H-OM: HADAMARD-ORTHOGONAL MODEL MERGING WITH RECOVERABLE TASK UPDATES

Anonymous authors
Paper under double-blind review

ABSTRACT

Model merging combines task-specific updates into a single model without joint retraining. However, existing independent and sequential methods for model merging cannot adapt to real-world deployments due to dependence on original adapters or preconditioned tasks. In this paper, to address the problem, we propose Hadamard Orthogonal Merging (H-OM) that constructs mutually orthogonal output subspaces for Low-Rank Adaptation (LoRA). H-OM learns to adapt to each task learns within its assigned subspace to simultaneously allow parallel adaptation and maintain orthogonal updates. Furthermore, we develop Coordinate-Exposure Uniformity (CEU) to guide subspace construction for effective adaptation, and induce subspaces with uniform coordinate exposure from disjoint sets of randomized Hadamard basis vectors. H-OM is flexible to enable component removal and reweighting without retaining the original adapters, since it can exactly recover the weight updates for tasks from their projection in the assigned subspaces under linear merging. Experiments on VTAB-1K and NLP-15 demonstrate that H-OM yields state-of-the-art accuracy in model merging with evidently fewer trainable parameters than standard LoRA. Remarkably, H-OM recovers all 19 VTAB-1K task updates from the merged model using the pretrained base model and reproducible basis metadata, without compromising single-task accuracy.

1 INTRODUCTION

The pretraining-finetuning paradigm has become prevailing to adapt large language models (LLMs) to varying tasks. Low-Rank Adaptation (LoRA) (Hu et al., 2022) achieves parameter-efficient task adaptation by learning compact updates under fixed pretrained weights. However, separately optimizing each adapter for its own task does not necessarily ensure that multiple updates remain compatible for various tasks and potential causes degraded performance on constituent tasks. To this end, model merging aggregates these updates into a single model without joint retraining on the combined data for tasks (Ilharco et al., 2023; Yadav et al., 2023).

Existing paradigms for model merging can be categorized into independent methods and sequential methods. Independent methods address merge interference by modifying or aligning task updates obtained through parallel task-specific fine-tuning (Yadav et al., 2023; Yu et al., 2024; Stoica et al., 2025; Panariello et al., 2025). These methods support parallel adaptation, but make individual components difficult to recover or edit without the original adapters. By contrast, sequential methods processes tasks sequentially using previously trained adapters or the current merged model to guide the adaptation or integration of new updates (Wang et al., 2023; Qiao and Mahdavi, 2026; Tang et al., 2025). They support continual task addition, but is restricted in adaptation, since they depend on earlier tasks for training a new task.

Moreover, both methods could be limited in real-world deployments that require to accommodate to changing tasks requirements by removing outdated task updates or editing current task updates. Independent methods learn merged parameters for multiple tasks in parallel, and cannot identify the contribution of an individual task to merged parameters and separate its updates. Sequential methods require to follow data dependence between tasks for discriminate a specific task and cannot break the dependence to make online removal or editing. Therefore, there still lacks of a model merging method that simultaneously achieves high-performance task adaptation and guarantees the flexibility to allow online removal and editing in real-world deployments.

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.

In this paper, we propose Hadamard Orthogonal Merging (H-OM) that projects the weight updates of different tasks to mutually orthogonal output subspaces spanned by Hadamard basis for parallel fine-tuning to accomplish these requirements. This allows tasks to train in parallel without accessing one another's learned updates. Under linear merging, the weight update of a task can be exactly recovered from the projection onto its assigned subspace. H-OM is flexible to remove or reweight the recovered component without access to original adapters, replace it with a new update trained within the same subspace, and introduce new tasks into the subspaces orthogonal to occupied subspaces.

Furthermore, we develop the Coordinate-Exposure Uniformity (CEU) metric to quantify the imbalanced contribution of Hadamard basis without using task data. CEU measures the uniformity of retention ratios of energy of projected gradients to output coordinates for guiding subspace construction for task adaptation. We construct mutually orthogonal task subspaces from disjoint sets of randomized Hadamard basis vectors by attaining maximum CEU with the same retention ratio for each output coordinate. This achieves higher mean merged accuracy than coordinate slicing and Gaussian QR bases, as demonstrated in Figure 2c.

Our main contributions are summarized as follows.

- We propose a model merging framework named H-OM that constructs mutually orthogonal task subspaces to support parallel adaptation for multiple tasks. Under linear merging, H-OM enables exact recovery of weighted task updates by projection to allow removing and reweighting existing components without accessing their original adapters.
- We reveal that uniform coordinate exposure is associated with better model merging performance and develop a CEU metric to guide subspace construction for effective task adaptation. CEU induces randomized Hadamard construction for subspaces to preserve task orthogonality and facilitate subspace-based learning for tasks.
- H-OM yields average accuracy gains of 2.72%–4.52% over the strongest evaluated VTAB-1k baseline across six merge operators and achieves 2.36 points above the strongest evaluated merging baseline on NLP-15, using only 52.6%–65.6% trainable adapter parameters of standard LoRA. After FP32 serialization and reload, H-OM recovers all 19 VTAB-1K components without original adapters and reproduces their corresponding single-task accuracies at the reported precision.

2 Related Work

Model merging. Model merging integrates information from multiple fine-tuned models at the parameter level. Model soups average checkpoints to improve performance on a target task (Wortsman et al., 2022), while Task Arithmetic combines task vectors to construct multitask models and edit model behavior (Ilharco et al., 2023). Subsequent methods address different sources of degradation after merging. TIES-Merging removes small updates and resolves sign conflicts (Yadav et al., 2023); DARE randomly drops and rescales update entries (Yu et al., 2024); and WuDi-Merging uses task vectors to guide interference reduction (Cheng et al., 2025). Spectral methods operate on singular components: AdaRank selects components using unlabeled test data (Lee et al., 2026), and SVC calibrates the singular values of a merged update to reduce over-accumulation in shared directions (Li et al., 2026). MaTS matches trained models within task parameter subspaces (Tam et al., 2024). These methods primarily operate on completed task models or updates, whereas H-OM specifies task subspaces before fine-tuning.

Merging LoRA adapters. LoRA adapters introduce factorized update geometry that motivates specialized merging strategies. KnOTS aligns LoRA updates through a joint singular value decomposition (Stoica et al., 2025). Core Space constructs a common representation from the low-rank factors, reducing the cost of alignment and merging (Panariello et al., 2025). These methods align, re-express, or calibrate LoRA updates after task-specific training. In sequential settings, O-LoRA constrains new task updates relative to previously learned low-rank subspaces (Wang et al., 2023), SLAO continually merges orthogonally initialized updates into a single LoRA representation (Qiao and Mahdavi, 2026), and OPCM incorporates new task updates with reference to the old merged model through orthogonal projection (Tang et al., 2025). These methods rely on learned adapters or merged states when incorporating new tasks. In contrast, H-OM preassigns task subspaces before fine-tuning, allowing task-specific adaptation to proceed independently.

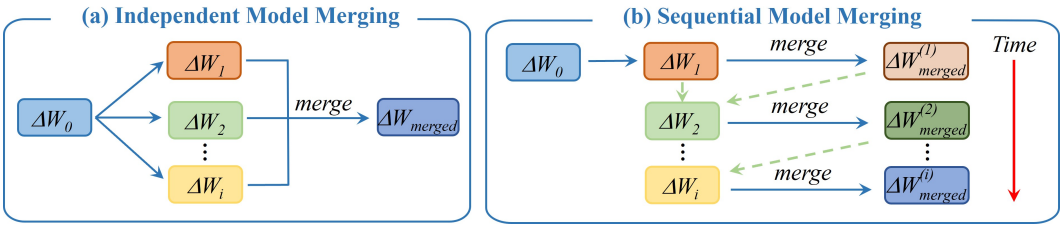


Figure 1: Illustrative task-adaptation and merging pipelines. (a) IMM learns task-specific updates independently from a shared pretrained model and combines them. (b) SMM uses information from preceding tasks during adaptation or merging. The green arrows indicate possible dependencies on previously learned task updates or merged models.

Adaptation for subsequent merging. OSRM uses information from other-task data distributions to construct task-specific LoRA subspaces (Zhang and Zhou, 2025). MergOPT instead improves robustness to subsequent merging by simulating anticipated merge-induced parameter shifts during independent fine-tuning (Yang et al., 2026). H-OM differs by preallocating fixed, mutually orthogonal task subspaces without task-data statistics or previously trained adapters. This structure additionally provides a known decomposition from which weighted task components can be exactly recovered under linear merging.

3 Proposed Method

3.1 Preliminaries

We consider adapting a shared pretrained model to T tasks on datasets $\{\mathcal{D}_i\}_{i=1}^T$, *i.e.*, $\mathcal{D}_i$ for task i. For a layer with the output dimension d_{out} and input dimension d_{in} to be adapted, let $W_0 \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ denote its pretrained weight and W_i the weight obtained by fine-tuning W_0 on $\mathcal{D}_i$. The task-specific update $\Delta W_i := W_i - W_0$ adapts the pretrained weight to task i. To reduce the number of trainable parameters required for task adaptation, Low-Rank Adaptation (LoRA) (Hu et al., 2022) parameterizes this update as the product of two trainable matrices $A_i \in \mathbb{R}^{r \times d_{\text{in}}}$ and $B_i \in \mathbb{R}^{d_{\text{out}} \times r}$ with much lower LoRA rank $r \ll \min(d_{\text{out}}, d_{\text{in}})$

$$W_i = W_0 + \Delta W_i, \qquad \Delta W_i = B_i A_i. \tag{1}$$

During adaptation, A_i and B_i are optimized with W_0 fixed. This formulation is applied for adaptation of each layer and we omit layer indices without introducing ambiguity.

Here, we focus on **model merging** that aims to integrate the T task-specific updates $\{\Delta W_i\}_{i=1}^T$ into a unified update ΔW_{merge}. The adapted weight becomes $W_{\text{merge}} = W_0 + \Delta W_{\text{merge}}$ for each layer. Existing methods for model merging can be roughly classified into independent model merging (IMM) and sequential model merging (SMM) according to information dependencies between tasks, as illustrated in Figure 1 and formalized below.

Definition 1 (Independent Model Merging (IMM))**.** Adaptation to a task does not access other tasks' data or data statistics, previously trained adapters, or merged models. Each task update ΔW_i is first learned independently from the shared pretrained model using its own dataset $\mathcal{D}_i$ and predefined task-specific constraints. Subsequently, task updates are combined with the Merge operator.

$$\Delta W_{\text{merge}} = \text{Merge}\left(\Delta W_1, \Delta W_2, \ldots, \Delta W_i\right). \tag{2}$$

Definition 2 (Sequential Model Merging (SMM))**.** SMM processes tasks sequentially to condition the adaptation or merging on information from earlier tasks, *e.g.*, previously trained adapters or current merged model. ΔW_i for adaptation to task i is learned on $\mathcal{D}_i$ under the constraint of the information $\mathcal{H}_{i-1}$. The merged update is then computed as

$$\Delta W^i_{\text{merged}} = \text{Merge}\Big(\Delta W^{i-1}_{\text{merged}}, \Delta W_i; \mathcal{H}_{i-1}\Big), \tag{3}$$

where $\Delta W^i_{\text{merged}}$ denotes the merged update of the first i tasks, with $\Delta W^0_{\text{merged}} = 0$ representing the state before any task is merged.

