MERGING WHAT MATTERS: INPUT-ADAPTIVE AND LAYER-AWARE MODEL MERGING

Anonymous authors
Paper under double-blind review

ABSTRACT

Model merging provides an efficient approach to consolidating the complementary capabilities of multiple task-specific models without retraining from scratch. Existing methods often combine complete task updates or weight entire experts without accounting for the different contributions of LoRA-adapted layers, hence introducing parameter conflicts and cross-task interference. To address these limitations, in this paper, we propose Merging What Matters (MWM), an input-adaptive, layer-aware model merging framework. MWM first identifies task-critical LoRA updates within each expert and attenuates less informative auxiliary updates. It then employs a lightweight router to estimate input-dependent expert relevance and select the most appropriate experts for each input. Finally, the selected experts are dynamically integrated using differentiated layer-wise rules that preserve the core knowledge of the primary expert while conservatively incorporating complementary information from other experts. Extensive experiments across natural language processing and computer vision benchmarks demonstrate that MWM consistently outperforms the SOTA model merging baselines, achieves performance comparable to or better than task-specific fine-tuned experts, and retains competitive out-of-domain generalization. Code is provided in the supplementary material.

1 INTRODUCTION

In recent years, large language models (LLMs) have demonstrated remarkable performance and served as backbones for many downstream tasks (Brown et al., 2020; Touvron et al., 2023). Parameter-efficient fine-tuning, most commonly LoRA (Hu et al., 2022), makes it practical to derive many task-specific experts from a shared backbone, each represented by a compact set of layer-wise parameter updates. However, the growing number of specialized experts makes independent deployment and maintenance increasingly costly, particularly when a single system must serve inputs from diverse tasks. This motivates consolidating their complementary capabilities without training a new model from scratch. Model merging offers a practical route by integrating the parameter updates of multiple fine-tuned experts through parameter-level operations.

Model-merging methods have evolved from fixed parameter aggregation to input-dependent expert composition. Weight-averaging methods, such as Fisher Merging (Matena & Raffel) and RegMean (Jin et al., 2022), improve naive averaging by incorporating parameter importance or activation statistics, but use globally fixed aggregation rules. Task-vector-based methods subsequently represent each expert as a parameter update relative to a shared base model. Task Arithmetic (He et al., 2024) composes task vectors, Ties-Merging (Yadav et al., 2023) addresses sign conflicts, and AdaMerging (Yang et al., 2023) learns adaptive coefficients. Preprocessing-based methods, such as DARE (Yu et al., 2024), further reduce interference by sparsifying task vectors and rescaling retained updates. Recently, router-based methods, including Twin-Merging (Lu et al., 2024) and Free-Merging (Zheng & Wang, 2025a), dynamically select or weight experts according to the input. Such routing is particularly useful when the task identity of a test input is unknown.

Despite this progress, merging specialized experts for heterogeneous inputs can still dilute task-specific knowledge and introduce interference. Three challenges remain. First, at the *intra-expert refinement level*, operations on whole parameter tensors or task vectors often fail to reveal which adapted layers sustain an expert's source-task capability. Treating critical and lower-contribution updates alike can weaken specialization and carry conflicts into the merge. Second, at the *inter-expert allocation level*, fixed coefficients cannot reflect the different expertise required by each input. When

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

task identity is unknown, expert relevance must be estimated efficiently without evaluating every full expert. Third, at the *layer-wise integration level*, selecting relevant experts does not determine how their layer updates should be combined. Uniform integration can dilute the most relevant expert's critical knowledge and obscure complementary information from the others.

To address these challenges, we propose **Merging What Matters (MWM)**, an input-adaptive, layer-aware framework that links expert refinement, lightweight routing, and asymmetric integration. First, *Layer Importance-based Expert Refinement (LIER)* measures the effect of ablating each LoRA-adapted layer on source-task validation performance. It partitions each expert into task-critical core updates and lower-contribution auxiliary updates, retaining the latter for selective use during merging. Second, *Strengthened Expert Routing (SER)* trains a lightweight MLP from scratch on fixed-dimensional input features, using each training sample's source expert as its label. For each input, it predicts expert weights and selects the top-k candidates without running every full expert. Third, *Layer-Adaptive Dynamic Merging (LDM)* combines LIER's layer partition with SER's input-dependent weights: the primary expert contributes its core updates, while the other selected experts contribute attenuated auxiliary updates. This asymmetric rule preserves the dominant task-specific capability while incorporating complementary information and reducing cross-task interference. The three stages address distinct decisions: which layer updates matter within an expert, which experts matter for an input, and how their updates should be combined.

Our main experiments use LoRA experts, the most common form of task-specific adaptation. We evaluate MWM on NLP and CV benchmarks, and further examine a larger language model, larger expert pools, out-of-domain tasks, and heterogeneous task types. Results consistently demonstrate its effectiveness in improving merging performance while preserving both task-specific capabilities and out-of-domain generalization. In addition, a supplementary study in Appendix C.5 shows that MWM remains effective when applied to fully fine-tuned experts.

Our contributions are summarized as follows:

- We propose an input-adaptive, layer-aware merging framework that unifies expert refinement, lightweight routing, and dynamic merging to enable effective multi-task model integration.
- We introduce LIER, SER, and LDM to identify core and auxiliary layer updates, route inputs with a lightweight model, and combine primary and secondary contributions asymmetrically.
- Experiments across modalities, model scales, and task settings demonstrate improved merging performance while retaining specialized capabilities and competitive generalization.

2 Related Work

Early data-aware methods, such as Fisher-Merging and RegMean, merge model parameters using additional statistics or calibration data. Another line of work studies model interpolation through linear mode connectivity (LMC) (Draxler et al., 2018; Frankle et al., 2020; Garipov et al., 2018), although the LMC assumption may not always hold for independently fine-tuned models (Hu et al., 2022; Liu et al., 2022; Pei & Wang, 2023; Pei et al., 2024). *Classical static parameter-space methods*, including Weight Averaging (Wortsman et al., 2022), Task Arithmetic , TIES-Merging, and DARE-TIES (Yu et al., 2023; Yadav et al., 2023), directly combine task-specific parameters but remain susceptible to cross-task interference. *Importance- and structure-aware methods*, such as SA-Merging (Park et al., 2026) and AdaRank (Lee et al., 2026), mitigate such interference by selecting salient parameters or singular components. *Low-rank LoRA merging methods*, including CircuMerge (Zhang et al., 2026) and CtM (He et al., 2026), preserve compact low-rank structures during merging. *Learned and alignment-based methods*, including AdaMerging, MetaMerging (Chen et al., 2026), Expert Merging++ (Zhang et al., 2025a), and Retrofitting (Luenam et al., 2025), learn merging coefficients or align task-specific representations. Finally, *input adaptive methods*, such as MASS (Crisostomi et al., 2025), Twin-Merging, FREE-Merging, and SyMerge (Jung et al., 2026), dynamically select or compose task knowledge according to each input. More detailed discussion is provided in Appendix B.

Unlike prior approaches that operate on complete updates or complete experts, MWM introduces input-dependent layer-wise integration. It explicitly separates the refinement of each expert from the allocation of expert contributions and the subsequent integration of their updates.

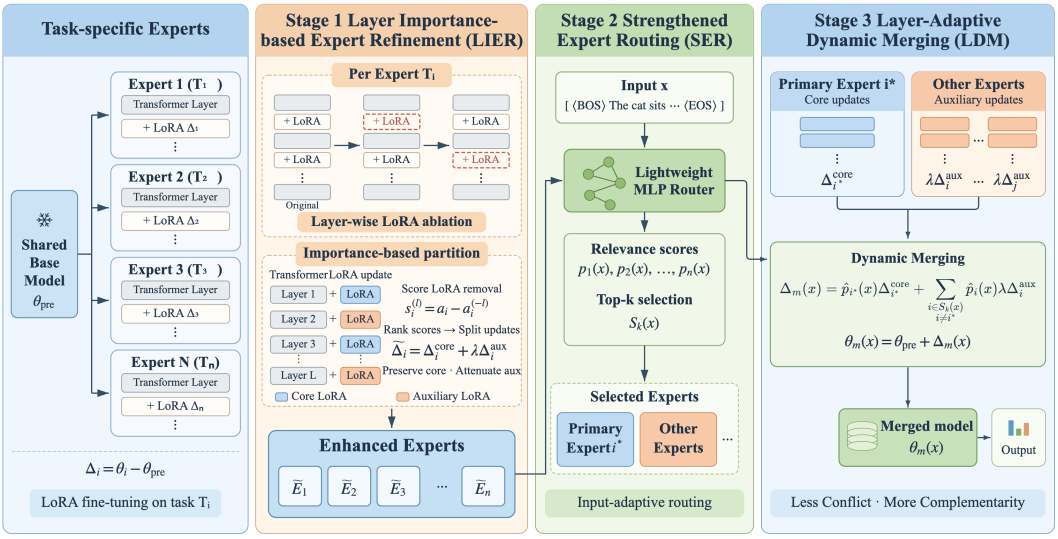


Figure 1: Overview of the proposed MWM framework.

3 METHOD

This section presents the workflow of MWM, as illustrated in Figure 1. Section 3.1 first formulates input-adaptive model merging under heterogeneous test distributions. Section 3.2 then analyzes the non-uniform contributions of LoRA-adapted layers and motivates selective, layer-wise integration. Based on this analysis, Section 3.3 identifies the task-critical components of each expert, and Section 3.4 selects the experts relevant to each input. Finally, Section 3.5 dynamically integrates the selected experts using an asymmetric layer-wise strategy. These components form a progressive pipeline from problem formulation and theoretical analysis to expert refinement, expert selection, and knowledge integration.

3.1 PROBLEM FORMULATION

Following the standard model-merging setting, we consider N tasks $\mathcal{T} = \{T_i\}_{i=1}^N$. For each task T_i, an expert $f(x; \boldsymbol{\theta}_i)$ is obtained by fine-tuning the same pretrained model $f(x; \boldsymbol{\theta}_{\text{pre}})$ on $\mathcal{D}_i^{\text{train}}$. Since LoRA is adopted for task-specific adaptation, expert i is represented by

$$\boldsymbol{\Delta}_i = \boldsymbol{\theta}_i - \boldsymbol{\theta}_{\text{pre}}, \qquad \boldsymbol{\theta}_i = \boldsymbol{\theta}_{\text{pre}} + \boldsymbol{\Delta}_i. \tag{1}$$

In practical scenarios, test inputs may originate from heterogeneous and unknown task distributions. We model the test distribution as

$$\mathcal{D}_{\text{test}} = \sum_{i=1}^{N} \pi_i \mathcal{D}_i^{\text{test}}, \qquad \pi_i > 0, \quad \sum_{i=1}^{N} \pi_i = 1. \tag{2}$$

The task identity of each input is unavailable at inference time. Therefore, a static combination shared by all inputs may be suboptimal under such a heterogeneous distribution.

Given an input x, we seek an input-dependent merging function $\mathcal{F}$:

$$\boldsymbol{\theta}_m(x) = \mathcal{F}\left(\boldsymbol{\theta}_{\text{pre}}, \boldsymbol{\Delta}_1, \ldots, \boldsymbol{\Delta}_N; x\right) = \boldsymbol{\theta}_{\text{pre}} + \sum_{i=1}^{N} \alpha_i(x) \boldsymbol{\Delta}_i, \quad \alpha_i(x) \geq 0, \quad \sum_{i=1}^{N} \alpha_i(x) = 1. \tag{3}$$

Conceptually, the desired merging function minimizes the overall expected risk over the heterogeneous test distribution:

$$\min_{\mathcal{F}} \ \mathbb{E}_{(x,y)\sim\mathcal{D}_{\text{test}}} \left[\ell\left(f(x; \boldsymbol{\theta}_m(x)), y\right)\right]. \tag{4}$$

Directly combining complete expert updates may introduce irrelevant or conflicting knowledge because different experts and adapted layers contribute unequally to each input. The following analysis motivates selective expert and layer integration.



Figure 2: Validation-performance variation after individually removing LoRA updates from different adapted layers. Positive values indicate performance degradation after removal, while negative values indicate performance improvement. The results demonstrate the non-uniform contribution of LoRA layers to task-specific expertise.

3.2 Theoretical Motivation and Analysis

Observation 1: LoRA layers contribute non-uniformly to expert specialization. Let $\mathbf{\Delta}_i^{(l)}$ denote the LoRA update of expert i at the l-th adapted layer, such that $\mathbf{\Delta}_i = \sum_{l=1}^{L} \mathbf{\Delta}_i^{(l)}$. If all adapted layers were equally important, removing any individual update would produce a similar performance change. However, our preliminary experiment reveals a substantially different pattern.

Specifically, we remove one LoRA-adapted layer at a time and evaluate the resulting expert on its task-specific validation set. As shown in Figure 2, different removals lead to markedly different performance changes. Removing some layers causes severe degradation, whereas removing others preserves or even improves performance. Moreover, the influential layers differ across experts. These results suggest that only a subset of layers is critical to the specialized capability of each expert. More detailed layer-wise results are reported in Tables 8 and 9.

This observation motivates preserving task-critical layers while attenuating those that contribute less to the source task. We do not regard the latter as entirely redundant: although their contribution to source-task specialization is limited, they may still encode transferable information that complements other experts. Appendix D.2 formalizes this observation. Under block smoothness, each ablation score decomposes into a gradient-alignment term and a curvature-weighted update term (Lemma D.2.1), so update norm or LoRA rank alone is not a valid proxy for importance (Proposition D.2.10). Removing a layer provably improves source-task risk when its adverse gradient alignment dominates the smoothness remainder (Theorem D.2.3), and retaining the top-scoring layers is optimal under an additive surrogate and near-optimal under bounded cross-layer interaction (Theorems D.2.5 and D.2.5). Finally, the validation-based ranking recovers the population ranking with $O(\gamma^{-2} \log L)$ samples, where γ is the score gap at the core boundary (Corollary D.2.6).

Observation 2: layer-wise merging provides a more expressive integration space. Model-level merging assigns one coefficient to the complete update of each expert, whereas layer-wise merging allows expert contributions to vary across layers:

$$\mathbf{\Delta}_{\text{model}}(x) = \sum_{l=1}^{L} \sum_{i=1}^{N} \alpha_i(x) \mathbf{\Delta}_i^{(l)}, \qquad \mathbf{\Delta}_{\text{layer}}(x) = \sum_{l=1}^{L} \sum_{i=1}^{N} \alpha_i^{(l)}(x) \mathbf{\Delta}_i^{(l)}. \tag{5}$$

Let $\mathcal{H}_{\text{model}}$ and $\mathcal{H}_{\text{layer}}$ denote their corresponding feasible parameter spaces. Model-level merging is recovered by setting $\alpha_i^{(l)}(x) = \alpha_i(x)$ for all l. Therefore,

$$\mathcal{H}_{\text{model}} \subseteq \mathcal{H}_{\text{layer}} \quad \Longrightarrow \quad \min_{\boldsymbol{\theta} \in \mathcal{H}_{\text{layer}}} \mathcal{R}(\boldsymbol{\theta}) \leq \min_{\boldsymbol{\theta} \in \mathcal{H}_{\text{model}}} \mathcal{R}(\boldsymbol{\theta}). \tag{6}$$

This inclusion indicates that layer-wise merging has greater representational flexibility, and Appendix D.3 makes the gain precise: under a local block-separable quadratic model, the excess risk of model-level merging equals the curvature-weighted dispersion of the expert mixtures preferred by individual layers (Theorem D.3.2). Layer-wise merging therefore helps exactly when different layers favor different experts, and offers no advantage when they agree. This flexibility is not free. Learning

all $N \times L$ input-dependent coefficients independently is expensive, and it inflates the uniform validation error from $O(\sqrt{\log M/n})$ to $O(\sqrt{L \log M/n})$ for M candidate mixtures (Theorem D.4.1). MWM therefore uses a structured approximation: it partitions each expert into core and auxiliary components, selects a small number of relevant experts, and assigns different merging roles to these components. Both restrictions are controlled: the top-k truncation error is bounded by the discarded routing mass (Proposition D.3.7), and the worst-case perturbation from auxiliary experts is bounded by a term linear and quadratic in λ (Theorem D.3.8).

3.3 Layer Importance-based Expert Refinement

We first identify the neuron groups that are important to each expert's specialized capability. In our implementation, a neuron group is instantiated as a LoRA-adapted layer/block, and importance estimation is therefore performed at the layer level. This block-wise instantiation provides a practical realization of layer-wise refinement while avoiding the excessive computational cost of evaluating individual neurons.

For expert i, we evaluate the complete model and its layer-ablated variant on $\mathcal{D}_i^{\mathrm{val}}$:

$$a_i = \mathcal{M}\left(\boldsymbol{\theta}_i; \mathcal{D}_i^{\mathrm{val}}\right), \qquad a_i^{(-l)} = \mathcal{M}\left(\boldsymbol{\theta}_i - \boldsymbol{\Delta}_i^{(l)}; \mathcal{D}_i^{\mathrm{val}}\right), \tag{7}$$

where $\mathcal{M}$ denotes the task-specific evaluation metric. The importance of layer l is defined by its removal-induced performance degradation:

$$s_i^{(l)} = a_i - a_i^{(-l)}, \qquad \mathcal{L}_i^{\mathrm{core}} = \mathrm{TopK}\left(\{s_i^{(l)}\}_{l=1}^{L}, \lceil \rho L \rceil\right). \tag{8}$$

A larger $s_i^{(l)}$ indicates that layer l is more important to expert i, and $\rho \in (0, 1]$ controls the proportion of retained core layers. The remaining layers constitute $\mathcal{L}_i^{\mathrm{aux}} = \{1, \ldots, L\} \setminus \mathcal{L}_i^{\mathrm{core}}$.

Let $\boldsymbol{M}_i^{\mathrm{core}}$ be the binary mask associated with $\mathcal{L}_i^{\mathrm{core}}$. We decompose the expert update as

$$\boldsymbol{\Delta}_i^{\mathrm{core}} = \boldsymbol{M}_i^{\mathrm{core}} \odot \boldsymbol{\Delta}_i, \qquad \boldsymbol{\Delta}_i^{\mathrm{aux}} = (\mathbf{1} - \boldsymbol{M}_i^{\mathrm{core}}) \odot \boldsymbol{\Delta}_i, \tag{9}$$

where $\odot$ denotes layer-wise masking. The core component preserves the expert's dominant task-specific capability, while the auxiliary component contains less task-critical parameters that can be incorporated conservatively as complementary information.

3.4 Strengthened Expert Routing

After decomposing the experts, we train a lightweight MLP router from scratch to estimate the relevance of each expert to the current input. Unlike routing methods that rely on hidden representations extracted from a pretrained backbone, our router operates independently of the expert models. Let $\boldsymbol{z}(x)$ denote the fixed-dimensional input feature provided to the router after task-specific preprocessing. The routing probabilities are computed as

$$\boldsymbol{p}(x) = \mathrm{softmax}(g_\phi(\boldsymbol{z}(x))), \qquad \mathcal{L}_{\mathrm{router}} = -\mathbb{E}_{(x,t)\sim\mathcal{D}_{\mathrm{route}}}\left[\log p_t(x)\right]. \tag{10}$$

Here, g_ϕ denotes a lightweight routing network parameterized by ϕ, and t is the identity of the expert associated with training sample x. The router is trained with cross-entropy loss while all expert parameters remain frozen.

