MULTI-TASK LOW-RANK MODEL ADAPTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Low-Rank Adaptation (LoRA) is the de-facto method for parameter-efficient fine-tuning Vision Transformers (ViTs). However, when applied to multi-task learning, the conventional method of training a LoRA module for each task independently leads to misaligned feature subspaces at inference, i.e., the semantic meanings of a feature dimension from two different LoRA modules are not aligned and may cancel each other in bad cases. Current solutions employ parameter regularization or feature routing, but they operate under the flawed assumption that task subspaces are independent, which is not the case in reality, resulting in limited improvements. In this paper, we first dive into the conflict problem on multiple multi-task datasets, and have two key observations. First, we reveal that LoRA's high singular value components encode discriminative information, while low singular value components accumulate noise. Second, we identify a critical source of feature misalignment from the perspective of the gradient: attaching LoRA modules to the wrong layers (within the attention module of ViT) may amplify conflicting gradients during backpropagation. Based on these, we develop our own add-on, plug-and-play solution for multi-task LoRA. Specifically, we propose 1) fine-grained routing with 2) spectrum-aware regularization, and 3) block-level LoRA adaptation. Their integration with the best baseline methods, such as HydraLoRA (Tian et al., 2024a), delivers large-margin improvements and state-of-the-art results. We name our final integrated approach `mt`LoRA. The efficacy of `mt`LoRA is validated through extensive experiments on a variety of multi-task benchmarks. These include natural language understanding (Dolly-15K (Conover et al., 2023)), cross-domain adaptation (DOTA (Xia et al., 2018)), and fine-grained classification (iNaturalist (Van Horn et al., 2018)), where it outperforms current multi-task LoRA variants. An ablation study further elucidates that our core contributions, spectrum-aware routing, adaptive regularization, and novel attachment locations, are instrumental in achieving these performance improvements. Our code is at this anonymous link.

1 INTRODUCTION

Low-Rank Adaptation (LoRA) (Hu et al., 2021) has emerged as the *de-facto* standard of parameter-efficient fine-tuning (PEFT), thanks to its minimal trainable parameters, zero inference latency overhead, and modular deployment (He et al., 2022; Zhang et al., 2023; Dettmers et al., 2023; Han et al., 2024; Ge et al.; Tian et al., 2025). Though LoRA achieves remarkable performance in single-task adaptation (Zhang et al., 2023; Liu et al., 2024; Tian et al., 2024b), real-world applications usually need multi-task LoRA adaptation, *i.e.*, using multiple task-specific LoRA modules to handle multiple tasks simultaneously (Stoica et al., 2025; Wu et al., 2024a; Ma et al., 2018). For instance, language models need to process multiple tasks (*e.g.*, mathematical reasoning, legal analysis, and ethical questions) concurrently (Hendrycks et al., 2020), and vision models need to adapt across multiple spectrums (*e.g.*, optical and radar imagery) (Tian et al., 2024b). However, existing multi-task LoRA adaptation methods suffer from catastrophic performance degradation as the number of tasks increases (Tian et al., 2024a; Wu et al., 2024a; Stoica et al., 2025).

The core challenges are two kinds of misalignment: parameter misalignment and representation misalignment (Stoica et al., 2025; Han et al., 2024). Specifically, **parameter misalignment** means different LoRA modules have conflicting weight updates (*i.e.*, weights have opposing signs and magnitudes). To address this, existing methods use regularizations to enforce orthogonality across LoRA parameters (Ilharco et al., 2022; Yadav et al., 2023; Yu et al., 2024). Another is **representation**

Table 1: **Key limitations of existing methods.** (A) We evaluate the impact of orthogonality regularization (λ). As λ increases, routing entropy (Ent.) steadily increases, indicating that features are less distinguishable. (B) SVD analysis reveals LoRA's spectral components are heterogeneous. High-SV components (Top-10%) are highly discriminative but also cause the most parameter conflicts. (C) Applying LoRA at the block-level outperforms component-level (*e.g.*, apply to individual linear layers W_q, W_k, etc). Please find detailed experimental setup in Sec 4.6.

(A) OrthoReg.

λ	Ent. ↓	Acc. (%)↑	Δ
0.00	1.72	78.5	—
0.25	1.89	79.8	+1.3%
0.50	2.04	79.2	+0.7%
0.75	2.18	77.6	-0.9%
1.00	2.29	75.3	-3.2%

(B) SVD Analysis.

