

MAPLE: Masked Aadapter Prototype Learning for OOD Generalization

Adarsh Kappiyath¹, Abhra Chaudhuri², Jiaxi Li¹, Ajay Kumar Jaiswal³, Xilu Wang¹,
Li Shen⁴, Souvik Kundu⁵, Xiatian Zhu¹, Lu Yin^{6*}

¹University of Surrey, United Kingdom ²Fujitsu Research of Europe, United Kingdom

³Apple, United States ⁴Sun Yat-sen University, China

⁵Intel Labs, United States ⁶Shenzhen University of Advanced Technology, China

*Corresponding author: yinlu@suat-sz.edu.cn

Abstract

Parameter-efficient fine-tuning with adapters (e.g., LoRA) equips LLMs with task-specific skills. However, utilizing multiple pretrained adapters for out-of-distribution (OOD) generalization remains challenging. Existing techniques for OOD generalization using multiple pretrained LoRAs route inputs using LoRA representations (*prototypes*) obtained independently, assuming these representations capture complementary information. However, we observe that for existing methods, in-distribution and OOD routing entropies are often comparable, thus bringing the complementarity assumption into question. We derive the theoretical conditions that could lead to a violation of such assumptions, distilling the cause down to the presence of shared, noisy prototype subspaces. Based on this, we introduce **MAPLE (Masked-Adapter Prototype Learning)**, a simple learning framework that refines LoRA prototypes by masking the target task’s LoRA during prototype learning. In doing so, it encourages prototypes to discard noisy attributes, improving prototype discriminability, routing, and OOD generalization. Extensive experiments on language models of varying size, such as Phi-2 (2.7B) and Llama 3 (8B) equipped with heterogeneous pools of pretrained LoRAs, show that MAPLE improves the LoRA representation and thus achieves state-of-the-art performance across multiple benchmarks.

1 Introduction

Large Language Models (LLMs) have emerged as versatile, general-purpose tools with a wide range of practical applications. However, their performance can vary substantially depending on the task under consideration (Moayeri et al., 2025). To mitigate this variability, lightweight fine-tuning techniques, such as LoRA (Hu et al., 2022), have gained traction, enabling the efficient adaptation of LLMs to diverse downstream tasks when task-specific

training data is available. With the growing availability of pretrained LoRA adapters, recent work studies a mixture-of-experts approach, *MoErging* (Yadav et al., 2025), which aims to generalize to out-of-distribution (OOD) inputs by dynamically combining adapters from a pretrained pool. Central to these methods are *prototypes*: representations of LoRAs which are used to compute routing weights for each input. Prior work constructs prototypes via heuristics (Muqeeth et al., 2024; Chronopoulou et al., 2023; Cheng et al., 2024; Durbin, 2024; Maxine, 2023) or learning-based objectives (Diao et al., 2023; Ostapenko et al., 2024), with the goal that prototype similarity reflects an adapter’s functional suitability for a given input. With appropriately determined routing weights, knowledge from pretrained LoRAs can dynamically augment the base model weights to generalize to OOD inputs.

Although existing prototypes have been shown to facilitate effective routing, they are constructed in isolation, assuming optimality under joint use. To probe this assumption, we compare the entropy of routing coefficients on ID versus OOD tasks. Ideally, well-discriminative LoRA representations should yield lower routing entropy for ID tasks than OOD tasks. However, we empirically observe comparable entropies across both settings (see Fig. 1), indicating that existing prototypes fail to identify relevant LoRAs effectively under joint use. From a representation learning perspective, this suggests that LoRA representations encode non-discriminative/noisy attributes. Under a relevance-coverage assumption, Theorem 2 shows that more restrictive subspaces are noisy with respect to a larger fraction of distributions. This is rooted in Theorem 1, where we show that low ID-OOD entropy gaps imply representations restricting the task subspace spanned by the adapters, degrading OOD generalization. Finally, we empirically observe that MAPLE increases the separation between ID and OOD routing entropies, which is

associated with improved OOD performance.

To enhance LoRA representations and thereby improve OOD generalization capabilities, we introduce a novel prototype refinement scheme: **MAPLE** (*M*asked *A*dapter *P*rototype *L*earning), a training framework that improves prototypes’ discriminability by simulating OOD conditions. During prototype refinement, MAPLE *masks the target task’s adapter*, forcing routing to rely on informative, task-specific cues rather than generic shared/noisy features. This simple intervention yields prototypes with improved discriminability and produces lower routing entropy on ID inputs. We empirically validate the effectiveness of the proposed method on two widely used LLMs, Phi-2 (2.7B)([Jawaheripi and Bubeck, 2023](#)) and Llama 3 (8B)([Grattafiori and Dubey, 2024](#)).

In summary, we make the following contributions. (1) We identify and formalize the inadequacy of LoRA prototypes learned in isolation for MoErging. We show that such prototypes exhibit dominant, noisy subspaces that influence routing plans irrespective of the input distribution (leading to similar in-distribution and OOD routing entropies), resulting in poor generalization. (2) We propose MAPLE, a masked-adapter prototype learning framework that enhances prototypes by discarding noisy attributes and thereby improving OOD generalisation. (3) Through extensive experiments and ablations across heterogeneous benchmarks, we demonstrate consistent improvements over state-of-the-art routing baselines on LLMs of varying sizes with diverse LoRA collections.

2 Related Work

MoErging techniques can be broadly classified by how they compute routing coefficients: *embedding-based*, *optimization-based*, and *learning-based*. **Embedding-based methods** build fixed expert representations and route by similarity to the input. Most generate task-level embeddings by averaging pretrained backbone features over task data, e.g., Token-Level Adaptation ([Belofsky, 2023](#)), LoRARetriever ([Zhao et al., 2024](#)), Mo’LoRA ([Maxine, 2023](#)), and DAM ([Cheng et al., 2024](#)). While simple, such averages encode generic pretrained factors, yielding weak separation among related adapters. [Ostapenko et al. \(2024\)](#) instead derive embeddings from LoRA weights via SVD, but because these are constructed *per expert* in isolation, they fail to capture inter-expert relation-

ships, reducing discriminability when experts are combined. MAPLE addresses these limitations by learning embeddings under simulated OOD conditions that suppress shared generic components and enhance cross-expert separability. **Optimization-based methods** such as LoRAHub ([Huang et al., 2024](#)), MoLE ([Wu et al., 2024](#)), and Lora-Flow ([Wang et al., 2024](#)) assume access to target-task data and directly optimize routing coefficients via gradient-based or gradient-free techniques. While these can capture inter-expert relationships, they maximize performance on a single known task rather than producing generalizable representations. MAPLE instead learns expert embeddings that generalize zero-shot to unseen tasks, pursuing a largely orthogonal objective. **Learning-based methods** train a router to minimize loss on in-distribution data. PHATGOOSE ([Muqeeth et al., 2024](#)) and Co-LLM ([Shen et al., 2024](#)) learn per-expert sigmoid gates for token-level routing, while MeteorA ([Xu et al., 2025](#)) optimizes the router on expert training data for automatic task switching. Though effective in-distribution, these methods are not explicitly designed for OOD generalization. MAPLE complements all three paradigms by learning embeddings tailored to discriminative routing under OOD conditions.