3.2 Motivation and Challenges

IMM supports independent adaptation from a shared pretrained model, allowing tasks to train in parallel. However, separately optimized updates may interfere when merged, reducing performance on the constituent tasks. SMM can mitigate this interference by using previous task updates or the current merged model to guide adaptation or merging. When such guidance is applied during adaptation, new task training depends on prior training results, limiting its independence.

After deployment, changing task requirements may call for adding, removing, or replacing task updates. Without the original adapters, removal or replacement becomes difficult because individual updates generally cannot be uniquely recovered from the merged parameters.

Within the independent adaptation setting, our goal is *to support parallel task training, mitigate merge interference, and enable subsequent task addition and removal without retaining the original adapters*. Achieving this goal requires addressing two related challenges.

Learning Orthogonal Task Updates in Parallel. Separately optimized task updates may interfere when merged, and their sum generally does not uniquely identify individual task updates. Orthogonal task subspaces separate these updates, but constructing them from trained adapters introduces training dependencies. The challenge is to establish separation without requiring other tasks' training results. Preallocating orthogonal subspaces supports parallel training and recovery of weighted task updates by projection after linear merging, without retaining the original adapters.

Effective Learning within Fixed Subspaces. Preallocating orthogonal subspaces enables parallel training, but also fixes the update directions available to each task. Effective adaptation therefore depends on both the choice of these directions and the flexibility to combine them during learning. The challenge is to design the task subspaces and the trainable parameters within them to support effective adaptation while preserving task separation.

3.3 H-OM: Hadamard Orthogonal Merging

To address these challenges, we propose Hadamard Orthogonal Merging (H-OM) that preallocates mutually orthogonal task subspaces to mitigate merge interference while allowing tasks to train in parallel. To avoid unequal restrictions on output coordinates within these fixed subspaces, we introduce Coordinate-Exposure Uniformity (CEU) as a criterion for subspace construction. Higher CEU indicates more uniform coordinate exposure and is associated with better merged performance in our controlled experiments. A randomized Hadamard basis attains maximum CEU, while trainable factors learn task-specific combinations of the assigned directions. The same orthogonal structure enables exact recovery of weighted task updates by projection after linear merging, supporting task removal or replacement without retaining the original adapters. The training and linear-merging procedure is summarized in Algorithm 1 in Appendix B.

3.3.1 Orthogonal Parameterization for Parallel Training

To mitigate interference in merging task updates, we constrain their output directions to mutually orthogonal subspaces. These subspaces are assigned before training to enforce the orthogonality constraint on each task without accessing learned updates for other tasks. For task i, let $Q_i \in \mathbb{R}^{d_{\text{out}} \times c_i}$ be an orthonormal basis for the assigned output subspace of dimension c_i, where the columns of Q_i specify the available output directions. For Q_i, $i = 1, \cdots, T$, it satisfies that $\sum_{i=1}^{T} c_i \leq d_{\text{out}}$ and

$$Q_i^\top Q_i = I_{c_i}, \qquad Q_i^\top Q_j = 0 \quad \text{for } i \neq j. \tag{4}$$

Therefore, the subspace Q_i provides c_i orthonormal output directions for task i and subspaces for different tasks are separated without overlapping.

To retain task adaptation within these fixed subspaces, we allow each task to learn combinations of its assigned directions. Specifically, we express the output-side LoRA factor as $B_i = Q_i \theta_i$, giving

$$\Delta W_i = Q_i \theta_i A_i, \tag{5}$$

where $\theta_i \in \mathbb{R}^{c_i \times r}$ learns the combination coefficients and $A_i \in \mathbb{R}^{r \times d_{\text{in}}}$ is the trainable input-side factor. Thus, Q_i determines which output directions the task may use, while θ_i and A_i determine the update learned from its data. During adaptation, each task optimizes θ_i and A_i on $\mathcal{D}_i$, keeping W_0

and Q_i fixed. Every column of ΔW_i remains in the subspace spanned by Q_i, so $\Delta W_i^\top \Delta W_j = 0$ for $i \neq j$ throughout training. Orthogonality is therefore preserved by the parameterization itself, allowing tasks to train in parallel without using one another's trained adapters.

3.3.2 Subspace Selection

The parameterization above keeps task updates orthogonal, but also limits the output directions available to each task. Effective adaptation therefore depends on both the choice of these directions through Q_i and the learning of θ_i and A_i within the assigned subspace. To guide subspace construction, we examine how Q_i affects the gradient information available for training these factors.

Let $\mathcal{L}_i$ denote the training loss for task i, and let $G_i = \partial \mathcal{L}_i / \partial W_i \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ be the full weight gradient at the current task model. The chain rule gives

$$\nabla_{\theta_i} \mathcal{L}_i = (Q_i^\top G_i) A_i^\top, \ \nabla_{A_i} \mathcal{L}_i = \theta_i^\top (Q_i^\top G_i). \tag{6}$$

Both factor gradients depend on $Q_i^\top G_i$. Let $P_i = Q_i Q_i^\top$ be the orthogonal projector onto the subspace. Since $Q_i^\top G_i = Q_i^\top (P_i G_i)$, only the projected gradient $P_i G_i$ can contribute to training. For $G_i \neq 0$, the fraction of gradient energy retained by the subspace is $\|P_i G_i\|_F^2 / \|G_i\|_F^2$. The factors determine how this information affects the optimization step.

The gradient retention ratio depends on how the task gradient aligns with the assigned subspace. The low absolute cosine similarities among the gradients of nine VTAB-1k tasks in Figure 2(a) illustrate the diversity of gradient directions that fixed subspace assignments must accommodate. Since these subspaces are chosen without task data, we seek a criterion that can be evaluated from Q_i alone. We therefore examine how the subspace restricts each output coordinate, with the aim of avoiding unequal access to coordinates whose task-specific importance is unknown before training.

Consider a gradient restricted to output coordinate k, $G_i^{(k)} = e_k g^\top$, where $e_k \in \mathbb{R}^{d_{\text{out}}}$ is the k-th standard basis and $g \in \mathbb{R}^{d_{\text{in}}}$ is nonzero. Its retained energy fraction defines the coordinate exposure:

$$\ell_k(Q_i) = \frac{\|P_i G_i^{(k)}\|_F^2}{\|G_i^{(k)}\|_F^2} = \|P_i e_k\|_2^2 = e_k^\top P_i e_k. \tag{7}$$

This is the leverage score of row k of Q_i. A smaller exposure means that less gradient energy from that coordinate is retained; in particular, $\ell_k(Q_i) = 0$ means that task i cannot modify output coordinate k. Because P_i is an orthogonal projector of rank c_i, the total exposure satisfies

$$\sum_{k=1}^{d_{\text{out}}} \ell_k(Q_i) = \operatorname{tr}(P_i) = c_i. \tag{8}$$

At a fixed subspace dimension, increasing exposure for some coordinates therefore reduces the amount available to others. Without task information to prioritize particular coordinates, we maximize the minimum coordinate exposure. Since $\min_k \ell_k(Q_i) \leq c_i / d_{\text{out}}$, the largest possible minimum is c_i / d_{out}, attained if and only if $\ell_k(Q_i) = c_i / d_{\text{out}}$ for every k.

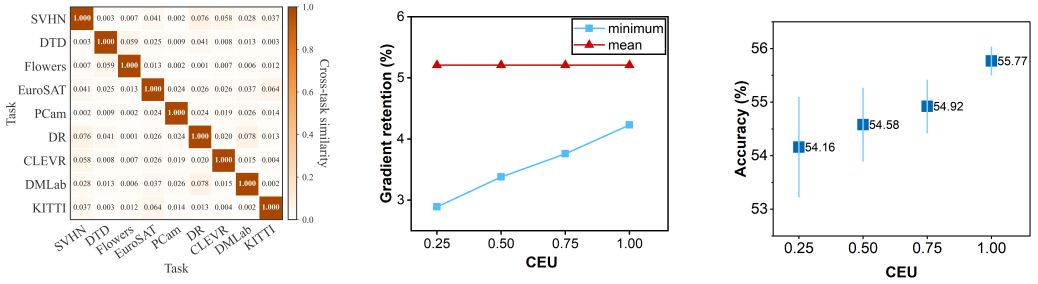
To quantify this balance, we apply Jain's fairness index (Jain et al., 1984) to the coordinate exposures and define Coordinate-Exposure Uniformity (CEU) as

$$\text{CEU}(Q_i) = \frac{\left(\sum_{k=1}^{d_{\text{out}}} \ell_k(Q_i)\right)^2}{d_{\text{out}} \sum_{k=1}^{d_{\text{out}}} \ell_k(Q_i)^2} = \frac{c_i^2}{d_{\text{out}} \sum_{k=1}^{d_{\text{out}}} \ell_k(Q_i)^2}. \tag{9}$$

Since $0 \leq \ell_k(Q_i) \leq 1$, Cauchy–Schwarz and $\ell_k(Q_i)^2 \leq \ell_k(Q_i)$ give

$$\frac{c_i^2}{d_{\text{out}}} \leq \sum_{k=1}^{d_{\text{out}}} \ell_k(Q_i)^2 \leq c_i.$$

Thus, $c_i / d_{\text{out}} \leq \text{CEU}(Q_i) \leq 1$, with equality at 1 if and only if all output coordinates receive equal exposure. Maximizing CEU therefore maximizes the minimum coordinate exposure. For a general gradient, however, retained energy also depends on correlations across output coordinates. We therefore test whether higher CEU leads to more balanced gradient retention. For each reference



(a) gradient direction similarity. (b) gradient retention. (c) Coordinate exposure

Figure 2: (a) Absolute pairwise cosine similarity of initial gradients across nine VTAB-1k tasks. (b) Mean and minimum projected-gradient energy retention versus mean CEU. (c) VTAB-1k merged accuracy at four mean CEU levels in a controlled ablation. Points show means and error bars indicate standard deviation. The protocol and full results for (c) are given in Appendix B.2.

gradient G, we measure retention in subspace i as $\|Q_i^\top G\|_F^2/\|G\|_F^2$. and compute its mean and minimum across the 19 candidate subspaces. As shown in Fig. 2(b), mean retention remains nearly unchanged, whereas the averaged minimum retention increases from 2.89% to 4.23% as mean CEU increases from 0.25 to 1.00, indicating less severe restriction by the least favorable subspace. This trend is also reflected in downstream performance. In the controlled VTAB-1k experiment of Fig. 2(c), where only subspace orientation is varied, mean merged accuracy increases from 54.16% to 55.77%, while the standard deviation over six runs decreases from 0.93 to 0.26. Together, these results support CEU as a practical criterion for selecting uniformly exposed task subspaces.