Freq. Band	Conflict ↓	Task Disc. ↑
Top-10%	0.68	2.14
10-50%	0.31	0.98
50-100%	0.19	0.67

(C) Attaching Level.

Attach Level	Acc. (%) ↑
Component	89.8
Block	
Attn	90.9
FFN	91.2
Transformer	92.0

misalignment, meaning that LoRA modules' output features are divergent, *e.g.*, centered kernel alignment (CKA) scores dropping to 0.09-0.37 (Stoica et al., 2025). To address this, existing methods use dynamic routing to weigh LoRAs' output features, *i.e.*, by using soft-/hard-gating networks to predict LoRA modules' weights (Ma et al., 2018; Wu et al., 2024a; Wei et al., 2025; Tian et al., 2024a) and then weighted-summing them together. The common limitation of existing methods is that they address the two misalignments independently, *i.e.*, either focus on regularization or dynamic routing. This is problematic because parameter and representation spaces are inherently entangled. **However, combining them straightforwardly reveals a key limitation**: stronger regularization alleviates conflict, but it harms routing. Specifically, Table 1(a) shows the feature discrimination (routing entropy, "Ent.", *i.e.* the entropy of the distribution over LoRA modules implied by the router, cf. Sec 4.6) versus regularization strength (λ). The results show that as λ increases, the accuracy improves to 79.8 (a $+1.3\%$ increase) due to reduced conflicts. However, stronger regularization harms feature discriminability (routing entropy increases to 2.29), resulting in a performance drop of -3.2% ($79.8 \rightarrow 75.3$).

This raises a key question: **why does this limitation exist?** We identify two root causes that stem from how LoRA modules are treated and placed, respectively.

First, uniformly treating LoRAs ignores their spectral heterogeneity. Table 1 shows how conflict (measured by SV similarity) and task discriminability (measured as Fisher Discrimination Ratio (FDA, by Fisher (1936)), *i.e.*, $d^{(i)}_{\text{inter}}/\sigma^{(i)}_{\text{intra}}$, where $d^{(i)}_{\text{inter}}$ is mean L2 distance between task centroids and $\sigma^{(i)}_{\text{intra}}$ is intra-task variance). Results show that high singular value (high-SV) components capture the task-discriminative knowledge, while low-SV components capture task-agnostic noise. Specifically, top-10% spectrum contain 68% of conflicts and encodes the most discriminative information (2.14), while bottom-50% spectrum captures negligible discriminativeness (0.67). This observation explains the limitation: uniform regularization applies strong regularization to high-SV components; hence, it suppresses discriminative information and harms routing. This motivates us to treat the spectral components differently.

Second, applying LoRA to component-level matrices amplifies gradient conflicts. We argue that applying LoRA to component-level matrices (*i.e.*, W_q, W_k, W_v, W_o) creates a multiplicative interference. Specifically, when the gradients ∇W_q and ∇W_k conflict, the resulting attention scores $\text{Softmax}(QK^T)$ combine both gradient errors, which then multiply with the errors ∇W_v in the output $\text{Attn}(Q, K, V)$. This motivates us to apply LoRA at the block-level (*e.g.*, to Attention/FFN/Transformer blocks).

Given these insights, we propose `mt`LoRA–a novel method that reconfigures LoRA modules. We introduce three key designs: 1) spectral-aware regularization, 2) fine-grained routing, and 3) block-level adaptation. **First, we design spectral-aware regularization.** We apply strong orthogonalization to low-SV components (bottom 50%, which contribute minimal discrimination but accumulate as noise) to prevent interference, while preserving high-SV components (top 10%, containing 2.14× discriminative information) to preserve task-specific information. We achieve this through a masking function $m(\sigma) = 1 - \exp(-\sigma/\bar{\sigma})$, where σ is the singular value and $\bar{\sigma}$ is the average of all singular