3 Methodology

In this section, we first examine the failure modes of two representative routing methods using *routing entropy* as a diagnostic metric. We then provide a theoretical analysis to elucidate the underlying causes of these behaviors. Building on these insights, we introduce a simple yet effective training paradigm, **MAPLE**, designed to improve the discriminability and adaptability of LoRA representations for OOD generalization.

3.1 Motivation

Problem Setting: We consider a large language model (LLM) equipped with a collection of M pretrained LoRA adapters, each fine-tuned on a distinct task from a training task set. The goal is to exploit this pool of adapters to achieve improved out-of-distribution (OOD) generalization on tasks unseen during adapter training. Specifically, given an input, the model must route among an appropriate subset of LoRAs from the pretrained pool such that their combined adaptation yields robust performance on OOD inputs. This formulation frames

OOD generalization as a routing problem over a fixed library of specialized, lightweight LoRAs. Samples from the tasks used to train these LoRAs constitute the in-distribution (ID) samples, while samples from any other task are treated as OOD.

As discussed in section 2, most existing routing schemes construct prototypes independently, which are then used jointly during inference. We investigate whether such prototypes that are learned in isolation remain discriminative under joint use. We assess the discriminability of LoRA prototypes via *routing entropy*, an information-theoretic measure of the router’s uncertainty over experts. The choice of this metric is motivated by its strong correlation with prediction loss as well as its use as a proxy objective for test-time estimation of routing coefficients (Yang et al., 2024). Here, e denotes an expert/LoRA adapter in the pool, x represents an input token, and M is the total number of pre-trained experts/LoRAs. Given the routing distribution $p(e | x)$ over M experts, we define the routing entropy as:

$$H(x) = - \sum_{e=1}^M p(e | x) \log p(e | x). \quad (1)$$

We compute $H(x)$ at the token level and report the average over tokens and samples. Lower routing entropy indicates a concentrated probability mass and, hence, a more decisive expert selection, while higher routing entropy reflects greater uncertainty in selecting an expert for a token. Ideally, for in-distribution (ID) tasks (*i.e.*, LoRA trained on the input task exists in the LoRA pool), the routing entropy should be lower. While for out-of-distribution (OOD) tasks, the routing entropy is expected to be higher, reflecting greater uncertainty. Routing entropy is computed for both ID and OOD tasks using prototypes obtained from high-performing embedding and learning-based methods. As shown in figure 1, across five ID/OOD tasks, the routing entropy estimated with Arrow (Ostapenko et al., 2024) and PHATGOOSE (Muqeeth et al., 2024) prototypes does not differ appreciably between ID and OOD samples. The overlapping routing-entropy distribution between ID and OOD samples suggests that these representations are insufficient to identify relevant LoRAs when used jointly effectively. This could be due to representations encoding certain noisy attributes, which blur inter-adapter distinctions and thus yield high routing uncertainty.

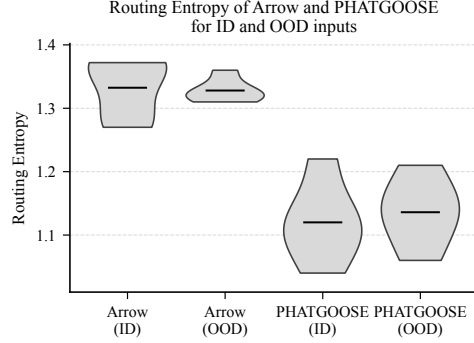


Figure 1: Distribution of routing entropy for Arrow and PHATGOOSE across five ID/OOD task pairs. ID and OOD distributions substantially overlap, indicating limited discriminability of prototypes under joint use. Routing entropy values are computed per layer, token, and sample using Eq. (1), and then averaged to obtain a single value per task. The distributions are visualized as violin plots.

3.2 Theoretical Formulation

Overview: We view routing as an embedding of the input into a vector space, of which the constituent bases are the LoRA prototype vectors. We show, via Theorem 1, that when the effective rank of this vector space is low, the model is forced to use similar routing plans for both ID and OOD samples, hindering OOD-specific adaptation. Under a relevance-coverage assumption, Theorem 2 then shows that more restrictive subspaces are noisy with respect to a larger fraction of distributions, for which we provide direct empirical evidence.

We start by formalizing our argument that for a set of experts to generalize OOD, it must be accompanied by an increased routing entropy, relative to ID samples, when applied to OOD samples. We emphasize that the following analysis considers the router, which is a linear function (dot product over learnable LoRA prototypes), and utilizes its linearity property throughout. We do not make any assumption that LoRA adapters induce a purely linear transformation of the network’s behavior. We then derive the conditions under which ID and OOD routing entropies might be similar, as observed by us in the default setting. As a remedy, we show that the desired ID-OOD entropy discrepancy can be achieved via a denoising mechanism that removes a specific type of subspace from the basis induced by the LoRA prototypes. To this end, we prove the following key results:

1. Theorem 1, where we introduce the idea of δ -

restrictivity, which implies that higher the difference in the ID and OOD routing entropies, *i.e.*, low ID and high OOD routing entropies, the better the OOD generalization.

2. Under a relevance-coverage assumption, Theorem 2 shows that more restrictive subspaces are noisy with respect to a larger fraction of distributions.

Therefore, better OOD generalization can be achieved by denoising the basis induced by the LoRA prototypes through the removal of irrelevant subspaces η such that $\|\eta x\|$ is low on average across the space of all distributions. We defer all proofs to Appendix A.

Definition 1 (Completeness). We call a function class of routers $\mathcal{F}$ acting on a data distribution $\mathcal{D}$ to be equipped with a complete basis $\mathcal{B}$ (corresponding to the set of LoRA prototypes) *iff*:

$$\begin{aligned} \forall (x, y) \in \mathcal{D}, \exists f \in \mathcal{F}, f(x) \in \text{span}(\mathcal{B}) \\ | g_{\theta+\phi(f(x))}(x) = y, \end{aligned}$$

i.e., $\mathcal{B}$ corresponds to the feature space under which the labels y of all points $x \in \mathcal{D}$ are predictable, where g is a non-linear model parameterized by weights θ , LoRA parameters ϕ and router $f(x)$.

Out-of-Distribution Risk: Consider the space of data distributions $\mathcal{D} = \{S\} \cup \mathcal{T}$, where $\mathcal{T}$ is arbitrarily discretized as $\mathcal{T} = \{T_1, T_2, \dots, T_t\}$. The risk of a predictor f_S trained on the source distribution S to generalise out-of-distribution to the target space $\mathcal{T}$ is given as follows:

$$R_{\mathcal{T}}(f_S) = \frac{1}{|\mathcal{T}|} \sum_{T \in \mathcal{T}} \mathbb{E}_{(X,Y) \sim T} [\mathcal{L}(f_S(X), Y)].$$

where $(X, Y) \in T$ are samples in a given target domain $T \in \mathcal{T}$.

Routing Plan: The routing plan corresponding to a router f with an associated basis $\mathcal{B}$ is a probability distribution $P_{\mathcal{B}}(X)$ over the basis vectors in $\mathcal{B}$ such that the mass on the i -th basis vector w_i is given by:

$$P_{\mathcal{B}}(X)\{w_i\} = \int_{x \in X} \frac{\exp(w_i^{\top} x)}{\sum_{j=1}^n \exp(w_j^{\top} x)} p(x) dx,$$

where X is some input distribution and $p(\cdot)$ is its probability density function.