At inference time, we select the top-k experts according to their routing probabilities and renormalize the corresponding weights:

$$\mathcal{S}_k(x) = \mathrm{TopK}\big(\boldsymbol{p}(x), k\big), \qquad \widehat{p}_i(x) = \frac{p_i(x)}{\sum_{j \in \mathcal{S}_k(x)} p_j(x)}, \quad i \in \mathcal{S}_k(x). \tag{11}$$

For each unselected expert $i \notin \mathcal{S}_k(x)$, we set $\widehat{p}_i(x) = 0$. This lightweight routing mechanism enables input-dependent expert selection without requiring an additional forward pass through the pretrained backbone, while restricting the subsequent merging operation to the most relevant experts.

3.5 Layer-Adaptive Dynamic Merging

Given $\mathcal{S}_k(x)$, we select the highest-ranked expert as the primary expert:

$$i^\star = \arg\max_{i \in \mathcal{S}_k(x)} \widehat{p}_i(x). \tag{12}$$

The primary expert contributes its core update to preserve the capability most relevant to the current input, while the remaining experts provide only attenuated auxiliary updates. These auxiliary contributions supply complementary information without directly combining highly task-specific core components across experts.

The dynamically merged update is

$$\boldsymbol{\Delta}_m(x) = \widehat{p}_{i^\star}(x)\boldsymbol{\Delta}_{i^\star}^{\mathrm{core}} + \lambda \sum_{i \in \mathcal{S}_k(x)\setminus\{i^\star\}} \widehat{p}_i(x)\boldsymbol{\Delta}_i^{\mathrm{aux}}, \qquad 0 \le \lambda \le 1, \tag{13}$$

and the final input-dependent model is

$$\boldsymbol{\theta}_m(x) = \boldsymbol{\theta}_{\mathrm{pre}} + \boldsymbol{\Delta}_m(x). \tag{14}$$

The resulting strategy is input-adaptive because both the selected experts and their coefficients vary with x, and layer-aware because core and auxiliary parameter groups follow different merging rules. Overall, importance estimation reduces the layer search space, top-k routing reduces the expert search space, and asymmetric integration preserves the dominant expertise while incorporating complementary knowledge conservatively.

4 Experiment

We comprehensively evaluate MWM in cross-task, cross-domain, cross-training-configuration, and domain-shift settings. Section 4.1 presents the experimental setup, including datasets, model configurations, baselines, and evaluation metrics. Section 4.2 presents comparative results on NLP and CV benchmarks. Section 4.3 further examines scalability to a larger expert pool, as well as cross-domain and cross-task merging. Section 4.4 analyzes the contribution of each proposed component through ablation studies.

4.1 Experimental Setup

We evaluate MWM on 8 NLP datasets—COLA, RTE, MNLI, QNLI, QQP, MRPC, SST-2 and SST-B(Wang et al., 2018)—and 10 CV datasets: MNIST (LeCun et al., 2010), EuroSAT (Helber et al., 2019), CIFAR-10 (Krizhevsky, 2009), CarBrands50, Fruits100, GTSRB (Stallkamp et al., 2011), DTD (Cimpoi et al., 2014), RESISC45 (Cheng et al., 2017), GRABAGE, and PLANTS. For NLP experiments, we merge task-specific Qwen2.5-0.5B-Instruct models (Yang et al., 2024; Team, 2024); for CV experiments, we merge fine-tuned ViT-Base-Patch16-224 models initialized from a shared pretrained checkpoint. All images are resized to 224×224 during both training and inference.

For LoRA fine-tuning, we use $(r, \alpha) = (8, 32)$ for NLP and $(16, 16)$ for CV, with a dropout rate of 0.1, targeting *gate-proj*, *up-proj*, and *down-proj* modules for Qwen2.5-0.5B-Instruct and *intermediate-dense* and *output-dense* modules for ViT-Base-Patch16-22. For MWM, a grid search on the validation sets yields $\lambda = 0.1$ and $k = 5$; the core-layer selections determined by ρ are reported in Tables 8 and 9, which are used across all experiments. We consider five groups of model-merging baselines. *Classical Static Parameter-Space Merging* includes **Weight Averaging** (Wortsman et al., 2022), **Task Arithmetic** (He et al., 2024), **TIES-Merging** (Yadav et al., 2023), and **DARE-TIES** (Yu et al., 2023; Yadav et al., 2023). *Importance and Structure-Aware Merging* includes **SA-Merging** (Park et al., 2026) and **AdaRank** (Lee et al., 2026). *Low-Rank and Efficient LoRA Merging* includes **CircuMerge** (Zhang et al., 2026) and **CtM** (He et al., 2026). *Learned and Alignment-Based Merging* includes **MetaMerging** (Chen et al., 2026), **Expert Merging++** (Zhang et al., 2025a), and **Retrofitting** (Luenam et al., 2025). Finally, *input-adaptive merging* includes **MASS** (Crisostomi et al., 2025), **Twin-Merging** (Lu et al., 2024), **FREE-Merging** (Zheng & Wang, 2025b), and **SyMerge** (Jung et al., 2026). Unless otherwise stated, all methods are evaluated under the same experimental protocol to ensure a fair comparison.

We use classification Accuracy as the evaluation metric for all tasks. We additionally measure throughput, model storage, and inference latency on a single NVIDIA A100 GPU using FP32 precision and a batch size of 32. Detailed dataset statistics, training configurations, and implementation details are provided in Appendix C.

4.2 Comparative Evaluation

For NLP experiments, we apply our proposed method and all baselines to the Qwen2.5-0.5B-Instruct model (Yang et al., 2024; Team, 2024), whereas for CV tasks, the same methods are evaluated using

Table 1: Accuracy comparison of pretrained, multi-task, task-specific, and model-merging methods across GLUE tasks (%).

MODEL	CoLA	MNLI	MRPC	QNLI	QQP	RTE	SST-2	STS-B	Avg.
Reference Models									
Pretrained	69.8	34.6	62.5	48.6	64.8	54.5	84.2	22.4	55.2
Multi-task	76.6	78.2	84.8	83.0	81.8	76.5	94.0	81.6	82.1
Individual	76.2	78.0	84.1	84.2	84.0	77.3	92.6	82.8	82.4
Classical Static Parameter-Space Merging									
Weight Averaging	71.5	45.5	68.1	57.5	69.6	60.2	86.7	37.5	62.1
Task Arithmetic	72.7	52.9	71.9	63.6	72.9	64.1	88.3	47.8	66.8
TIES-Merging	72.9	54.7	72.8	65.0	73.6	65.0	88.7	50.2	67.9
DARE-TIES	73.2	56.4	73.7	66.4	74.4	65.9	89.1	52.6	69.0
Importance and Structure-Aware Merging									
SA-Merging	73.3	57.3	74.1	67.1	74.8	66.3	89.3	53.8	69.5
AdaRank	74.6	65.1	78.1	73.5	78.2	70.4	91.1	64.7	74.5
Low-Rank and Efficient LoRA Merging									
CircuMerge	72.2	49.9	70.3	61.1	71.5	62.5	87.6	43.5	64.8
CtM	73.5	58.6	74.8	68.2	75.4	67.0	89.6	55.6	70.3
Learned and Alignment-Based Merging									
MetaMerging	75.1	68.6	79.9	76.4	79.8	72.3	91.8	69.5	76.7
Expert Merging++	75.2	69.5	80.3	77.1	80.2	72.7	92.0	70.7	77.2
Retrofitting	74.7	66.0	78.6	74.2	78.6	70.9	91.3	65.9	75.0
Input Adaptive Merging									
MASS	75.7	72.5	81.9	79.6	81.5	74.3	92.7	74.9	79.1
Twin-Merging	75.9	73.8	82.6	80.6	82.1	75.0	93.0	76.8	80.0
FREE-Merging	76.1	75.1	83.2	81.7	82.7	75.7	93.3	78.6	80.8
SyMerge	76.3	76.5	83.9	82.8	83.2	76.3	93.6	80.4	81.6
Layer-Aware Input-Adaptive Merging									
MWM/Ours	76.8 ↑0.6	79.5 ↑1.5	85.5 ↑1.4	85.3 ↑1.1	84.6 ↑0.6	77.9 ↑0.6	94.3 ↑1.7	84.6 ↑1.8	83.6 ↑1.2

the pretrained ViT-Base-Patch16-224 architecture. Unless stated otherwise, all input images are uniformly resized to 224×224 resolution for both training and inference.

Results. As shown in Tables 1 and 2, our method not only consistently outperforms existing model-merging baselines on both NLP and CV tasks, but also notably surpasses task-specific fine-tuned models in single-task performance. Moreover, as reported in Table 12, our method requires less storage and achieves shorter inference time than Twin-Merging and FREE-Merging. Although it introduces a modest efficiency overhead compared with static merging methods, this overhead primarily arises from the lightweight router used for input-adaptive expert selection. These results demonstrate that our approach achieves a favorable balance between predictive performance and computational efficiency, while effectively mitigating parameter conflicts and task interference in model merging. In contrast to prior methods that often fail to scale due to these challenges, our approach exhibits superior robustness and scalability.

4.3 Comparative Analysis

Task Diversity. Tables 1 and 2 demonstrate the effectiveness of MWM across diverse NLP and CV tasks spanning digits, remote sensing, objects, vehicles, plants, waste, and traffic signs. Table 3 further shows that MWM can integrate generative and discriminative tasks within a unified framework while outperforming the standalone fine-tuned experts on average. Together, these results demonstrate the scalability of MWM across model sizes, modalities, domains, and task paradigms.

Large-Model Scalability and OOD Generalization. We evaluate MWM on Qwen2.5-7B-Instruct using the same setting as Qwen2.5-0.5B-Instruct. As shown in Table 4, MWM consistently outperforms the fine-tuned baselines, demonstrating its scalability to larger models. Moreover, MWM achieves 67.5% accuracy on MMLU, only 0.4 percentage points below the pretrained model while substantially outperforming Weight Averaging and Task Arithmetic (Tables 5 and 7). This result shows that MWM largely preserves the generalization capability of the pretrained model after integrating multiple task experts.

Table 2: Accuracy comparison of pretrained, multi-task, task-specific, and model-merging methods across CV tasks (%).

MODEL	MNIST	EuroSAT	CIFAR-10	CAR	FRUITS100	GTSRB	DTD	RESISC45	GRABAGE	PLANTS	Avg.
Reference Models											
Pretrained	56.5	63.9	92.8	23.0	78.6	37.8	60.7	61.6	72.0	87.9	63.5
Multi-task	95.5	93.9	94.9	32.7	80.2	92.2	73.6	83.9	68.5	88.1	80.4
Individual	98.9	95.1	98.2	19.0	83.1	91.0	68.8	83.8	76.4	90.4	80.5
Classical Static Parameter-Space Merging											
Weight Averaging	72.8	69.8	94.5	18.0	80.4	48.9	62.6	67.9	73.6	89.1	67.8
Task Arithmetic	80.9	73.0	96.8	20.0	81.1	55.7	65.2	71.0	75.9	89.2	70.9
TIES-Merging	84.5	77.6	97.1	23.0	81.5	63.5	67.3	74.0	76.0	89.4	73.4
DARE-TIES	84.9	78.1	97.1	23.3	81.5	64.2	67.6	74.3	76.0	89.5	73.7
Importance and Structure-Aware Merging											
SA-Merging	85.6	79.0	97.2	23.9	81.6	65.8	68.0	75.0	76.0	89.5	74.2
AdaRank	96.7	93.4	98.0	33.2	82.9	89.8	74.6	84.4	76.3	90.3	82.0
Low-Rank and Efficient LoRA Merging											
CircuMerge	82.3	74.9	96.9	21.2	81.3	58.8	66.1	72.2	75.9	89.3	71.9
CtM	83.8	76.7	97.0	22.4	81.4	61.9	66.9	73.4	76.0	89.4	72.9
Learned and Alignment-Based Merging											
MetaMerging	90.8	85.8	97.6	28.3	82.2	77.0	71.1	79.4	76.2	89.9	77.8
Expert Merging++	92.6	88.1	97.7	29.8	82.4	80.9	72.2	80.9	76.2	90.0	79.1
Retrofitting	88.5	82.7	97.4	26.3	81.9	72.0	69.7	77.4	76.1	89.7	76.2
Input Adaptive Merging											
MASS	97.1	92.9	98.1	31.4	82.9	88.6	72.8	82.6	76.4	90.3	81.3
Twin-Merging	95.3	90.7	97.9	30.2	82.7	84.9	71.9	81.3	76.3	90.2	80.1
FREE-Merging	98.2	94.2	98.1	32.2	83.0	90.7	73.3	83.4	76.4	90.4	82.0
SyMerge	98.5	94.7	98.2	32.4	83.1	91.5	73.4	83.6	76.4	90.4	82.2
Layer-Aware Input-Adaptive Merging											
MWM/Ours	98.9 ↑0.0	95.8 ↑0.7	98.3 ↑0.1	34.8 ↑15.8	83.2 ↑0.1	94.2 ↑3.2	74.5 ↑5.7	85.0 ↑1.2	76.5 ↑0.1	90.5 ↑0.1	83.2 ↑2.7

Table 3: Accuracy across different task types (%).

Method	Classification	Generation
Individual	77.3	49.5
MWM/ours	79.1 ↑1.8	51.5 ↑2.0

Scalability to More Tasks. As shown in Figure 3, MWM maintains stable normalized performance as the number of merged tasks increases, whereas competing methods gradually degrade. It also requires less storage than Twin-Merging and achieves shorter inference times, as reported in Table 12. These results demonstrate that MWM can efficiently integrate a larger number of experts.

4.4 Ablation Studies

To assess the contribution of each component in our method, we conduct ablation studies across 8 NLP tasks and 10 CV tasks, focusing on the effects of three core modules: LIER, SER, and LDM.

LIER Analysis. LIER identifies task-critical LoRA-adapted layers and partitions each expert update into core and auxiliary components. As shown in Table 6, this variant reduces the average performance from 83.6% to 79.5% on NLP tasks and from 83.2% to 78.9% on CV tasks. The resulting performance is also lower than the task-specific fine-tuned references of 82.4% and 80.5%, respectively. These results indicate that explicitly preserving task-critical updates before merging helps retain expert specialization and reduce interference from less informative components.

SER Analysis. SER estimates the relevance of each expert to the current input and provides input-dependent coefficients for subsequent dynamic merging. To evaluate its contribution, we remove the router and replace its input-dependent coefficients with a fixed expert combination. As shown in Table 6, the average performance decreases substantially from 83.6% to 74.5% on NLP tasks and from 83.2% to 74.6% on CV tasks. This degradation demonstrates that a fixed expert combination cannot adequately accommodate heterogeneous input requirements. By contrast, SER identifies the top-k experts most relevant to each input and supplies their normalized routing coefficients to LDM, enabling more precise expert allocation while limiting interference from unrelated experts.

Table 4: Accuracy comparison of our method on Qwen2.5-7B-Instruct models (%).

MODEL	MNLI	MRPC	QNLI	QQP	RTE	Avg.
Pretrained	53.8	62.0	50.0	68.6	58.4	58.6
MULTI-TASK	84.8	86.4	86.4	88.8	86.6	86.6
Individual	89.2	89.0	92.0	86.0	91.7	89.6
MWM/ours	90.1 ↑0.9	89.2 ↑0.2	92.1 ↑0.1	86.3 ↑0.3	91.7 ↑0.0	89.9 ↑0.3

Table 5: Average accuracy on out-of-domain MMLU benchmark tasks (%).

MMLU Task	Qwen2.5-7B-Instruct	Weight Averaging	Task Arithmetic	MWM/ours
Avg	67.9	67.3 ↓0.6	63.6 ↓4.3	67.5 ↓0.4

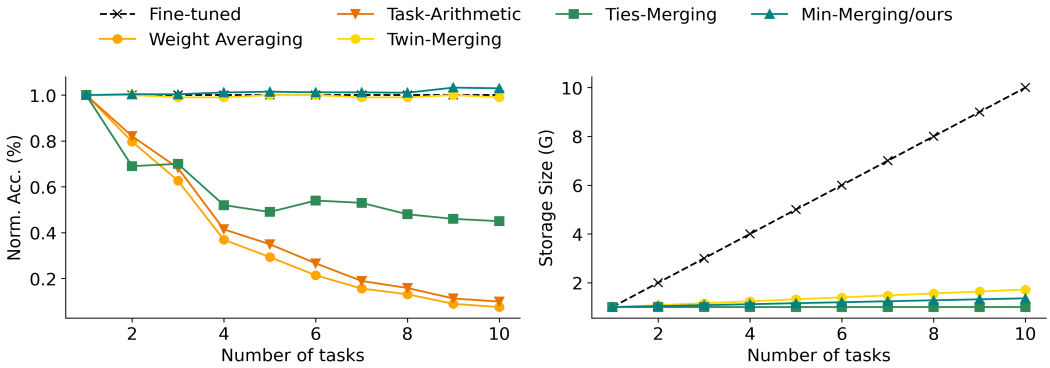


Figure 3: Scalability analysis of normalized task performance and storage footprint as the number of merged tasks increases. Performance is normalized by the corresponding standalone fine-tuned expert, which is set to 100%; values above 100% indicate improvements over the fine-tuned reference.

Table 6: Accuracy ablation study of LIER, SER, and LDM across NLP and CV tasks (%).

Method	NLP (Avg.)	CV (Avg.)
Individual	82.4	80.5
w/o LIER	79.5 ↓4.1	78.9 ↓4.3
w/o SER	74.5 ↓7.1	74.6 ↓8.6
w/o LDM	81.3 ↓2.5	80.6 ↓2.6
MWM/ours	83.6	83.2

LDM Analysis. LDM integrates the experts selected by SER using differentiated rules for core and auxiliary updates. Removing LDM replaces this asymmetric, layer-aware strategy with uniform merging, thereby treating all selected expert updates equivalently. As reported in Table 6, this modification decreases the average performance from 83.6% to 81.3% on NLP tasks and from 83.2% to 80.6% on CV tasks. Although the reductions are smaller than those caused by removing SER, the consistent degradation across both modalities confirms the benefit of differentiated integration. In particular, LDM preserves the core updates of the primary expert while conservatively incorporating auxiliary updates from the remaining selected experts, improving the retention of task-specific knowledge and reducing cross-expert interference.

5 Conclusion

We presented MWM, an input-adaptive layer-wise framework for merging task-specific LoRA experts. Its central idea is to determine which layers matter within each expert and which experts matter for each input, then use both decisions to guide merging. MWM refines each expert by emphasizing task-relevant updates, uses a lightweight router to select relevant experts for each input, and integrates the selected updates with differentiated layer-wise rules. Extensive experiments on NLP and CV benchmarks show that MWM consistently outperforms the evaluated merging baselines. Additional analyses indicate that the method scales to larger language models, supports a larger pool of experts, and retains competitive out-of-domain performance.