values. For noise components (*i.e.*, $\sigma \ll \bar{\sigma}$), $m(\sigma)$ approaches 0, this enforces strong orthogonality; for discriminative components (*i.e.*, $\sigma \gg \bar{\sigma}$), $m(\sigma)$ approaches 1, maintaining task-specific expression. **Second, we propose fine-grained routing.** Unlike standard routing that assigns one scalar weight per LoRA (forcing uniform combination across all dimensions), We use a router network to produce a d-dimensional weight vector per LoRA. This addresses our observed heterogeneous conflict pattern (*i.e.*, in Table 1(b) conflicts concentrate in certain dimensions while others remain compatible). Consider an example, given Dolly-15k's prompt "Write a creative story about why the sky is blue with scientific accuracy" and two task LoRAs: brainstorm LoRA and QA LoRA. Answering this prompt requires more of the "creativity" dimension from brainstorm LoRA and more of the "accuracy" dimension from QA LoRA, respectively. However, traditional methods (Tian et al., 2024a; Wu et al., 2024a) use a scalar weight for all dimensions of each LoRA module, *e.g.*, all the dimensions are weighted combination of $p\%$ brainstorm and $(1-p)\%$ QA LoRAs. Our fine-grained routing breaks this constraint. For example, the 256th dimension (handling metaphors) assigns weight 0.9 to brainstorming LoRA for vivid imagery, while the 384th dimension (handling scientific terms) assigns weight 0.9 to QA LoRA for factual correctness. **Third, we propose block-level adaptation.** To resolve the multiplicative effect of gradient conflicts, we apply LoRA to block-level (*e.g.*, attention block, FFN block or entire transformer block). Specifically, for block F, our module is applied directly as $F(x) + \Delta(x)$, where Δ learns block-level input-output mapping. This avoids going through attention's Softmax normalization. Table 1 validates this design. Block-level adaptation (*e.g.*, transformer-level reaches 92.0%) outperforms component-level (89.8%).

We validate `mt`LoRA on three benchmarks across vision and NLP domains. We establish three key findings. First, our block-level adaptation largely eliminates gradient conflict amplification. Specifically, Table 1 shows Transformer-level (92.0%) outperforms component-level (89.8%) on DOTA. Second, fine-grained routing captures dimension-specific task requirements. Table 5 shows channel-wise routing improves MMLU accuracy from 44.5% to 47.1%. Third, even vanilla orthogonal regularization, when combined with dynamic routing, improves performance (Table 2: 78.5%→79.8% on iNat2018). For extreme multi-task scenarios (N=25 for iNat2018, N=16 for Dolly-15k), `mt`LoRA surpasses SOTA by 3.4% and 4.4% respectively (Table 2: 81.9% vs. 78.5% for iNat2018, 47.1% vs. 42.7% for Dolly-15k).

Our contributions are three-fold. 1) We identify spectral heterogeneity as the key limitation in multi-task LoRA: high singular value components encode both conflicts and critical task discrimination. 2) We make three key technical contributions. Spectral-aware regularization selectively orthogonalizes low-SV noise while preserving high-SV discrimination, fine-grained routing assigns dimension-specific weights instead of scalar weights, and block-level adaptation mitigates gradient conflict amplification. 3) We demonstrate consistent improvements across vision (DOTA, iNat2018) and language (Dolly-15k) benchmarks with 15-25 tasks, achieving up to 4.4% absolute performance improvement over state-of-the-art.

2 Related Works

Multi-Task LoRA Adaptation. Multi-task LoRA adapation aims to compose multiple task-specific LoRA modules to handle multiple tasks, simultaneously. The key challenge is the misalignment between LoRA modules. Such misalignment can be categorised into parameter misalignment or representation misalignment. Existing studies can be categorised into regularization methods and dynamic routing methods, tackling the two misalignments respectively. Specificially, **Regularization methods** addresses the *parameter misaglinment* issue. Existing methods impose regularization to enforce orthogonality across LoRA parameters (Ilharco et al., 2022; Yadav et al., 2023; Yu et al., 2024). For instance, Task Arithmetic (Ilharco et al., 2022) linearly combines task vectors; TIES-Merging (Yadav et al., 2023) resolves sign conflicts through majority voting; DARE (Yu et al., 2024) applies stochastic masking to enforce sparsity. However, these methods are input-independent and ignore input dynamics. **Dynamic routing methods** address *representation misaglinment*, existing methods use dynamic routing to route LoRA's output features. Such methods rely on a soft-/hard-gating network to predict combination weights for LoRA modules. MMoE (Ma et al., 2018) first proposed gating networks for expert selection. MoLE (Wu et al., 2024a) extends this to LoRA adaptation, introducing Top-K routing and balancing losses to prevent degeneration (*i.e.*, 68% probability on single LoRA module). HydraLoRA (Tian et al., 2024a) combines routing with asymmetric LoRA structure (*i.e.*, a single shared A, with multiple task-specific B_k). LoRAMoE force

part of LoRA experts to maintain the foundation model's knowledge to guard against catastrophyic forgetting. However, existing works only offer independent, partial solutions. Our preliminary study shows that combination of the two methods could be beneficial. Based on our insights, we the first to propose joint regularization and dynamic routing.