3.2.1 Convergence States

Below, we enumerate the states to which the optimization over $\mathcal{F}$ might converge in terms of the relationship of the basis associated to the solution f with the input distributions.

(State 1) $R_S(f) > K, R_{\mathcal{T}}(f) > K; R_{\mathcal{T}}(f) - R_S(f) < \epsilon$: High source and target risks, all basis vectors are irrelevant. Corresponds to a high entropy routing plan for both ID and OOD samples.

(State 2) $R_S(f) < k, R_{\mathcal{T}}(f) > K; R_{\mathcal{T}}(f) - R_S(f) > \gamma$: Low source but high target risk, basis vectors are aligned to the source distribution. Routing plan has low entropy for the source but high entropy for the target.

(State 3) $R_S(f) > K, R_{\mathcal{T}}(f) < k; R_S(f) - R_{\mathcal{T}}(f) > \gamma$: High source but low target risk, highly unlikely, not possible with ERM, only possible if f is sampled randomly from the associated hypothesis class. Routing plan has high entropy for the source but low entropy for the target.

(State 4) $R_S(f) < k, R_{\mathcal{T}}(f) < k; R_{\mathcal{T}}(f) - R_S(f) < \epsilon$: Low source and target risks, relevant orthogonal basis, best OOD generalization among all. The routing plan has low entropy for the source but high entropy for the target.

R_S above is simply the source/training distribution risk. Also, K, k, γ, ϵ are constants such that $K > k$ and $\gamma > \epsilon$. Given a model with high source and target risk and a high-entropy routing plan, *i.e.*, state 1, the objective is to transition to a basis in state 4. However, note that the same ratios of relative entropy characterize the routing plans for both state 2 and state 4 and are only distinguishable by relative risk, the latter being determined by the level of noise in the solution. Thus, although ensuring that the routing plan induces a low source and high target entropy is associated with more discriminative routing in our formulation, the role of noisy subspaces requires the relevance-coverage condition formalized in Assumption 2. In the following section, we expound the scenario where despite the associated function class $\mathcal{F}$ being equipped with a complete basis (Definition 1), certain members of $\mathcal{F}$ might get restricted to a feature domain where a noisy subspace dominates, the elimination of which could lead to better generalization.

3.2.2 Pseudo-Incompleteness

Assumption 1 (Rank-entropy ordering). For the family of prototype bases considered in our analysis, let B_1 and B_2 be two admissible bases. We

assume that

$$\text{erank}(B_1) > \text{erank}(B_2)$$

implies

$$\begin{aligned} H(P_{B_1}(X^o)) &\geq H(P_{B_2}(X^o)), \\ H(P_{B_1}(X^i)) &\leq H(P_{B_2}(X^i)), \end{aligned}$$

with at least one inequality strict. This assumption captures the regime of interest in which higher effective rank is associated with increased ID–OOD routing discrimination. We empirically examine this behavior in Appendix A.1.

Theorem 1 (Restrictivity). *Let $\mathcal{B}$ be a matrix of basis vectors associated with router f . Let η be a subspace that is removed from $\mathcal{B}$, resulting in $\mathcal{B}' = \mathcal{B} - \eta$. Let X^i and X^o respectively be source (ID) and target (OOD) distributions, and assume $H(P_{\mathcal{B}}(X^o)) - H(P_{\mathcal{B}}(X^i)) > 0$. Let the restrictivity ratio of η be defined by:*

$$r(\eta) = \frac{H(P_{\mathcal{B}'}(X^o)) - H(P_{\mathcal{B}'}(X^i))}{H(P_{\mathcal{B}}(X^o)) - H(P_{\mathcal{B}}(X^i))} \leq \delta,$$

where $P_{\mathcal{B}}(X)$ and $P_{\mathcal{B}'}(X)$ represent the corresponding routing plans, $H(\cdot)$ is their entropy, and δ is some constant, such that the subspace η is called δ -restrictive, i.e., the higher the value of δ , the more restrictive the effect of η on $\mathcal{B}$. Then, for subspaces $\eta_1, \eta_2 \in \mathcal{B}$, the following holds:

$$\begin{aligned} \text{erank}(\mathcal{B} - \eta_1) &> \text{erank}(\mathcal{B} - \eta_2) \\ \implies r(\eta_1) &> r(\eta_2), \end{aligned}$$

where erank denotes the effective rank (Roy and Vetterli, 2007), i.e., the higher the effective rank of $\mathcal{B} - \eta$, the more restrictive the subspace η .

As a result, the presence of subspaces with high δ in $\mathcal{B}$ renders the latter pseudo-incomplete for the associated $f \in \mathcal{F}$.

If ID and OOD routing distributions are roughly similar, it implies the existence of a dominant subspace that is equally similar or dissimilar to both ID and OOD samples. If the subspace is strongly relevant to both ID and OOD samples, then its removal should mean that $\mathcal{B} - \eta$ would contain irrelevant basis vectors for both ID and OOD samples, leading to high entropies that are still roughly similar across both ID and OOD. In this case $X^o \cdot \eta \approx X^i \cdot \eta > \gamma$, i.e., η is very similar to both ID and OOD samples. On the other hand, and upon the removal of η , for all ID assignments, if the ID distributions lose

entropy, then it implies that $\mathcal{B} \setminus \eta$ is closer to the canonical basis of distributions than $\mathcal{B}$, implying that η was noisy. In this case $X^o \cdot \eta \approx X^i \cdot \eta < \epsilon$, where $\gamma \gg \epsilon$, i.e., η is highly dissimilar to both ID and OOD samples.

Assumption 2 (Relevance–coverage ordering). For a distribution $D \in \mathcal{D}$, define the relevance of a subspace η as

$$s_D(\eta) = \left| \frac{1}{|D|} \int_{x \in D} x^\top \eta \, dx \right|.$$

For constants $0 \leq \epsilon < \gamma$, let

$$\begin{aligned} \mathcal{R}_\gamma(\eta) &= \{D \in \mathcal{D} : s_D(\eta) > \gamma\}, \\ \mathcal{N}_\epsilon(\eta) &= \{D \in \mathcal{D} : s_D(\eta) < \epsilon\}. \end{aligned}$$

The former contains the distributions for which η is relevant, and the latter those for which η is noisy. We assume that the considered regime has a relevance gap, such that every $D \in \mathcal{D}$ belongs to exactly one of these sets. Writing $\delta(\eta)$ for the restrictivity coefficient of η , we further assume that more restrictive subspaces have no broader relevance coverage:

$$\delta(\eta_1) > \delta(\eta_2) \implies \mathcal{R}_\gamma(\eta_1) \subseteq \mathcal{R}_\gamma(\eta_2).$$

Theorem 2 (Noisiness). *Under Assumption 2, if $\delta(\eta_1) > \delta(\eta_2)$, then*

$$N(\eta_1) \geq N(\eta_2), \quad N(\eta) = \frac{|\mathcal{N}_\epsilon(\eta)|}{|\mathcal{D}|}.$$

If the relevance-set inclusion is strict, then $N(\eta_1) > N(\eta_2)$.