3.3.3 Hadamard Subspace Construction

We next construct task subspaces that attain this uniformity while preserving mutual orthogonality. For an output dimension d_{out} admitting a Hadamard matrix, let $H_{d_{\text{out}}} \in \{-1,+1\}^{d_{\text{out}}\times d_{\text{out}}}$ satisfy $H_{d_{\text{out}}} H_{d_{\text{out}}}^\top = d_{\text{out}} I_{d_{\text{out}}}$. The normalized matrix $\Phi_{d_{\text{out}}} = H_{d_{\text{out}}}/\sqrt{d_{\text{out}}}$ has orthonormal columns and $|(\Phi_{d_{\text{out}}})_{kj}|^2 = 1/d_{\text{out}}$ for every entry. Using a fixed seed ξ, we construct

$$U = D_\xi P_{\text{row},\xi} \Phi_{d_{\text{out}}} P_{\text{col},\xi}, \tag{10}$$

where D_ξ is a diagonal matrix with entries in $\{-1,+1\}$, and $P_{\text{row},\xi}$ and $P_{\text{col},\xi}$ are permutation matrices. These transformations preserve both $U^\top U = I_{d_{\text{out}}}$ and $|U_{kj}|^2 = 1/d_{\text{out}}$. The seed fixes the signs and permutations before training, yielding a data-free reproducible construction

Task i is assigned a column-index set $\mathcal{I}_i$ with $|\mathcal{I}_i| = c_i$, and its basis is $Q_i = U[:,\mathcal{I}_i]$. The sets $\{\mathcal{I}_i\}$ are mutually disjoint, so the resulting task bases satisfy Eq. (4). Moreover,

$$\ell_k(Q_i) = \sum_{j\in\mathcal{I}_i} |U_{kj}|^2 = \frac{c_i}{d_{\text{out}}}, \qquad k = 1,\dots,d_{\text{out}}. \tag{11}$$

Thus, every assigned subspace attains $\text{CEU}(Q_i) = 1$, independently of which columns it receives. The construction provides uniform coordinate exposure within each task subspace and orthogonality between different subspaces.Beyond uniform coordinate exposure, the construction allows subspaces to be regenerated when needed. Once the Hadamard factorization and randomization convention are fixed, the seed ξ determines U, and the assigned indices $\mathcal{I}_i$ select $Q_i = U[:,\mathcal{I}_i]$. Thus, neither the full dictionary nor the task-basis matrices need to be retained for later use. This property supports component recovery in Sec. 3.3.5; the editing operations are detailed in Appendix F.

3.3.4 Training and Linear Merging

Once the task bases have been assigned, each task trains its factors while keeping W_0 and Q_i fixed. For $c_i \geq r$, we initialize

$$\theta_i^{(0)} = \frac{c_i}{d_{\text{out}}} \begin{bmatrix} I_r \\ 0_{(c_i-r)\times r} \end{bmatrix}, \qquad A_i^{(0)} = 0_{r\times d_{\text{in}}}. \tag{12}$$

The zero initialization of A_i gives $\Delta W_i^{(0)} = 0$, so adaptation starts from the pretrained model. The nonzero initialization of θ_i allows A_i to receive gradients at the first optimization step. The initialization scale c_i/d_{out} is examined in Appendix A.4.

After training, the task updates are combined by linear merging,

$$\Delta W_{\text{merge}} = \sum_{i \in \mathcal{T}} \lambda_i \Delta W_i, \qquad W_{\text{merge}} = W_0 + \Delta W_{\text{merge}}, \tag{13}$$

where $\mathcal{T}$ is the set of merged tasks and λ_i is the coefficient of task i.

3.3.5 Exact Task-Component Recovery and Model Editing

Under the linear merge in Eq. (13), the preassigned orthogonal subspaces also allow each weighted task component to be isolated by projection after merging.

Proposition 1 (Exact Task-Component Recovery). *Let $\Delta W_{\text{merge}} = \sum_{i \in \mathcal{T}} \lambda_i \Delta W_i$, where $P_t = Q_t Q_t^\top$, $\Delta W_i = Q_i \theta_i A_i$, $Q_i^\top Q_i = I_{c_i}$, and $Q_i^\top Q_j = 0$ for $i \neq j$. For any task $t \in \mathcal{T}$, its weighted component is exactly recovered by*

$$P_t \Delta W_{\text{merge}} = Q_t Q_t^\top \Delta W_{\text{merge}} = \lambda_t \Delta W_t, \tag{14}$$

For a saved merge, let $M = W_{\text{merge}} - W_0$. Because Q_t is deterministically regenerated from the Hadamard construction metadata and its assigned column indices, $P_t M$ can be extracted without the original task adapter. Appendix F verifies this procedure after FP32 serialization and reload.

4 Experiments

4.1 Experimental Setup

Benchmarks and models. For vision, we evaluate CLIP ViT-B/32 (Radford et al., 2021) on the 19-task VTAB-1k benchmark (Zhai et al., 2019), which contains seven Natural, four Specialized, and eight Structured tasks. For language, we follow the 15-task setting used by SLAO (Qiao and Mahdavi, 2026) with LLaMA-2-7B-Chat (Touvron et al., 2023); we refer to this suite as *NLP-15*. For the task-order configuration study, the 15 tasks are grouped into Classification (C), Question Answering (Q), and Semantic Reasoning (R), and O1–O6 enumerate the six permutations of these three groups. The complete task list and order configurations are given in Appendix E. A supplemental Qwen2.5-1.5B-Instruct experiment is reported in Appendix D.

Baselines. We compare H-OM with methods under both paradigms defined in Sec. 3.1. IMM baselines include Task Arithmetic, TIES, DARE, DARE-TIES, KnOTS, Core Space, AdaRank, WuDi-Merging, SVC, and MergOPT, while SMM baselines include OSRM, OPCM, O-LoRA, and SLAO. Comparisons within each benchmark use the same backbone, task data, adapted modules, and evaluation split, following each method's information-access protocol. NLP-15 includes IMM and SMM methods, whereas VTAB-1k focuses on independently trained adapters and merge operators.

Implementation details. Unless otherwise stated, we adapt query and value projections with rank-8 adapters using 1,000 training examples per task. H-OM assigns $c = \lfloor d_{\text{out}}/T \rfloor$ columns per task ($c = 40$ for VTAB-1k and $c = 273$ for LLaMA) and initializes θ_i with scale c/d_{out}. We report mean task accuracy. VTAB-1k results are averaged over seeds $\{0, 1, 42\}$, whereas NLP-15 uses seed 42 for six training orders. The NLP-15 standard deviation is computed across these permutations rather than seeds. Experimental settings and metric definitions are provided in Appendix B.

4.2 Main Results

Results on VTAB-1k. Table 1 shows that H-OM achieves the highest mean merged accuracy under Task Arithmetic and each of the five additional operators, exceeding the strongest alternative under the same operator by 2.72–4.52 percentage points. With Task Arithmetic, its gains over MergOPT are 3.09, 1.54, and 4.61 points on Natural, Specialized, and Structured tasks, respectively. The consistent gains after merging are compatible with reduced overlap between updates assigned to orthogonal output subspaces. In a matched dictionary comparison, the Hadamard basis also yields a modest 0.44-point gain over seeded Gaussian QR (Appendix G.1).

Table 1: VTAB-1k results obtained by applying six downstream merging operators to the task updates produced by each adaptation method. Task Arithmetic is the default linear merge for H-OM; the other operators evaluate its compatibility with existing merging methods. Results are averaged over training seeds $\{0, 1, 42\}$.

Merge Operator	Method	Natural	Specialized	Structured	Avg. Acc.
Single-task	LoRA (Hu et al., 2022)	80.90±0.32	83.97±0.28	49.58±0.85	68.36±0.21
Task Arithmetic	LoRA (Hu et al., 2022)	73.94±0.61	72.52±0.58	23.23±1.02	52.29±0.55
	KnOTS (Stoica et al., 2025)	73.86±0.59	72.40±0.56	23.28±0.98	52.26±0.53
	MergOPT (Yang et al., 2026)	74.00±0.62	72.56±0.59	23.35±1.01	52.37±0.57
	H-OM	**77.09**±0.30	**74.10**±0.27	**27.96**±0.55	**55.77**±0.28
TIES	LoRA (Hu et al., 2022)	71.64±0.12	69.35±0.11	22.35±0.78	50.41±0.30
	KnOTS (Stoica et al., 2025)	71.71±0.11	69.30±0.10	22.39±0.76	50.44±0.29
	MergOPT (Yang et al., 2026)	72.04±0.13	69.66±0.12	22.45±0.77	50.66±0.35
	H-OM	**76.23**±0.14	**73.93**±0.12	**26.87**±0.24	**54.96**±0.13
DARE	LoRA (Hu et al., 2022)	73.96±0.65	72.51±0.60	23.37±0.92	52.36±0.60
	KnOTS (Stoica et al., 2025)	73.90±0.63	72.41±0.58	23.20±0.90	52.24±0.58
	MergOPT (Yang et al., 2026)	74.09±0.66	72.62±0.61	23.44±0.93	52.45±0.62
	H-OM	**76.66**±0.32	**74.19**±0.29	**28.33**±0.60	**55.79**±0.31
DARE-TIES	LoRA (Hu et al., 2022)	71.64±0.27	69.28±0.25	22.23±0.62	50.34±0.25
	KnOTS (Stoica et al., 2025)	71.74±0.26	69.19±0.24	22.36±0.61	50.41±0.24
	MergOPT (Yang et al., 2026)	72.10±0.28	69.61±0.26	22.42±0.63	50.66±0.26
	H-OM	**76.39**±0.06	**74.18**±0.05	**27.11**±0.38	**55.18**±0.14
WuDi-Merging	LoRA (Hu et al., 2022)	74.13±0.53	72.60±0.51	24.35±0.88	52.85±0.48
	KnOTS (Stoica et al., 2025)	74.08±0.52	72.55±0.50	24.42±0.87	52.85±0.47
	MergOPT (Yang et al., 2026)	74.25±0.54	72.68±0.52	24.48±0.89	52.97±0.50
	H-OM	**77.94**±0.18	**75.10**±0.16	**26.52**±0.38	**55.69**±0.19
SVC	LoRA (Hu et al., 2022)	74.43±0.08	72.15±0.07	24.56±1.06	52.95±0.43
	KnOTS (Stoica et al., 2025)	74.35±0.07	72.12±0.06	24.60±1.05	52.93±0.46
	MergOPT (Yang et al., 2026)	74.52±0.08	72.24±0.07	24.65±1.07	53.04±0.47
	H-OM	**78.57**±0.05	**73.27**±0.04	**27.43**±0.31	**55.92**±0.14

Table 2: NLP-15 accuracy across six training orders (O1–O6). The six configurations permute the three task groups; the same 15 tasks are retrained in each configuration using global seed 42. Methods are grouped according to the IMM and SMM paradigms defined in Sec. 3.1.