LoRA Placement Strategies. Prior work explores where to place LoRA modules in transformers. Ada-Merging (Yang et al., 2024) and MoLE (Wu et al., 2024a) assign different merging weights to different layers. They find that uniform treatment across layers is suboptimal. MTLoRA (Agiza et al., 2024) places task-irrelevant modules early and task-specific modules late in the network. MixLoRA (Wu et al., 2024b) only inserts LoRA into FFN blocks, avoiding attention layers completely. However, all these methods apply LoRA to individual weight matrices (W_q, W_k, W_v, W_o). When multiple LoRAs update these matrices, gradients conflict occurs and amplifies through attention's Softmax. Instead, we apply LoRA at block-level (*i.e.*, around entire blocks like Attention/FFN/Transformer).

3 METHOD

In this section, we provide problem formulation for multi-task LoRA adaptation in Sec 3.1, and three key designs of our `mt`LoRA: Spectral-aware regularization in Sec. 3.2, fine-grained routing in Sec. 3.3, and block-level adaptation in 3.4

3.1 PROBLEM FORMULATION

We address the challenge of multi-task LoRA adaptation. Specifically, consider a frozen pretrained model with parameters W_0, we have N task-specific LoRA modules $\{\Delta_N\}$, where each module Δ_i is a low-rank update of W_0 parameterized as $\Delta_i = B_i A_i$. For an input x, multi-task LoRA adaptation combines these LoRA modules:

$$f(x) = f_{W_0}(x) + \sum_{i=1}^{N} \pi_i(x) \cdot \Delta_i(x). \tag{1}$$

Existing methods can be categorised as two kinds. **1) Dynamic routing** uses a soft-/hard-gated network $g(\cdot)$ to weigh LoRA modules, *i.e.*, $\pi(x) = \text{Softmax}\,(g(x))$. **2) Orthogonal regularization** forces parameter orthogonality during training. A basic orthogonal regularization is $\mathcal{L}_{\text{ortho}} = \lambda \sum_{i<j} \|\Delta_i^T \Delta_j\|_F^2$, where $\|\cdot\|_F$ is the Frobenius norm. Note that it is impractical to calculate on the full-size update (Δ). In our implementation, we apply the regularization directly to the B matrices (following (Tian et al., 2024a)).

3.2 SPECTRAL-AWARE REGULARIZATION

Our key insight is that not all parameters contribute equally to conflicts or discrimination. Standard orthogonal regularization suppresses all parameters equally, harming both conflicts and task-specific information (as explained in the introduction). We apply selective regularization based on LoRA modules' singular value magnitude. For LoRA module Δ_i with SVD decomposition

$$\Delta_i = U_i \Sigma_i V_i^T, \tag{2}$$

we define regularization strength as $w(\sigma) = \exp(-\sigma/\bar{\sigma})$, where $\bar{\sigma}$ is the mean singular value. Low-SV components (noise) get strong regularization ($w \to 1$), while high-SV components (task-specific) are preserved ($w \to 0$). The spectral-aware loss thus becomes:

$$\mathcal{L}_{\text{spectral}} = \lambda \sum_{i<j} \sum_{k} w(\sigma_k) \cdot (\vec{u}_{i,k}^T \vec{u}_{j,k})^2 \tag{3}$$

where $\vec{u}_{i,k}$ is the k-th singular vector of module i.

3.3 FINE-GRAINED ROUTING

Unlike conventional dynamic routing that assigns one weight per LoRA ($\pi_i \in \mathbb{R}$), we assign dimension-specific weights $\Pi_i \in \mathbb{R}^{d/g}$, where g is group size. The combination becomes:

$$f(x) = f_{W_0}(x) + \sum_{i=1}^{N} \Pi_i(x) \odot \Delta_i(x), \tag{4}$$

where $\odot$ denotes grouped element-wise multiplication after broadcasting. This allows different dimensions to use different LoRA combinations.

