NULL-SPACE ORTHOGONAL FINE-TUNING FOR MULTIMODAL CONTINUAL INSTRUCTION TUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal Continual Instruction Tuning (MCIT) requires Multimodal Large Language Models (MLLMs) to adapt to new tasks while retaining previously learned knowledge, making the balance between stability and plasticity a central challenge. Orthogonal Fine-Tuning (OFT), traditionally applied to parameter-efficient downstream adaptation, offers a potential starting point for MCIT through orthogonal transformations. In MCIT, orthogonality alone does not explicitly control interference with previous tasks. A null-space constraint can reduce such interference, but simply imposing it on OFT is insufficient to guarantee orthogonality. Therefore, we propose Null-space Orthogonal Fine-Tuning (NOFT), which parameterizes task-specific transformations to jointly satisfy the null-space and orthogonality constraints, thereby improving stability on continual adaptation. To enhance plasticity under OFT's fixed block-diagonal structure, NOFT further introduces plasticity-aware permutation, which rearranges the trainable block positions to capture more current-task gradient information without increasing the parameter budget. Experiments with LLaVA-v1.5-7B and Qwen3-VL-8B on two MCIT benchmarks demonstrate that NOFT effectively balances stability and plasticity and achieves state-of-the-art performance.

1 INTRODUCTION

Multimodal Large Language Models (MLLMs) have demonstrated strong generalization across diverse multimodal tasks, enabling a broad range of downstream applications (Liu et al., 2023; Bai et al., 2025; Ma et al., 2026). Despite these advances, pre-trained MLLMs may still require further adaptation to meet the performance demands of specific downstream tasks. In practical deployments, new multimodal tasks can emerge over time, necessitating sequential adaptation and motivating the study of Multimodal Continual Instruction Tuning (MCIT) (Chen et al., 2024; He et al., 2026). Effective MCIT requires models to acquire knowledge from new tasks (*plasticity*) while retaining knowledge learned from previous tasks (*stability*). However, catastrophic forgetting can substantially degrade performance on previously learned tasks (Zhou et al., 2025; Yu et al., 2026). Consequently, balancing stability and plasticity remains a central challenge in MCIT.

To mitigate catastrophic forgetting, recent studies introduce lightweight task-specific modules, such as prompt pools (Zeng et al., 2025; Fu et al., 2026) and Mixture-of-Experts (MoE) architectures (Zhao et al., 2025; Xie et al., 2026), while keeping the pre-trained MLLM backbone frozen. Although these approaches can reduce interference across sequential tasks, accumulating additional modules may increase inference overhead and deployment complexity (Chen et al., 2025a; Jie et al., 2025). Since these costs arise from continually expanding task-specific components, a desirable approach is to incorporate new-task knowledge directly into the existing model parameters without expanding the model architecture. Following this direction, we focus on parameter-efficient fine-tuning (PEFT) (Hu et al., 2022; Liu et al., 2024) approaches that adapt the model using few trainable parameters.

Orthogonal Fine-Tuning (OFT) (Liu et al., 2021; Qiu et al., 2023) is a parameter-efficient approach for downstream task adaptation that modifies existing model weights through orthogonal transformations. Such transformations preserve the norms and pairwise angles of the weight vectors, helping retain the geometric relations associated with pre-trained knowledge. Preserving these relations has been shown to help maintain pre-training performance (Qiu et al., 2023) and mitigate catastrophic forgetting (Qiu et al., 2025b). However, MCIT imposes a stronger stability requirement that the

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.

model should preserve knowledge accumulated across previously learned tasks rather than only from pre-training. Although orthogonality preserves the weight geometry, vanilla OFT does not explicitly account for knowledge acquired from previous tasks when determining its weight updates. Consequently, the resulting transformation may still interfere with previously acquired knowledge. Moreover, from the plasticity perspective, parameter-efficient OFT typically adopts a block-diagonal orthogonal structure (Qiu et al., 2023), which restricts trainable parameters to fixed matrix positions regardless of the incoming task, potentially limiting adaptation when informative gradient components of the current task fall outside the trainable positions.