Method	O1	O2	O3	O4	O5	O6	Avg.	Std.
Single-task LoRA	87.25	87.36	87.28	87.33	87.23	87.35	87.30	0.05
Sequential Model Merging (SMM)								
OSRM (Zhang and Zhou, 2025)	76.85	77.55	76.54	78.30	77.29	77.99	77.42	0.61
OPCM (Tang et al., 2025)	78.24	78.53	79.64	78.44	79.64	79.35	78.97	0.58
O-LoRA (Wang et al., 2023)	77.95	78.33	78.54	79.43	80.40	80.73	79.23	1.05
SLAO Qiao and Mahdavi (2026)	81.82	81.08	80.98	79.87	78.89	78.56	80.20	1.19
Independent Model Merging (IMM)								
Task Arithmetic (Ilharco et al., 2023)	74.12	74.78	74.36	74.64	74.22	74.76	74.48	0.26
TIES-Merging (Yadav et al., 2023)	74.02	74.48	73.92	74.58	73.71	74.61	74.22	0.35
DARE-TIES	74.41	74.89	74.56	74.84	74.62	74.94	74.71	0.19
KnOTS (Stoica et al., 2025)	73.62	74.08	73.76	73.98	73.71	74.01	73.86	0.17
Core Space (Panariello et al., 2025)	78.24	79.12	78.55	79.31	78.41	78.93	78.76	0.39
AdaRank (Lee et al., 2026)	78.95	78.12	78.64	77.91	78.53	78.43	78.43	0.34
WuDi-Merging (Cheng et al., 2025)	76.58	77.11	76.65	77.04	76.47	76.95	76.80	0.24
SVC (Li et al., 2026)	78.52	79.24	78.61	79.35	78.48	79.20	78.90	0.37
MergOPT (Yang et al., 2026)	78.77	78.11	77.74	78.02	77.81	78.46	78.15	0.36
H-OM	**82.49**	**82.63**	**82.42**	**82.67**	**82.45**	**82.71**	**82.56**	**0.11**

Results on NLP-15. Table 2 shows that H-OM achieves the highest merged accuracy in the six task-group configurations. Its mean accuracy of 82.56% exceeds SLAO, the strongest merging baseline by mean accuracy, by 2.36 percentage points. H-OM also has the smallest standard deviation across configurations among merging methods (0.11 percentage points). These results show consistent performance under the evaluated protocol. Because task adapters are retrained for each configuration, the reported variation may include training effects alongside any effect of task-group order.

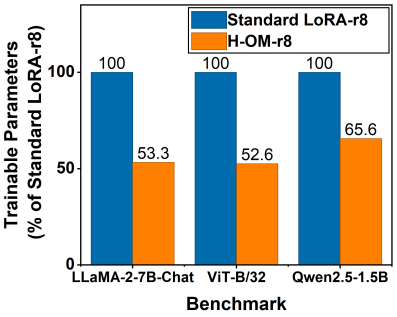


Figure 3: Trainable adapter parameters relative to standard LoRA-r8, normalized to 100% on each backbone.

Table 3: Deployment-style task-component recovery on VTAB-1k. We serialize the complete merged model in FP32 and reload it in a fresh process using only the pretrained base model and reproducible basis metadata, while excluding the original task adapters and stored task bases from the recovery procedure.

Deployment diagnostic	Result
FP32 reload rel. error	5.81×10^{-7}
Recovery rel. error (mean / max.)	$3.00/4.63 \times 10^{-4}$
Recovered accuracy gap (max.)	**0.00** pp
Deletion rel. error (mean / max.)	$6.72/7.74 \times 10^{-5}$
Delete–restore rel. error (max.)	1.18×10^{-8}

4.3 Analysis and Ablations

Trainable-parameter efficiency. Figure 3 shows that H-OM reduces trainable adapter parameters by 34.4%–47.4% relative to rank-8 LoRA across the three evaluated backbones, while retaining trainable adaptation within each assigned subspace. Absolute parameter counts and the fixed-θ_i control are provided in Appendix G.

Task-component recovery after deployment. We serialize the 19-task VTAB-1k merged model in FP32 and reload it in a fresh process with access to the serialized merged model, the pretrained base model, and basis-generation metadata, but not the original task adapters or stored task bases. All 19 task updates are reconstructed, and the recovered task models match their corresponding original H-OM task models with a maximum accuracy gap of 0.00 percentage points. The mean (maximum) relative recovery error is 3.00×10^{-4} (4.63×10^{-4}), while projection-based deletion has a mean (maximum) relative error of 6.72×10^{-5} (7.74×10^{-5}) against an oracle re-merge. Full protocols and editing results are reported in Appendix F.1.

5 Conclusion

We introduced H-OM, a LoRA-based model merging framework that preallocates fixed, mutually orthogonal task subspaces before independent fine-tuning. By combining a randomized Hadamard construction with Coordinate-Exposure Uniformity, H-OM provides uniform output-coordinate exposure while preserving task separation and trainable adaptation within each assigned subspace. Under linear merging, the orthogonal structure enables exact recovery of weighted task components by projection in exact arithmetic. In a deployment-style FP32 serialization-and-reload evaluation, all 19 VTAB-1k task updates are reconstructed without access to the original adapters, with a maximum accuracy gap of 0.00 percentage points from their corresponding original H-OM task models. Experiments on vision and language benchmarks further demonstrate improved merged performance with fewer trainable adapter parameters than standard LoRA, supporting preallocated orthogonal subspaces as a practical basis for independent and editable model merging.

Discussion and Limitations. The main limitations of H-OM concern the scale and flexibility of subspace allocation. First, the number of orthogonal directions is limited by the layer width, restricting how many task subspaces H-OM can allocate. However, only a modest performance drop under this setting while reducing the subspace width from $c = 40$ to $c = 20$ doubles the maximum number of disjoint task subspaces. For larger task collections, sharing subspaces among related tasks could help accommodate more tasks within the available space. Second, the uniform allocation $c_i = \lfloor d_{\text{out}}/T \rfloor$ assigns the same subspace dimension to every task, which may not reflect their different learning requirements. Future work could adapt subspace dimensions to these requirements to make better use of the available directions.

AI USE STATEMENT

Generative AI was used to assist with manuscript writing and polishing, including sentence rephrasing, grammar checking, and improving clarity and readability. It was also used as an auxiliary tool to inspect mathematical derivations, formulas, and experimental descriptions for potential inconsistencies. The AI tools were not involved in the methodology, experimental design, implementation, or scientific conclusions. All research content was developed and verified by the authors.

ETHICS STATEMENT

This work uses only publicly available benchmark datasets and does not involve human-subject or animal experiments. No personally identifiable information was collected or used in this study.

REPRODUCIBILITY STATEMENT

Section 3 and Appendix A specify the H-OM parameterization, assumptions, initialization, and recovery identities. Appendix B provides the construction details of task subspaces, including Hadamard factorizations, sign and permutation conventions, and task column indices. The experimental sections and appendices report the training configurations, task configurations, evaluation protocols, hyperparameters, and merging coefficients required for numerical reproduction.

REFERENCES

Cheng, Runxi, Xiong, Feng, Wei, Yongxian, Zhu, Wanyun, and Yuan, Chun. Whoever Started the Interference Should End It: Guiding Data-Free Model Merging via Task Vectors. In *International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 10121–10143, 2025.

Gorbunov, Mikhail, Yudin, Nikolay, Soboleva, Vera, Alanov, Aibek, Naumov, Alexey, and Rakhuba, Maxim. Group and Shuffle: Efficient Structured Orthogonal Parametrization. In *Advances in Neural Information Processing Systems*, 2024.

Hu, Edward J., Shen, Yelong, Wallis, Phillip, Allen-Zhu, Zeyuan, Li, Yuanzhi, Wang, Shean, Wang, Lu, and Chen, Weizhu. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*, 2022.

Ilharco, Gabriel, Ribeiro, Marco Tulio, Wortsman, Mitchell, Gururangan, Suchin, Schmidt, Ludwig, Hajishirzi, Hannaneh, and Farhadi, Ali. Editing Models with Task Arithmetic. In *International Conference on Learning Representations*, 2023.

Jain, Raj, Chiu, Dah-Ming, and Hawe, William. A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer Systems. Technical Report DEC-TR-301, Digital Equipment Corporation, 1984. https://www.cs.wustl.edu/~jain/papers/ftp/fairness.pdf.

Lee, Chanhyuk, Choi, Jiho, Lee, Chanryeol, Kim, Donggyun, and Hong, Seunghoon. AdaRank: Adaptive Rank Pruning for Enhanced Model Merging. In *International Conference on Learning Representations*, 2026.

Li, Yayuan, Peng, Ze, Zhang, Jian, Guo, Jintao, Duan, Yue, and Shi, Yinghuan. When Shared Knowledge Hurts: Spectral Over-Accumulation in Model Merging. In *International Conference on Machine Learning*, 2026.

Panariello, Aniello, Marczak, Daniel, Magistri, Simone, Porrello, Angelo, Twardowski, Bartłomiej, Bagdanov, Andrew D., Calderara, Simone, and van de Weijer, Joost. Accurate and Efficient Low-Rank Model Merging in Core Space. In *Advances in Neural Information Processing Systems*, 2025.

Qiao, Fuli and Mahdavi, Mehrdad. Merge before Forget: A Single LoRA Continual Learning via Continual Merging. In *International Conference on Learning Representations*, 2026.

Qwen Team. Qwen2.5 Technical Report. *arXiv preprint arXiv:2412.15115*, 2024.

Radford, Alec, Kim, Jong Wook, Hallacy, Chris, Ramesh, Aditya, Goh, Gabriel, Agarwal, Sandhini, Sastry, Girish, Askell, Amanda, Mishkin, Pamela, Clark, Jack, Krueger, Gretchen, and Sutskever, Ilya. Learning Transferable Visual Models From Natural Language Supervision. In *International Conference on Machine Learning*, 2021.

Stoica, George, Ramesh, Pratik, Ecsedi, Boglarka, Choshen, Leshem, and Hoffman, Judy. Model merging with SVD to tie the KnOTS. In *International Conference on Learning Representations*, 2025.

Tang, Anke, Yang, Enneng, Shen, Li, Luo, Yong, Hu, Han, Zhang, Lefei, Du, Bo, and Tao, Dacheng. Merging on the Fly Without Retraining: A Sequential Approach to Scalable Continual Model Merging. In *Advances in Neural Information Processing Systems*, 2025.

Tam, Derek, Bansal, Mohit, and Raffel, Colin. Merging by Matching Models in Task Parameter Subspaces. *Transactions on Machine Learning Research*, 2024.

Touvron, Hugo, Martin, Louis, Stone, Kevin, Albert, Peter, Almahairi, Amjad, Babaei, Yasmine, Bashlykov, Nikolay, Batra, Soumya, Bhargava, Prajjwal, Bhosale, Shruti, et al. Llama 2: Open Foundation and Fine-Tuned Chat Models. *arXiv preprint arXiv:2307.09288*, 2023.

Wang, Xiao, Chen, Tianze, Ge, Qiming, Xia, Han, Bao, Rong, Zheng, Rui, Zhang, Qi, Gui, Tao, and Huang, Xuanjing. Orthogonal Subspace Learning for Language Model Continual Learning. In *Findings of the Association for Computational Linguistics: EMNLP*, 2023.

Wortsman, Mitchell, Ilharco, Gabriel, Gadre, Samir Yitzhak, Roelofs, Rebecca, Gontijo-Lopes, Raphael, Morcos, Ari S., Namkoong, Hongseok, Farhadi, Ali, Carmon, Yair, Kornblith, Simon, and Schmidt, Ludwig. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning*, 2022.

Yadav, Prateek, Tam, Derek, Choshen, Leshem, Raffel, Colin, and Bansal, Mohit. TIES-Merging: Resolving Interference When Merging Models. In *Advances in Neural Information Processing Systems*, 2023.