3.4 BLOCK-LEVEL ADAPTATION

Instead of modifying individual weight matrices within each transformer block (*e.g.*, W_q, W_k, W_v), we apply LoRA as a **residual adapter** at the block-level, operating on the entire block's input-output transformation (*i.e.*, Attention block, FFN block, or entire Transformer block).

Specifically, for a frozen Transformer block $W : h_{in} \to h_{out}$, we create a parallel residual path that

$$h_{out} = W(h_{in}) + \Delta(h_{in}), \quad \text{where} \Delta(h_{in}) = BA \cdot h_{in}. \tag{5}$$

Here, $A \in \mathbb{R}^{r \times d}$ and $B \in \mathbb{R}^{d \times r}$ form a low-rank bottleneck. Unlike traditional LoRA that decomposes weight updates, our proposed block-level adaptation learns a direct input-to-output mapping.

Why does our design work? Compared with conventional LoRA, our block-level adaptation offers a key advantage: **it avoids gradient conflicts rooted in attention.** In traditional LoRA, gradients flow through the Softmax in attention. This creates cross-token dependencies. Changing attention to one token position affects all other positions through normalization. This effect amplifies task conflicts. For example, consider the input "The bank is steep". For finance tasks, the model needs high attention on "bank"→"money". For geography tasks, the model needs "bank"→"river". These conflicting attention patterns interfere through Softmax. In traditional LoRA, updating B to increase "bank"→"money" attention automatically decreases "bank"→"river" attention due to Softmax normalization, as they compete for the same probability mass. Our block-level adaptation avoids this competition. The two adapters can add the "money" and "river" feature **independently** to "bank" representation.

4 EXPERIMENTS

We validate mtLoRA through comprehensive experiments across vision and NLP benchmarks, demonstrating that our three key designs, spectral-aware regularization, fine-grained routing, and block-level adaptation, collectively address the multi-task collapse problem.

4.1 DATASETS

We evaluate on three benchmarks spanning vision and NLP domains: DOTA (Xia et al., 2018) for cross-domain adaptation (15 tasks), iNaturalist 2018 (Van Horn et al., 2018) for fine-grained classification (25-100 tasks), and Dolly-15k (Conover et al., 2023) for instruction following (16 tasks).

iNat2018 (Fine-Grained Classification) To simulate a high-conflict scenario with a large number of fine-grained classes, we construct a benchmark from the iNat2018 dataset. Our methodology is designed to be principled and reproducible, leveraging the dataset's inherent biological taxonomy to create semantically coherent yet potentially conflicting tasks. **1) Hierarchical Task Definition:** We define each LoRA expert's task at the taxonomic rank of **Order**. This is a principled choice, as classes within the same order (e.g., different species of songbirds) are visually similar and thus create high inter-task conflict, while being semantically distinct from other orders (e.g., raptors). **2) Data Partitioning:** We first select a high-level super-category, the class *Aves*, to ensure all tasks are within the same broad domain. We then identify the N most populous Orders within this class (e.g., *Passeriformes*, *Accipitriformes*, *Charadriiformes*). For each of these N Orders, we assign all of its

constituent species-level classes to a single corresponding LoRA expert. This creates N disjoint sets of classes for training. **3) Training and Evaluation:** Each LoRA expert is trained exclusively on the images of species belonging to its assigned Order. The composed model is then evaluated on its ability to classify species across the union of all N Orders, using a unified classification head on a held-out test set. This setup directly tests the model's ability to resolve conflicts among many fine-grained, visually similar experts.

Dolly-15k (Natural Language Understanding) For the language domain, we use the Dolly-15k instruction-following dataset. To create distinct experts, we perform K-Means clustering on the instruction embeddings to group them into N semantically related categories (e.g., summarization, creative writing, question-answering). Each LoRA is then trained on the data from one cluster, making it an expert in a specific type of instruction.

4.2 Implementation Details

Spectral-Aware Regularization. Computing SVD on full $\Delta = BA$ is expensive. We apply SVD to B matrices as proxy: $B_i = U_i \Sigma_i V_i^T$. We construct weighted matrices $B_i' = U_i \Sigma_i' V_i^T$ where $\Sigma_{kk}' = \sqrt{w(\sigma_k)} \cdot \sigma_k$, then compute $\mathcal{L}_{\text{spectral}} = \lambda \sum_{i<j} \|(B_i')^T B_j'\|_F^2$. SVD is performed every epoch.