AI USE STATEMENT

Generative AI tools were used only to assist with language polishing, including grammar correction, wording refinement, and improving readability. They were not used for research ideation, experimental design, data analysis, or scientific interpretation. All AI-assisted content was carefully reviewed and verified by the authors, who take full responsibility for the final content of this work.

ETHICS STATEMENT

This work uses publicly available datasets and pretrained models and does not involve human-subject studies or the collection of personal data. Model merging may preserve or amplify biases and harmful behaviors inherited from the constituent models. We therefore recommend evaluating the merged model for fairness, safety, and privacy risks before deployment. All datasets and models are used in accordance with their applicable licenses and terms.

REPRODUCIBILITY STATEMENT

We provide the model configurations, datasets, training settings, hyperparameters, and evaluation protocols in the main text and Appendix C. Detailed results, ablation studies, and full-parameter fine-tuning experiments are also included in the appendix. The assumptions and proofs supporting the theoretical analysis are provided in Appendix D. Anonymous source code is provided in the supplementary material to facilitate reproducibility.

REFERENCES

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. URL https://arxiv.org/abs/2005.14165.

Jun Chen, Qin Zhang, Weizhi Zhang, Xiao Luo, Philip S. Yu, and Ziyue Qiao. Learn to merge: Meta-learning for adaptive multi-task model merging. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://github.com/cjcj46262/MetaMerging.

Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883, 2017.

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3606–3613, 2014.

Donato Crisostomi, Alessandro Zirilli, Antonio Andrea Gargiulo, Maria Sofia Bucarelli, Simone Scardapane, Fabrizio Silvestri, Iacopo Masi, and Emanuele Rodolà. MASS: MoErging through adaptive subspace selection. *arXiv preprint arXiv:2504.05342*, 2025. URL https://arxiv.org/abs/2504.05342.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Felix Draxler, Kambis Veschgini, Manfred Salmhofer, and Fred Hamprecht. Essentially no barriers in neural network energy landscape. In *International conference on machine learning*, pp. 1309–1318. PMLR, 2018.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *International Conference on Machine Learning*, pp. 3259–3269. PMLR, 2020.

Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry P Vetrov, and Andrew G Wilson. Loss surfaces, mode connectivity, and fast ensembling of dnns. *Advances in neural information processing systems*, 31, 2018.

Yifei He, Yuzheng Hu, Yong Lin, Tong Zhang, and Han Zhao. Localize-and-stitch: Efficient model merging via sparse task arithmetic. *Transactions on Machine Learning Research*, 2024, 2024. ISSN 2835-8856.

Zhengbao He, Ruiqi Ding, Zhehao Huang, Ruikai Yang, Tao Li, and Xiaolin Huang. Compress then merge: From multiple LoRAs into one low-rank adapter. *arXiv preprint arXiv:2606.03723*, 2026. URL https://arxiv.org/abs/2606.03723.

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2019.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

Xisen Jin, Xiang Ren, Daniel Preotiuc-Pietro, and Pengxiang Cheng. Dataless knowledge fusion by merging weights of language models. *arXiv preprint arXiv:2212.09849*, 2022.

Aecheon Jung, Seunghwan Lee, Dongyoon Han, and Sungeun Hong. SyMerge: From non-interference to synergistic merging via single-layer adaptation. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2412.19098. Also available as arXiv:2412.19098.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*, 2015.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

Rudolf Kruse, Sanaz Mostaghim, Christian Borgelt, Christian Braune, and Matthias Steinbrecher. Multi-layer perceptrons. In *Computational intelligence: a methodological introduction*, pp. 53–124. Springer, 2022.

Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

Chanhyuk Lee, Jiho Choi, Chanryeol Lee, Donggyun Kim, and Seunghoon Hong. AdaRank: Adaptive rank pruning for enhanced model merging. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2503.22178. Also available as arXiv:2503.22178.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin A Raffel. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems*, 35:1950–1965, 2022.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

Zhenyi Lu, Chenghao Fan, Wei Wei, Xiaoye Qu, Dangyang Chen, and Yu Cheng. Twin-merging: Dynamic integration of modular expertise in model merging. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://arxiv.org/abs/2406.15479.

Phoomraphee Luenam, Andreas Spanopoulos, Amit Sant, Thomas Hofmann, Sotiris Anagnostidis, and Sidak Pal Singh. Model fusion via retrofitting. *arXiv preprint arXiv:2507.00037*, 2025. URL https://arxiv.org/abs/2507.00037.

Anqi Mao, Mehryar Mohri, and Yutao Zhong. Cross-entropy loss functions: Theoretical analysis and applications. In *International conference on Machine learning*, pp. 23803–23828. PMLR, 2023.

Michael Matena and Colin Raffel. Merging models with fisher-weighted averaging, 2021. *arXiv preprint arXiv:2111.09832*.

Jungin Park, Jiyoung Lee, and Kwanghoon Sohn. Saliency-aware model merging. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2606.00511. Also available as arXiv:2606.00511.

Zhengqi Pei and Shuhui Wang. Dynamics-inspired neuromorphic visual representation learning. In *International Conference on Machine Learning*, pp. 27521–27541. PMLR, 2023.

Zhengqi Pei, Anran Zhang, Shuhui Wang, Xiangyang Ji, and Qingming Huang. Data-free neural representation compression with riemannian neural dynamics. In *Forty-first International Conference on Machine Learning*, 2024.

Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. The german traffic sign recognition benchmark: a multi-class classification competition. In *The 2011 international joint conference on neural networks*, pp. 1453–1460. IEEE, 2011.

Qwen Team. Qwen2.5: A party of foundation models, September 2024. URL https://qwenlm.github.io/blog/qwen2.5/.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023. URL https://arxiv.org/abs/2302.13971.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: Averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 23965–23998. PMLR, 2022. URL https://proceedings.mlr.press/v162/wortsman22a.html.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. Ties-merging: Resolving interference when merging models. *Advances in Neural Information Processing Systems*, 36:7093–7115, 2023.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.

Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. Adamerging: Adaptive model merging for multi-task learning. *arXiv preprint arXiv:2310.02575*, 2023.

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. *arXiv preprint arXiv:2311.03099*, 2023. URL https://arxiv.org/abs/2311.03099.

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 *Forty-first International Conference on Machine Learning*, 2024.

Dengming Zhang, Xiaowen Ma, Zhenliang Ni, Zhenkai Wu, Han Shu, Xin Jiang, and Xinghao Chen. Expert merging: Model merging with unsupervised expert alignment and importance-guided layer chunking. *arXiv preprint arXiv:2509.25712*, 2025a. URL https://arxiv.org/abs/2509.25712.

Jihai Zhang, Xiaoye Qu, Tong Zhu, and Yu Cheng. Clip-moe: Towards building mixture of experts for clip with diversified multiplet upcycling, 2025b. URL https://arxiv.org/abs/2409.19291.

Zhiming Zhang, Rong Yin, Xiaoshuai Hao, Hansong Zhang, Hao Peng, Yong Liu, Can Ma, and Dan Meng. Sketch-based low-rank model merging with shared circulant transforms. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, 2026. URL https://openreview.net/forum?id=ytlRCCl7Pf.

Shenghe Zheng and Hongzhi Wang. Free-merging: Fourier transform for efficient model merging. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3863–3873, 2025a.

Shenghe Zheng and Hongzhi Wang. FREE-Merging: Fourier transform for efficient model merging. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025b. URL https://arxiv.org/abs/2411.16815. Also available as arXiv:2411.16815.

Tong Zhu, Daize Dong, Xiaoye Qu, Jiacheng Ruan, Wenliang Chen, and Yu Cheng. Dynamic data mixing maximizes instruction tuning for mixture-of-experts, 2024a. URL https://arxiv.org/abs/2406.11256.

Tong Zhu, Xiaoye Qu, Daize Dong, Jiacheng Ruan, Jingqi Tong, Conghui He, and Yu Cheng. Llama-moe: Building mixture-of-experts from llama with continual pre-training, 2024b. URL https://arxiv.org/abs/2406.16554.

A Limitations and Future Work

Although MWM demonstrates consistent effectiveness across the evaluated natural language processing and computer vision benchmarks, the current framework assumes that all expert models share the same pretrained backbone and have compatible parameter structures. Because MWM estimates the importance of corresponding LoRA-adapted components and dynamically combines their parameter updates, it cannot be directly applied to models with heterogeneous architectures or incompatible parameterizations. Extending MWM to experts with different architectures or independently pretrained initializations would require additional mechanisms for parameter alignment, representation mapping, or knowledge transfer. We leave this more general setting for future investigation.

Several directions may further extend the scope and applicability of MWM:

- **Scaling to Larger Expert Pools:** Within the range evaluated in this study, MWM maintains stable performance as the number of merged experts increases. Experiments involving substantially larger expert pools would provide a more comprehensive understanding of its performance and computational trade-offs at scale. Future work may also explore more efficient importance estimation and expert retrieval strategies to support large-scale deployment.
- **Finer-Grained Expert Refinement:** The current implementation estimates importance at the level of LoRA-adapted layers, providing a practical balance between refinement granularity and computational cost. Finer-grained parameter grouping may enable more precise knowledge selection, but may also introduce additional estimation and storage overhead. Developing efficient refinement strategies at finer levels of granularity could further improve the flexibility and effectiveness of MWM.
- **Integration with Mixture-of-Experts Architectures:** Mixture-of-Experts (MoE) architectures (Zhang et al., 2025b; Zhu et al., 2024a;b) offer a complementary paradigm for modular model design. An interesting direction is to investigate whether the core–auxiliary decomposition and input-dependent integration strategy of MWM can be adapted to MoE systems. Such an extension may facilitate the efficient reuse and composition of specialized expert modules, although its predictive and computational benefits require dedicated empirical evaluation.

B Additional Related Work

Static Parameter-Space Merging. Early model-merging methods construct a unified model by directly combining models with the same architecture and initialization. Weight Averaging and Model Soups aggregate model parameters using fixed coefficients, providing simple and computationally

efficient baselines (Wortsman et al., 2022). Fisher-Merging instead weights parameters according to their estimated Fisher information, thereby emphasizing parameters that are more important to the predictive distribution of each expert. RegMean formulates merging as a regression problem and derives layer-wise solutions using activation statistics. Related studies motivated by linear mode connectivity further interpolate models within a shared low-loss basin (Draxler et al., 2018; Frankle et al., 2020; Garipov et al., 2018), sometimes using weight matching or optimal transport to improve parameter alignment. However, these approaches produce a static model whose expert contributions cannot adapt to individual inputs.

Task-Vector and Conflict-Aware Merging. Task Arithmetic represents task-specific knowledge as a task vector, defined as the parameter difference between a fine-tuned model and its shared pretrained initialization. Multiple capabilities can then be composed through arithmetic operations over task vectors. Nevertheless, directly adding task vectors may accumulate redundant updates and cause interference when different tasks modify the same parameters in conflicting directions.

TIES-Merging addresses this problem by removing low-magnitude updates, resolving sign conflicts, and merging only parameters consistent with the elected direction (Yadav et al., 2023). DARE randomly removes a large fraction of delta parameters and rescales the retained updates, exploiting the redundancy of fine-tuned models (Yu et al., 2023). DARE-TIES combines these two strategies by applying DARE sparsification before sign-aware merging. Although effective, these methods primarily rely on parameter magnitude, sign consistency, or random sparsification rather than directly measuring the functional contribution of individual adapted components.

Importance- and Structure-Aware Merging. Importance-aware methods seek to preserve task-relevant updates while suppressing interfering components. SA-Merging estimates the saliency of task-vector components and assigns non-uniform contributions according to their importance (Park et al., 2026). AdaRank decomposes task vectors into singular components and adaptively prunes directions that introduce cross-task interference (Lee et al., 2026). These methods demonstrate the benefit of exploiting internal update structures rather than treating all parameters uniformly. However, their importance estimation and component selection are not explicitly conditioned on each test input.

Low-Rank and Efficient LoRA Merging. The increasing use of LoRA motivates merging methods that preserve a compact low-rank representation. CircuMerge reduces redundant LoRA updates while retaining task-relevant structures, aiming to improve both merging efficiency and scalability (Zhang et al., 2026). CtM adopts a compress-then-merge strategy: it first constructs shared low-dimensional subspaces, projects individual adapters into these subspaces, and subsequently performs merging in the compressed coordinate space (He et al., 2026). This design guarantees a fixed-rank merged adapter and avoids the information loss caused by post-hoc compression. Nevertheless, these methods primarily focus on structural compactness and do not explicitly adapt the merged parameters to individual inputs.

Learned and Alignment-Based Merging. Learned merging methods replace manually specified coefficients with data-driven optimization. AdaMerging learns task-wise or layer-wise coefficients from unlabeled data by minimizing prediction entropy. MetaMerging further learns how task-specific updates should be integrated through a meta-learning procedure (Chen et al., 2026). Expert Merging++ introduces importance-aware merging at different granularities, allowing different layers or parameter groups to receive distinct contributions (Zhang et al., 2025a). Retrofitting instead addresses representation and neuron misalignment before parameter integration, enabling models with less directly aligned parameter spaces to be merged (Luenam et al., 2025). Although these approaches improve merging flexibility, their learned coefficients or alignments are generally shared across inputs after optimization.

Input and Test-Time Adaptive Merging. Input-adaptive methods dynamically determine which task knowledge should be used for each sample. MASS identifies task-relevant low-rank subspaces and selects them according to the input (Crisostomi et al., 2025). Twin-Merging decomposes model knowledge into shared and task-specific components, compresses the task-specific components, and dynamically combines them at inference time (Lu et al., 2024). FREE-Merging performs flexible input-conditioned expert composition while reducing the dependence on explicit task identifiers (Zheng & Wang, 2025b). SyMerge further adapts task representations and merging coefficients at test time to improve performance under heterogeneous inputs (Jung et al., 2026). These approaches improve adaptability, but their routing and integration commonly operate over complete experts

or compressed expert components without first removing functionally redundant units within each expert.

Relation to MWM. Existing methods improve model merging from several complementary perspectives, including statistics-aware aggregation, task-vector conflict resolution, low-rank compression, learned coefficients, and input-dependent expert composition. However, these aspects are typically addressed separately. Static methods cannot respond to input-specific requirements; parameter-level sparsification does not directly measure functional importance; and router-based methods commonly allocate complete expert updates before removing redundant or conflicting components within them.

MWM integrates these directions through a progressive framework. It first evaluates the functional contribution of LoRA-adapted units and separates task-critical updates from less informative auxiliary components. It then learns an input-aware router over the refined experts, enabling expert relevance to be estimated from more task-discriminative representations. Finally, MWM performs asymmetric layer-wise integration, preserving the core updates of the primary expert while conservatively incorporating complementary information from other selected experts. Consequently, MWM jointly addresses intra-expert redundancy, input-dependent expert allocation, and cross-expert interference within a unified framework.

C Experiment Details

C.1 Employed Datasets and Associated Licences

Discriminative Tasks.

- **MRPC**. A binary paraphrase detection task from the Microsoft Research Paraphrase Corpus. Each example consists of a pair of sentences, and the model must determine if they are semantically equivalent. It has 3,668 training examples, 408 validation examples, and 1,725 test examples.
- **QQP**. A paraphrase detection task on Quora Question Pairs. The model must decide whether two questions are semantically identical. The training set contains 363,846 examples, with 40,430 for validation, and 390,965 for testing (test labels are not publicly available).
- **MNLI**. A natural language inference (NLI) task with three labels: entailment, neutral, and contradiction. The dataset includes multiple genres of text. It contains 392,702 training examples, 9,815 matched validation, 9,832 mismatched validation, and 20,000 test examples.
- **QNLI**. A binary classification task converted from the Stanford Question Answering Dataset. The model determines whether a given context sentence contains the answer to a question. It consists of 104,743 training examples, 5,463 validation examples, and 5,463 test examples.
- **RTE**. A binary entailment task combining data from multiple RTE challenges. The task is to determine if a hypothesis sentence can be inferred from a given premise. The dataset contains 2,490 training examples, 277 validation examples, and 3,000 test examples.
- **CoLA**. A linguistic acceptability task based on the Corpus of Linguistic Acceptability. Given an English sentence, the model must determine whether it is grammatically acceptable. The dataset contains 8,551 training examples, 1,043 validation examples, and 1,063 test examples.
- **SST-2**. A binary sentiment classification task derived from the Stanford Sentiment Treebank. The model classifies each sentence as either positive or negative. It contains 67,349 training examples, 872 validation examples, and 1,821 test examples.
- **STS-B**. A semantic textual similarity task based on the Semantic Textual Similarity Benchmark. We convert the original similarity scores, which range from 0 to 5, into three categorical labels using predefined thresholds, representing low, moderate, and high semantic similarity. The dataset contains 5,749 training examples, 1,500 validation examples, and 1,379 test examples.

The GLUE datasets are distributed under heterogeneous licenses and source-specific terms. QNLI is derived from SQuAD and follows its CC BY-SA license, whereas QQP and MRPC are subject to the release terms of Quora and Microsoft, respectively. MNLI and RTE similarly retain the terms associated with their original sources. For the additional datasets, CoLA retains the copyrights of its original authors and publishers, while SST-2 does not specify a uniform license in its public distribution. STS-B combines data from multiple SemEval sources and is therefore subject to their

respective terms. We use all datasets solely for academic research and comply with the applicable licenses and source-specific conditions.

Generation and Math Tasks. We also incorporate a dataset designed for generative tasks, specifically targeting mathematical reasoning. The **MAWPS** dataset consists of 1,772 examples of math word problems, requiring models to generate the correct mathematical expressions or answers based on natural language descriptions.

Vision Tasks.

- **MNIST**. A benchmark dataset for image classification, containing grayscale images of handwritten digits across 10 classes. The training set has 60,000 images, and the test set has 10,000 images, with a balanced distribution among classes.
- **EuroSAT**. A satellite image classification dataset consisting of 27,000 labeled and geo-referenced images across 10 classes.
- **CIFAR-10**. A benchmark for object recognition tasks in computer vision. It consists of 60,000 32x32 color images in 10 different classes, with 6,000 images per class. The dataset is divided into 50,000 training images and 10,000 test images.
- **CarBrands50**. A car classification dataset comprising 50 classes. The dataset contains a total of 4,500 labeled images, which are partitioned into 4,400 images for training, and 100 for validation.
- **FRUITS100**. A fruit classification dataset comprising 100 classes. The dataset contains a total of 50,000 labeled images, which are partitioned into 40,000 images for training, 5,000 for validation, and 5,000 for test.
- **GTSRB**. A traffic sign classification dataset containing over 50,000 images across 43 classes of traffic signs.
- **DTD**. A texture classification dataset with 47 classes and a total of 5,640 images, with approximately 120 images per class.
- **RESISC45**. A remote sensing image scene classification dataset with 45 classes and 31,500 images, approximately 700 per class.
- **GRABAGE**. A grabage classification dataset. The dataset contains a total of 147,674 labeled images, which are partitioned into 133,038 images for training, and 14,642 for test.
- **PLANTS**. A plant classification dataset comprising 30 classes. The dataset contains a total of 30,000 labeled images, which are partitioned into 24,000 images for training, 3,000 for validation, and 3,000 for testing.