To overcome these limitations, we propose Null-space Orthogonal Fine-Tuning (NOFT) for MCIT. A natural way to reduce interference with previous tasks is to constrain the weight update to a null space constructed from previous-task features. However, directly imposing such a constraint on OFT does not guarantee that the resulting transformation remains orthogonal. To resolve this issue, we derive a null-space constrained orthogonal parameterization that restricts the induced weight update to this subspace while preserving the orthogonality of the task-specific transformation. By confining the update to the null space, NOFT limits changes to layer responses to previous-task features during new-task learning, thereby reducing interference with previously acquired knowledge. Meanwhile, maintaining orthogonality preserves the geometric relations of the weight vectors throughout continual adaptation. Together, these properties improve stability during continual adaptation. Moreover, to enhance plasticity under the parameter-efficient block-diagonal structure, NOFT introduces plasticity-aware permutation. Since the fixed partition is independent of the incoming task, informative gradient components from the current task may fall outside the trainable block positions, limiting adaptation to the new task. At the beginning of each task, NOFT uses the orthogonality-constrained gradient computed from the current task to construct a permutation matrix, which reorganizes the block positions to capture more informative gradient components. This reorganization improves plasticity without increasing the parameter budget.

Our main contributions are summarized as follows. *First*, we propose Null-space Orthogonal Fine-Tuning (NOFT) for multimodal continual instruction tuning. NOFT constrains task-specific orthogonal updates to a null space constructed from previous task features, reducing interference with previously acquired knowledge while preserving the geometric properties of orthogonal fine-tuning. *Second*, we introduce plasticity-aware permutation to improve task adaptation under the parameter-efficient block diagonal structure. The gradient of the current task is used to construct a permutation matrix that reorganizes the trainable block positions without increasing the parameter budget. *Third*, extensive experiments on LLaVA-v1.5-7B and Qwen3-VL-8B backbones validate that NOFT effectively balances stability and plasticity, achieving state-of-the-art performance on two MCIT benchmarks.

2 Related work

Continual Instruction Tuning. With the development of MLLMs, continual learning has been extended to continual instruction tuning, where models sequentially adapt to vision-language tasks while retaining knowledge acquired from previous tasks (Chen et al., 2024; He et al., 2026). Existing methods largely rely on expanded modules to reduce interference across sequential tasks, with MoE-based designs being particularly common. For example, CoIN (Chen et al., 2024) introduces MoELoRA to capture task-specific knowledge with multiple LoRA experts, while ProgLoRA (Yu et al., 2025) progressively adds LoRA blocks for incoming tasks. HiDe-LLaVA (Guo et al., 2025a) combines task-specific expansion with task-general fusion, whereas LLaVA-CMoE (Zhao et al., 2025) adaptively expands and routes LoRA experts. ModalPrompt (Zeng et al., 2025) learns task-specific prompts for continual adaptation. More recently, SAME (Xie et al., 2026) identifies router drift and expert drift as two major sources of forgetting in MoE-based adaptation, and mitigates them through orthogonal routing subspaces and drift-aware expert updates. Beyond these parameter expansion designs, some approaches instead mitigate forgetting by regularizing or constraining parameter updates. SEFE (Chen et al., 2025b) identifies critical elements in previous weight update matrices and regularizes subsequent updates at these important positions. KeepLoRA (Luo et al., 2026) constrains the gradients of LoRA updates onto a residual subspace orthogonal to both pre-trained weights and previous knowledge features. In comparison, NOFT introduces a transformation that jointly ensures the null-space and orthogonality constraints to improve stability. After each task, it can be merged

into the model weights, avoiding the additional memory overhead and routing complexity associated with the expanded modules at inference.

Orthogonal Fine-Tuning. As a paradigm of parameter-efficient fine-tuning, OFT adapts pre-trained models through multiplicative orthogonal transformations (Liu et al., 2021; Qiu et al., 2023). These transformations preserve the geometric relations among weight vectors (Qiu et al., 2023), helping retain the knowledge encoded in the pre-trained model. To reduce the cost of high-dimensional orthogonal transformations, OFT (Qiu et al., 2023) adopts a block-diagonal structure, while subsequent methods improve the efficiency and expressiveness of this parameterization through butterfly factorization (Liu et al., 2024), Givens rotations (Ma et al., 2024), Cayley–Neumann parameterization (Qiu et al., 2025b), and group-and-shuffle structures (Gorbunov et al., 2024). Moreover, PSOFT (Wu et al., 2026) decomposes pre-trained weights into principal and residual components and applies orthogonal adaptation to the principal component for downstream adaptation. Beyond parameter-efficient fine-tuning, orthogonal transformations have also been explored for large-scale LLM pre-training. POET (Qiu et al., 2025a) introduces orthogonal equivalence transformations that preserve spectral properties during training, while POET-X (Qiu et al., 2026) further improves the scalability and memory efficiency of this paradigm. In MCIT, preserving weight geometry alone does not explicitly control interference with previous tasks. NOFT addresses this limitation by constraining the induced weight updates to limit changes in layer responses to previous-task features, thereby mitigating forgetting. To improve plasticity, NOFT further uses the current-task gradients to reassign coordinates across trainable orthogonal blocks, capturing more informative gradient components without increasing the parameter budget.