Fine-Grained Routing. The router is a 2-layer MLP: $g(x) = \text{Linear}_2(\text{ReLU}(\text{Linear}_1(\bar{h})))$ where $\bar{h}$ is mean-pooled input. Output dimension is $N \times (d/g)$, reshaped and normalized: $\Pi = \text{Softmax}(\text{reshape}(g(x)), \text{dim} = 0)$. Each row Π_i is broadcast by repeating elements g times before multiplication with $\Delta_i(x)$. Larger g means finer grained routing.

4.3 Main Results

mtLoRA achieves state-of-the-art performance across all benchmarks, with particularly strong gains in extreme multi-task scenarios.

Performance under extreme multi-tasking. Table 2 shows our main results. mtLoRA improves over HydraLoRA baseline by 3.4% on iNat2018 (25 tasks) and 4.4% on Dolly-15k (16 tasks). The gains are most pronounced when all three components work together: orthogonal regularization, block-level scope, and channel-wise routing achieve 81.9% on iNat2018 and 47.1% on Dolly-15k.

Table 2: **SOTA Comparison.** We compare against the baseline (*i.e.*, uniformly weighted combination) and HydraLoRA (Tian et al., 2024a). Our method builds upon HydraLoRA by adding Orthogonality Regularization (Orth. Reg.), Block-level Scope (Scope), and Channel-wise Routing (Channel). All tasks are measured in average accuracy (%).

Method	Dyn. Routing	Orth. Reg.	Scope	Channel	DOTA	iNat2018	Dolly-15K
Baseline					18.0	8.5	19.5
HydraLoRA	✓				89.1	78.5	42.7
	✓	✓			89.8	79.8	43.5
mtLoRA (Ours)	✓	✓	✓		**92.0**	81.3	44.5
	✓	✓	✓	✓	91.0	**81.9**	**47.1**

Addressing the collapse problem. Table 3 reveals the severity of multi-task collapse. Naive averaging catastrophically fails as tasks increase: accuracy drops from 88.2% to 2.0% on DOTA when scaling from 5 to 15 tasks. The conflict score reaches 97.9%, indicating severe parameter interference. mtLoRA maintains stable performance even with 100 tasks on iNat2018.

4.4 Component Analysis

We ablate each design choice to understand their individual and combined contributions.

Table 3: **Performance Collapse Issue.** Results reported on Vision (ViT-B/16→{DOTA, iNat2018}) and NLP (LLaMA2-7B→Dolly-15k) tasks. Model collapses as the number of LoRA modules (N) increase.

	ViT-B/16 → DOTA			ViT-B/16 → INat2018				LLaMA2-7B → Dolly-15k		
	5	10	15	15	25	80	100	4	8	16
Single LoRA	94.5%	94.5%	94.5%	87.0%	87.0%	87.0%	87.0%	45.45%	45.45%	45.45%
Naive Averaging	88.2%	12.0%	2.0%	3.5%	1.0%	0.5%	0.3%	46.14%	40.50%	16.03%
Conflict Score	6.7%	87.3%	97.9%	96.0%	98.9%	99.4%	99.7%	−1.5%	10.9%	64.7%

Spectral heterogeneity drives design choices. Table 1(B) validates our core insight: LoRA's spectral components are heterogeneous. Top-10% singular values contain 68% of conflicts but also 3.2× higher task discriminability (2.14 vs 0.67). This motivates selective regularization: preserving high-SV components while orthogonalizing low-SV noise.

Regularization helps only with routing. Table 1(A) shows the regularization-routing interaction. Moderate regularization ($\lambda = 0.25$) improves accuracy (+1.3%), but stronger regularization ($\lambda = 1.0$) causes -3.2% drop. The routing entropy increases from 1.72 to 2.29, indicating reduced feature discriminability. Table 4 confirms that regularization benefits disappear without dynamic routing—orthogonal regularization alone achieves only 20.5% on DOTA versus 89.8% with routing.

Table 4: **Compare Uniform and Dynamic Routing.** We ablate combination strategies on the high-conflict DOTA (N=15), the extreme-conflict iNat2018 (N=25), and the Dolly-15k (N=16) settings. The results highlight the pivotal role of Dynamic Routing across domains.

Method	DOTA	iNat2018	Dolly-15k
Uniform Routing			
HydraLoRA†	18.0	8.5	19.5
+ Sparsity Reg.	16.5	7.2	18.0
+ Orthogonality Reg.	20.5	10.1	21.0
Dynamic Routing			
HydraLoRA	89.1	78.5	42.7
+ Sparsity Reg.	87.9	77.2	41.5
+ Orthogonality Reg.	**89.8**	**79.8**	**43.5**

†Implemented with static, uniform weight.