C.2 Comparative Evaluation Details

Implementation details For **NLP** tasks, we adopt the standard LoRA (Hu et al., 2022) training configuration with a rank of 8 and a scaling factor of 32. The ViT (Dosovitskiy et al., 2020) model is initialized with pretrained weights and fine-tuned separately on 8 datasets—RTE, MNLI, QNLI, QQP, COLA, SST-2, SST-B and MRPC resulting in 8 task-specific models. We use the AdamW optimizer with a learning rate of 0.0001. For **CV** tasks, we follow the standard LoRA training protocol with a rank of 16 and a scaling factor of 16. For CV fine-tuning, we employ the AdamW optimizer with a learning rate of 0.005. Across **both** domains, LoRA adapters are injected into the MLP layers with a dropout rate of 0.1, and all bias terms remain unchanged. We use cross-entropy loss and apply weight decay of 0.01 to the optimizer. A warm-up learning rate schedule is used to stabilize training. To ensure reproducibility, we fix the random seed for NumPy, PyTorch, and Python's random module. For model merging baselines, we empirically find that setting the task arithmetic coefficient to 0.3 yields the best performance. Inference time is measured by averaging over 100 independent runs on the full dataset to ensure robust evaluation. All experiments are conducted on a single NVIDIA A100 GPU. The software environment includes CUDA 12.4, cuDNN 9.1.0, and PyTorch 2.1.2.

The Router layer in the **Qwen2.5-0.5B-Instruct** model consists of a single attention layer followed by a multilayer perceptron (MLP) (Kruse et al., 2022). It is trained for 3 epochs on 8 NLP datasets using the cross-entropy loss function (Mao et al., 2023). Optimization is performed using the AdamW optimizer (Loshchilov & Hutter, 2019; Kingma & Ba, 2015) with a learning rate of 0.0002. In contrast, the Router layer in the **ViT-Base-Patch16-224** model comprises convolutional layers and is trained

for 15 epochs on 10 computer vision (CV) datasets. The training also employs the cross-entropy loss function, with the Adam optimizer and a learning rate of 0.001.

Individual Model. Each task is evaluated using its corresponding fine-tuned model. This provides a task-specific reference without cross-task interference.

Multi-task Model. involving mixing datasets from multiple tasks and training the model jointly, representing one of the earliest solutions for multitask learning.

Merging Model. This term denotes algorithms aimed at combining multiple models into a unified, consolidated model, including approaches exemplified by methods Weight Averaging, Task Arithmetic, Twin-Merging, MWM and more.

C.3 Large-Model Scalability and Out-of-Domain Generalizability Details

As shown in Tables 4, we apply our model merging approach to larger models(Qwen2.5-7B-Instruct), demonstrating that our method scales effectively to models of increased size. Furthermore, as shown in Tables 7, we evaluate the merged models on out-of-domain MMLU benchmark tasks, providing evidence that our approach exhibits strong generalization capabilities beyond the training domains.

C.4 Details of LIER

As shown in Table 8 9, we present the analysis of core and auxiliary layers in the Qwen2.5-0.5B-Instruct and ViT-Base-Patch16-224 models, along with the corresponding performance before and after expert enhancement.

C.5 Extension to Full-Parameter Fine-Tuning

We further evaluate whether MWM can be extended beyond parameter-efficient fine-tuning to the full-parameter fine-tuning setting. In this setting, all model parameters are updated during task-specific fine-tuning, and the update of expert i is defined as the complete parameter difference $\boldsymbol{\Delta}_i = \boldsymbol{\theta}_i - \boldsymbol{\theta}_{\text{pre}}$. We treat the parameter update associated with each Transformer layer or block as the basic refinement unit and apply the same importance estimation, expert routing, and dynamic merging procedures used in the main experiments. Results on the NLP and CV tasks are reported in Tables 10 and 10, respectively.

NLP Setup. For the NLP experiments, the Qwen model is fully fine-tuned using AdamW with a learning rate of 2×10^{-5}, an effective batch size of 32, and a weight decay of 0.01. We use a warmup ratio of 0.1 followed by a linear learning-rate schedule. The maximum sequence length is set to either 256 or 512 according to the input length distribution of each dataset. Training is performed using BF16 precision with gradient clipping set to 1.0. The random seed is fixed to 42.

CV Setup. For the CV experiments, the ViT model is fully fine-tuned using AdamW with a learning rate of 5×10^{-5}, an effective batch size of 32, and a weight decay of 0.05. We use a warmup ratio of 0.1 followed by a cosine learning-rate schedule. All input images are resized to 224×224, and mixed-precision training is employed. The random seed is fixed to 42.

The results demonstrate that MWM remains effective when applied to fully fine-tuned experts, indicating that its refinement and dynamic integration mechanisms are not restricted to LoRA-based task updates.

C.6 Memory and Inference Efficiency.

We evaluate the memory footprint and inference efficiency of each method using a fixed set of 500 samples. Each inference experiment is repeated 100 times, and the average execution time is reported to reduce the influence of runtime fluctuations. As shown in Table 12, MWM has a smaller model storage footprint and achieves faster inference than both Twin-Merging and FREE-Merging on the NLP and CV benchmarks. This improvement results from retaining and dynamically integrating only compact expert components. Static merging methods remain more efficient because they produce a single fixed model and introduce no input-dependent computation. The additional overhead of MWM mainly arises from its lightweight router and dynamic expert composition. Nevertheless, MWM achieves a favorable efficiency trade-off compared with existing input-adaptive merging methods.

Table 7: Performance comparison of our method on out-of-domain MMLU benchmark tasks

MMLU Task	Qwen2.5-7B-Instruct	Weight Averaging	Task Arithmetic	MWM/ours
management	78.6%	79.6%	75.7%	**78.6%**
high_school_world_history	78.9%	78.9%	75.5%	**78.9%**
college_mathematics	45.0%	46.0%	36.0%	**45.0%**
high_school_us_history	79.4%	80.4%	77.0%	**79.4%**
sociology	82.1%	82.1%	77.1%	**80.6%**
astronomy	77.0%	74.3%	69.1%	**76.3%**
moral_disputes	67.1%	69.4%	65.9%	**67.1%**
high_school_government_and_politics	**89.1%**	88.6%	84.5%	88.1%
medical_genetics	**73.0%**	71.0%	67.0%	73.0%
high_school_macroeconomics	72.1%	72.8%	70.0%	**71.8%**
international_law	76.9%	78.5%	72.7%	**76.9%**
high_school_geography	83.3%	83.8%	80.3%	**83.8%**
electrical_engineering	**63.4%**	60.0%	57.9%	62.8%
virology	48.8%	50.0%	47.6%	**50.0%**
high_school_european_history	74.5%	76.4%	70.3%	**74.5%**
elementary_mathematics	60.6%	62.2%	56.3%	**59.8%**
moral_scenarios	**22.5%**	20.1%	22.0%	22.5%
formal_logic	50.8%	49.2%	42.1%	**49.2%**
machine_learning	40.2%	44.6%	46.4%	**41.1%**
us_foreign_policy	**86.0%**	85.0%	82.0%	86.0%
high_school_psychology	85.7%	85.1%	81.5%	**85.0%**
high_school_chemistry	**61.6%**	58.1%	55.7%	60.6%
computer_security	**78.0%**	76.0%	72.0%	76.0%
college_physics	53.9%	54.9%	50.0%	**52.9%**
professional_law	**45.8%**	43.9%	38.9%	45.8%
marketing	**89.7%**	88.5%	85.9%	89.3%
prehistory	76.5%	76.9%	74.1%	**76.2%**
college_biology	80.6%	83.3%	78.5%	**79.2%**
nutrition	70.6%	71.2%	65.4%	**69.6%**
professional_medicine	78.7%	76.8%	74.6%	**77.6%**
human_sexuality	75.6%	69.5%	64.9%	**72.5%**
philosophy	67.2%	69.1%	63.3%	**65.6%**
high_school_statistics	71.8%	71.8%	66.7%	**69.4%**
business_ethics	68.0%	72.0%	68.0%	**68.0%**
professional_accounting	54.3%	52.8%	52.8%	**55.0%**
high_school_mathematics	45.6%	43.0%	43.0%	**43.7%**
global_facts	40.0%	32.0%	36.0%	**38.0%**
miscellaneous	81.4%	81.7%	78.5%	**81.1%**
anatomy	71.1%	70.4%	70.4%	**69.6%**
security_studies	67.8%	69.0%	64.5%	**68.6%**
public_relations	67.3%	65.5%	60.9%	**66.4%**
clinical_knowledge	76.6%	73.6%	72.8%	**77.4%**
high_school_physics	57.0%	52.3%	48.3%	**57.0%**
econometrics	56.1%	54.4%	51.8%	**58.8%**
conceptual_physics	69.4%	70.6%	66.4%	**67.7%**
high_school_computer_science	78.0%	76.0%	70.0%	**77.0%**
college_chemistry	47.0%	46.0%	49.0%	**46.0%**
high_school_biology	81.6%	81.9%	78.4%	**81.9%**
world_religions	83.0%	79.5%	75.4%	**83.0%**
human_aging	69.1%	69.1%	65.9%	**68.6%**
college_medicine	68.2%	70.5%	64.7%	**67.6%**
college_computer_science	60.0%	54.0%	47.0%	**59.0%**
jurisprudence	75.9%	75.0%	71.3%	**75.0%**
high_school_microeconomics	81.1%	84.5%	74.8%	**81.9%**
abstract_algebra	49.0%	48.0%	38.0%	**46.0%**
professional_psychology	70.4%	70.9%	64.4%	**69.3%**
Avg	67.9%	67.3%	63.6%	**67.4%**

Table 8: Core and Auxiliary Layers in Qwen2.5-0.5B-Instruct Models and Performance Impact of Expert Enhancement

Expert Model	Core Layers	Auxiliary Layers	Finetune Performance	Performance after Enhancement
COLA	0-7, 11-23	8-10	76.2%	76.9%
MNLI	0-10, 12-23	11	78.0%	78.9%
MRPC	0-12, 22-23	13-21	84.1%	85.1%
QNLI	0-13, 17-23	14-16	84.2%	85.3%
QQP	0-5, 9-23	6-8	84.0%	85.2%
RTE	0-5, 7-23	6	77.3%	78.9%
SST-2	0-13, 16-23	14-15	92.6%	94.5%
SST-B	0-10, 16-23	11-15	82.8%	83.7%

Table 9: Core and Auxiliary Layers in ViT-Base-Patch16-224 Models and Performance Impact of Expert Enhancement

Expert Model	Core Layers	Auxiliary Layers	Finetune Performance	Performance after Enhancement
MNIST	0-1, 3-11	2	98.9%	98.9%
EuroSAT	0-1, 3-11	2	95.1%	95.9%
CIFAR-10	0-5, 7-11	6	98.2%	98.4%
Car	0-2, 4-11	3	19.0%	21.0%
FRUITS100	0-7	8-11	83.1%	83.3%
GTSRB	1-11	0	91.0%	94.0%
DTD	1-11	0	68.8%	70.3%
RESISC45	0-2, 4-11	3	83.8%	85.1%
GRABAGE	0-5	6-11	76.4%	76.6%
PLANTS	0-5	6-11	90.4%	90.8%

D A Theory of Layer-Importance Refinement and Layer-Wise Merging

This appendix provides a formal justification for the two observations that motivate MWM:

1. *LoRA layers contribute non-uniformly to expert specialization*;
2. *layer-wise merging provides a more expressive integration space than expert-wise merging with a single coefficient per expert*.

The first statement is partly empirical: without assumptions on the data, optimization trajectory, or curvature, no theorem can require two arbitrary layers to have different effects. We therefore establish (i) exact identities for layer ablation, (ii) deterministic sufficient conditions under which removing a LoRA block improves subdomain performance, and (iii) finite-sample guarantees showing when measured non-uniformity and the induced layer ranking generalize from validation data. The second statement admits a stronger, assumption-light treatment: the hypothesis class induced by layer-wise mixing contains the class induced by globally shared expert coefficients. We prove this containment, characterize when it is strict, and quantify the resulting optimization gap under locally quadratic and smooth nonconvex objectives.

The results are intentionally stated for parameter blocks. A block may be a Transformer layer, a LoRA module, an attention/MLP submodule, or a neuron group. Consequently, the same analysis applies to the layer-level implementation used in our experiments and to a finer neuron-level implementation.

Table 10: Accuracy comparison on NLP tasks under full-parameter fine-tuning (%).

MODEL	CoLA	MNLI	MRPC	QNLI	QQP	RTE	SST-2	STS-B	Avg.
Individual	78.0	78.9	84.0	84.9	83.8	78.3	91.6	82.5	82.8
MWM/Ours	78.8 ↑0.8	79.3 ↑0.4	85.0 ↑1.0	85.5 ↑0.6	84.7 ↑0.9	78.5 ↑0.2	93.3 ↑1.7	84.1 ↑1.8	83.7 ↑0.9

Table 11: Accuracy comparison on CV tasks under full-parameter fine-tuning (%).

MODEL	MNIST	EuroSAT	CIFAR-10	CAR	FRUITS100	GTSRB	DTD	RESISC45	GRABAGE	PLANTS	Avg.
Individual	98.7	95.3	98.1	20.0	83.3	92.0	69.1	84.1	77.1	90.6	80.8
MWM/Ours	98.9 ↑0.2	95.7 ↑0.4	98.3 ↑0.2	33.1 ↑13.1	83.7 ↑0.4	94.1 ↑2.1	73.5 ↑4.4	85.3 ↑1.2	77.6 ↑0.5	90.6 ↑0.0	83.1 ↑2.3

D.1 Notation and setup

Let $f(\cdot;\boldsymbol{\theta})$ be a model parameterized by $\boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^d$, and let $\boldsymbol{\theta}_0$ denote a shared pretrained model. We consider N task-specific experts. Expert i is obtained by fine-tuning the base model on task T_i and is represented as

$$\boldsymbol{\theta}_i = \boldsymbol{\theta}_{\text{pre}} + \boldsymbol{\Delta}_i, \qquad \boldsymbol{\Delta}_i = \big(\boldsymbol{\Delta}_i^{(1)}, \ldots, \boldsymbol{\Delta}_i^{(L)}\big), \tag{15}$$

where $\boldsymbol{\Delta}_i^{(l)} \in \mathbb{R}^{d_l}$ is the update associated with block l and $\sum_{l=1}^{L} d_l = d$. For LoRA fine-tuning, $\boldsymbol{\Delta}_i^{(l)}$ denotes the effective low-rank update $\boldsymbol{B}_i^{(l)}\boldsymbol{A}_i^{(l)}$ inserted into block l; the arguments below do not require explicitly materializing this matrix.

For a distribution P over examples $z = (x, y)$, define the population risk

$$\mathcal{R}_P(\boldsymbol{\theta}) = \mathbb{E}_{z\sim P}[\ell(z;\boldsymbol{\theta})]\,. \tag{16}$$

For a validation sample $S = (z_1, \ldots, z_n) \sim P^n$, define the empirical risk $\widehat{\mathcal{R}}_S(\boldsymbol{\theta}) = n^{-1}\sum_{j=1}^{n} \ell(z_j;\boldsymbol{\theta})$. Lower risk is better. When the evaluation metric is an accuracy-like score $\mathcal{M}$, all risk differences below can be converted by taking $\mathcal{R} = -\mathcal{M}$.

For expert i, we distinguish its training distribution P_i^{tr} from a target subdomain distribution P_i. This distinction is useful because an expert can be trained on a noisy, heterogeneous, or finite sample while it is evaluated on a more specific subdomain. We abbreviate $\mathcal{R}_i = \mathcal{R}_{P_i}$ and let $\boldsymbol{\theta}_i^{-l} = \boldsymbol{\theta}_i - \boldsymbol{E}_l\boldsymbol{\Delta}_i^{(l)}$ be the model obtained by ablating block l, where $\boldsymbol{E}_l$ embeds a d_l-dimensional block into the full parameter vector. The population and empirical ablation importances are

$$I_i^{(l)} = \mathcal{R}_i(\boldsymbol{\theta}_i^{-l}) - \mathcal{R}_i(\boldsymbol{\theta}_i), \qquad \widehat{I}_i^{(l)} = \widehat{\mathcal{R}}_{S_i}(\boldsymbol{\theta}_i^{-l}) - \widehat{\mathcal{R}}_{S_i}(\boldsymbol{\theta}_i). \tag{17}$$

Thus, $I_i^{(l)} > 0$ means that block l is beneficial to the target subdomain, $I_i^{(l)} < 0$ means that removing it improves target risk, and $|I_i^{(l)}|$ measures sensitivity to its removal. This signed definition is more informative than clipping negative values to zero.

For merging, let $\mathcal{S}_k(x)$ be the experts selected for input x. Global expert-wise merging uses one coefficient per selected expert,

$$\boldsymbol{\Delta}_{\text{g}}(x) = \sum_{i\in\mathcal{S}_k(x)} \alpha_i(x)\boldsymbol{\Delta}_i, \qquad \boldsymbol{\alpha}(x) \in \Delta_k, \tag{18}$$

where $\Delta_k = \{\boldsymbol{a} \in \mathbb{R}_+^k : \mathbf{1}^\top\boldsymbol{a} = 1\}$ is the probability simplex. Layer-wise merging allows a separate simplex vector for each block,

$$\boldsymbol{\Delta}_{\text{lw}}^{(l)}(x) = \sum_{i\in\mathcal{S}_k(x)} \alpha_i^{(l)}(x)\boldsymbol{\Delta}_i^{(l)}, \qquad \boldsymbol{\alpha}^{(l)}(x) \in \Delta_k. \tag{19}$$

D.2 Observation 1: LoRA layers contribute non-uniformly to expert specialization

D.2.1 Exact layer-ablation identity

We first express the contribution of a layer without a local linearity approximation.

Table 12: Comparison of memory footprint and inference time across model-merging methods on NLP and CV tasks. N denotes the number of merged experts.