Yang, Enneng, Yang, Qun, Wang, Peng, Tang, Anke, Guo, Guibing, Cao, Xiaochun, and Shen, Li. MergOPT: A Merge-Aware Optimizer for Robust Model Merging. In *International Conference on Learning Representations*, 2026.

Yu, Le, Yu, Bowen, Yu, Haiyang, Huang, Fei, and Li, Yongbin. Language Models are Super Mario: Absorbing Abilities from Homologous Models as a Free Lunch. In *International Conference on Machine Learning*, 2024.

Zhai, Xiaohua, Puigcerver, Joan, Kolesnikov, Alexander, Ruyssen, Pierre, Riquelme, Carlos, Lucic, Mario, Djolonga, Josip, Pinto, Andre Susano, Neumann, Maxim, Dosovitskiy, Alexey, Beyer, Lucas, Bachem, Olivier, Tschannen, Michael, Michalski, Marcin, Bousquet, Olivier, and Gelly, Sylvain. A Large-scale Study of Representation Learning with the Visual Task Adaptation Benchmark. *arXiv preprint arXiv:1910.04867*, 2019.

Zhang, Haobo and Zhou, Jiayu. Unraveling LoRA Interference: Orthogonal Subspaces for Robust Model Merging. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, pp. 26459–26472, 2025.

A Supporting Derivations

A.1 Gradient Access and Leverage Scores

We provide the derivation underlying the gradient-access analysis in Sec. 3.3.2. For task i, let $\Delta W_i = Q_i \theta_i A_i$, where $Q_i^\top Q_i = I_{c_i}$ and Q_i is fixed. Denote the full weight gradient by $G_i = \nabla_{\Delta W_i} \mathcal{L}_i$ and the projector onto the assigned output subspace by $P_i = Q_i Q_i^\top$.

Factor gradients and projected energy. Since $d(\Delta W_i) = Q_i \, d\theta_i \, A_i + Q_i \theta_i \, dA_i$, the chain rule gives

$$\nabla_{\theta_i} \mathcal{L}_i = Q_i^\top G_i A_i^\top, \ \nabla_{A_i} \mathcal{L}_i = \theta_i^\top Q_i^\top G_i. \tag{15}$$

Both factor gradients depend on $Q_i^\top G_i = Q_i^\top P_i G_i$. Since P_i is an orthogonal projector,

$$\|Q_i^\top G_i\|_F^2 = \mathrm{tr}(G_i^\top P_i G_i) = \|P_i G_i\|_F^2. \tag{16}$$

For $G_i \neq 0$, the accessible fraction of full gradient energy is therefore

$$R_i(G_i) := \frac{\|P_i G_i\|_F^2}{\|G_i\|_F^2}. \tag{17}$$

This quantity measures subspace-level gradient access, not the fraction necessarily realized in an actual parameter update, which also depends on the trainable factors.

Coordinate-wise gradient access. Consider a nonzero gradient supported only on output coordinate k, namely $G_i^{(k)} = e_k g^\top$, where e_k is the k-th standard basis vector and $g \neq 0$. Its retained energy fraction is

$$\begin{aligned} R_i(G_i^{(k)}) &= \frac{\|Q_i^\top e_k g^\top\|_F^2}{\|e_k g^\top\|_F^2} \\ &= \|Q_i^\top e_k\|_2^2 = e_k^\top P_i e_k =: \ell_k(Q_i). \end{aligned} \tag{18}$$

Thus, the leverage score $\ell_k(Q_i)$ equals the retained energy fraction for a gradient supported on coordinate k, motivating its interpretation as coordinate exposure.

General gradients and cross-coordinate interactions. Let $g_{t,k}$ denote row k of G_i, and let $(P_i)_{kl}$ denote entry (k, l) of P_i. Expanding the projected energy gives

$$\begin{aligned} \|P_i G_i\|_F^2 = &\sum_{k=1}^{d_{\text{out}}} \ell_k(Q_i) \|g_{t,k}\|_2^2 \\ &+ \sum_{\substack{k,l=1 \\ k \neq l}}^{d_{\text{out}}} (P_i)_{kl} \langle g_{t,k}, g_{t,l} \rangle. \end{aligned} \tag{19}$$

If $\ell_k(Q_i) = c_i / d_{\text{out}}$ for all k, the diagonal contribution equals $(c_i / d_{\text{out}}) \|G_i\|_F^2$, independently of the gradient-energy distribution across output coordinates. However, the cross-coordinate term remains dependent on the projector and correlations between gradient rows. Equal leverage therefore does not guarantee identical or optimal full-gradient retention for arbitrary task gradients.

Finally, orthonormality implies

$$\sum_{k=1}^{d_{\text{out}}} \ell_k(Q_i) = \mathrm{tr}(P_i) = \mathrm{tr}(Q_i^\top Q_i) = c_i. \tag{20}$$

The total coordinate exposure is fixed by the subspace dimension. Since task-specific gradient requirements are unknown at allocation, uniform exposure provides a task-independent criterion that avoids structurally favoring particular output coordinates.

A.2 CEU Bounds and the Minimax Property

Let $Q \in \mathbb{R}^{d_{\text{out}} \times c}$ satisfy $Q^\top Q = I_c$, with $1 \le c \le d_{\text{out}}$. Write $P = QQ^\top$ and $\ell_k = \ell_k(Q) = P_{kk}$. Because P is an orthogonal projector, $0 \le \ell_k \le 1$ and $\sum_{k=1}^{d_{\text{out}}} \ell_k = \operatorname{tr}(P) = c$. The CEU defined in Eq. (9) is

$$\text{CEU}(Q) = \frac{c^2}{d_{\text{out}} \sum_{k=1}^{d_{\text{out}}} \ell_k^2}. \tag{21}$$

Bounds and equality conditions. By Cauchy–Schwarz, $\sum_k \ell_k^2 \ge c^2/d_{\text{out}}$, with equality if and only if all leverage scores equal c/d_{out}. Conversely, $0 \le \ell_k \le 1$ implies $\sum_k \ell_k^2 \le \sum_k \ell_k = c$. Substituting these bounds into Eq. (21) yields

$$\boxed{\frac{c}{d_{\text{out}}} \le \text{CEU}(Q) \le 1.} \tag{22}$$

The upper bound is attained exactly when $\ell_k = c/d_{\text{out}}$ for every k. The lower bound is attained by a coordinate-aligned c-dimensional subspace, whose leverage scores consist of c ones and $d_{\text{out}} - c$ zeros.

The relation between CEU and deviations from equal exposure follows from $\sum_k (\ell_k - c/d_{\text{out}}) = 0$, which gives

$$\text{CEU}(Q) = \left[1 + \frac{d_{\text{out}}}{c^2} \sum_{k=1}^{d_{\text{out}}} \left(\ell_k - \frac{c}{d_{\text{out}}}\right)^2\right]^{-1}. \tag{23}$$

Thus, at fixed c, CEU decreases with the aggregate squared deviation from equal leverage. An ordering by CEU alone does not necessarily imply the same ordering of minimum coordinate exposure for arbitrary intermediate configurations.

Maximin and minimax properties. Since the leverage scores have fixed mean c/d_{out},

$$\min_k \ell_k(Q) \le \frac{c}{d_{\text{out}}} \le \max_k \ell_k(Q). \tag{24}$$

An equal-leverage subspace attains both bounds simultaneously. Such subspaces are realized by the Hadamard construction in Sec. 3.3.3 for the dimensions used in our experiments. Consequently,

$$\max_{Q^\top Q = I_c} \min_k \ell_k(Q) = \frac{c}{d_{\text{out}}}, \tag{25}$$

$$\min_{Q^\top Q = I_c} \max_k \ell_k(Q) = \frac{c}{d_{\text{out}}}. \tag{26}$$

Equality in either objective requires $\ell_k = c/d_{\text{out}}$ for every k, because a set of leverage scores with fixed mean cannot have its minimum or maximum equal to that mean unless all scores are equal.

Scope of the guarantee. These results establish optimality of equal leverage at the coordinate level. They do not establish optimal retention of arbitrary full task gradients: as shown in Eq. (19), general-gradient retention also depends on cross-coordinate interactions. The projected-gradient retention in Fig. 2(b), together with the merged accuracy and performance variability in Appendix B.2, provides empirical evidence for the benefits of higher CEU within the evaluated constructions.

A.3 Proof of Component Recovery

Let $M = W_{\text{merge}} - W_0 = \sum_{i \in \mathcal{T}} \lambda_i Q_i C_i$, where $C_i = \theta_i A_i$. Fix a task $t \in \mathcal{T}$. Since $Q_t^\top Q_t = I_{c_t}$ and $Q_t^\top Q_i = 0$ for $i \ne t$,

$$Q_t^\top M = \sum_{i \in \mathcal{T}} \lambda_i Q_t^\top Q_i C_i = \lambda_t C_t.$$

Consequently,

$$P_t M = Q_t Q_t^\top M = \lambda_t Q_t C_t = \lambda_t \Delta W_t.$$

Thus, $M - P_t M$ removes task t while leaving the remaining weighted components unchanged. If $\lambda_t \ne 0$, then $\Delta W_t = P_t M / \lambda_t$.

This proves Proposition 1.

Table 4: Initialization-scale ablation on VTAB-1k. The H-OM initialization uses $c_i/d_{\text{out}} = 40/768 \approx 0.05$. Results are mean $\pm$ standard deviation over the shared seeds.

Initialization scale	Avg. Acc.
Fixed 0.01	55.23 ± 0.28
c_i/d_{out} (≈ 0.05)	$\mathbf{55.77 \pm 0.28}$
Fixed 0.1	55.08 ± 0.24
Fixed 0.5	54.28 ± 0.19

A.4 Initialization Scale and Factor Gradients

The LoRA scaling factor $s = \alpha_{\text{LoRA}}/r$ and the initialization prefactor c_i/d_{out} play different roles. H-OM sets $A_i^{(0)} = 0$ and initializes

$$\theta_i^{(0)} = \frac{c_i}{d_{\text{out}}} \begin{bmatrix} I_r \\ 0_{(c_i - r)\times r} \end{bmatrix}.$$

Let $B_{t,\text{eff}} := Q_i \theta_i$ denote the resulting output-side factor. Since $Q_i^\top Q_i = I_{c_i}$,

$$\begin{aligned} B_{t,\text{eff}}^{(0)\top} B_{t,\text{eff}}^{(0)} &= \theta_i^{(0)\top} Q_i^\top Q_i \theta_i^{(0)} \\ &= \theta_i^{(0)\top} \theta_i^{(0)} = \left(\frac{c_i}{d_{\text{out}}}\right)^2 I_r. \end{aligned}$$

Hence, c_i/d_{out} directly determines the initial singular values of $Q_i\theta_i$, while $A_i^{(0)} = 0$ guarantees $\Delta W_i^{(0)} = 0$.

Writing $G_i = \nabla_{\Delta W_i} L_i$, the gradients of the two trainable factors are

$$\begin{aligned} \nabla_{\theta_i} L_i &= Q_i^\top G_i A_i^\top, \\ \nabla_{A_i} L_i &= \theta_i^\top Q_i^\top G_i. \end{aligned}$$

At initialization, $\nabla_{\theta_i} L_i = 0$ because $A_i^{(0)} = 0$, whereas $\nabla_{A_i} L_i$ depends on the scale of $\theta_i^{(0)}$. The initialization prefactor can therefore affect early optimization even though every configuration begins from the same pretrained model.