Fine-grained routing captures dimension-specific patterns. Table 5 demonstrates channel-wise routing's advantage. Full channel-wise routing (g=768) achieves 47.1% on Dolly-15k, outperforming module-wise routing (44.5%) by 2.6%. This confirms that different dimensions require different LoRA combinations.

Table 5: **Comprehensive Ablation on Routing Granularity for NLP (N=16).** Performance on Dolly-15k evaluated by MMLU accuracy. Empty cells denote planned experiments.

Routing Strategy	Hyperparameter	MMLU Acc. (%)
Module-Wise	-	44.5
Fine-Grained		
Grouped	g=32	45.3
	g=64	45.2
	g=128	44.3
Full	g=768	**47.1**

Block-level adaptation mitigates gradient conflicts. Tables 1(C) and 6 compare attachment strategies. Transformer-level attachment (92.0%) consistently outperforms component-level (89.8%)

across all data splits. The improvement is most pronounced in challenging tail categories (87.1% vs 85.7% for attention-only).

Table 6: **Compare LoRA Attaching Scope.** Results reported on N=15 categories on DOTA dataset.

Block	Head	Mid	Tail	Average
Attn	95.5	91.5	85.7	90.9
FFN	95.7	92.0	85.9	91.2
Transformer	**96.0**	**92.9**	**87.1**	**92.0**

4.5 Configuration Analysis

We identify optimal configurations for different domains.

Vision tasks prefer block-level without fine-grained routing. Table 7 shows vision models benefit most from transformer-level attachment (92.0%) but not from fine-grained routing (91.0%). This suggests vision features are more homogeneous across channels.

Table 7: **Optimal Configuration on Vision Tasks.** We report the performance gains from progressively incorporating our proposed techniques. Results reported in Top-1 Average Accuracy (%) on DOTA (N=15).

Method	DOTA (%)
Block-Level Adaptation	
HydraLoRA	89.8
+ Attn-level	90.9
+ FFN-level	91.2
+ Transformer-level	**92.0**
Routing Granularity	
Module-Wise	**92.0**
Fine-Grained	91.0

NLP tasks benefit from both block-level and fine-grained routing. Table 8 reveals NLP models gain from both transformer-level attachment (43.5%→44.5%) and fine-grained routing (44.5%→47.1%). Language representations appear more heterogeneous, requiring dimension-specific combinations.

Table 8: **Optimal Configuration on NLP Tasks.** We report the performance gains from progressively incorporating our proposed techniques. Experts learned on Dolly-15k (N=16) and evaluated on MMLU.

Method	MMLU Acc. (%)
Block-Level Adaptation	
HydraLoRA	43.5
+ Transformer-level	**44.5**
Routing Granularity	
Module-Wise	44.5
Fine-Grained	**47.1**

Module complexity offers diminishing returns. Table 9 shows that increasing LoRA module complexity yields marginal gains. Even 2.11× parameters (Transformer) only improves 0.2% over standard LoRA. This validates our focus on structural changes rather than capacity increases.

Table 9: **Ablation on LoRA Structure.** Increasing local module complexity yields only marginal gains. Baseline refers to the standard LoRA.

Method	Avg Acc. (%)	Relative Params
LoRA	89.8	$1\times$
MLP	89.9	$\approx 1.0\times$
Deep MLP	90.1	$\approx 1.07\times$
Attention	90.1	$\approx 1.24\times$
Transformer	90.0	$\approx 2.11\times$

4.6 Discussion

OrthoReg: Performance-Discriminability Limitation. We demonstrate that orthogonal regularization (applied to parameters) creates a key limitation: it reduces parameter conflict but harms feature discriminability. We measure routing entropy $\mathbb{E}_x\left[-\sum_{i=1}^{N}\pi_i(x)\log\pi_i(x)\right]$, *i.e.*, the average per-sample entropy of the router's output distribution $\pi(x)$. Higher entropy means more router uncertainty. Table 1(A) shows 25 LoRA experts on iNat2018 with orthogonality regularization $\lambda \in \{0, 0.25, 0.5, 0.75, 1.0\}$. Results confirm that while strong regularization ($\lambda \geq 0.5$) severely degrades performance by increasing router uncertainty, a smaller regularization ($\lambda = 0.25$) improves accuracy. Notably, *in our experiments, reaching this optimal point requires a 1.4× more training iterations*. It indicates the increased difficulty of optimizing.