Method	NLP		CV	
	Model storage	Time	Model storage	Time
Pretrained	2010M	195s	805M	158s
Individual	2010M	195s	805M	158s
Multi-task	2010M	195s	805M	158s
Weight Averaging	2010M	195s	805M	158s
Task Arithmetic	2010M	195s	805M	158s
TIES-Merging	2010M	195s	805M	158s
Twin-Merging	2010M+(N-1)*34.2M	275s	805M+(N-1)*7M	223s
FREE-Merging	2010M+(N-1)*30M	275s	805M+(N-1)*6M	223s
MWM/Ours	2010M+(N-1)*20M	235s	805M+(N-1)*5M	179s

[Exact path-integral identity] Assume that $\mathcal{R}_i$ is continuously differentiable on the line segment between $\boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_i^{-l}$. Then

$$I_i^{(l)} = -\int_0^1 \left\langle \nabla_l \mathcal{R}_i\big(\boldsymbol{\theta}_i - t\boldsymbol{E}_l\boldsymbol{\Delta}_i^{(l)}\big), \boldsymbol{\Delta}_i^{(l)} \right\rangle dt. \tag{20}$$

Define $h(t) = \mathcal{R}_i(\boldsymbol{\theta}_i - t\boldsymbol{E}_l\boldsymbol{\Delta}_i^{(l)})$. By the chain rule, $h'(t) = -\langle \nabla_l \mathcal{R}_i(\boldsymbol{\theta}_i - t\boldsymbol{E}_l\boldsymbol{\Delta}_i^{(l)}), \boldsymbol{\Delta}_i^{(l)} \rangle$. The fundamental theorem of calculus gives $I_i^{(l)} = h(1) - h(0) = \int_0^1 h'(t)dt$, which is equation 20.

Lemma D.2.1 shows that importance depends on a layer-specific alignment between the update and the target-risk gradient along the entire ablation path. Equal parameter norms therefore do not imply equal functional contributions.

[Second-order ablation formula] Assume that $\mathcal{R}_i$ is twice continuously differentiable on the same segment. There exists $\tau_{i,l} \in (0, 1)$ such that

$$I_i^{(l)} = -\left\langle \boldsymbol{g}_i^{(l)}, \boldsymbol{\Delta}_i^{(l)} \right\rangle + \frac{1}{2}\left(\boldsymbol{\Delta}_i^{(l)}\right)^\top \boldsymbol{H}_{i,l}(\tau_{i,l})\boldsymbol{\Delta}_i^{(l)}, \tag{21}$$

where $\boldsymbol{g}_i^{(l)} = \nabla_l \mathcal{R}_i(\boldsymbol{\theta}_i)$ and $\boldsymbol{H}_{i,l}(t) = \nabla_{ll}^2 \mathcal{R}_i(\boldsymbol{\theta}_i - t\boldsymbol{E}_l\boldsymbol{\Delta}_i^{(l)})$.

Apply the one-dimensional Taylor theorem with Lagrange remainder to $h(t)$ from Lemma D.2.1 around $t = 0$ and evaluate it at $t = 1$. The first derivative is $h'(0) = -\langle \boldsymbol{g}_i^{(l)}, \boldsymbol{\Delta}_i^{(l)} \rangle$, and the second derivative is $h''(t) = (\boldsymbol{\Delta}_i^{(l)})^\top \boldsymbol{H}_{i,l}(t)\boldsymbol{\Delta}_i^{(l)}$.

Equation 21 separates two mechanisms. The gradient-alignment term measures whether retaining the learned update moves the expert toward or away from the target subdomain at the final iterate. The curvature term measures how strongly the subdomain risk responds to movement in the layer's update direction. Both quantities vary across layers.

D.2.2 Deterministic bounds and sources of non-uniformity

[Block smoothness] For each expert i and block l, the block gradient is $\beta_{i,l}$-Lipschitz on the ablation segment:

$$\|\nabla_l \mathcal{R}_i(\boldsymbol{\theta} + \boldsymbol{E}_l\boldsymbol{u}) - \nabla_l \mathcal{R}_i(\boldsymbol{\theta})\|_2 \le \beta_{i,l}\|\boldsymbol{u}\|_2. \tag{22}$$

[Two-sided layer-importance bound] Under Assumption D.2.2,

$$-\langle \boldsymbol{g}_i^{(l)}, \boldsymbol{\Delta}_i^{(l)} \rangle - \frac{\beta_{i,l}}{2}\|\boldsymbol{\Delta}_i^{(l)}\|_2^2 \le I_i^{(l)} \le -\langle \boldsymbol{g}_i^{(l)}, \boldsymbol{\Delta}_i^{(l)} \rangle + \frac{\beta_{i,l}}{2}\|\boldsymbol{\Delta}_i^{(l)}\|_2^2. \tag{23}$$

The descent lemma applied to the displacement $-\boldsymbol{E}_l\boldsymbol{\Delta}_i^{(l)}$ yields the upper bound. Applying the same lemma after exchanging the two endpoints yields the lower bound. Equivalently, bound the operator norm of the Hessian term in Lemma D.2.1 by $\beta_{i,l}$.

[Approximate stationarity] If $\|\boldsymbol{g}_i^{(l)}\|_2 \le \varepsilon_{i,l}$, then

$$\left| I_i^{(l)} - \frac{1}{2}(\boldsymbol{\Delta}_i^{(l)})^\top \boldsymbol{H}_{i,l}(\tau_{i,l})\boldsymbol{\Delta}_i^{(l)} \right| \le \varepsilon_{i,l}\|\boldsymbol{\Delta}_i^{(l)}\|_2, \quad |I_i^{(l)}| \le \varepsilon_{i,l}\|\boldsymbol{\Delta}_i^{(l)}\|_2 + \frac{\beta_{i,l}}{2}\|\boldsymbol{\Delta}_i^{(l)}\|_2^2. \tag{24}$$

If, additionally, the curvature along the ablation path obeys $\mu_{i,l}\boldsymbol{I} \preceq \boldsymbol{H}_{i,l}(t) \preceq \beta_{i,l}\boldsymbol{I}$, then

$$-\varepsilon_{i,l}\|\boldsymbol{\Delta}_i^{(l)}\|_2 + \frac{\mu_{i,l}}{2}\|\boldsymbol{\Delta}_i^{(l)}\|_2^2 \le I_i^{(l)} \le \varepsilon_{i,l}\|\boldsymbol{\Delta}_i^{(l)}\|_2 + \frac{\beta_{i,l}}{2}\|\boldsymbol{\Delta}_i^{(l)}\|_2^2. \tag{25}$$

The identity follows directly from Lemma D.2.1. Apply Cauchy–Schwarz to its first term and the Rayleigh-quotient bounds to its second term.

Corollary D.2.2 explains why the Frobenius norm of a LoRA update is at best an incomplete proxy for importance: even near a stationary point, importance is a curvature-weighted norm, not an ordinary Euclidean norm. Different layers generally have different curvatures, update directions, and update magnitudes.

[A sufficient condition for unequal contributions] For two blocks $l \ne r$, define

$$q_{i,l} = -\langle \boldsymbol{g}_i^{(l)}, \boldsymbol{\Delta}_i^{(l)} \rangle, \qquad u_{i,l} = \frac{\beta_{i,l}}{2}\|\boldsymbol{\Delta}_i^{(l)}\|_2^2. \tag{26}$$

If $|q_{i,l} - q_{i,r}| > u_{i,l} + u_{i,r}$, then $I_i^{(l)} \ne I_i^{(r)}$. More precisely, $q_{i,l} > q_{i,r} + u_{i,l} + u_{i,r}$ implies $I_i^{(l)} > I_i^{(r)}$.

By Theorem D.2.2, $I_i^{(l)} \ge q_{i,l} - u_{i,l}$ and $I_i^{(r)} \le q_{i,r} + u_{i,r}$. The stated condition makes the former lower bound larger than the latter upper bound. The opposite case is symmetric.

[Generic non-uniformity] Suppose the vector of layer statistics $\boldsymbol{V} = (I_i^{(1)}, \dots, I_i^{(L)})$ has a joint distribution absolutely continuous with respect to Lebesgue measure on $\mathbb{R}^L$. Then

$$\Pr\left(I_i^{(1)} = \cdots = I_i^{(L)}\right) = 0, \qquad L \ge 2. \tag{27}$$

More generally, the probability of any exact tie $I_i^{(l)} = I_i^{(r)}$ is zero whenever $I_i^{(l)} - I_i^{(r)}$ has an absolutely continuous marginal.

The set $\{\boldsymbol{v} \in \mathbb{R}^L : v_1 = \cdots = v_L\}$ is a one-dimensional linear subspace and has L-dimensional Lebesgue measure zero for $L \ge 2$. Absolute continuity assigns probability zero to measure-zero sets. The pairwise claim follows by the same argument for a codimension-one hyperplane.

Proposition D.2.2 is not a statement that every trained network must display a large importance gap; it states that exact uniformity is nongeneric once the layer statistics vary continuously. The magnitude and practical relevance of the gaps remain empirical questions and are measured by the layer-ablation experiment in Figure 2.

D.2.3 When removing a LoRA block improves subdomain performance

An ablation with $I_i^{(l)} < 0$ improves target performance. The next result gives a verifiable sufficient condition.

[Safe improvement by removing a detrimental block] Under Assumption D.2.2, removal of block l strictly decreases the target-subdomain risk whenever

$$\left\langle \boldsymbol{g}_i^{(l)}, \boldsymbol{\Delta}_i^{(l)} \right\rangle > \frac{\beta_{i,l}}{2}\|\boldsymbol{\Delta}_i^{(l)}\|_2^2. \tag{28}$$

In particular,

$$\mathcal{R}_i(\boldsymbol{\theta}_i^{-l}) \le \mathcal{R}_i(\boldsymbol{\theta}_i) - \left\langle \boldsymbol{g}_i^{(l)}, \boldsymbol{\Delta}_i^{(l)} \right\rangle + \frac{\beta_{i,l}}{2}\|\boldsymbol{\Delta}_i^{(l)}\|_2^2. \tag{29}$$

The upper bound in Theorem D.2.2 equals the last two terms of equation 29. Condition equation 28 makes this upper bound strictly negative, hence $I_i^{(l)} < 0$.

The condition has a simple interpretation. At the final expert parameters, the update $\mathbf{\Delta}_i^{(l)}$ points in an ascent direction for the target subdomain, and the adverse first-order alignment dominates the worst-case second-order remainder. Such a situation can arise from finite-sample noise, overfitting, mismatch between P_i^{tr} and P_i, or negative transfer within a heterogeneous training set.

The framework in the main paper attenuates rather than necessarily deletes an auxiliary block. Let $\boldsymbol{\theta}_i(\lambda) = \boldsymbol{\theta}_i^{-l} + \lambda \boldsymbol{E}_l \mathbf{\Delta}_i^{(l)}$, where $\lambda = 1$ retains the original block and $\lambda = 0$ removes it.

[Optimal attenuation on a smooth upper bound] Assume $\mathcal{R}_i$ is $\beta_{i,l}$-smooth in block l around $\boldsymbol{\theta}_i^{-l}$. Let $a_{i,l} = \langle \nabla_l \mathcal{R}_i(\boldsymbol{\theta}_i^{-l}), \mathbf{\Delta}_i^{(l)} \rangle$ and $b_{i,l} = \beta_{i,l} \|\mathbf{\Delta}_i^{(l)}\|_2^2$. Then

$$\mathcal{R}_i(\boldsymbol{\theta}_i(\lambda)) \leq \mathcal{R}_i(\boldsymbol{\theta}_i^{-l}) + \lambda a_{i,l} + \frac{\lambda^2}{2} b_{i,l}. \tag{30}$$

The minimizer of the right-hand side over $[0, 1]$ is

$$\lambda_{i,l}^{\mathrm{ub}} = \Pi_{[0,1]}\left(-\frac{a_{i,l}}{b_{i,l}} \right), \tag{31}$$

with the convention $\lambda_{i,l}^{\mathrm{ub}} = 0$ if $b_{i,l} = 0$ and $a_{i,l} \geq 0$.

Equation 30 is the descent lemma. Its right-hand side is a convex quadratic in λ with unconstrained minimizer $-a_{i,l}/b_{i,l}$; projection onto $[0, 1]$ gives equation 31.

This result formally supports soft suppression: if the evidence that a block is detrimental is uncertain, a coefficient in $(0, 1)$ can improve the smoothness-based guarantee while retaining complementary information.

D.2.4 Specialization as a contrastive property

Good average task risk is not identical to specialization. To formalize the latter, let P_i be expert i's target distribution and let Q_i be a reference mixture of other tasks. Define the specialization margin

$$\Gamma_i(\boldsymbol{\theta}) = \mathcal{R}_{Q_i}(\boldsymbol{\theta}) - \mathcal{R}_{P_i}(\boldsymbol{\theta}). \tag{32}$$

A larger Γ_i means that the model is relatively more suited to P_i than to the reference tasks.

[Exact effect of ablation on specialization] For any block l,

$$\Gamma_i(\boldsymbol{\theta}_i^{-l}) - \Gamma_i(\boldsymbol{\theta}_i) = I_{Q_i}^{(l)} - I_{P_i}^{(l)}, \tag{33}$$

where $I_P^{(l)} = \mathcal{R}_P(\boldsymbol{\theta}_i^{-l}) - \mathcal{R}_P(\boldsymbol{\theta}_i)$. Thus, ablation increases specialization if and only if $I_{Q_i}^{(l)} > I_{P_i}^{(l)}$.

Expand both margins using Definition equation 32 and rearrange the four risk terms.

The proposition covers two relevant cases. First, a block can be actively harmful to P_i ($I_{P_i}^{(l)} < 0$), so its removal directly improves the target expert. Second, a block can help both P_i and Q_i, but help the other tasks more ($I_{Q_i}^{(l)} > I_{P_i}^{(l)}$); its removal then sharpens specialization even if target risk slightly increases. In practice, MWM uses target validation degradation to preserve capability and suppresses only the lower-ranked blocks, favoring the first, conservative notion.

[A sufficient condition for specialization enhancement] Suppose $\mathcal{R}_{P_i}$ and $\mathcal{R}_{Q_i}$ are block-smooth with constants $\beta_{P,l}$ and $\beta_{Q,l}$. Define $\boldsymbol{g}_{P,l} = \nabla_l \mathcal{R}_{P_i}(\boldsymbol{\theta}_i)$ and $\boldsymbol{g}_{Q,l} = \nabla_l \mathcal{R}_{Q_i}(\boldsymbol{\theta}_i)$. If

$$\left\langle \boldsymbol{g}_{P,l} - \boldsymbol{g}_{Q,l}, \mathbf{\Delta}_i^{(l)} \right\rangle > \frac{\beta_{P,l} + \beta_{Q,l}}{2} \|\mathbf{\Delta}_i^{(l)}\|_2^2, \tag{34}$$

then removing block l strictly increases Γ_i.

Apply the upper importance bound to P_i and the lower importance bound to Q_i. Their difference satisfies

$$I_{Q_i}^{(l)} - I_{P_i}^{(l)} \geq \langle \boldsymbol{g}_{P,l} - \boldsymbol{g}_{Q,l}, \mathbf{\Delta}_i^{(l)} \rangle - \frac{\beta_{P,l} + \beta_{Q,l}}{2} \|\mathbf{\Delta}_i^{(l)}\|_2^2. \tag{35}$$

Condition equation 34 makes this lower bound positive; Proposition D.2.4 completes the proof.

D.2.5 WHY TOP-IMPORTANCE SELECTION IS PRINCIPLED

Let $\boldsymbol{m} = (m_1, \ldots, m_L) \in \{0,1\}^L$ be a mask, where $m_l = 1$ retains block l. The masked expert is

$$\boldsymbol{\theta}_i(\boldsymbol{m}) = \boldsymbol{\theta}_{\mathrm{pre}} + \sum_{l=1}^{L} m_l \boldsymbol{E}_l \boldsymbol{\Delta}_i^{(l)}. \tag{36}$$

[Additive ablation surrogate] For masks in the considered budget class, the risk increase is approximated by

$$\mathcal{R}_i(\boldsymbol{\theta}_i(\boldsymbol{m})) - \mathcal{R}_i(\boldsymbol{\theta}_i) = \sum_{l:m_l=0} I_i^{(l)}. \tag{37}$$

This equality is exact for a block-separable quadratic objective and otherwise serves as a first-order block-additive surrogate.

[Optimality of retaining the most important blocks] Under Assumption D.2.5, among all masks that retain exactly q blocks, every minimizer of target risk retains a set of the q largest values of $I_i^{(l)}$.

Minimizing the right-hand side of equation 37 is equivalent to maximizing $\sum_{l:m_l=1} I_i^{(l)}$ subject to $\sum_l m_l = q$. If a retained block r has smaller importance than a removed block l, swapping them increases the retained sum. Repeating this exchange produces a top-q set.

To quantify deviations from additivity, define the interaction residual

$$\varepsilon_i(\boldsymbol{m}) = \mathcal{R}_i(\boldsymbol{\theta}_i(\boldsymbol{m})) - \mathcal{R}_i(\boldsymbol{\theta}_i) - \sum_{l:m_l=0} I_i^{(l)}. \tag{38}$$

[Near-optimality under bounded cross-block interactions] Suppose $|\varepsilon_i(\boldsymbol{m})| \le \eta_i$ for every mask retaining q blocks. Let $\widehat{\boldsymbol{m}}_{\mathrm{top}}$ retain the q largest population importances, and let $\boldsymbol{m}^*$ minimize the true masked risk. Then

$$\mathcal{R}_i(\boldsymbol{\theta}_i(\widehat{\boldsymbol{m}}_{\mathrm{top}})) - \mathcal{R}_i(\boldsymbol{\theta}_i(\boldsymbol{m}^*)) \le 2\eta_i. \tag{39}$$

Let $A_i(\boldsymbol{m}) = \sum_{l:m_l=0} I_i^{(l)}$. By Theorem D.2.5, $A_i(\widehat{\boldsymbol{m}}_{\mathrm{top}}) \le A_i(\boldsymbol{m}^*)$. Therefore,

$$\mathcal{R}_i(\boldsymbol{\theta}_i(\widehat{\boldsymbol{m}}_{\mathrm{top}})) - \mathcal{R}_i(\boldsymbol{\theta}_i(\boldsymbol{m}^*)) \tag{40}$$

$$= A_i(\widehat{\boldsymbol{m}}_{\mathrm{top}}) - A_i(\boldsymbol{m}^*) + \varepsilon_i(\widehat{\boldsymbol{m}}_{\mathrm{top}}) - \varepsilon_i(\boldsymbol{m}^*) \le 2\eta_i. \tag{41}$$

For twice differentiable risks, η_i can be connected to off-diagonal Hessian blocks. If A denotes the set of ablated blocks and $\boldsymbol{v}_A = -\sum_{l\in A} \boldsymbol{E}_l \boldsymbol{\Delta}_i^{(l)}$, the second-order cross term contains $\sum_{l\neq r\in A} (\boldsymbol{\Delta}_i^{(l)})^\top \boldsymbol{H}_{lr} \boldsymbol{\Delta}_i^{(r)}$. Hence a small interaction residual is expected when cross-layer curvature is weak or when LoRA update directions are nearly orthogonal in the Hessian metric.

D.2.6 FINITE-SAMPLE VALIDITY OF LAYER IMPORTANCE

The scores used by LIER are computed on a finite validation set. We now show that the paired ablation estimator concentrates uniformly over layers.