Among the four evaluated initialization scales, c_i/d_{out} gives the highest reported mean accuracy.

A.5 Parameter Accounting

For one task and one adapted layer, standard LoRA trains $r(d_{\text{out}} + d_{\text{in}})$ parameters, whereas H-OM trains $r(c_i + d_{\text{in}})$ parameters because Q_i is fixed.

Across tasks and adapted layers $\mathcal{L}$, let r_ℓ, $d_{\text{out},\ell}$, and $d_{\text{in},\ell}$ denote the rank, output dimension, and input dimension of layer ℓ, respectively, and let $c_{t,\ell}$ denote the assigned subspace dimension. The total numbers of trainable adapter parameters are

$$\begin{aligned} N_{H-OM} &= \sum_{\ell \in \mathcal{L}} r_\ell \left(T d_{\text{in},\ell} + \sum_{t \in \mathcal{T}} c_{t,\ell} \right), \\ N_{\text{LoRA}} &= \sum_{\ell \in \mathcal{L}} T r_\ell \left(d_{\text{out},\ell} + d_{\text{in},\ell} \right). \end{aligned}$$

These counts exclude the fixed Hadamard bases and the metadata required to regenerate the task assignments.

Algorithm 1 Hadamard Orthogonal Merging (H-OM)

Require: Pretrained weight $W_0 \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$; task datasets $\{\mathcal{D}_t\}_{t=1}^{T}$
Ensure: Merged weight W_{merge}
1: Generate a randomized Hadamard basis $U \in \mathbb{R}^{d_{\text{out}} \times d_{\text{out}}}$
2: **for** $t = 1, \ldots, T$ **independently do**
3: Select the preassigned columns $Q_t = U[:, \mathcal{I}_t] \in \mathbb{R}^{d_{\text{out}} \times c}$
4: Initialize $\theta_t = \dfrac{c}{d_{\text{out}}} \begin{bmatrix} I_r \\ 0_{(c-r)\times r} \end{bmatrix} \in \mathbb{R}^{c \times r}, \quad A_t = 0_{r \times d_{\text{in}}}$
5: Train θ_t and A_t on $\mathcal{D}_t$ with W_0 and Q_t frozen
6: Compute $\Delta W_t = Q_t \theta_t A_t$
7: **end for**
8: Set $\lambda = T^{-p}$
9: Merge $\Delta W_{\text{merge}} = \lambda \sum_{t=1}^{T} \Delta W_t$
10: **return** $W_{\text{merge}} = W_0 + \Delta W_{\text{merge}}$

B Construction and Experimental Protocol

B.1 Hadamard Matrices for the Evaluated Dimensions

For the output dimensions used in the experiments, the normalized Hadamard bases are constructed from the following factorizations:

$$\Phi_{768} = \overline{H}_{12} \otimes \overline{H}_{64}, \quad \Phi_{4096} = \overline{H}_{4096},$$
$$\Phi_{1536} = \overline{H}_{12} \otimes \overline{H}_{128}, \quad \Phi_{256} = \overline{H}_{256},$$

where $\overline{H}_n$ denotes a normalized $n \times n$ real Hadamard matrix and $\otimes$ denotes the Kronecker product.

The first two dimensions correspond to the main ViT-B/32 and LLaMA-2-7B-Chat experiments, while the latter two correspond to the Qwen2.5-1.5B-Instruct query and value projections. All evaluated output dimensions therefore admit the exact Hadamard constructions used in the experiments.

The seed ξ, the Hadamard factors, the sign and permutation conventions, and the assigned column indices determine each task basis. The basis can therefore be regenerated without explicitly storing the complete $d_{\text{out}} \times d_{\text{out}}$ matrix.

B.2 Controlled Construction of Different CEU Levels

We conduct a controlled ablation to examine the relation between leverage-score uniformity and merging performance while keeping subspace capacity and merge interference orthogonality unchanged. The experiment uses the full VTAB-1k setting with $T = 19$ tasks, output dimension $d_{\text{out}} = 768$, and a fixed task-subspace dimension $c = 40$. The LoRA rank, initialization, optimizer, training configuration, task allocation, and merge configuration are kept fixed across all CEU levels. Only the orientation of the mutually orthogonal task subspaces is varied.

Orthogonal dictionary interpolation. Let $U_{\text{H}} \in \mathbb{R}^{d_{\text{out}} \times d_{\text{out}}}$ denote the normalized Hadamard dictionary used by the full H-OM configuration, including the fixed sign and permutation transformations in Eq. (10). To obtain a controlled family of orientations between a coordinate-aligned basis and the Hadamard basis, we form

$$M_\alpha = (1 - \alpha) I_{d_{\text{out}}} + \alpha U_{\text{H}}, \qquad \alpha \in [0, 1]. \tag{27}$$

For each candidate α, we compute a full QR factorization

$$M_\alpha = U_\alpha R_\alpha, \qquad U_\alpha^\top U_\alpha = I_{d_{\text{out}}}, \tag{28}$$

with the signs of the QR factors fixed by requiring the diagonal entries of R_α to be nonnegative. Candidate values for which M_α is numerically rank deficient are excluded. The resulting U_α is therefore a full orthogonal dictionary for every evaluated configuration. At $\alpha = 0$, the construction gives the coordinate basis; at $\alpha = 1$, it recovers the Hadamard dictionary up to column-wise sign changes introduced by the QR convention. Such sign changes do not alter the corresponding subspaces or their leverage scores.

Fixed task-subspace allocation. The same disjoint column-index sets used by the main H-OM configuration are retained for every α. Let $\mathcal{I}_i \subset \{1, \ldots, d_{\text{out}}\}$ denote the column set assigned to task t, with

$$|\mathcal{I}_i| = c = 40, \qquad \mathcal{I}_i \cap \mathcal{I}_j = \varnothing \quad \text{for } i \neq j. \tag{29}$$

The task basis at interpolation level α is

$$Q_{i,\alpha} = (U_\alpha)_{\mathcal{I}_i}. \tag{30}$$

Because all task bases are formed from disjoint columns of the same orthogonal dictionary,

$$Q_{i,\alpha}^\top Q_{i,\alpha} = I_c, \qquad Q_{i,\alpha}^\top Q_{j,\alpha} = 0 \quad (i \neq j) \tag{31}$$

for every evaluated α. Thus, varying α leaves both subspace dimension and merge interference orthogonality unchanged and modifies only the orientation of the assigned subspaces relative to the output coordinates.

Mean CEU across task subspaces. For each task basis $Q_{i,\alpha}$, We compute coordinate leverage scores $\ell_k(Q_i) = e_k^\top Q_i Q_i^\top e_k$ and evaluate CEU using Eq. (9). A non-Hadamard orthogonal dictionary does not generally give identical leverage distributions to all disjoint column blocks. We therefore summarize each complete 19-task allocation by the mean CEU across its task subspaces:

$$\overline{\text{CEU}}(\alpha) = \frac{1}{T} \sum_{t=1}^{T} \text{CEU}(Q_{i,\alpha}), \qquad T = 19. \tag{32}$$

This definition prevents the reported value from depending on one arbitrarily selected task block and measures the average leverage-score uniformity of the complete allocation.

We sweep α and select orthogonal realizations whose measured $\overline{\text{CEU}}(\alpha)$ values are closest to the target levels 0.25, 0.50, 0.75, and 1.00. The interpolation parameter α is used only to generate controlled orthogonal dictionaries and is not treated as an evaluation metric. The horizontal axis and table entries report the measured mean task-subspace CEU, rounded to two decimal places. The $\overline{\text{CEU}} = 1$ endpoint corresponds to the equal-leverage Hadamard construction used by the full H-OM configuration.

Controlled training protocol. For every selected dictionary, task adapters are trained with the same H-OM parameterization $\Delta W_i = Q_{i,\alpha} \theta_i A_i$. Across CEU levels, we keep $d_{\text{out}} = 768$, $c = 40$, LoRA rank, initialization of θ_i and A_i, training data, optimizer and optimization settings, task-to-column allocation, and merge coefficients unchanged. The only controlled change is the orientation of U_α. Each CEU level is evaluated over six repeated runs under the same protocol.

Table 5: Controlled CEU ablation on VTAB-1k. CEU denotes the mean across the 19 assigned task subspaces as defined in Eq. (32). All configurations use $d_{\text{out}} = 768$, $c = 40$, and the same rank, initialization, optimization settings, task allocation, and merge configuration; only the orthogonal subspace orientation changes. Statistics are computed over six runs.

Mean CEU	Mean Acc. (%)	Sample SD	Worst (%)	Best (%)	Range (pp)
0.25	54.16	0.93	52.88	55.33	2.45
0.50	54.58	0.68	53.71	55.44	1.73
0.75	54.92	0.49	54.52	55.77	1.25
1.00	**55.77**	**0.26**	55.47	56.10	**0.63**

The mean merged accuracy increases monotonically from 54.16 at mean CEU 0.25 to 55.77 at mean CEU 1.00, while the sample standard deviation decreases from 0.93 to 0.26. This controlled trend supports the use of uniform leverage as the task-independent subspace-orientation criterion within the evaluated family of mutually orthogonal subspaces.

For six merged-accuracy observations $a_1, \ldots, a_6$ at a given CEU level, the reported sample standard deviation is

$$s_a = \sqrt{\frac{1}{5} \sum_{i=1}^{6} (a_i - \bar{a})^2}, \qquad \bar{a} = \frac{1}{6} \sum_{i=1}^{6} a_i. \tag{33}$$

The "Worst" and "Best" columns report $\min_i a_i$ and $\max_i a_i$, respectively, and $\text{Range} = \max_i a_i - \min_i a_i$.

B.3 Subspace Dimensions, Initialization, and Merge Coefficients

For the main experiments, H-OM uses the allocation rule $c = \lfloor d_{\text{out}}/T \rfloor$ within each adapted projection. This gives $c = 40$ for VTAB-1k with $d_{\text{out}} = 768$ and $T = 19$, and $c = 273$ for the LLaMA-2-7B-Chat projections with $d_{\text{out}} = 4096$ and $T = 15$.

For Qwen2.5-1.5B-Instruct, the rule is applied independently to the two adapted projection widths, giving $c_q = 102$ for $d_q = 1536$ and $c_v = 17$ for $d_v = 256$ with $T = 15$. Any remaining Hadamard columns are left unassigned.

The initialization prefactor is c_i/d_{out} for the corresponding projection. The default merge coefficient follows Sec. 3.3.4:

$$\lambda_i = T^{-0.3}.$$

This gives $\lambda = 19^{-0.3} \approx 0.4134$ for VTAB-1k and $\lambda = 15^{-0.3} \approx 0.4438$ for the two language-model evaluations.

B.4 Training Configuration

Table 6: Recorded training configurations for the main experiments.