SVD spectrum analysis. We perform a Singular Value Decomposition (SVD) to decompose $\Delta W = U\Sigma V^T$ for $N = 25$ LoRA modules trained on iNat2018. Singular values are partitioned into three bands by cumulative energy, high (top-10%), mid (10-50%), and low (bottom-50%). For each band B = $\{i_m\}$, we measure: 1) Parameter conflict: $\frac{1}{N(N-1)}\sum_i\sum_{j\neq k}\sigma_{j,i}\sigma_{k,i}|\cos(\vec{u}_{j,i},\vec{u}_{k,i})|$ where the outer sum averages over all singular value positions i within band B, and the inner sum computes pairwise LoRA conflicts at position i. 2) task discriminability as $d^{(i)}_{\text{inter}}/\sigma^{(i)}_{\text{intra}}$, where $d^{(i)}_{\text{inter}}$ is the mean L2 distance between task centroids and $\sigma^{(i)}_{\text{intra}}$ is within-task variance. We show results in Table 1(B). High-frequency components contain 68% of conflicts while comprising only 10% of parameters. These same components show 3.2× higher discriminability (2.14 vs 0.67) than low-frequency components.

5 Conclusion

We presented mtLoRA, which enables stable multi-task adaptation even with 25+ tasks by addressing the fundamental spectral heterogeneity in LoRA modules. Our key insight (*i.e.*, high-SV components encode both conflicts and discrimination) motivates treating different spectral bands differently rather than uniformly. The combination of spectral-aware regularization, fine-grained routing, and block-level adaptation achieves up to 4.4% improvement over state-of-the-art, making multi-task LoRA practical for real-world deployments.

REFERENCES

Ahmed Agiza, Marina Neseem, and Sherief Reda. Mtlora: Low-rank adaptation approach for efficient multi-task learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16196–16205, June 2024.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. Free dolly: Introducing the world's first truly open instruction-tuned llm, 2023. URL https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems*, 36:10088–10115, 2023.

Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7 (2):179–188, 1936.

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*.

Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*, 2024.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. Towards a unified view of parameter-efficient transfer learning. In *International Conference on Learning Representations*, 2022.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

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. *arXiv preprint arXiv:2106.09685*, 2021.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*, 2022.

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

Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H. Chi. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1930–1939. Association for Computing Machinery, 2018. doi: 10.1145/3219819.3220007.

George Stoica, Pratik Ramesh, Boglarka Ecsedi, Leshem Choshen, and Judy Hoffman. Model merging with svd to tie the knots. *ICLR*, 2025.

Chunlin Tian, Zhan Shi, Zhijiang Guo, Li Li, and Cheng-Zhong Xu. Hydralora: An asymmetric lora architecture for efficient fine-tuning. *Advances in Neural Information Processing Systems*, 37: 9565–9584, 2024a.

Zichen Tian, Zhaozheng Chen, and Qianru Sun. Learning de-biased representations for remote-sensing imagery. In *Advances in Neural Information Processing Systems*, 2024b.

Zichen Tian, Yaoyao Liu, and Qianru Sun. Meta-learning hyperparameters for parameter efficient fine-tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2025.

Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8769–8778, 2018.

Xuyang Wei, Chunlin Tian, and Li Li. Asymlora: Harmonizing data conflicts and commonalities in mllms, 2025.

Xun Wu, Shaohan Huang, and Furu Wei. Mixture of lora experts, 2024a.

Xun Wu, Shaohan Huang, and Furu Wei. Mixture of lora experts. In *The Twelfth International Conference on Learning Representations*, 2024b.

Gui-Song Xia, Xiang Bai, Jian Ding, Zhen Zhu, Serge Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, and Liangpei Zhang. Dota: A large-scale dataset for object detection in aerial images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3974–3983, 2018.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. TIES-merging: Resolving interference when merging models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=xtaX3WyCj1.

Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. Adamerging: Adaptive model merging for multi-task learning. In *The Twelfth International Conference on Learning Representations*, 2024.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 57755–57775. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/yu24p.html.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Nikos Karampatziakis, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adalora: Adaptive budget allocation for parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.10512*, 2023.