[Bounded paired loss difference] For every l and $z \sim P_i$,

$$D_{i,l}(z) = \ell(z; \boldsymbol{\theta}_i^{-l}) - \ell(z; \boldsymbol{\theta}_i) \in [a_{i,l}, b_{i,l}], \qquad b_{i,l} - a_{i,l} \le B_i. \tag{42}$$

[Uniform concentration of ablation importance] Under Assumption D.2.6, for any $\delta \in (0,1)$, with probability at least $1-\delta$ over $S_i \sim P_i^n$,

$$\max_{1\le l\le L} \left| \widehat{I}_i^{(l)} - I_i^{(l)} \right| \le B_i \sqrt{\frac{\log(2L/\delta)}{2n}}. \tag{43}$$

For a fixed layer, $\widehat{I}_i^{(l)}$ is the sample mean of the paired differences $D_{i,l}(z_j)$. Hoeffding's inequality gives $\Pr(|\widehat{I}_i^{(l)} - I_i^{(l)}| > \epsilon) \le 2\exp(-2n\epsilon^2/B_i^2)$. Apply the union bound over L layers and solve for ϵ.

[Recovery of the core-layer set] Let $I_{i,(1)} \geq \cdots \geq I_{i,(L)}$ be the ordered population importances and suppose the desired core size is q. Define the boundary gap $\gamma_i = I_{i,(q)} - I_{i,(q+1)} > 0$. If

$$n > \frac{2B_i^2}{\gamma_i^2} \log \frac{2L}{\delta}, \tag{44}$$

then the top-q set induced by $\widehat{I}_i^{(l)}$ equals the population top-q set with probability at least $1 - \delta$.

Under equation 44, the uniform error in Theorem D.2.6 is strictly smaller than $\gamma_i/2$. Therefore no layer below the population boundary can overtake a layer above it.

If the loss is unbounded, the same proof can be replaced by a Bernstein bound under sub-exponential tails. Because the full and ablated models are evaluated on the same examples, the paired difference can have substantially lower variance than two independently estimated validation risks.

[False-pruning control with a margin] Let ϵ_n denote the right-hand side of equation 43. On the corresponding high-probability event, every layer satisfying $\widehat{I}_i^{(l)} > \epsilon_n$ has positive population importance, and every layer satisfying $\widehat{I}_i^{(l)} < -\epsilon_n$ is population-detrimental.

If $\widehat{I}_i^{(l)} > \epsilon_n$, then $I_i^{(l)} \geq \widehat{I}_i^{(l)} - \epsilon_n > 0$. The negative case is identical.

This proposition suggests a confidence-aware variant of LIER: preserve unambiguously positive blocks, suppress unambiguously negative blocks, and attenuate statistically ambiguous blocks.

D.2.7 Connection to the experiment in Figure 2

Figure 2 reports the change in validation performance after ablating individual LoRA blocks. Let the plotted metric be $\mathcal{M}$ and define $s_i^{(l)} = \mathcal{M}(\boldsymbol{\theta}_i) - \mathcal{M}(\boldsymbol{\theta}_i^{-l})$. For the risk convention $\mathcal{R} = -\mathcal{M}$, $s_i^{(l)} = I_i^{(l)}$. Therefore:

- variation across bars is a direct estimate of non-uniform $I_i^{(l)}$;
- positive bars identify blocks whose removal damages the expert and which should preferentially be retained;
- negative bars identify blocks whose removal improves the validation metric, consistent with Theorem D.2.3;
- the separation between the retained and suppressed groups determines ranking stability through Corollary D.2.6.

The figure is thus empirical evidence for the premise of LIER, whereas the results above explain when the measured effect is reliable and why selective suppression can improve specialization. The theory does not infer numerical effect sizes from smoothness alone; those quantities must be reported from the actual experiment.

D.2.8 A function-space interpretation of LoRA importance

Parameter distance alone is not a reliable measure of functional change. We make this distinction explicit by linearizing the model outputs. Let $\boldsymbol{f}_{\boldsymbol{\theta}}(x) \in \mathbb{R}^m$ denote the vector of logits and let $\boldsymbol{J}_l(x;\boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}^{(l)}} \boldsymbol{f}_{\boldsymbol{\theta}}(x)$ be the block Jacobian. The first-order output change caused by ablating block l is

$$\boldsymbol{f}_{\boldsymbol{\theta}_i^{-l}}(x) - \boldsymbol{f}_{\boldsymbol{\theta}_i}(x) \approx -\boldsymbol{J}_l(x;\boldsymbol{\theta}_i)\boldsymbol{\Delta}_i^{(l)}. \tag{45}$$

Define the functional energy of the block on subdomain P_i as

$$\mathcal{E}_i^{(l)} = \mathbb{E}_{x\sim P_i}\left[\|\boldsymbol{J}_l(x;\boldsymbol{\theta}_i)\boldsymbol{\Delta}_i^{(l)}\|_2^2\right]. \tag{46}$$

[Jacobian-metric form of functional energy] Let $\boldsymbol{K}_{i,l} = \mathbb{E}_{x\sim P_i}[\boldsymbol{J}_l(x;\boldsymbol{\theta}_i)^\top \boldsymbol{J}_l(x;\boldsymbol{\theta}_i)]$. Then

$$\mathcal{E}_i^{(l)} = (\boldsymbol{\Delta}_i^{(l)})^\top \boldsymbol{K}_{i,l} \boldsymbol{\Delta}_i^{(l)}. \tag{47}$$

Moreover,

$$\lambda_{\min}(\boldsymbol{K}_{i,l})\|\boldsymbol{\Delta}_i^{(l)}\|_2^2 \le \mathcal{E}_i^{(l)} \le \lambda_{\max}(\boldsymbol{K}_{i,l})\|\boldsymbol{\Delta}_i^{(l)}\|_2^2. \tag{48}$$

Expand the squared norm in equation 46, interchange the expectation and the quadratic form, and apply the Rayleigh-quotient bounds.

Two equally sized LoRA blocks can therefore have very different functional effects because they are filtered through different subdomain Jacobian metrics. A large update in a near-null direction of $\boldsymbol{K}_{i,l}$ can be functionally negligible, whereas a small update aligned with a high-eigenvalue direction can substantially change the predictions.

[Exact ablation importance in a linearized squared-loss model] Suppose the logits are linear in block l on the ablation path and the target risk is

$$\mathcal{R}_i(\boldsymbol{\theta}) = \frac{1}{2}\mathbb{E}_{(x,y)\sim P_i}\|\boldsymbol{f}_{\boldsymbol{\theta}}(x) - \boldsymbol{y}\|_2^2. \tag{49}$$

Let $\boldsymbol{r}_i(x) = \boldsymbol{f}_{\boldsymbol{\theta}_i}(x) - \boldsymbol{y}$. Then

$$I_i^{(l)} = -\mathbb{E}\left[\boldsymbol{r}_i(x)^\top \boldsymbol{J}_l(x)\boldsymbol{\Delta}_i^{(l)}\right] + \frac{1}{2}\mathcal{E}_i^{(l)}. \tag{50}$$

If the residual is orthogonal in expectation to the block-induced output change, then $I_i^{(l)} = \mathcal{E}_i^{(l)}/2$.

Under block linearity, $\boldsymbol{f}_{\boldsymbol{\theta}_i^{-l}}(x) = \boldsymbol{f}_{\boldsymbol{\theta}_i}(x) - \boldsymbol{J}_l(x)\boldsymbol{\Delta}_i^{(l)}$. Substitute this expression into the squared loss, expand the square, subtract the original loss, and take expectations.

Theorem D.2.8 is the function-space analogue of Lemma D.2.1. It makes clear that specialization is not encoded by update magnitude alone; it is encoded by the interaction between the learned direction, the expert residual, and the subdomain's Jacobian geometry.

D.2.9 Joint ablation and cross-layer interaction

Single-layer scores are attractive because they are simple and interpretable, but jointly suppressing several blocks can produce interactions. For a set $A \subseteq \{1, \dots, L\}$, define

$$\boldsymbol{v}_A = \sum_{l\in A} \boldsymbol{E}_l \boldsymbol{\Delta}_i^{(l)}, \qquad I_i(A) = \mathcal{R}_i(\boldsymbol{\theta}_i - \boldsymbol{v}_A) - \mathcal{R}_i(\boldsymbol{\theta}_i). \tag{51}$$

[Second-order decomposition of set importance] If $\mathcal{R}_i$ is twice continuously differentiable on the joint ablation segment, then for some $\tau_A \in (0,1)$,

$$\begin{aligned} I_i(A) = &-\sum_{l\in A}\langle \boldsymbol{g}_i^{(l)}, \boldsymbol{\Delta}_i^{(l)}\rangle \\ &+ \frac{1}{2}\sum_{l\in A}(\boldsymbol{\Delta}_i^{(l)})^\top \boldsymbol{H}_{ll}(\tau_A)\boldsymbol{\Delta}_i^{(l)} \\ &+ \frac{1}{2}\sum_{\substack{l,r\in A\\ l\neq r}}(\boldsymbol{\Delta}_i^{(l)})^\top \boldsymbol{H}_{lr}(\tau_A)\boldsymbol{\Delta}_i^{(r)}. \end{aligned} \tag{52}$$

Apply the multivariate Taylor theorem to displacement $-\boldsymbol{v}_A$ and expand the resulting quadratic form by parameter blocks.

The last line is precisely the cross-layer interaction ignored by an additive ranking. Its sign can be positive or negative, so two blocks that appear safe individually may be harmful or beneficial when removed together.

[Bounded interaction error from off-diagonal curvature] Suppose that on all relevant joint ablation paths, $\|\boldsymbol{H}_{lr}(\boldsymbol{\theta})\|_{2\to 2} \le \kappa_{lr}$ for $l \neq r$. Let $\widetilde{I}_i(A)$ be the second-order surrogate that drops all off-diagonal Hessian blocks. Then

$$|I_i(A) - \widetilde{I}_i(A)| \le \sum_{\substack{l,r\in A\\ l<r}} \kappa_{lr}\|\boldsymbol{\Delta}_i^{(l)}\|_2\|\boldsymbol{\Delta}_i^{(r)}\|_2. \tag{53}$$

By Lemma D.2.9, the difference is the sum of cross-block quadratic terms. Pair the (l, r) and (r, l) terms, use Hessian symmetry, and apply $|\boldsymbol{a}^\top \boldsymbol{H}_{lr}\boldsymbol{b}| \le \|\boldsymbol{H}_{lr}\|_{2\to 2}\|\boldsymbol{a}\|_2\|\boldsymbol{b}\|_2$.

[Exact additivity under block-diagonal curvature] If $\boldsymbol{H}_{lr} = \boldsymbol{0}$ for all $l \neq r$ on the ablation region and the risk is quadratic there, then set importance is additive and $I_i(A) = \sum_{l\in A} I_i^{(l)}$.

All off-diagonal terms in equation 52 vanish, while quadraticity makes the second-order expansion exact with a constant Hessian.

This analysis identifies when individual ablations are sufficient and when a more expensive group-aware score may be useful. It also supplies an empirical diagnostic: pairwise ablations can estimate the largest interaction terms, and the additive LIER rule is justified if those interactions are small relative to the margin between retained and suppressed layers.

D.2.10 What the low-rank LoRA structure does and does not imply

For a linear map $\boldsymbol{W}^{(l)}$, LoRA writes the update as $\boldsymbol{\Delta}_i^{(l)} = \boldsymbol{B}_i^{(l)}\boldsymbol{A}_i^{(l)}$ with rank at most r_l. Low rank reduces the dimension of the adapted subspace, but it does not make different layers equally important.

[Norm and output bounds for a LoRA block] For any activation vector $\boldsymbol{h}$,

$$\|\boldsymbol{B}_i^{(l)}\boldsymbol{A}_i^{(l)}\|_F \le \|\boldsymbol{B}_i^{(l)}\|_2\|\boldsymbol{A}_i^{(l)}\|_F, \tag{54}$$

$$\|\boldsymbol{B}_i^{(l)}\boldsymbol{A}_i^{(l)}\boldsymbol{h}\|_2 \le \|\boldsymbol{B}_i^{(l)}\|_2\|\boldsymbol{A}_i^{(l)}\|_2\|\boldsymbol{h}\|_2. \tag{55}$$

Neither bound is determined by rank alone.

Both inequalities follow from submultiplicativity of the spectral and Frobenius norms. Matrices of the same rank can have arbitrarily different singular values, so rank alone cannot determine either right-hand side.

[Rank equality does not imply importance equality] Fix any $r \ge 1$ and two blocks with nonzero Jacobian metrics $\boldsymbol{K}_{i,l}$ and $\boldsymbol{K}_{i,s}$. There exist rank-r LoRA updates at the two blocks having the same Frobenius norm but different functional energies whenever the relevant Jacobian metrics have different attainable Rayleigh quotients. The energy ratio can be as large as the ratio of the selected eigenvalues and becomes unbounded when one update lies in a metric null space while the other has positive energy.

Choose unit update directions aligned with eigenvectors of $\boldsymbol{K}_{i,l}$ and $\boldsymbol{K}_{i,s}$ having eigenvalues λ_l and λ_s. Scale the two updates to have equal Frobenius norm c. By Proposition D.2.8, their energies are $c^2\lambda_l$ and $c^2\lambda_s$. Selecting unequal eigenvalues gives unequal energies; selecting one direction in a null space gives an unbounded positive- to-zero ratio. Each direction can be embedded in a rank-r factorization by adding $r-1$ singular directions of arbitrarily small magnitude.

Thus, the use of a common LoRA rank across layers is not evidence for uniform functional importance. This distinction is important in experimental analysis: rank and parameter count are architectural quantities, whereas the ablation score is a data- and task-conditioned functional quantity.

D.3 Observation 2: layer-wise merging provides a more expressive integration space

D.3.1 Global and layer-wise feasible sets

Fix an input x and its selected expert set $\mathcal{S}_k(x)$; suppress x from the notation. Let

$$\mathcal{C}_l = \operatorname{conv}\{\boldsymbol{\Delta}_i^{(l)} : i \in \mathcal{S}_k\} \tag{56}$$

be the convex hull of selected updates at block l. The global and layer-wise feasible update sets are

$$\mathcal{F}_{\mathrm{g}} = \left\{ \left(\sum_i \alpha_i \boldsymbol{\Delta}_i^{(1)}, \ldots, \sum_i \alpha_i \boldsymbol{\Delta}_i^{(L)} \right) : \boldsymbol{\alpha} \in \Delta_k \right\}, \tag{57}$$

$$\mathcal{F}_{\mathrm{lw}} = \mathcal{C}_1 \times \cdots \times \mathcal{C}_L. \tag{58}$$

[Expressivity containment] For any selected experts and any block partition,

$$\mathcal{F}_{\mathrm{g}} \subseteq \mathcal{F}_{\mathrm{lw}}. \tag{59}$$

Consequently, for any objective J defined on merged updates,

$$\inf_{\boldsymbol{\Delta}\in\mathcal{F}_{\mathrm{lw}}} J(\boldsymbol{\Delta}) \le \inf_{\boldsymbol{\Delta}\in\mathcal{F}_{\mathrm{g}}} J(\boldsymbol{\Delta}). \tag{60}$$

Take any $\boldsymbol{\Delta} \in \mathcal{F}_{\mathrm{g}}$ generated by $\boldsymbol{\alpha} \in \Delta_k$. In the layer-wise construction choose $\boldsymbol{\alpha}^{(l)} = \boldsymbol{\alpha}$ for every l. This reproduces every block of $\boldsymbol{\Delta}$, proving containment. The infimum of a function over a superset cannot exceed its infimum over a subset.

Theorem D.3.1 is unconditional: it does not require convexity, smoothness, or a probabilistic model. It states that layer-wise merging can always reproduce a global merge and can additionally choose different expert proportions in different layers.

[A sufficient condition for strict containment] Assume $L \ge 2$ and $k \ge 2$. Suppose there exist blocks $l \ne r$ and two distinct coefficient vectors $\boldsymbol{a}, \boldsymbol{b} \in \Delta_k$ such that the affine maps induced by the selected expert updates are injective on the simplex at both blocks—that is,

$$\boldsymbol{D}_s\boldsymbol{u} = \boldsymbol{D}_s\boldsymbol{v}, \quad \boldsymbol{u}, \boldsymbol{v} \in \Delta_k \Longrightarrow \boldsymbol{u} = \boldsymbol{v}, \qquad s \in \{l, r\}, \tag{61}$$

then $\mathcal{F}_{\mathrm{g}} \subsetneq \mathcal{F}_{\mathrm{lw}}$.

Construct a layer-wise update using $\boldsymbol{a}$ at block l and $\boldsymbol{b}$ at block r, with $\boldsymbol{a} \ne \boldsymbol{b}$. If it belonged to $\mathcal{F}_{\mathrm{g}}$, a single coefficient vector $\boldsymbol{c}$ would satisfy $\boldsymbol{D}_l\boldsymbol{c} = \boldsymbol{D}_l\boldsymbol{a}$ and $\boldsymbol{D}_r\boldsymbol{c} = \boldsymbol{D}_r\boldsymbol{b}$. Injectivity implies $\boldsymbol{c} = \boldsymbol{a}$ and $\boldsymbol{c} = \boldsymbol{b}$, a contradiction.

Equation 61 holds, for example, when the selected updates are affinely independent at the two blocks. For $k = 2$, it reduces to requiring the two experts to have different updates at each block.

Strict containment can fail in degenerate cases: all experts may have identical updates in every but one block, or different coefficient vectors may represent the same update because of affine dependence. These cases do not invalidate containment; they only remove the possibility of a strict expressivity gain for that particular expert set.

D.3.2 Exact optimization gap for a local quadratic model

We next quantify the advantage when different blocks prefer different expert mixtures. Let $\boldsymbol{D}_l = [\boldsymbol{\Delta}_1^{(l)}, \dots, \boldsymbol{\Delta}_k^{(l)}] \in \mathbb{R}^{d_l \times k}$ and suppose the desired local update at block l is $\boldsymbol{u}_l$. Consider the block-separable quadratic surrogate

$$J(\{\boldsymbol{\alpha}^{(l)}\}) = \frac{1}{2}\sum_{l=1}^{L} \|\boldsymbol{D}_l\boldsymbol{\alpha}^{(l)} - \boldsymbol{u}_l\|_{\boldsymbol{H}_l}^2, \qquad \boldsymbol{H}_l \succ 0, \tag{62}$$

where $\|\boldsymbol{v}\|_{\boldsymbol{H}}^2 = \boldsymbol{v}^\top\boldsymbol{H}\boldsymbol{v}$. For a transparent closed form, first ignore simplex boundaries and assume the relevant optima lie in the simplex interior.