Item	VTAB-1k (ViT-B/32)	LLaMA-2-7B-Chat NLP-15	Qwen2.5-1.5B NLP-15
Optimizer	AdamW	AdamW	AdamW
Learning rate	1×10^{-4}	1×10^{-4}	1×10^{-4}
Batch size	32	1/device; grad. accum. 4	1/device; grad. accum. 4
Steps / epochs	20 epochs	1 epoch	1 epoch
Rank r	8	8	8
Subspace dimension	$c = 40$	$c = 273$	$c_q = 102,\ c_v = 17$
Initialization prefactor	c/d_{out}	c/d_{out}	c/d_{out}
Training seed	$\{0, 1, 42\}$	42 for O1–O6	42 for O1
Merge coefficient	$19^{-0.3} \approx 0.4134$	$15^{-0.3} \approx 0.4438$	$15^{-0.3} \approx 0.4438$

All experiments use up to 1,000 training examples per task, with all available samples used when fewer examples exist. Language experiments use a maximum sequence length of 1024. Experiments are run on an NVIDIA RTX 3090, with DeepSpeed ZeRO enabled for the 7B language model.

B.5 Evaluation Metrics

Task accuracy is reported as a percentage. For VTAB-1k, the overall average is the arithmetic mean across the evaluated tasks. Differences between two accuracies are reported in percentage points (pp).

For NLP-15, each O1–O6 entry is the arithmetic mean across the 15 tasks. The Avg. column is the mean across the six configurations, and the Std. column is the population standard deviation across O1–O6.

C Effect of the Number of Merged Tasks

We evaluate nested VTAB task subsets containing $T \in \{10, 15, 19\}$ tasks to examine the effect of increasing the number of merged tasks. The subsets satisfy $\mathcal{T}_{10} \subset \mathcal{T}_{15} \subset \mathcal{T}_{19}$.

To isolate the effect of task count from that of per-task subspace dimension, we fix the H-OM subspace dimension to $c = 40$ for all three settings. This matches the subspace dimension of the full 19-task configuration and keeps the dimension assigned to each task unchanged across the comparison. Since $Tc \leq d_{\text{out}} = 768$ for all three settings, the assigned task subspaces remain mutually orthogonal.

This controlled experiment differs from the default rule $c = \lfloor d_{\text{out}}/T \rfloor$ used in the main experiments; the fixed value $c = 40$ is used here specifically to vary the number of merged tasks without simultaneously changing the H-OM subspace dimension.

Table 7: Nested VTAB task subsets used in the task-count experiment. Long task names are shortened for readability.

T	Group	N	Tasks
10	Natural	4	caltech101, cifar100, dtd, flowers
10	Specialized	2	diabetic retinopathy, eurosat
10	Structured	4	CLEVR-Distance, CLEVR-Count, dmlab, DSprites-Orientation
15	Natural	5	caltech101, cifar100, dtd, flowers, pets
15	Specialized	3	diabetic retinopathy, eurosat, pcam
15	Structured	7	CLEVR-Distance, CLEVR-Count, dmlab, DSprites-Orientation, DSprites-X-Position, KITTI-Distance, SmallNORB-Azimuth
19	Natural	7	caltech101, cifar100, dtd, flowers, pets, sun397, svhn
19	Specialized	4	diabetic retinopathy, eurosat, pcam, resisc45
19	Structured	8	CLEVR-Distance, CLEVR-Count, dmlab, DSprites-Orientation, DSprites-X-Position, KITTI-Distance, SmallNORB-Azimuth, SmallNORB-Elevation

Table 8: VTAB accuracy as the number of merged tasks increases over the nested task subsets in Table 7. H-OM uses the same subspace dimension $c = 40$ for all three settings. Bold entries indicate the best result among the merged models within each task-count setting and column.

T	Method	Natural	Specialized	Structured	Avg.
10	Single-task LoRA	82.78	83.60	55.60	72.07
	Task Arithmetic	79.53	79.10	28.15	58.89
	TIES	79.48	79.50	28.12	58.94
	DARE	78.72	79.65	28.75	58.92
	DARE-TIES	78.15	**80.10**	28.85	58.82
	MergOPT	79.60	78.60	28.12	58.81
	SVC	79.75	78.00	24.25	57.20
	WuDi	78.50	78.50	28.50	58.50
	H-OM	**83.84**	77.73	**33.69**	**62.56**
15	Single-task LoRA	84.10	83.07	51.52	68.69
	Task Arithmetic	80.96	70.80	26.43	53.48
	TIES	80.62	71.43	26.49	53.52
	DARE	80.26	71.57	26.76	53.56
	DARE-TIES	79.80	71.93	26.74	53.47
	MergOPT	80.96	70.60	26.37	53.41
	SVC	80.80	69.67	26.14	53.07
	WuDi	79.80	69.67	26.00	52.67
	H-OM	**84.95**	**74.13**	**29.95**	**57.12**
19	Single-task LoRA	80.90	83.97	49.58	68.36
	Task Arithmetic	73.94	72.52	23.23	52.29
	TIES	71.64	69.35	22.35	50.41
	DARE	73.96	72.51	23.37	52.36
	DARE-TIES	71.64	69.28	22.23	50.34
	MergOPT	74.00	72.56	23.35	52.37
	SVC	74.43	72.15	24.56	52.95
	WuDi	74.13	72.60	24.35	52.85
	H-OM	**77.09**	**74.10**	**27.96**	**55.77**

With the H-OM subspace dimension fixed at $c = 40$, H-OM achieves the highest overall average accuracy among the merged models for all three task-set sizes. Its margin over the strongest comparison is 3.62 percentage points for $T = 10$, 3.56 points for $T = 15$, and 2.82 points for $T = 19$. Thus, the advantage of the fixed orthogonal subspaces is maintained across the three evaluated task counts under the same H-OM subspace dimension.

C.1 Sensitivity to the Merge-Scaling Exponent

We examine the sensitivity of H-OM to the exponent p in the shared merge coefficient $\lambda_i = T^{-p}$. Due to the orthogonal task decomposition of H-OM, the merged update magnitude is explicitly controlled:

$$\|\Delta W_{\text{merge}}\|_F^2 = \sum_{t=1}^{T} \lambda_i^2 \|\Delta W_i\|_F^2 , \tag{34}$$

Table 9: Sensitivity to the merge-scaling exponent p on the nested VTAB task subsets. The merge coefficient is $\lambda_i = T^{-p}$ and H-OM uses $c = 40$ for all three task-set sizes. Entries are average merged accuracies (%).

p	$T = 10$	$T = 15$	$T = 19$
0.0	59.02	53.95	50.60
0.1	60.63	55.33	53.55
0.2	61.17	55.80	54.55
0.3	**62.56**	**57.12**	**55.77**
0.4	60.70	54.73	54.17
0.5	59.32	53.52	52.86

where cross-task interaction terms vanish under orthogonal subspaces. Therefore, the scaling coefficient directly controls the update magnitude without being affected by task interference.

For task updates with comparable magnitudes, $\lambda = 1/\sqrt{T}$ preserves the overall update scale as the number of merged tasks increases, while $\lambda = 1/T$ corresponds to more conservative linear averaging. We therefore parameterize the shared coefficient as $\lambda_i = T^{-p}$ and evaluate $p \in [0, 0.5]$, where $p = 0$ denotes no scaling and larger p applies stronger attenuation.

Using the same nested VTAB task subsets with $T \in \{10, 15, 19\}$ and $c = 40$, we vary $p \in \{0, 0.1, 0.2, 0.3, 0.4, 0.5\}$.

Table 9 shows a consistent intermediate-scaling regime across all three task-set sizes. Increasing p from 0 initially improves merged accuracy, while stronger attenuation at $p = 0.4$ and $p = 0.5$ reduces it again. Among the evaluated values, $p = 0.3$ gives the highest average accuracy for $T = 10$, $T = 15$, and $T = 19$, yielding 62.56, 57.12, and 55.77, respectively. This supports the use of $p = 0.3$ as the default merge-scaling exponent in the main experiments without implying that it is globally optimal.

D Additional Evaluation on Qwen2.5-1.5B-Instruct

The supplemental Qwen2.5-1.5B-Instruct experiment (Qwen Team, 2024) evaluates H-OM with grouped-query attention and asymmetric adapted projection dimensions, $q_{proj} \in \mathbb{R}^{1536\times1536}$ and $v_{proj} \in \mathbb{R}^{256\times1536}$. The same allocation rule is applied separately to the two output dimensions, giving $c_q = 102$ for q_{proj} and $c_v = 17$ for v_{proj}. The comparison uses the O1 configuration and training seed 42.

Table 10: Qwen2.5-1.5B-Instruct evaluation on NLP-15. Accuracy is averaged over all 15 tasks. The merged methods use the O1 task configuration with training seed 42.

Method	Protocol	Avg. Acc. ↑
Single-task LoRA	Per-task	80.86
Task Arithmetic	O1	66.78
DARE-TIES	O1	67.80
SLAO	O1	73.22
H-OM	O1	**76.45**

E NLP-15 Task Groups and Order Permutations

The NLP-15 benchmark contains AG News, Amazon, Yelp, DBpedia, Yahoo, MNLI, QQP, RTE, SST-2, WiC, CB, COPA, MultiRC, BoolQ, and IMDB. For the order-configuration experiment, these tasks are grouped into Classification (C), Question Answering (Q), and Semantic Reasoning (R), as shown in Table 11. O1–O6 enumerate all six permutations of these three groups.

Table 11: Task groups and order permutations used in the NLP-15 configuration study.

Task groups

Group	Tasks
Classification (C)	ag_news, amazon, yelp, dbpedia, sst2, imdb
Question Answering (Q)	yahoo, boolq
Semantic Reasoning (R)	mnli, qqp, rte, multirc, wic, cb, copa

Order permutations

Order	Group order
O1	C → Q → R
O2	C → R → Q
O3	Q → C → R
O4	Q → R → C
O5	R → C → Q
O6	R → Q → C

F Task Component Editing and Recovery Diagnostics

This section evaluates the post-merge editing capability enabled by the recovery identity in Sec. 3.3.5. We first describe the editing operations and then evaluate recovery and removal under an FP32 serialization-and-reload workflow in which the original task adapters are unavailable to the deployment process. All parameter discrepancies are measured using relative Frobenius errors.

Task-component editing. The following operations are optional and take place after the initial linear merge. Write $M_T = W_{\text{merge}}^{(T)} - W_0 = \lambda_T \sum_{i=1}^{T} \Delta W_i$, where $\lambda_T = T^{-p}$. To remove task j after its adapter is discarded, we need $W_{\text{merge}}^{(T)}$, W_0, the Hadamard-construction metadata, and the task's assigned column indices $\mathcal{I}_j$. With the fixed construction of Eq. (10), these regenerate U and $Q_j = U[:, \mathcal{I}_j]$. The task projector is $P_j = Q_j Q_j^\top$, and its action can be computed without explicitly storing P_j:

$$P_j M_T = Q_j (Q_j^\top M_T) = \lambda_T \Delta W_j.$$

Removing task j while keeping the remaining coefficients unchanged gives

$$W_{-j}^{\text{fixed}} = W_0 + M_T - P_j M_T. \tag{35}$$

To subsequently restore the removed component, the extracted component $C_j := P_j M_T$ is retained at removal time, giving

$$W_{\text{restore}} = W_0 + (M_T - C_j) + C_j.$$

The component C_j cannot in general be recovered from the residual $M_T - C_j$ alone. If the shared coefficient is instead recomputed for $T-1$ tasks, the resulting model is

$$W_{\text{merge}}^{(T-1;-j)} = W_0 + \frac{\lambda_{T-1}}{\lambda_T}(M_T - P_j M_T). \tag{36}$$

If task $T+1$ is assigned an unused subspace orthogonal to the existing ones, it can be added by

$$W_{\text{merge}}^{(T+1)} = W_0 + \frac{\lambda_{T+1}}{\lambda_T} M_T + \lambda_{T+1} \Delta W_{T+1}. \tag{37}$$

Adding a task additionally requires an independently trained update in unused orthogonal columns. These editing operations do not require the earlier task adapters or explicitly stored task-basis matrices.