3 PRELIMINARY

Problem Formulation. We consider a multi-modal continual instruction tuning setting, where a pre-trained MLLM encounters a sequence of T tasks $\{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_T\}$. Each task $\mathcal{D}_t = \{(x_i^v, x_i^q, y_i)\}_{i=1}^{N_t}$ for $t \in \{1, 2, \ldots, T\}$ contains N_t training examples, where x_i^v, x_i^q and y_i denote the input image, the instruction and the target answer. The goal of continual instruction tuning is to learn new knowledge while preserving the knowledge from previously learned tasks.

Orthogonal Fine-Tuning. Let $\boldsymbol{W}_{t-1} \in \mathbb{R}^{d\times n}$ denote the weight matrix obtained after learning the previous tasks in MCIT. To adapt to the t-th task, Orthogonal Fine-Tuning (Liu et al., 2021) (OFT) freezes $\boldsymbol{W}_{t-1}$ and learns an orthogonal transformation $\boldsymbol{R}_t \in \mathbb{R}^{d\times d}$. The updated weight is given by $\boldsymbol{W}_t = \boldsymbol{R}_t\boldsymbol{W}_{t-1}$, and the corresponding forward pass can be written as

$$\boldsymbol{h} = \boldsymbol{x}\boldsymbol{R}_t\boldsymbol{W}_{t-1} = \boldsymbol{x}\boldsymbol{W}_t, \qquad \text{s.t.} \quad \boldsymbol{R}_t^\top\boldsymbol{R}_t = \boldsymbol{R}_t\boldsymbol{R}_t^\top = \boldsymbol{I}_d. \tag{1}$$

The transformation $\boldsymbol{R}_t$ is initialized as the identity matrix $\boldsymbol{I}_d$, ensuring that adaptation to the t-th task starts from $\boldsymbol{W}_{t-1}$. Since $\boldsymbol{R}_t$ is orthogonal, the Gram matrix of the weight vectors is preserved:

$$\boldsymbol{W}_t^\top\boldsymbol{W}_t = \boldsymbol{W}_{t-1}^\top\boldsymbol{R}_t^\top\boldsymbol{R}_t\boldsymbol{W}_{t-1} = \boldsymbol{W}_{t-1}^\top\boldsymbol{W}_{t-1}. \tag{2}$$

Consequently, OFT maintains the norms and pairwise angles of the weight vectors throughout adaptation. Moreover, directly parameterizing $\boldsymbol{R}_t \in \mathbb{R}^{d\times d}$ incurs a high parameter cost. To improve parameter efficiency, OFT adopts a block-diagonal structure (Qiu et al., 2023) and constructs $\boldsymbol{R}_t$ as $\boldsymbol{R}_t = \operatorname{diag}(\boldsymbol{R}_{t,1}, \ldots, \boldsymbol{R}_{t,K})$, where each block $\boldsymbol{R}_{t,i} \in \mathbb{R}^{r\times r}$ is a small orthogonal matrix, with $K = d/r$. This block-diagonal parameterization substantially reduces the number of trainable parameters while retaining the geometric preservation induced by orthogonal transformations.

Discussion. OFT mitigates catastrophic forgetting by constraining adaptation to structure-preserving orthogonal transformations (Qiu et al., 2023; 2025b). Unlike additive low-rank updates, OFT preserves the structural properties of pretrained weights, which empirically improves training stability and preserves the pretrained capabilities during fine-tuning on a single downstream task.

4 METHOD

Despite the effectiveness of OFT for fine-tuning weights through orthogonal transforms, extending OFT to continual instruction tuning is non-trivial. The first challenge is that orthogonal fine-tuning cannot prevent forgetting of previously learned knowledge during sequential instruction-tuning, since