[Weighted-variance characterization of the global-merging gap] Let $\boldsymbol{Q}_l = \boldsymbol{D}_l^\top\boldsymbol{H}_l\boldsymbol{D}_l \succ 0$, and let $\boldsymbol{\alpha}_l^* = \boldsymbol{Q}_l^{-1}\boldsymbol{D}_l^\top\boldsymbol{H}_l\boldsymbol{u}_l$ be the unconstrained layer-wise optimizer. The best shared global coefficient is

$$\boldsymbol{\alpha}_{\mathrm{g}}^* = \left(\sum_{l=1}^{L}\boldsymbol{Q}_l\right)^{-1}\sum_{l=1}^{L}\boldsymbol{Q}_l\boldsymbol{\alpha}_l^*. \tag{63}$$

Moreover, the exact excess objective of global merging is

$$J_{\mathrm{g}}^* - J_{\mathrm{lw}}^* = \frac{1}{2}\sum_{l=1}^{L}\|\boldsymbol{\alpha}_l^* - \boldsymbol{\alpha}_{\mathrm{g}}^*\|_{\boldsymbol{Q}_l}^2. \tag{64}$$

Hence the gap is zero if and only if all block-wise optima coincide, and it is strictly positive whenever at least two identifiable blocks prefer different coefficients.

Completing the square in each block gives

$$\frac{1}{2}\|\boldsymbol{D}_l\boldsymbol{\alpha} - \boldsymbol{u}_l\|^2_{\boldsymbol{H}_l} = c_l + \frac{1}{2}\|\boldsymbol{\alpha} - \boldsymbol{\alpha}_l^*\|^2_{\boldsymbol{Q}_l}, \tag{65}$$

where c_l is the layer-wise minimum. Summing and differentiating with respect to the shared $\boldsymbol{\alpha}$ yields equation 63. Substituting it into the completed-square form gives equation 64. Positive definiteness implies that the sum vanishes exactly when every $\boldsymbol{\alpha}_l^*$ equals $\boldsymbol{\alpha}_{\mathrm{g}}^*$.

Equation 64 gives a precise meaning to "heterogeneous layer importance." The advantage of layer-wise merging is the weighted dispersion of the coefficients preferred by individual blocks. If all layers prefer the same mixture, global merging loses nothing. If early layers prefer one expert and later layers prefer another, the compromise imposed by a single coefficient produces a positive regret.

[Lower bound from coefficient disagreement] If $\lambda_{\min}(\boldsymbol{Q}_l) \geq \mu_l > 0$, then

$$J_{\mathrm{g}}^* - J_{\mathrm{lw}}^* \geq \frac{1}{2}\sum_{l=1}^{L} \mu_l \|\boldsymbol{\alpha}_l^* - \boldsymbol{\alpha}_{\mathrm{g}}^*\|_2^2. \tag{66}$$

Apply the Rayleigh-quotient bound to each term in equation 64.

With simplex constraints, the same geometric interpretation holds after replacing unconstrained least squares with projections in the metrics $\boldsymbol{Q}_l$. Although a single closed form is generally unavailable at the boundary, Theorem D.3.1 still guarantees non-inferiority, and strict improvement follows whenever the layer-wise constrained minimizer is not representable by a shared coefficient vector.

D.3.3 Risk improvement beyond quadratic objectives

The local quadratic model is natural near a pretrained or merged parameter vector. We now bound the discrepancy for a general smooth objective. Let $\boldsymbol{\delta}_{\mathrm{g}}$ and $\boldsymbol{\delta}_{\mathrm{lw}}$ be candidate global and layer-wise merged updates, and let $\mathcal{R}_x$ denote the conditional risk for input or context x.

[Local Hessian Lipschitzness] On a ball of radius r around $\boldsymbol{\theta}_{\mathrm{pre}}$, the Hessian of $\mathcal{R}_x$ is ρ_x-Lipschitz in operator norm.

[Third-order Taylor remainder] Under Assumption D.3.3, for $\|\boldsymbol{\delta}\|_2 \leq r$,

$$\left|\mathcal{R}_x(\boldsymbol{\theta}_{\mathrm{pre}} + \boldsymbol{\delta}) - \left[\mathcal{R}_x(\boldsymbol{\theta}_{\mathrm{pre}}) + \boldsymbol{g}_x^\top \boldsymbol{\delta} + \frac{1}{2}\boldsymbol{\delta}^\top \boldsymbol{H}_x \boldsymbol{\delta}\right]\right| \leq \frac{\rho_x}{6}\|\boldsymbol{\delta}\|_2^3, \tag{67}$$

where $\boldsymbol{g}_x = \nabla\mathcal{R}_x(\boldsymbol{\theta}_{\mathrm{pre}})$ and $\boldsymbol{H}_x = \nabla^2\mathcal{R}_x(\boldsymbol{\theta}_{\mathrm{pre}})$.

Use Taylor's theorem with integral remainder and the Hessian Lipschitz bound.

[Transfer of a quadratic layer-wise advantage] Let $Q_x(\boldsymbol{\delta})$ denote the quadratic expression in brackets in equation 67. Suppose $Q_x(\boldsymbol{\delta}_{\mathrm{g}}) - Q_x(\boldsymbol{\delta}_{\mathrm{lw}}) = \zeta_x > 0$ and both updates have norm at most r. Then

$$\mathcal{R}_x(\boldsymbol{\theta}_{\mathrm{pre}} + \boldsymbol{\delta}_{\mathrm{g}}) - \mathcal{R}_x(\boldsymbol{\theta}_{\mathrm{pre}} + \boldsymbol{\delta}_{\mathrm{lw}}) \geq \zeta_x - \frac{\rho_x}{6}\left(\|\boldsymbol{\delta}_{\mathrm{g}}\|_2^3 + \|\boldsymbol{\delta}_{\mathrm{lw}}\|_2^3\right). \tag{68}$$

Thus, the true risk is lower for the layer-wise solution whenever the quadratic advantage exceeds the two remainder terms.

Apply Lemma D.3.3 with the lower remainder sign to the global candidate and the upper remainder sign to the layer-wise candidate, then subtract.

This theorem clarifies the local nature of parameter merging. Because LoRA updates are low-rank and typically small relative to the pretrained weights, the cubic remainder can be controlled in a neighborhood where the quadratic surrogate is informative. The theorem does not claim that arbitrary large parameter interpolations are well approximated by a quadratic model.

D.3.4 Cross-expert interference and block-wise control

Consider a merged update $\boldsymbol{\delta} = \sum_i \alpha_i \boldsymbol{\Delta}_i$ and a positive semidefinite curvature matrix $\boldsymbol{H}$. Its quadratic energy is

$$\|\boldsymbol{\delta}\|_{\boldsymbol{H}}^2 = \sum_i \alpha_i^2 \|\boldsymbol{\Delta}_i\|_{\boldsymbol{H}}^2 + 2\sum_{i<j} \alpha_i\alpha_j \langle \boldsymbol{\Delta}_i, \boldsymbol{\Delta}_j\rangle_{\boldsymbol{H}}. \tag{69}$$

Negative cross terms indicate destructive alignment in the local metric. Partitioning by layers gives

$$\sum_{l=1}^{L} \left\| \sum_i \alpha_i^{(l)} \boldsymbol{\Delta}_i^{(l)} \right\|_{\boldsymbol{H}_l}^2, \tag{70}$$

which exposes each block's conflict separately.

[Pairwise block conflict] For experts i, j at block l, define

$$c_{ij}^{(l)} = \left[- \left\langle \boldsymbol{\Delta}_i^{(l)}, \boldsymbol{\Delta}_j^{(l)} \right\rangle_{\boldsymbol{H}_l} \right]_+, \qquad [a]_+ = \max(a, 0). \tag{71}$$

[Conflict exposure under layer-wise coefficients] The magnitude of the negative pairwise contribution to equation 70 is bounded by

$$\mathsf{C}(\boldsymbol{\alpha}) \leq 2 \sum_{l=1}^{L} \sum_{i<j} \alpha_i^{(l)} \alpha_j^{(l)} c_{ij}^{(l)}. \tag{72}$$

Global merging is the special case $\alpha_i^{(l)} = \alpha_i$. Layer-wise merging can set $\alpha_i^{(l)} \alpha_j^{(l)} = 0$ only on blocks where the pair conflicts while retaining both experts on nonconflicting blocks; a global rule cannot generally do so.

Expand each squared norm in equation 70; retain the negative parts of pairwise cross terms and upper-bound their magnitude by the definition of $c_{ij}^{(l)}$. The final statement follows because the layer-wise simplex variables are independent across l, whereas global variables are tied.

[Strict conflict reduction in a two-layer, two-expert case] Suppose experts 1 and 2 conflict at block l but are complementary at block r, with $c_{12}^{(l)} > 0$ and $\langle \boldsymbol{\Delta}_1^{(r)}, \boldsymbol{\Delta}_2^{(r)} \rangle_{\boldsymbol{H}_r} > 0$. A layer-wise rule may select only one expert at l and mix both at r, thereby eliminating the pairwise conflict at l without discarding complementarity at r. Any global rule that assigns both experts positive mass retains a positive conflict term at l.

Choose $(\alpha_1^{(l)}, \alpha_2^{(l)})$ at a simplex vertex and choose both coefficients positive at r. The conflict multiplier is zero at l. Under a global rule the product $\alpha_1 \alpha_2$ is shared and positive whenever both experts are retained.

D.3.5 Input-adaptive routing and conditional expressivity

So far, coefficients were optimized for a fixed input. Let $\mathcal{A}_{\mathrm{g}}$ be a class of measurable routing functions $x \mapsto \boldsymbol{\alpha}(x) \in \Delta_k$, and define the corresponding global hypothesis class

$$\mathcal{H}_{\mathrm{g}} = \left\{ x \mapsto f\left(x; \boldsymbol{\theta}_{\mathrm{pre}} + \sum_i \alpha_i(x) \boldsymbol{\Delta}_i \right) : \boldsymbol{\alpha} \in \mathcal{A}_{\mathrm{g}} \right\}. \tag{73}$$

Let $\mathcal{A}_{\mathrm{lw}}$ contain tuples of routing functions $(\boldsymbol{\alpha}^{(1)}, \ldots, \boldsymbol{\alpha}^{(L)})$ and define $\mathcal{H}_{\mathrm{lw}}$ analogously.

[Conditional hypothesis-class containment] If $\mathcal{A}_{\mathrm{lw}}$ contains every tied tuple $(\boldsymbol{\alpha}, \ldots, \boldsymbol{\alpha})$ with $\boldsymbol{\alpha} \in \mathcal{A}_{\mathrm{g}}$, then $\mathcal{H}_{\mathrm{g}} \subseteq \mathcal{H}_{\mathrm{lw}}$. Therefore, for any data distribution P and loss,

$$\inf_{h \in \mathcal{H}_{\mathrm{lw}}} \mathbb{E}_P[\ell(h(x), y)] \leq \inf_{h \in \mathcal{H}_{\mathrm{g}}} \mathbb{E}_P[\ell(h(x), y)]. \tag{74}$$

For every global routing function, choose the tied layer-wise tuple. This produces exactly the same input-dependent parameters and predictions. The risk inequality follows from class containment.

The result separates two benefits that are sometimes conflated. Input-adaptive routing allows the expert mixture to change with x; layer-wise integration allows the mixture to change with l. Combining them permits coefficients $\alpha_i^{(l)}(x)$ to adapt along both axes.

D.3.6 APPROXIMATE ROUTING AND ROBUSTNESS TO COEFFICIENT ERROR

The router predicts coefficients rather than optimizing them exactly. Let $\boldsymbol{\alpha}_*^{(l)}(x)$ be oracle layer-wise coefficients and $\widehat{\boldsymbol{\alpha}}^{(l)}(x)$ be predicted coefficients. Define $\boldsymbol{e}_l(x) = \widehat{\boldsymbol{\alpha}}^{(l)}(x) - \boldsymbol{\alpha}_*^{(l)}(x)$.

[Parameter and risk sensitivity to routing error] Let $\|\boldsymbol{D}_l\|_{2\to 2} \le \sigma_l$. Then the induced parameter error satisfies

$$\|\widehat{\boldsymbol{\delta}}(x) - \boldsymbol{\delta}_*(x)\|_2 \le \left(\sum_{l=1}^{L} \sigma_l^2 \|\boldsymbol{e}_l(x)\|_2^2\right)^{1/2}. \tag{75}$$

If $\mathcal{R}_x$ is G_x-Lipschitz in parameters on the segment between the two merged models, then

$$|\mathcal{R}_x(\boldsymbol{\theta}_{\mathrm{pre}} + \widehat{\boldsymbol{\delta}}(x)) - \mathcal{R}_x(\boldsymbol{\theta}_{\mathrm{pre}} + \boldsymbol{\delta}_*(x))| \le G_x \left(\sum_{l} \sigma_l^2 \|\boldsymbol{e}_l(x)\|_2^2\right)^{1/2}. \tag{76}$$

The error in block l is $\boldsymbol{D}_l \boldsymbol{e}_l$. Orthogonality of disjoint parameter blocks yields $\|\widehat{\boldsymbol{\delta}} - \boldsymbol{\delta}_*\|_2^2 = \sum_l \|\boldsymbol{D}_l \boldsymbol{e}_l\|_2^2 \le \sum_l \sigma_l^2 \|\boldsymbol{e}_l\|_2^2$. The risk bound follows from Lipschitzness.

Theorem D.3.6 motivates strengthening experts before router training. Redundant directions can enlarge $\|\boldsymbol{D}_l\|_{2\to 2}$ and make routing errors more costly, while LIER suppresses weak or conflicting components before SER estimates expert relevance.

D.3.7 TOP-k ROUTING APPROXIMATION

Let a dense router output $\boldsymbol{p}(x) \in \Delta_N$ and define the dense update $\boldsymbol{\delta}_{\mathrm{dense}}^{(l)} = \sum_i p_i \boldsymbol{\Delta}_i^{(l)}$. Let S_k contain the k largest probabilities, let $\tau_k = \sum_{i \notin S_k} p_i$, and renormalize $\widehat{p}_i = p_i/(1-\tau_k)$ for $i \in S_k$.

[Error induced by Top-k truncation] If $\|\boldsymbol{\Delta}_i^{(l)}\|_2 \le R_l$ for all i, then

$$\|\boldsymbol{\delta}_{\mathrm{top}k}^{(l)} - \boldsymbol{\delta}_{\mathrm{dense}}^{(l)}\|_2 \le 2R_l\tau_k. \tag{77}$$

Across all blocks,

$$\|\boldsymbol{\delta}_{\mathrm{top}k} - \boldsymbol{\delta}_{\mathrm{dense}}\|_2 \le 2\tau_k \left(\sum_l R_l^2\right)^{1/2}. \tag{78}$$

At one block,

$$\boldsymbol{\delta}_{\mathrm{top}k}^{(l)} - \boldsymbol{\delta}_{\mathrm{dense}}^{(l)} = \frac{\tau_k}{1-\tau_k} \sum_{i\in S_k} p_i \boldsymbol{\Delta}_i^{(l)} - \sum_{i \notin S_k} p_i \boldsymbol{\Delta}_i^{(l)}. \tag{79}$$

The triangle inequality bounds the first term by $R_l\tau_k$ and the second by $R_l\tau_k$. Sum squared block bounds and take the square root.

Thus, Top-k selection is a controlled approximation whenever the discarded routing mass is small. Combined with a Lipschitz risk bound, the excess risk is at most $2G_x\tau_k(\sum_l R_l^2)^{1/2}$.

D.3.8 PRIMARY-CORE AND AUXILIARY-UPDATE INTEGRATION

MWM uses a structured subset of the full layer-wise coefficient space. For input x, let $i^* = \arg\max_i \widehat{p}_i(x)$ be the primary expert and write

$$\boldsymbol{\delta}_m(x) = \widehat{p}_{i^*}(x)\boldsymbol{\Delta}_{i^*}^{\mathrm{core}} + \lambda \sum_{i \in \mathcal{S}_k(x)\setminus\{i^*\}} \widehat{p}_i(x)\boldsymbol{\Delta}_i^{\mathrm{aux}}. \tag{80}$$

Let $\boldsymbol{\theta}_c(x) = \boldsymbol{\theta}_{\mathrm{pre}} + \widehat{p}_{i^*}(x)\boldsymbol{\Delta}_{i^*}^{\mathrm{core}}$ and $\boldsymbol{A}(x) = \sum_{i\neq i^*} \widehat{p}_i(x)\boldsymbol{\Delta}_i^{\mathrm{aux}}$.

[Controlled influence of auxiliary experts] Assume the conditional risk $\mathcal{R}_x$ is H_x-smooth and $\|\nabla\mathcal{R}_x(\boldsymbol{\theta}_c(x))\|_2 \le G_x$. Then, for $\lambda \in [0,1]$,

$$\mathcal{R}_x(\boldsymbol{\theta}_c + \lambda\boldsymbol{A}) \le \mathcal{R}_x(\boldsymbol{\theta}_c) + \lambda G_x\|\boldsymbol{A}\|_2 + \frac{H_x\lambda^2}{2}\|\boldsymbol{A}\|_2^2. \tag{81}$$

If the auxiliary direction is beneficial with $\langle \nabla \mathcal{R}_x(\boldsymbol{\theta}_c), \boldsymbol{A} \rangle = -b_x < 0$, then

$$\mathcal{R}_x(\boldsymbol{\theta}_c + \lambda \boldsymbol{A}) \leq \mathcal{R}_x(\boldsymbol{\theta}_c) - \lambda b_x + \frac{H_x \lambda^2}{2} \|\boldsymbol{A}\|_2^2, \tag{82}$$

and the right-hand side is minimized on $[0, 1]$ at $\lambda^* = \min\{1, b_x/(H_x \|\boldsymbol{A}\|_2^2)\}$.

The smoothness inequality gives $\mathcal{R}_x(\boldsymbol{\theta}_c + \lambda \boldsymbol{A}) \leq \mathcal{R}_x(\boldsymbol{\theta}_c) + \lambda \langle \nabla \mathcal{R}_x(\boldsymbol{\theta}_c), \boldsymbol{A} \rangle + (H_x \lambda^2/2) \|\boldsymbol{A}\|_2^2$. Apply Cauchy–Schwarz for equation 81. Substitute the beneficial-alignment assumption and minimize the resulting scalar quadratic for equation 82.

The theorem captures the purpose of asymmetric integration. The core update of the most relevant expert supplies the dominant task direction. Contributions from secondary experts are permitted when they are complementary, while λ limits their worst-case perturbation. This is a constrained, regularized use of the additional expressivity established by Theorem D.3.1; it does not require estimating an unconstrained coefficient for every layer and expert.

D.4 From the two observations to the three-stage MWM pipeline

The preceding results yield the method in three logical steps.

Step 1: Layer Importance-based Expert Refinement (LIER). Lemma D.2.1 and Theorem D.2.2 show that the functional effect of a LoRA block is governed by layer-specific gradient alignment, curvature, direction, and magnitude. Proposition D.2.2 explains why exact uniformity is nongeneric, while Figure 2 measures the effect in the trained experts. Theorem D.2.3 proves that removing a block can improve subdomain risk under a concrete alignment condition. Theorems D.2.5 and D.2.5 justify retaining high-importance blocks, and Theorem D.2.6 establishes when the validation ranking is reliable. These results motivate splitting each expert into core and auxiliary updates and attenuating the latter.