Theorem 1 shows that comparable ID and OOD entropies can arise from a dominant subspace that is similarly aligned with both ID and OOD samples. Under Assumption 2, Theorem 2 shows that more restrictive subspaces are noisy with respect to a larger fraction of distributions. In the next section, we present a simple method to remove such noisy components and thereby improve OOD generalization.

3.3 MAPLE: Masked Adapter Prototype Learning

Theorem 1 characterizes the restrictive effect of prototype subspaces on routing distributions, while Theorem 2, conditional on Assumption 2, relates greater restrictivity to noisiness across a larger fraction of distributions. If we could attenuate these

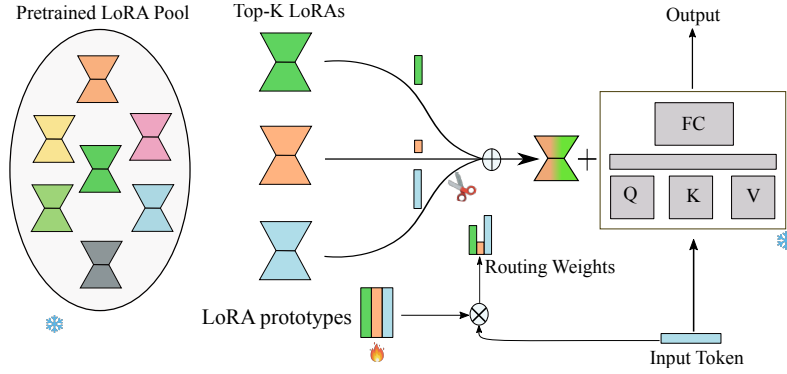


Figure 2: **MAPLE overview:** MAPLE improves OOD generalization of a pretrained LLM using a library of pretrained LoRA adapters. For each input, per-layer routing weights are computed as dot products between hidden states and learned prototypes. From the top- k candidates, the LoRA associated with the input’s source domain is masked out (if present). The remaining LoRAs are linearly combined according to the routing weights and used to augment the base model.

noisy components within the LoRA prototypes, routing would improve, and so would OOD performance. A natural idea is to jointly train all LoRA prototypes so that noisy components contribute to higher loss and are therefore eliminated; in practice, however, joint training yields little to no improvement as shown in Table 2. We hypothesize that this failure stems from including the target-task LoRA in the routing during training. Here, the “target-task LoRA” refers to the LoRA adapter that was trained on the same task as the current training sample, *i.e.*, the adapter specifically associated with the data point’s source domain. Because the target LoRA alone can substantially reduce the loss, causing gradients that would otherwise pressure non-target prototypes to unlearn their noise to become very small. To address this and to facilitate efficient removal of noisy attributes from LoRA prototypes, we propose **Masked Adapter Prototype Learning (MAPLE)**, a training paradigm that explicitly excludes the target-task LoRA from the forward pass during training. By masking the target adapter, the model must route through the remaining prototypes, increasing the training loss and producing stronger, more informative gradients on those prototypes. This promotes the removal of spurious, task-specific noise from their representations, leading to cleaner prototypes, more reliable routing, and improved OOD generalisation. For each input, MAPLE computes routing scores in every layer, selects the top- k candidate LoRAs, and explicitly *masks* the adapter trained on the input’s target task. The remaining adapters are then combined using their routing coefficients to augment the base LLM. This procedure is applied layer-wise, and training

proceeds with a causal language modeling (next-token prediction) objective. A schematic of the MAPLE training paradigm is shown in figure 2 and a detailed pseudocode is given in algorithm 1. At evaluation time, all adapters are used for computing routing coefficients, and routing is performed on top- k LoRAs.

4 Experimental Setup

Models, Data and LoRA Pools: We use **Phi-2** (2.7B; [Jawaheripi and Bubeck 2023](#)) with the 256-adapter Flan-v2 pool from [Ostapenko et al. \(2024\)](#), and **Llama 3** (8B; [Grattafiori and Dubey 2024](#)) with the 28-adapter pool from [Xu et al. \(2025\)](#). We reuse pretrained checkpoints where available and otherwise train adapters with the original hyperparameters. During MAPLE training, only the LoRA prototypes are updated, so all improvements stem from better routing. Additionally, for larger adapter pools, such as the 256-adapter pool used with Phi-2, we require only a small amount of task-specific data (~ 200 samples per task). This high data efficiency is possible due to the large diversity of LoRA, resulting in diverse gradients, which makes it easier to eliminate noisy attributes. Additional implementation details are specified in Appendix D.

ID and OOD Definitions: In our experiments, the LoRA pool is constructed by training LoRAs on 256 tasks from FlanV2 datasets (for Phi-2); any samples from these tasks are considered as in-distribution (ID) samples. For evaluation, we consider datasets such as ARC-Easy, ARC-Challenge, HumanEval, etc., whose tasks are not part of the LoRA training set. Samples from these evaluation tasks are treated as OOD.

Model	Method	ARC-C	ARC-E	BBH	BoolQ	HSWAG	HE	MBPP	OQA	PIQA	WG	Mean
Phi-2 (2.7B)	Base	52.90	77.50	48.00	82.70	72.50	45.10	56.00	49.80	79.20	75.70	63.94
	Shared	55.80	83.20	48.40	82.40	73.40	46.30	58.40	50.40	80.40	76.60	65.53
	Uniform	53.90	78.70	47.90	83.30	73.10	44.50	59.90	48.60	79.40	75.20	64.46
	Arrow	56.40	84.80	47.90	83.90	72.80	45.70	60.30	51.40	80.20	76.00	65.94
	Arrow + MAPLE	56.70	84.50	48.90	83.60	73.10	48.80	59.50	51.60	80.50	76.90	66.41
	PHATGOOSE	56.70	84.20	48.70	83.90	73.10	48.80	58.00	52.20	80.60	76.40	66.26
	PHATGOOSE + MAPLE	56.80	84.20	47.00	83.40	73.30	50.00	61.10	52.40	80.60	77.60	66.64
Llama 3 (8B)	Base	51.80	73.30	48.20	68.40	78.10	36.60	52.90	41.60	80.40	73.00	60.43
	Shared	54.30	82.70	51.30	81.20	79.30	40.20	55.90	45.30	80.30	73.50	64.40
	Uniform	53.90	78.20	48.90	71.10	79.10	38.40	55.30	42.60	80.50	73.40	62.14
	Arrow	56.40	84.00	52.70	82.90	80.10	39.60	55.30	43.20	81.10	74.40	64.97
	Arrow + MAPLE	57.40	83.20	52.30	83.60	79.50	41.50	56.80	47.60	81.30	74.20	65.74
	PHATGOOSE	57.60	83.30	52.30	83.60	79.50	42.70	56.40	47.60	81.20	74.00	65.82
	PHATGOOSE + MAPLE	58.50	84.80	53.10	84.60	79.40	42.10	57.20	46.80	80.70	75.00	66.22

Table 1: Zero-shot accuracy (%) of **Phi-2 (2.7B)** and **Llama 3 (8B)** across 10 benchmarks under different routing strategies in comparison with MAPLE.

Baselines: To evaluate the effectiveness of MAPLE in eliminating generic (noisy) factors encapsulated in prototypes and improving routing, we refine LoRA prototypes obtained using two popular and representative routing methods: ARROW (embedding-based; Ostapenko et al., 2024) and PHATGOOSE (learning-based; Muqeeth et al., 2024). We also report results of **Base LLM** (no adapters), **Uniform** (equal routing weights over all K adapters, i.e., $p_k = 1/K$), and **Shared** (a single multi-task LoRA jointly fine-tuned on all training tasks) for comparison.

