems. Potential negative impacts are similar to those of general-purpose language models, such as misuse in generating harmful content, and our method does not introduce new risks beyond these existing concerns.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

Justification: NA.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We make sure to cite the original sources of the models and datasets throughout the paper.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification: NA.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: NA.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: NA.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: NA.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.