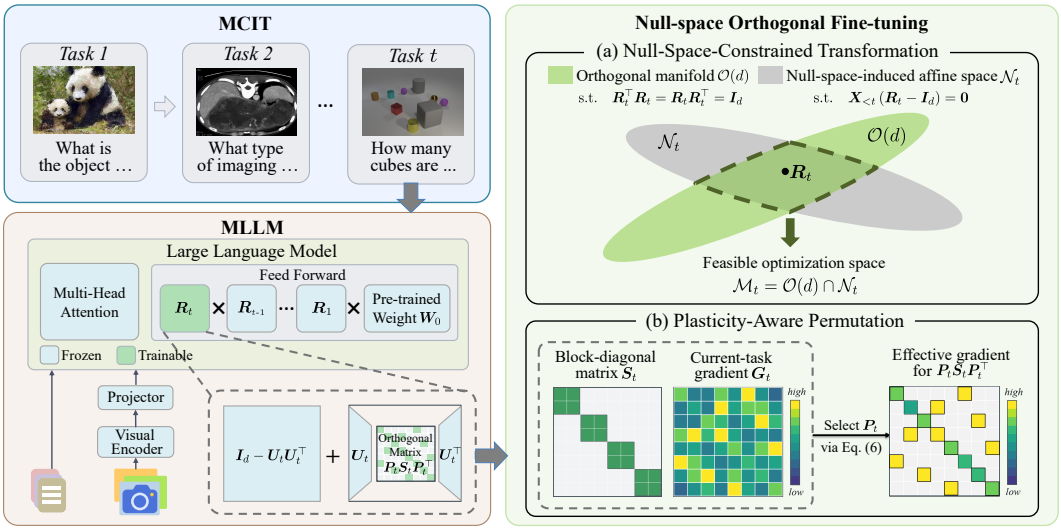


Figure 1: Overview of NOFT. NOFT balances stability and plasticity in continual adaptation by (a) constructing a transformation $\boldsymbol{R}_t$ in $\mathcal{O}(d) \cap \mathcal{N}_t$ to jointly satisfy the orthogonality and null-space constraints, thereby mitigating interference with previous tasks, and (b) permuting the block-diagonal orthogonal matrix to reorganize trainable block positions toward more informative current-task gradients. In (a), $\mathcal{O}(d) = \{\boldsymbol{R}_t \in \mathbb{R}^{d\times d} : \boldsymbol{R}_t^\top \boldsymbol{R}_t = \boldsymbol{R}_t \boldsymbol{R}_t^\top = \boldsymbol{I}_d\}$ denotes the orthogonal manifold, and $\mathcal{N}_t = \{\boldsymbol{R}_t \in \mathbb{R}^{d\times d} : \boldsymbol{X}_{<t}(\boldsymbol{R}_t - \boldsymbol{I}_d) = \boldsymbol{0}\}$ denotes the null-space-induced affine space. In (b), colored and gray entries denote trainable and frozen positions, respectively.

successive OFT updates can alter model responses to features associated with earlier tasks. A natural way to mitigate such interference is to restrict parameter updates to the null space of previous-task features (Wang et al., 2025). However, incorporating this constraint into OFT is a challenge, as the induced weight update must satisfy the null-space constraint while the transformation matrix $\boldsymbol{R}_t$ must remain orthogonal. The second challenge is that the predefined block-diagonal structure of parameter-efficient OFT restricts model plasticity. Each orthogonal block operates within a predefined partition of the feature space. Such a fixed partition limits the flexibility of orthogonal transformations over feature space and may hinder adaptation to continual instruction-tuning tasks.

To address these challenges, we propose null-space orthogonal fine-tuning (NOFT), which extends OFT to continual instruction tuning through two key designs. The overall framework of NOFT is illustrated in Fig. 1. First, through a rigorous derivation, we obtain a null-space-constrained orthogonal transformation $\boldsymbol{R}_t$ that preserves its orthogonality while ensuring that the induced weight update lies in the null space of previous-task features, thereby reducing interference with previously acquired knowledge. Second, we leverage task-specific gradient information to rearrange the block-diagonal structure, yielding a plasticity-aware partition for each task that provides flexibility for learning new knowledge. Specifically, taking $\boldsymbol{U}_t$ as the basis of the null space of previous-task features, the resulting null-space-constrained orthogonal transformation $\boldsymbol{R}_t$ is

$$\boldsymbol{R}_t = \boldsymbol{I}_d + \boldsymbol{U}_t \left(\bar{\boldsymbol{S}}_t - \boldsymbol{I}_m\right) \boldsymbol{U}_t^\top, \tag{3}$$

where $\bar{\boldsymbol{S}}_t = \boldsymbol{P}_t \boldsymbol{S}_t \boldsymbol{P}_t^\top$, with $\boldsymbol{P}_t$ denoting a plasticity-aware permutation matrix and $\boldsymbol{S}_t$ the block-diagonal orthogonal matrix. In the remainder of this section, we first derive the null-space-constrained orthogonal transformation $\boldsymbol{R}_t$, and then construct the plasticity-aware permutation matrix $\boldsymbol{P}_t$.