Benchmarks: We assess effectiveness on ten widely used tasks spanning four categories: (1) *Common-sense reasoning*—WinoGrande (Sakaguchi et al., 2019), HellaSwag (Zellers et al., 2019), PIQA (Bisk et al., 2020); (2) *Question answering*—BoolQ (Clark et al., 2019), OpenBookQA (Mihaylov et al., 2018), ARC-Easy and ARC-Challenge (Clark et al., 2018); (3) *Coding*—HumanEval (Chen et al., 2021), MBPP (Austin et al., 2021); (4) *General-purpose reasoning*—BBH (Suzgun et al., 2023). We report zero-shot accuracy on these benchmarks, along with average accuracy.

4.1 Results

We empirically validate the effectiveness of MAPLE and compare it against various baselines with results summarized in table 1. As observed in table 1, applying MAPLE improves the prototype representations obtained by state-of-the-art

routing methods (Arrow and PHATGOOSE), reflected by higher average zero-shot accuracy across various benchmarks. Because training updates only the LoRA prototypes, these gains arise from the improvement of the LoRA prototypes. The magnitude of improvement varies by model and routing scheme, likely reflecting different noise levels in prototypes learned by each method. The $\sim 0.5\%$ gain from MAPLE’s noisy-attribute elimination matches the $\sim 0.6\%$ gap between strong baselines (Arrow vs. PHATGOOSE), suggesting that removing noisy attributes alone yields a meaningful improvement. Furthermore, improvements hold for both a small model (Phi-2) and a larger model (Llama 3), indicating robustness across parameter scales. Different benchmarks exhibit varying degrees of improvement, with the highest performance gain (4.4%) observed for OpenBookQA with Arrow prototypes obtained on Llama 3. This variability indicates that different OOD distributions affect noisy attributes to different degrees.

4.2 Analysis and Ablations

In this section, we validate the design choices behind MAPLE and also re-examine the difference in routing entropy using refined prototypes for ID and OOD tasks, which was the primary indicator of inadequacy of LoRA prototypes.

4.2.1 Effect of masking

Masking was introduced in MAPLE to enhance the elimination of noisy attributes embedded in the LoRA prototypes. To validate that this mask-

	Avg. Accuracy (%)
Arrow	65.94
MAPLE (Without masking)	66.06
MAPLE	66.41

Table 2: Ablation on the effect of masking during MAPLE training. Masking yields a +0.41% improvement, validating our design choice and its significance in prototype refinement.

ing is necessary for the elimination of noisy attributes, we perform an ablation study wherein we disable masking (i.e., the target-task LoRA participates in routing and loss minimisation), which is compared against the case in which target-task LoRA is masked during training (MAPLE). We also report the results obtained via routing using the prototypes before any training for reference. The result of this ablation is summarised in table 2.

In the absence of masking, zero-shot accuracy shows only a marginal improvement of 0.12% with respect to initialization (Arrow), while enabling masking of the target task LoRA resulted in an improvement of 0.41%, validating the design choice of MAPLE as well as providing supportive evidence for our initial hypothesis of gradient suppression on inclusion of target-task LoRA. We provide a detailed explanation of why masking leads to denoising in Appendix B.2: Without masking, the target-task LoRA matches the input distribution, yielding low loss and weak gradients that limit noise removal; with masking, the remaining LoRAs’ noise patterns do not align with the current input, resulting in higher loss, stronger gradients, and more effective denoising. This is empirically validated via the loss comparison in Fig. 6, where MAPLE exhibits higher ID loss (stronger gradient signal) but lower OOD loss (better generalization).

4.2.2 Routing entropy analysis

In section 3.1, we used routing entropy as a proxy for prototype quality and showed that routing entropy using LoRA prototypes obtained from popular methods (ARROW and PHATGOOSE) is similar for ID and OOD tasks. In this section, we revisit this analysis to assess how prototypes refined by MAPLE alter the routing entropy. Although MAPLE does not explicitly optimise routing entropy, improved representations should manifest as (i) lower absolute entropy and (ii) a larger ID–OOD separation. With MAPLE-trained prototypes, figure 3 shows a clear separation in routing en-

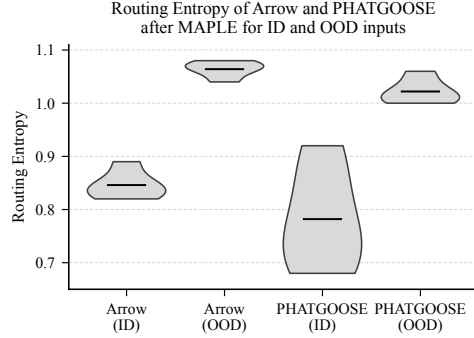


Figure 3: Routing entropy of learned representations with MAPLE using ARROW and PHATGOOSE initialisations across five tasks. ID and OOD distributions are clearly separated; ID routing entropy is consistently lower than OOD, indicating stronger discriminative capacity and more effective routing.

trophy for ID and OOD inputs for both ARROW and PHATGOOSE initialisations. ID entropy is consistently lower than OOD entropy, indicating effective identification of relevant LoRA as a result of improved discriminability of LoRA prototypes. Moreover, relative to the corresponding initialisations in figure 1, both ID and OOD entropies decrease, indicating more confident routing overall. Taken together, these results suggest that MAPLE yields more discriminative prototypes, validating its effectiveness in refining LoRA prototypes. We note that MAPLE implicitly routes samples to the most relevant experts and does not explicitly aim to identify samples as OOD; the entropy separation is an emergent property of the denoised prototypes.

5 Conclusion and Future Work

We show that LoRA prototypes learned in isolation fail to discriminate under joint routing, with ID and OOD inputs exhibiting similar routing entropy, and formalize this through a theoretical analysis exposing noisy subspaces. To address this, we introduce **MAPLE**, which masks the target adapter during prototype learning to eliminate noisy attributes from LoRA representations. MAPLE yields more discriminative prototypes, lowers routing entropy, and improves OOD routing across model scales and heterogeneous adapter pools. The method is model-agnostic and refines representations from existing techniques. Future directions include designing objectives that explicitly penalize noisy subspaces, extending to multi-task and continual settings with evolving pools, and scaling to larger backbones and additional modalities.

6 Limitations

MAPLE is effective in eliminating noisy attributes encoded in LoRA prototypes, but the sole improvement arises from the elimination of noisy components involved in routing. The amount of noise encoded in LoRA prototypes depends on various factors, such as the quality of data used for LoRA training, the similarity of the tasks used for training LoRAs, and the technique used to obtain LoRA representations. In case the LoRA prototypes don't encode many noisy attributes, the performance improvement with MAPLE would be marginal. Since MAPLE is a training-based method, it requires data (only a small fraction), which demands access to the LoRA training data, which may not always be accessible, restricting its utility.