Step 2: Strengthened Expert Routing (SER). Routing complete experts forces a relevance score to summarize both task-critical and redundant directions. After LIER, the expert representations contain a larger proportion of validated task-specific signal. Theorem D.3.6 shows that the risk induced by routing error scales with the operator norms of the update dictionaries, providing a formal reason to suppress unnecessary components before training the router. Proposition D.3.7 further shows that Top-k routing is accurate when the discarded probability mass is small.

Step 3: Layer-Adaptive Dynamic Merging (LDM). Theorem D.3.1 proves that layer-wise mixing weakly dominates globally tied coefficients in representation space, and Theorem D.3.2 identifies the exact advantage as disagreement among block-wise preferred mixtures. Proposition D.3.4 shows that layer-wise coefficients can suppress conflicts only where they occur, without discarding an expert's complementary blocks elsewhere. Finally, Theorem D.3.8 justifies the primary-core/secondary-auxiliary asymmetry and quantifies how λ controls secondary-expert perturbations.

D.4.1 Statistical price of the larger layer-wise space

Theorem D.3.1 concerns oracle optimization. A larger class can overfit finite validation data, so expressivity must be paired with capacity control. A simple finite-class argument makes the tradeoff explicit. Suppose a coefficient vector is selected from a finite candidate set $\mathcal{A} \subset \Delta_k$ of size M. Global merging chooses one element of $\mathcal{A}$, while unconstrained layer-wise merging chooses one independently for each of L blocks, yielding at most M^L configurations.

[Uniform validation bound for discretized merging rules] Assume the per-example loss lies in $[0, 1]$ and a validation set of size n is independent of expert training. With probability at least $1 - \delta$, simultaneously for every global candidate,

$$|\widehat{\mathcal{R}}(h) - \mathcal{R}(h)| \leq \sqrt{\frac{\log(2M/\delta)}{2n}}, \tag{83}$$

and simultaneously for every unconstrained layer-wise candidate,

$$|\widehat{\mathcal{R}}(h) - \mathcal{R}(h)| \leq \sqrt{\frac{L \log M + \log(2/\delta)}{2n}}. \tag{84}$$

Apply Hoeffding's inequality to each fixed candidate. A union bound over M global candidates gives equation 83; a union bound over at most M^L layer-wise configurations gives equation 84.

The $L \log M$ term is a worst-case price for completely independent coefficients. MWM deliberately occupies a much smaller structured subclass: the router supplies shared input-dependent expert scores, the core/auxiliary mask restricts where those scores apply, and a single scalar λ controls auxiliary contributions. This design seeks much of the oracle benefit of layer-wise control without paying for an unconstrained $N \times L$ coefficient table.

[Excess population risk of empirical selection] Let $\widehat{h}$ minimize validation risk in a finite class $\mathcal{H}$ and let h^* minimize its population risk. On an event where $\sup_{h \in \mathcal{H}} |\widehat{\mathcal{R}}(h) - \mathcal{R}(h)| \le \epsilon$,

$$\mathcal{R}(\widehat{h}) - \mathcal{R}(h^*) \le 2\epsilon. \tag{85}$$

$\mathcal{R}(\widehat{h}) \le \widehat{\mathcal{R}}(\widehat{h}) + \epsilon \le \widehat{\mathcal{R}}(h^*) + \epsilon \le \mathcal{R}(h^*) + 2\epsilon$.

D.4.2 A TETHERED LAYER-WISE OBJECTIVE

One way to interpolate between global routing and unrestricted layer-wise mixing is to tether each layer's coefficient to the router prediction $\boldsymbol{p}(x)$. Consider

$$J_\gamma(\{\boldsymbol{\alpha}^{(l)}\}) = \frac{1}{2} \sum_{l=1}^{L} \|\boldsymbol{D}_l \boldsymbol{\alpha}^{(l)} - \boldsymbol{u}_l\|^2_{\boldsymbol{H}_l} + \frac{\gamma}{2} \sum_{l=1}^{L} \|\boldsymbol{\alpha}^{(l)} - \boldsymbol{p}\|_2^2. \tag{86}$$

[Closed-form tethered coefficients] Ignoring simplex boundaries and assuming $\boldsymbol{Q}_l = \boldsymbol{D}_l^\top \boldsymbol{H}_l \boldsymbol{D}_l \succ 0$, the unique minimizer of equation 86 is

$$\boldsymbol{\alpha}_\gamma^{(l)} = (\boldsymbol{Q}_l + \gamma \boldsymbol{I})^{-1} (\boldsymbol{Q}_l \boldsymbol{\alpha}_l^* + \gamma \boldsymbol{p}). \tag{87}$$

Furthermore, $\lim_{\gamma \to 0} \boldsymbol{\alpha}_\gamma^{(l)} = \boldsymbol{\alpha}_l^*$ and $\lim_{\gamma \to \infty} \boldsymbol{\alpha}_\gamma^{(l)} = \boldsymbol{p}$.

Differentiate equation 86 with respect to $\boldsymbol{\alpha}^{(l)}$, set the derivative to zero, and solve the resulting linear system. The limits follow from continuity and the resolvent form of $(\boldsymbol{Q}_l + \gamma \boldsymbol{I})^{-1}$.

This result gives a useful conceptual reading of LDM. The router supplies a global relevance prior $\boldsymbol{p}(x)$, whereas neuron/layer importance determines where that prior is trusted, attenuated, or overridden. The practical binary core/auxiliary construction is a computationally inexpensive structured approximation to a fully optimized tethered solution.

D.4.3 ROBUSTNESS OF TOP-LAYER SELECTION TO SCORE PERTURBATIONS

Finite-sample noise, numerical precision, and validation distribution shift perturb the importance scores. The next deterministic result complements the probabilistic ranking guarantee.

[Value robustness under uniformly perturbed scores] Suppose $|\widehat{I}_i^{(l)} - I_i^{(l)}| \le \epsilon$ for every layer. Let $\widehat{C}$ be the q layers with largest estimated scores and C^* the q layers with largest true scores. Then

$$\sum_{l \in C^*} I_i^{(l)} - \sum_{l \in \widehat{C}} I_i^{(l)} \le 2q\epsilon. \tag{88}$$

Under the additive surrogate, the corresponding excess masked risk is at most $2q\epsilon$.

Optimality of $\widehat{C}$ under estimated scores gives $\sum_{l \in \widehat{C}} \widehat{I}_i^{(l)} \ge \sum_{l \in C^*} \widehat{I}_i^{(l)}$. Replace each estimated score on the left by at most $I_i^{(l)} + \epsilon$ and each score on the right by at least $I_i^{(l)} - \epsilon$, then rearrange.

Unlike exact ranking recovery, Proposition D.4.3 does not require a nonzero boundary gap. Even if several layers exchange order, the selected set remains near-optimal in total importance when estimation error is small.

Together, the two observations do not merely suggest three independent heuristics. They establish a progressive argument: heterogeneous layer contributions motivate expert refinement; refined experts improve the objects on which routing operates; and the heterogeneous optima across layers motivate an input-adaptive, layer-aware integration rule.

Table 13: Scope of the theoretical claims. "Deterministic" means that the result holds for any fixed trained experts satisfying the listed analytical condition; it does not average over training randomness.

Claim	Type	Main requirement
Exact layer-ablation identity	Unconditional analytical identity	Continuous differentiability on the ablation path
Layer-importance bounds	Deterministic	Block smoothness
Removing a block improves target risk	Deterministic sufficient condition	Adverse gradient alignment dominates curvature remainder
Empirical importance recovers population ranking	Statistical	I.i.d. validation sample, bounded or suitably light-tailed paired differences, and a nonzero ranking margin
Global feasible set is contained in layer-wise set	Unconditional geometric	Same selected experts and block partition
Strictly larger layer-wise set	Geometric sufficient condition	At least two informative blocks and identifiable convex representations
Exact Exact global-vs-layer-wise optimization gap	Deterministic local model	Block-separable quadratic surrogate and interior optima
Transfer of quadratic advantage to true risk	Deterministic local	Hessian Lipschitzness and sufficiently small updates
Top-k routing approximation	Deterministic	Bounded expert updates and small discarded routing mass
Auxiliary-update perturbation bound	Deterministic	Smooth conditional risk

D.5 Assumption audit and scope of the claims

For clarity, we summarize which conclusions are unconditional and which are conditional.

Three limitations are worth emphasizing. First, non-uniformity is measured, not created, by ablation; if a particular expert truly has uniform contributions, LIER cannot obtain a strict ranking benefit. Second, the larger layer-wise class can increase statistical or selection complexity. MWM controls this by using a shared lightweight router, Top-k sparsity, a binary core/auxiliary partition, and the scalar attenuation λ, rather than learning an unconstrained coefficient for every parameter. Third, parameter-space bounds are local. They justify the design in the regime of moderate LoRA updates but do not imply that arbitrarily distant networks can be interpolated without loss barriers.

D.6 Worked constructions, counterexamples, and sanity checks

This section gives small constructions that isolate the mechanisms in the theorems. They are not intended to model the full geometry of a Transformer; their purpose is to show that the inequalities can be tight, that the claimed benefits can be strict, and that the stated assumptions cannot simply be removed.

D.6.1 A block whose removal strictly improves the target expert

Consider two scalar parameter blocks and the target risk

$$\mathcal{R}(\theta^{(1)}, \theta^{(2)}) = \frac{1}{2}(\theta^{(1)} - 1)^2 + \frac{q}{2}(\theta^{(2)})^2, \qquad q > 0. \tag{89}$$

Let the base be $(0, 0)$ and the learned expert update be $\mathbf{\Delta} = (1, 1)$, so the complete expert is $(1, 1)$. Its risk is $q/2$. Ablating the first block produces $(0, 1)$ with risk $(1 + q)/2$, hence

$$I^{(1)} = \frac{1}{2} > 0. \tag{90}$$

Ablating the second block produces $(1, 0)$, the target optimum, and therefore

$$I^{(2)} = -\frac{q}{2} < 0. \tag{91}$$

The two equal-magnitude parameter updates have opposite functional roles. The first is essential to specialization; the second is detrimental and its removal gives a strict improvement of $q/2$. At the full expert, $g^{(2)} = q$, $\Delta^{(2)} = 1$, and $\beta_2 = q$, so condition equation 28 becomes $q > q/2$ and is satisfied. The smoothness bound certifies the observed improvement.

The construction also illustrates distribution mismatch. The second update could be useful on a training distribution that rewards $\theta^{(2)} \approx 1$, while the target subdomain in equation 89 rewards $\theta^{(2)} \approx 0$. Ablation detects the mismatch without requiring access to the original training trajectory.

D.6.2 A NON-UNIFORM EXPERT NEAR A STATIONARY POINT

Let

$$\mathcal{R}(\boldsymbol{\theta}) = \frac{1}{2}\sum_{l=1}^{L} h_l(\theta^{(l)} - u_l)^2, \qquad h_l > 0, \tag{92}$$

and let the expert be the minimizer $\boldsymbol{\theta}_i = \boldsymbol{u}$. If the base is zero, then $\Delta_i^{(l)} = u_l$ and the ablated model sets coordinate l to zero. The exact importance is

$$I_i^{(l)} = \frac{1}{2} h_l u_l^2. \tag{93}$$

The expert gradient is exactly zero, yet contributions remain non-uniform whenever the curvature-weighted energies $h_l u_l^2$ differ. This example shows why a first-order saliency score can be uninformative at convergence and why the ablation score retains meaningful second-order information.

Uniformity is possible but highly structured: if $h_l u_l^2 = c$ for every layer, then all scores equal $c/2$. This is a counterexample to any unconditional claim that trained layers must have unequal importance. It is precisely why Observation 1 should be presented as an empirical observation supported by conditional theory rather than as a universal law.

D.6.3 STRICT ADVANTAGE OF LAYER-WISE MERGING

Consider two experts and two scalar blocks with updates

$$\boldsymbol{\Delta}_1 = (1, 1), \qquad \boldsymbol{\Delta}_2 = (-1, -1), \tag{94}$$

and target update $\boldsymbol{u} = (1, -1)$. A global convex mixture with coefficient α on expert 1 produces

$$\boldsymbol{\delta}_{\mathrm{g}}(\alpha) = (2\alpha - 1, 2\alpha - 1). \tag{95}$$

For squared error $J(\boldsymbol{\delta}) = \frac{1}{2}\|\boldsymbol{\delta} - \boldsymbol{u}\|_2^2$, the best global coefficient is $\alpha = 1/2$, producing $(0, 0)$ and risk

$$J_{\mathrm{g}}^* = 1. \tag{96}$$

Layer-wise merging selects expert 1 at the first block and expert 2 at the second, producing $(1, -1)$ and

$$J_{\mathrm{lw}}^* = 0. \tag{97}$$

The oracle gap is therefore exactly one. In the notation of Theorem D.3.2, the two block-wise preferred coefficients are $\alpha_1^* = 1$ and $\alpha_2^* = 0$, while the global coefficient is their weighted mean $1/2$. The gap is the resulting weighted variance.

The example also illustrates why expert selection alone is insufficient. Both experts are relevant: expert 1 supplies the desired first block and expert 2 supplies the desired second block. Selecting one complete expert yields loss two, and averaging them yields loss one. Only block-wise integration recovers the desired complementary composition.

D.6.4 A CASE IN WHICH LAYER-WISE MERGING GIVES NO BENEFIT

Let the two experts instead be $\boldsymbol{\Delta}_1 = (1, 1)$ and $\boldsymbol{\Delta}_2 = (-1, -1)$ with target $\boldsymbol{u} = (c, c)$ for some $c \in [-1, 1]$. Every block prefers the same coefficient $\alpha^* = (c + 1)/2$. Global merging reaches the target exactly, so both global and layer-wise optima have zero risk. This realizes the equality condition of Theorem D.3.2: greater expressivity does not imply a strict gain on every problem.

D.6.5 CROSS-LAYER INTERACTION CAN INVALIDATE NAIVE ADDITIVITY

Consider

$$\mathcal{R}(\theta^{(1)}, \theta^{(2)}) = \frac{1}{2}(\theta^{(1)} + \theta^{(2)} - 1)^2 \tag{98}$$

and expert $(\theta^{(1)}, \theta^{(2)}) = (1, 0)$. The full model has zero risk. Ablating the first block increases risk by $1/2$, whereas ablating the second has no effect. Now perturb the expert to $(1, -\varepsilon)$. The two blocks partly cancel, and the off-diagonal Hessian blocks are nonzero. The importance of jointly ablating both blocks differs from the sum of the individual scores by a term proportional to the product of the two updates. This is the interaction term in Lemma D.2.9.

An even sharper example is the expert $(1, -1)$ under target zero and risk $\mathcal{R} = \frac{1}{2}(\theta^{(1)} + \theta^{(2)})^2$. The complete expert has zero risk, but ablating either block individually increases risk to $1/2$, while ablating both returns to zero risk. Consequently,

$$I(\{1\}) + I(\{2\}) = 1, \qquad I(\{1, 2\}) = 0. \tag{99}$$

No additive single-block score can represent this cancellation exactly. The example justifies the bounded-interaction qualification in Theorem D.2.5 and suggests checking pairwise interactions when ablation curves display unstable rankings.

D.6.6 A NUMERICAL READING OF THE RANKING BOUND

Suppose an expert has $L = 32$ adapted blocks, the paired per-example loss difference has range at most $B_i = 1$, and we seek confidence $1 - \delta = 0.95$. If the population gap at the core-set boundary is $\gamma_i = 0.1$, Corollary D.2.6 requires

$$n > \frac{2}{0.1^2} \log \frac{64}{0.05} \approx 1431 \tag{100}$$

validation examples for the simple Hoeffding guarantee. This number is a conservative worst-case bound, not a recommended universal validation size. The paired estimator may concentrate much faster when the loss differences have low variance; a Bernstein analysis can exploit that variance. The example is useful mainly because it reveals the dependencies: logarithmic in the number of blocks and confidence, quadratic in the inverse score gap.

D.6.7 WHY A LEARNED ROUTER SHOULD ACT ON REFINED EXPERTS

Let a router make a coefficient error $\boldsymbol{e}$ between two experts. At a block where their updates differ by $\boldsymbol{d}_l = \boldsymbol{\Delta}_1^{(l)} - \boldsymbol{\Delta}_2^{(l)}$, a scalar coefficient error e_l induces parameter error $e_l \boldsymbol{d}_l$. If an auxiliary component enlarges $\|\boldsymbol{d}_l\|_2$ without improving target discrimination, it increases the sensitivity bound in Theorem D.3.6 but supplies no compensating routing signal. Suppressing that component reduces the worst-case downstream effect of the same routing error. This calculation does not guarantee that every low-importance block harms routing, but it establishes a concrete mechanism by which refinement before routing can improve robustness.

D.6.8 BOUNDARY CASES AND FALSIFIABLE PREDICTIONS

The theory yields several predictions that can be checked experimentally:

1. If the ablation-score variance across layers approaches zero, the benefit of LIER over uniform retention should diminish.
2. If all layers prefer nearly identical expert coefficients, the gap between global and layer-wise merging should diminish according to equation 64.
3. If the discarded Top-k routing mass τ_k is large, aggressive expert truncation should become less reliable according to equation 78.
4. If pairwise ablation interactions are large, independent layer ranking should become unstable and group-aware refinement should help.
5. If the auxiliary alignment is beneficial but weak, an intermediate λ should outperform both hard pruning and full retention, as in Theorem D.3.8.

These statements turn the appendix into more than a post-hoc explanation: they identify conditions under which MWM should succeed, cases in which its advantage should vanish, and diagnostics that can falsify the proposed mechanism.

D.7 Additional results for layer-wise refinement

Although the implementation can score LoRA-adapted layers, the analysis extends to neurons or groups. Partition block l into groups $g = 1, \dots, G_l$ and write $\boldsymbol{\Delta}_i^{(l)} = (\boldsymbol{\Delta}_i^{(l,1)}, \dots, \boldsymbol{\Delta}_i^{(l,G_l)})$. Define neuron-group ablation importance

$$I_i^{(l,g)} = \mathcal{R}_i(\boldsymbol{\theta}_i - \boldsymbol{E}_{l,g}\boldsymbol{\Delta}_i^{(l,g)}) - \mathcal{R}_i(\boldsymbol{\theta}_i). \tag{101}$$

Every result in Section D.2 applies after replacing l by (l, g). The finer partition yields a superset of the coarse mask class.

[Refinement monotonicity of mask expressivity] Let $\mathcal{P}_c$ be a coarse partition of parameter coordinates and let $\mathcal{P}_f$ refine $\mathcal{P}_c$. If every coarse mask can be represented by tying the masks of its fine subgroups, then the set of updates reachable under $\mathcal{P}_c$ is contained in the set reachable under $\mathcal{P}_f$. The same statement holds for merging coefficients.

Assign every fine subgroup inside a coarse block the coarse block's mask or coefficient. This reproduces any coarse operation exactly.

The proposition parallels Theorem D.3.1: finer control cannot worsen the best achievable training objective, but it can raise estimation and storage costs. Layer-wise scoring is therefore a practical point on the granularity–complexity spectrum, while the name "neuron-adaptive" emphasizes that the retained masks act on the underlying adapted units.