F.1 Deployment-Faithful 19-Task Recovery and Removal

We evaluate recovery and removal using the full 19-task VTAB-1k configuration with $\lambda_{19} = 19^{-0.3}$ under a deployment-style workflow. During export, the trained task updates are linearly merged into W_{merge}, and the complete merged backbone is serialized in FP32. We additionally retain only the deterministic metadata required to reproduce the Hadamard construction, including the construction seed, permutation and sign conventions, and the task column assignments. The complete dictionary and individual task-basis matrices are not required by the subsequent deployment-stage recovery.

Deployment is executed in a fresh process. This process loads the pretrained model W_0, the serialized merged model $\widehat{W}_{\text{merge}}$, and the basis-generation metadata, but does not load the original task adapters or stored task updates. The merged update available after serialization is

$$\widehat{M} = \widehat{W}_{\text{merge}} - W_0. \tag{38}$$

Table 12: Deployment-faithful recovery and removal on all 19 VTAB-1k tasks after FP32 serialization and reload. Parameter errors are relative Frobenius errors aggregated over the adapted matrices. Original task adapters are used only for post-hoc reference comparisons and are not available to the deployment-stage recovery procedure.

Diagnostic	Statistic	Result
Basis regeneration	Max. abs. difference	0.00
Merged checkpoint reload	Relative error	5.81×10^{-7}
Task-component recovery	Mean / max. relative error	3.00×10^{-4} / 4.63×10^{-4}
Recovered task accuracy	Max. gap over 19 tasks	0.00 pp
Component deletion	Mean / max. relative error	6.72×10^{-5} / 7.74×10^{-5}
Delete–restore	Max. relative error	1.18×10^{-8}

For each task i, its basis Q_i is regenerated from the stored metadata, and its unscaled task update is recovered as

$$\widehat{\Delta W}_i = \lambda_{19}^{-1} Q_i Q_i^\top \widehat{M}. \tag{39}$$

Deletion is performed directly on the reloaded merged update:

$$\widehat{M}_{-i} = \widehat{M} - Q_i Q_i^\top \widehat{M}. \tag{40}$$

For restoration, the extracted weighted component $Q_i Q_i^\top \widehat{M}$ is added back to $\widehat{M}_{-i}$.

The original task adapters are accessed only after deployment-stage recovery and evaluation have completed, and are used solely as reference oracles for computing parameter errors. They do not participate in the tested recovery or deletion procedure. Task-specific classifier heads are used only to evaluate accuracy and likewise do not enter component reconstruction.

Before deployment, we verify that the exported metadata regenerates the training-time basis with zero maximum absolute difference. The deployment process itself does not load the stored training-time basis. Serializing and reloading the complete merged model introduces a relative round-trip error of 5.81×10^{-7}. After this FP32 round trip, the mean relative error of the recovered task updates is 3.00×10^{-4}, with a maximum of 4.63×10^{-4} across the 19 tasks. Despite these finite-precision parameter differences, every recovered task model matches the accuracy of its corresponding original H-OM task model, giving a maximum accuracy gap of 0.00 percentage points over all 19 tasks.

Projection-based deletion closely reproduces an oracle re-merge that excludes the designated task while keeping the original merge coefficient λ_{19} unchanged. The mean and maximum relative deletion errors are 6.72×10^{-5} and 7.74×10^{-5}, respectively. Adding the extracted component back restores the reloaded merged update with a maximum relative error of 1.18×10^{-8}.

We additionally evaluate all 19 deletion models on all 19 VTAB-1k tasks. Removing a task's designated component decreases that task's accuracy by 4.80 percentage points on average. Across the remaining 342 retained-task evaluations, the mean signed accuracy change relative to the original merged model is $+0.035$ percentage points and the mean absolute change is 0.395 percentage points; the largest observed absolute change is 6.50 percentage points. Thus, orthogonal projection provides parameter-component separability, but does not imply complete prediction-level independence between tasks.

These results validate the practical realization of the exact recovery identity in Proposition 1 under a standard serialization-and-reload workflow. Although FP32 serialization introduces small numerical discrepancies, individual task updates can be reconstructed or removed from a saved merged model using only the pretrained base model and reproducible basis metadata, without retaining the original task adapters.

F.2 Task-Component Reweighting

We further examine whether a recovered task component can be used to modulate target-task performance. For each edited task i, we replace its original merge coefficient λ_i with $\lambda_i' = \alpha \lambda_i$, where $\alpha \in \{0, 0.5, 1, 2\}$, while keeping all other task coefficients unchanged.

Target-task accuracy increases monotonically over the tested coefficients for all five edited tasks, although the response magnitude differs across tasks. These results demonstrate behavioral modula-

Table 13: Target-task accuracy (%) after reweighting the extracted component in the five-task merge. The coefficient of the edited task is set to $\lambda_i' = \alpha\lambda_i$; all other coefficients remain unchanged.

Edited task	$\alpha = 0$	$\alpha = 0.5$	$\alpha = 1$	$\alpha = 2$
CIFAR100	54.2	61.8	66.8	67.2
EuroSAT	64.6	80.8	91.5	94.2
Diabetic Retinopathy	73.1	73.3	73.8	74.1
DMLab	25.9	32.8	39.5	44.2
SmallNORB Elevation	19.2	21.1	27.0	31.0

Table 14: Matched orthogonal-dictionary comparison on VTAB-1k. Merged accuracy is averaged over six runs; Sample SD is the sample standard deviation of the six run-level merged accuracies. All methods use $d_{\text{out}} = 768$ and $c = 40$ under matched training and merging settings. Reference CEU values are rounded to two decimal places.

Dictionary	Reference CEU ↑	Merged Acc. (%) ↑	Sample SD
Coordinate slicing	0.05	51.81	1.85
Seeded Gaussian QR	0.95	55.33	0.31
Randomized Hadamard (H-OM)	1	**55.77**	0.26

tion through component reweighting in the evaluated setting. They do not imply that the predictive performance of other tasks remains unchanged.

F.3 Task-Address Recycling

We evaluate reuse of a released task address in a five-task configuration with subspace width $c = 96$. The address previously assigned to CIFAR100 is reassigned to a newly trained DTD adapter, which is merged with four retained task updates without retraining those adapters. DTD achieves 72.2% accuracy in the resulting merge, compared with 73.0% in single-task evaluation. The recycled component remains projection-addressable: removing it yields a relative parameter error of 1.48×10^{-4} against a reference merge of the four retained tasks, with a maximum displayed accuracy gap of 0.00 pp.

G Additional Ablations

G.1 Effect of the Orthogonal Dictionary

Effect of the orthogonal dictionary. We compare three orthogonal dictionary constructions under a matched VTAB-1k diagnostic. As shown in Table 14, H-OM achieves the highest mean merged accuracy, exceeding seeded Gaussian QR by 0.44 percentage points and coordinate slicing by 3.96 percentage points.

G.2 Effect of Training θ_i

Freezing θ_i while keeping the assigned basis, rank, data, optimizer, and training seeds fixed isolates the effect of the trainable output-side factor.

G.3 Single-task Adaptation Reference

Although H-OM is designed for multi-task merging rather than isolated task optimization, we report single-task adaptation results as a reference. The fixed orthogonal subspaces constrain the adaptation space compared with unconstrained LoRA, which may reduce single-task flexibility while improving merge compatibility.

The results show the expected trade-off introduced by fixed task subspaces: compared with unconstrained LoRA, H-OM sacrifices some single-task adaptation flexibility in exchange for a structured parameter space that enables interference-controlled merging and task-component editing.

Table 15: Matched VTAB-1k control over seeds $\{0, 1, 42\}$. The two configurations differ only in whether θ_i is trainable.

Variant	Q_i	θ_i	Single Avg. ↑
H-OM with fixed θ_i	Fixed	Fixed	57.12
H-OM	Fixed	Trainable	62.98

Table 16: Single-task adaptation results on VTAB-1k. Results are averaged over the same training seeds $\{0, 1, 42\}$ used in the main experiments.

Method	Single-task Avg. Acc. (%) ↑
LoRA	68.36 ± 0.21
H-OM	62.98 ± 0.24

Table 17: Performance–capacity trade-off on VTAB-1k. Both configurations merge the same 19 tasks. Used and Free denote the numbers of assigned and unassigned Hadamard columns. Accuracy is averaged over training seeds $\{0, 1, 42\}$.

Width	Used	Free	Natural	Specialized	Structured	Avg.
$c = 40$	760	8	77.09	74.10	27.96	55.77
$c = 20$	380	388	75.86	72.98	26.74	54.57

Table 18: Trainable adapter parameters under the fixed task-subspace widths used in our experiments. Both methods use rank-8 adapters on the query and value projections. H-OM uses $c = 40$ for VTAB-1k, $c = 273$ for LLaMA-2-7B-Chat, and $c_q = 102$, $c_v = 17$ for Qwen2.5-1.5B-Instruct. Unassigned Hadamard columns are not included.

Backbone	LoRA	H-OM	H-OM / LoRA
ViT-B/32	5,603,328	2,947,584	52.60%
LLaMA-2-7B-Chat	62,914,560	33,553,920	53.33%
Qwen2.5-1.5B-Instruct	16,343,040	10,721,760	65.60%

G.4 Reserved Capacity for Future Task Admission

To examine the performance–capacity trade-off, we compare the default VTAB-1k allocation ($c = 40$) with a narrower allocation ($c = 20$), keeping the other experimental settings unchanged and following the default c/d_{out} initialization rule. Reducing the task-subspace width decreases the mean merged accuracy from 55.77% to 54.57% (a drop of 1.20 percentage points), illustrating the adaptation-capacity cost of narrower task subspaces in this setting. Meanwhile, the number of unassigned Hadamard columns increases from 8 to 388, providing sufficient orthogonal capacity for up to 19 additional tasks of width 20 without modifying the existing task subspaces.

G.5 Rank and Parameter-Budget Control

Table 19 compares rank-4 and rank-8 configurations. The LoRA-r4 and H-OM-r8 configurations also provide an approximately parameter-matched comparison.

Table 19: Rank- and parameter-controlled comparison on VTAB-1k. All configurations adapt query/value projections.

Method	Rank	Trainable params	Natural	Specialized	Structured	Avg.
Standard LoRA	4	2.80M	73.81	72.39	23.10	52.16
H-OM	4	1.47M	77.72	76.12	25.27	55.30
Standard LoRA	8	5.60M	73.94	72.52	23.23	52.29
H-OM	8	2.95M	77.09	74.10	27.96	55.77