7 Potential Risks

MAPLE refines routing among existing pretrained LoRA adapters and therefore inherits risks from the base model, the adapters, and their training data, including biased, harmful, or factually incorrect generations. Improved routing may also increase reliance on an unsuitable adapter when the input is ambiguous or differs substantially from the distributions represented in the adapter pool. Because prototype training requires access to samples from the LoRA training tasks, deployments must consider data licensing, privacy, and the possibility that sensitive attributes are encoded in learned prototypes. These risks can be reduced through careful dataset and adapter curation, access controls, evaluation across relevant demographic and task groups, monitoring of routing behavior, and application-specific output safeguards. MAPLE should not be deployed in high-stakes settings without additional validation and human oversight.

Author Contributions

Adarsh Kappiyath conceived the idea, implemented the method, and conducted all experiments. Abhra Chaudhuri developed the theoretical contributions. Xiatian Zhu and Lu Yin were the main supervisors and participated in weekly discussions. The remaining authors contributed through discussions and editing the paper.

Acknowledgments

We acknowledge the EuroHPC Joint Undertaking for awarding us access to Leonardo at CINECA, Italy.

References

- Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. 2021. [Program synthesis with large language models](#). *Preprint*, arXiv:2108.07732.
- Joshua Belofsky. 2023. Token-level adaptation of LoRA adapters for downstream task generalization. In *Proceedings of the 2023 6th Artificial Intelligence and Cloud Computing Conference*.
- Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. PIQA: Reasoning about physical commonsense in natural language. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence*.
- Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, and 39 others. 2021. [Evaluating large language models trained on code](#). *Preprint*, arXiv:2107.03374.
- Feng Cheng, Ziyang Wang, Yi-Lin Sung, Yan-Bo Lin, Mohit Bansal, and Gedas Bertasius. 2024. Dynamic adapter merging for continual video question-answering learning.
- Alexandra Chronopoulou, Matthew Peters, Alexander Fraser, and Jesse Dodge. 2023. AdapterSoup: Weight averaging to improve generalization of pre-trained language models. In *Findings of the Association for Computational Linguistics: EACL 2023*. Association for Computational Linguistics.
- Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics.
- Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try ARC, the AI2 reasoning challenge.
- Shizhe Diao, Tianyang Xu, Ruijia Xu, Jiawei Wang, and T. Zhang. 2023. Mixture-of-domain-adapters: Decoupling and injecting domain knowledge to pre-trained language models' memories. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.
- Jon Durbin. 2024. airoboros: Customizable implementation of the self-instruct paper.
- Aaron Grattafiori and Abhimanyu Dubey. 2024. [The Llama 3 herd of models](#). *Preprint*, arXiv:2407.21783.

- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.
- Chengsong Huang, Qian Liu, Bill Yuchen Lin, Tianyu Pang, Chao Du, and Min Lin. 2024. LoraHub: Efficient cross-task generalization via dynamic LoRA composition. In *First Conference on Language Modeling*.
- Mojan Javaheripi and Sébastien Bubeck. 2023. [Phi-2: The surprising power of small language models](#). Microsoft Research Blog.
- Maxine. 2023. Llama-2, mo' LoRA. <https://crumbly.medium.com/llama-2-molara-f5f909434711>.
- Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.
- Mazda Moayeri, Vidhisha Balachandran, Varun Chandrasekaran, Safoora Yousefi, Thomas Fel, Soheil Feizi, Besmira Nushi, Neel Joshi, and Vibhav Vineet. 2025. Unearthing skill-level insights for understanding trade-offs of foundation models. In *The Thirteenth International Conference on Learning Representations*.
- Mohammed Muqeeth, Haokun Liu, Yufan Liu, and Colin Raffel. 2024. Learning to route among specialized experts for zero-shot generalization. In *Proceedings of the 41st International Conference on Machine Learning*. PMLR.
- Oleksiy Ostapenko, Zhan Su, Edoardo Ponti, Laurent Charlin, Nicolas Le Roux, Lucas Caccia, and Alessandro Sordoni. 2024. Towards modular LLMs by building and reusing a library of LoRAs. In *Proceedings of the 41st International Conference on Machine Learning*.
- Olivier Roy and Martin Vetterli. 2007. The effective rank: A measure of effective dimensionality. *2007 15th European Signal Processing Conference*.
- Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. WinoGrande: An adversarial Winograd schema challenge at scale. In *arXiv preprint arXiv:1907.10641*.
- Zejiang Shen, Hunter Lang, Bailin Wang, Yoon Kim, and David Sontag. 2024. Learning to decode collaboratively with multiple language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.
- Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and Jason Wei. 2023. Challenging BIG-Bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*.
- Hanqing Wang, Bowen Ping, Shuo Wang, Xu Han, Yun Chen, Zhiyuan Liu, and Maosong Sun. 2024. LoRA-Flow: Dynamic LoRA fusion for large language models in generative tasks. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.
- Xun Wu, Shaohan Huang, and Furu Wei. 2024. MoLE: Mixture of LoRA experts. In *International Conference on Learning Representations*.
- Jingwei Xu, Junyu Lai, and Yunpeng Huang. 2025. Me-teoRA: Multiple-tasks embedded LoRA for large language models. In *The Thirteenth International Conference on Learning Representations*.
- Prateek Yadav, Colin Raffel, Mohammed Muqeeth, Lucas Caccia, Haokun Liu, Tianlong Chen, Mohit Bansal, Leshem Choshen, and Alessandro Sordoni. 2025. A survey on model MoErging: Recycling and routing among specialized experts for collaborative learning. *Transactions on Machine Learning Research*.
- Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. 2024. [AdaMerging: Adaptive model merging for multi-task learning](#). In *The Twelfth International Conference on Learning Representations*.
- Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.
- Ziyu Zhao, Leilei Gan, Guoyin Wang, Wangchunshu Zhou, Hongxia Yang, Kun Kuang, and Fei Wu. 2024. LoraRetriever: Input-aware LoRA retrieval and composition for mixed tasks in the wild. In *Findings of the Association for Computational Linguistics: ACL 2024*.

A Proofs and Derivations

A.1 Intuition behind Theoretical Formulation

We view routing as an embedding of the input into a vector space, of which, the constituent bases are the LoRA prototype vectors. We show, via Theorem 1, that when the rank of this vector space is low, one is likely to observe similar source and target routing entropies. The intuition behind this is that a low rank prototype basis "restricts" the span of the vector space in which the input could be embedded, meaning that the model is forced to use similar routing plans for both ID and OOD samples, hindering OOD-specific adaptation.

The fact that existing methods exhibit similar source and target entropies is illustrated empirically in Fig. 1. To provide empirical evidence to the premise that such overlap in entropy in fact occurs due to low-rank LoRA prototypes, we conduct further experiments, where we show that the rank of the LoRA prototype basis is noticeably lower for existing methods as compared to our proposed MAPLE. Furthermore, since Theorem 1 equivalently predicts an increase in rank to lead to a low source and high target entropy, we conduct the experiment shown in Fig. 3 where such a phenomenon is indeed observed to happen. Source entropy is significantly lower than the target for MAPLE, which is seen to yield higher rank prototypes based on our recent experiments. The fact that LoRA prototypes become more orthogonal (implying higher rank) with MAPLE fine-tuning can also be observed in Fig. 5, where the cosine similarity between different prototypes is seen to generally drop, justifying the applicability of Theorem 1.

Following from the above, Theorem 2 characterizes the relationship between restrictivity (as described in Theorem 1) and noisiness in the LoRA prototype space. Under Assumption 2, it shows that more restrictive subspaces are noisy with respect to a larger fraction of distributions. We aim to provide direct empirical evidence for this relationship. To this end, we performed the following experiment: Consider the initial (pre-finetuning) prototype vector to be A and the post-finetuning prototype vector to be B . As A gets closer to B , the difference vector $(A - B)$, which corresponds to the subspace being removed from A to reach B , gets closer to B too. As a result, $B \cdot (A - B)$ increases. Therefore, the magnitude of the dot-product of $(A - B)$ with any vector X , i.e., $X \cdot (A - B)$, can be used as

a measure of the relevance (or inverse measure of noisiness) of the removed subspace $(A - B)$ to that of X , which reaches its maximum when both $(A - B)$ and X are equal to B .

In summary, the difference vector $(A - B)$ can be thought of as the subspace removed under denoising and consequently, the dot product $X \cdot (A - B)$ can be used as an inverse measure of the noisiness of the eliminated subspace with respect to X . The lower the dot-product, the more noisy the removed subspace, and vice versa.

We leverage this idea to estimate the degree of denoising achieved by our method. To this end, we computed the dot product of the difference vector (pre-finetuning - post-finetuning) with each of the source and target distributions. We compared these dot products with the dot products between the inputs and the MAPLE prototypes and observed the former to be significantly lower than the latter.

Prototype	$a \cdot b$
MAPLE	0.272
Δ	0.136

This implies the noisiness of the subspaces eliminated through fine-tuning, relative to the actual prototype bases that our proposed MAPLE converges to.

A.2 Proof of Theorem 1

Proof. Let $B_1 = \mathcal{B} - \eta_1$ and $B_2 = \mathcal{B} - \eta_2$. We start with the premise that $\text{erank}(B_1) > \text{erank}(B_2)$. By Assumption 1,

$$\begin{aligned} H(P_{B_1}(X^o)) - H(P_{B_1}(X^i)) \\ > H(P_{B_2}(X^o)) - H(P_{B_2}(X^i)) \end{aligned} \quad (2)$$

Dividing Eq. (2) by $H(P_{\mathcal{B}}(X^o)) - H(P_{\mathcal{B}}(X^i))$:

$$\begin{aligned} \frac{H(P_{B_1}(X^o)) - H(P_{B_1}(X^i))}{H(P_{\mathcal{B}}(X^o)) - H(P_{\mathcal{B}}(X^i))} \\ > \frac{H(P_{B_2}(X^o)) - H(P_{B_2}(X^i))}{H(P_{\mathcal{B}}(X^o)) - H(P_{\mathcal{B}}(X^i))} \\ \implies r(\eta_1) > r(\eta_2) \end{aligned}$$

This completes the proof of the theorem. $\square$

A.3 Proof of Theorem 2

Proof. By Assumption 2,

$$\mathcal{R}_\gamma(\eta_1) \subseteq \mathcal{R}_\gamma(\eta_2).$$

Since $\mathcal{R}_\gamma(\eta)$ and $\mathcal{N}_\epsilon(\eta)$ partition $\mathcal{D}$,

$$|\mathcal{N}_\epsilon(\eta)| = |\mathcal{D}| - |\mathcal{R}_\gamma(\eta)|.$$

Therefore,

$$|\mathcal{N}_\epsilon(\eta_1)| \geq |\mathcal{N}_\epsilon(\eta_2)|.$$

Dividing by $|\mathcal{D}|$ gives $N(\eta_1) \geq N(\eta_2)$. Strict inclusion gives a strict inequality. $\square$

Algorithm 1: Pseudocode for MAPLE

Input: Collection of pretrained LoRAs
 $\mathcal{A} = \{A_1, \dots, A_L\}$; routing parameters
 $\{W_\ell \in \mathbb{R}^{L \times d}\}_{\ell=1}^N$, where N is the number of layers; Task specific data
 $\mathcal{D} = \{\mathcal{D}_1, \dots, \mathcal{D}_L\}$; K -Number of LoRAs to select

Output: Updated routing parameters $\{W_\ell\}$

// Notation: $h_\ell \in \mathbb{R}^d$ is the hidden state at layer ℓ with dimension d

while training do
 Sample task $k \sim \text{Unif}\{1, \dots, L\}$; sample $x \sim \mathcal{D}_k$;
 $h_0 \leftarrow \text{ENCODE}(\text{TOKENIZE}(x))$;
 for $\ell \leftarrow 1$ **to** N **do**
 $z \leftarrow W_\ell h_{\ell-1} \in \mathbb{R}^L$; // z_i scores adapter A_i
 $\mathcal{I} \leftarrow \text{top-K}(z, K)$;
 $\mathcal{I} \leftarrow \mathcal{I} \setminus \{k\}$; // mask target-task LoRA
 $p \leftarrow -\infty \cdot \mathbf{1}_L$; $p_{\mathcal{I}} \leftarrow z_{\mathcal{I}}$; // top- K selection
 $\alpha \leftarrow \text{softmax}(p)$;
 $h_\ell \leftarrow \text{LLMFORWARD}(\theta, \{A_i\}_{i=1}^L, \alpha)(h_{\ell-1})$;
 $\mathcal{L} \leftarrow \text{MLM.Loss}(h_N, x)$;
 Update $\{W_\ell\}_{\ell=1}^N$ by backprop on $\mathcal{L}$;
 // frozen base LLM and LoRAs $\mathcal{A}$

B Additional Results

B.1 MAPLE training dynamics

To provide further insights into the training dynamics of MAPLE, we provide the evolution of validation and testing loss of MAPLE across training iterations in Fig. 4. As shown in Fig. 4, both losses decrease relative to the Arrow initialization at iteration 0, exhibiting clear and structured trajectories rather than stochastic fluctuations. These trends indicate that MAPLE’s improvements arise from meaningful optimization behavior and are not artifacts of random variation.

B.1.1 Impact on prototype similarity

To assess how relationships between prototypes change under the proposed training scheme, we randomly sample nine LoRA prototypes and track how their cosine similarity evolves post-training for a randomly chosen layer. We summarise the

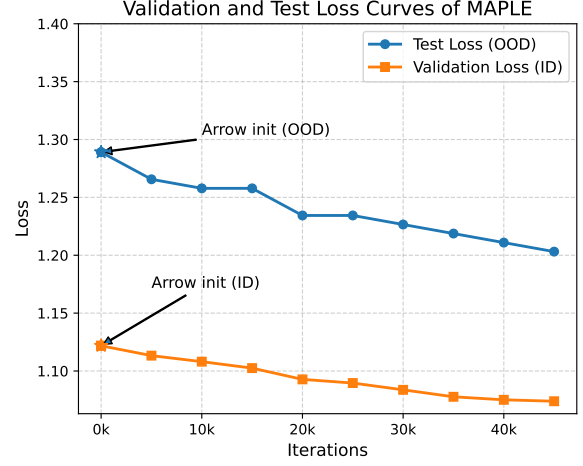


Figure 4: Validation (ID) and test (OOD) loss curves for MAPLE training. Iteration 0 is the Arrow initialization, marked by stars. MAPLE fine-tuning steadily reduces both losses relative to initialization.

pairwise cosine similarity between prototypes before and after training in figure 5. Comparing these similarities, we observe that after refining the prototypes with MAPLE, the pairwise cosine similarity between prototypes generally decreases by a small margin. We hypothesize that this results from removing shared noisy attributes encoded in LoRA prototypes, thereby reducing spurious dependencies between prototypes. The modest shift in cosine similarity suggests that noise encoded in LoRA prototypes is limited, which aligns with the average performance gain of 0.5% achieved by MAPLE.

B.2 Effect of masking in denoising LoRA prototypes

To illustrate the role of masking, we compare two settings: MAPLE and a baseline without masking. Consider k pretrained LoRAs, each trained on a corresponding task and therefore encoding both useful task structure and task-specific noise.

Without masking, an input from task i will most likely route to LoRA i . Because LoRA i closely matches the data distribution of task i , the model attains low loss and therefore receives weak gradients. This limits its ability to remove task-specific noise embedded within LoRA i . Moreover, because the noise in LoRA i aligns with noise in the task- i inputs, the routing mechanism does not incur penalties that would otherwise encourage denoising.

With masking applied, LoRA i is excluded when processing inputs from task i , forcing the model to rely on LoRAs trained on other tasks. The noise

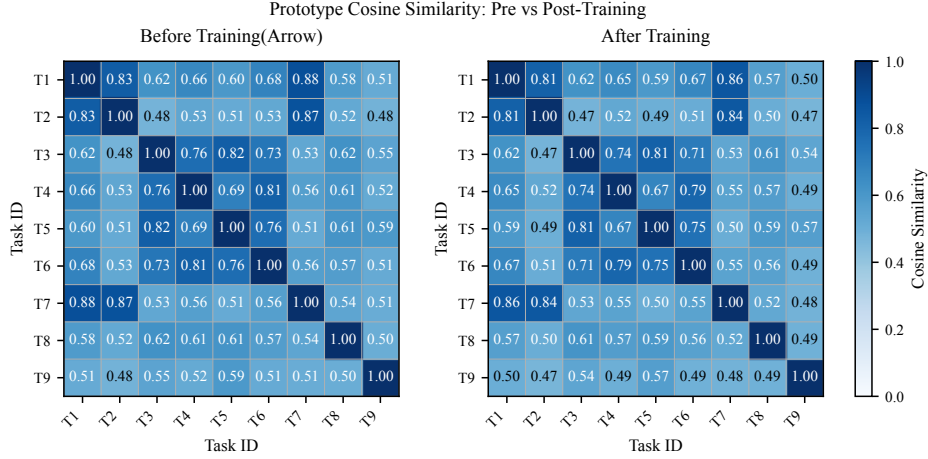


Figure 5: Pairwise cosine similarity of LoRA prototypes before and after training. Similarity generally decreases after training, suggesting that removing noisy attributes makes the prototypes more orthogonal.

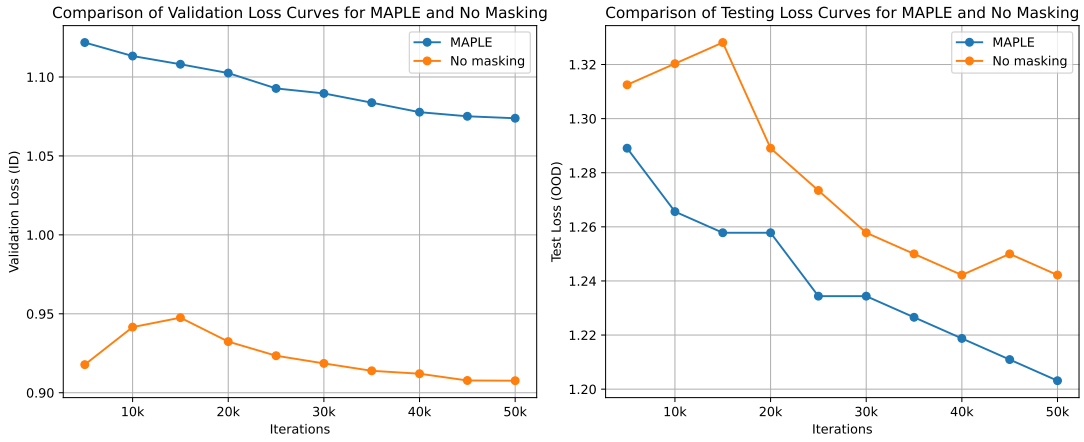


Figure 6: Validation and test loss curves comparing the MAPLE masking strategy with a no-masking baseline. The left panel shows in-distribution validation behavior across training iterations, while the right panel presents out-of-distribution test performance. The higher loss on ID samples of MAPLE under masking indicates stronger gradient signals that drive denoising, while the lower OOD loss confirms improved generalization.

patterns in these LoRAs do not match the noise present in task i , leading to routing errors and higher loss. These errors produce stronger gradients, which push the shared representation space toward reducing noise. Consequently, masking enables MAPLE to learn more task-agnostic and better-generalizing prototypes.

Empirical evidence for this mechanism is shown in Fig. 6. During training, MAPLE exhibits higher ID loss (Fig. 6 (Left)) under masking than in the no-masking baseline. On OOD data(Fig. 6 (Right)), MAPLE achieves consistently lower loss, demonstrating more effective removal of task-specific artifacts. This behavior aligns with the improvements reported in Table 2 of the main text, where masking outperforms the no-masking variant.

Method	$k = 2$	$k = 3$	$k = 4$
Arrow + MAPLE	66.20	66.52	66.41

Table 3: Sensitivity of MAPLE to the selection of top- k on Phi-2 with Arrow initialization. Performance varies mildly across values of k .

B.2.1 Sensitivity to top- k

To assess the sensitivity of MAPLE to the top- k hyperparameter, we evaluate MAPLE (Arrow + MAPLE on Phi-2) with three different values of k . As shown in Table 3, performance varies mildly with k , and choosing an optimal k can further enhance the gains of MAPLE.

C LLM Usage

LLMs were utilized in refining the paper, primarily for rephrasing, restructuring, and condensing sentences. They were also queried for suggestions on additional experiments; however, none of these suggestions were deemed suitable for inclusion, limiting the use of LLMs to writing assistance only.

D Reproducibility

In this section, we report all model- and dataset-specific hyperparameters in Table 4. Experiments were executed on two NVIDIA A100 GPUs. To enable exact replication, we release a clean, user-oriented codebase and pretrained checkpoints. All datasets used in this work are publicly available. For all methods, we use the same top- k as the original routing scheme ($k = 2$ for PHATGOOSE and $k = 4$ for Arrow) to ensure the same inference compute capacity. In zero-shot downstream evaluation, we append an EOS token to the target options to indicate the end of a sentence. For multiple-choice task evaluations, we select continuations using scores normalized by token length.

Table 4: Hyperparameters for the Phi-2 and Llama 3 models.

Parameter	Phi-2	Llama 3
Learning rate (LR)	$1e-4$	$1e-3$
Batch size	16	16
Epochs	1	1
Warm-up proportion	0.06	0.06

For all LoRA experts trained in this work, we use a LoRA rank of 4, a dropout probability of 0.05, an α value of 16, and a learning rate of 10^{-4} , along with learning-rate warm-up and annealing phases, following Ostapenko et al. (2024). For all models, we modify the attention layers and attention output projection, with weight sharing among LoRAs applied to the attention layers.