4.1 NULL-SPACE-CONSTRAINED ORTHOGONAL TRANSFORMATION

Let $\boldsymbol{X}_{<t}$ denote the feature matrix of the preceding tasks $1, \ldots, t-1$, and let $\boldsymbol{U}_t \in \mathbb{R}^{d\times m}$ denote an orthonormal basis of the null space of $\boldsymbol{X}_{<t}$. We use $\mathrm{col}(\boldsymbol{U}_t)$ to denote the column space of $\boldsymbol{U}_t$, which corresponds to the null space of $\boldsymbol{X}_{<t}$. Thus, $\boldsymbol{X}_{<t}\boldsymbol{U}_t = \boldsymbol{0}$ and $\boldsymbol{U}_t^\top \boldsymbol{U}_t = \boldsymbol{I}_m$. With $\boldsymbol{U}_t$, we seek to parameterize the orthogonal transformation $\boldsymbol{R}_t$ such that the induced weight update is constrained to this null space $\mathrm{col}(\boldsymbol{U}_t)$ while $\boldsymbol{R}_t$ remains orthogonal.

Specifically, given $\boldsymbol{W}_t = \boldsymbol{R}_t \boldsymbol{W}_{t-1}$, the induced weight update at the t-th task is $\Delta \boldsymbol{W}_t = \boldsymbol{W}_t - \boldsymbol{W}_{t-1} = (\boldsymbol{R}_t - \boldsymbol{I}_d)\boldsymbol{W}_{t-1}$. The null-space constraint on the induced update is given by $\boldsymbol{X}_{<t}\Delta \boldsymbol{W}_t = \boldsymbol{0}$. Furthermore, the transformation matrix $\boldsymbol{R}_t$ is constrained to be orthogonal, which amounts to enforcing $\boldsymbol{R}_t^\top \boldsymbol{R}_t = \boldsymbol{I}_d$. We next derive a parameterization of $\boldsymbol{R}_t$ that simultaneously satisfies these two constraints in Proposition 1.

Proposition 1. *Given the null-space basis $\boldsymbol{U}_t$ defined above, if $\boldsymbol{R}_t$ admits the parameterization in Eq. 3, where $\bar{\boldsymbol{S}}_t \in \mathbb{R}^{m\times m}$ is orthogonal, then the transformation $\boldsymbol{R}_t \in \mathbb{R}^{d\times d}$ is orthogonal and the induced weight update $\Delta \boldsymbol{W}_t = (\boldsymbol{R}_t - \boldsymbol{I}_d)\boldsymbol{W}_{t-1}$ lies in the null space of previous-task features, i.e., $\Delta \boldsymbol{W}_t \in \mathrm{col}(\boldsymbol{U}_t)$.*

The detailed proof of Proposition 1 can be found in Sec. A.1. Proposition 1 provides a parameterization that simultaneously preserves the orthogonality of $\boldsymbol{R}_t$ and constrains the induced weight update to the null space of the previous-task features. Since $\boldsymbol{U}_t^\top \boldsymbol{U}_t = \boldsymbol{I}_m$ and $\bar{\boldsymbol{S}}_t$ is orthogonal, we have

$$\boldsymbol{R}_t^\top \boldsymbol{R}_t = \boldsymbol{I}_d + \boldsymbol{U}_t \left(\bar{\boldsymbol{S}}_t^\top \bar{\boldsymbol{S}}_t - \boldsymbol{I}_m\right) \boldsymbol{U}_t^\top = \boldsymbol{I}_d, \tag{4}$$

which verifies the orthogonality of $\boldsymbol{R}_t$. Substituting Eq. 3 into the induced weight update gives

$$\Delta \boldsymbol{W}_t = \boldsymbol{U}_t \left(\bar{\boldsymbol{S}}_t - \boldsymbol{I}_m\right) \boldsymbol{U}_t^\top \boldsymbol{W}_{t-1}, \tag{5}$$

which directly ensures that $\Delta \boldsymbol{W}_t \in \mathrm{col}(\boldsymbol{U}_t)$.

Rather than directly learning a dense orthogonal matrix $\bar{\boldsymbol{S}}_t$, we construct it as $\bar{\boldsymbol{S}}_t = \boldsymbol{P}_t \boldsymbol{S}_t \boldsymbol{P}_t^\top$, where $\boldsymbol{S}_t$ is a learnable block-diagonal orthogonal matrix and $\boldsymbol{P}_t$ is a plasticity-aware permutation matrix. The block-diagonal parameterization of $\boldsymbol{S}_t$ preserves the parameter efficiency of OFT by restricting learning to a collection of small orthogonal blocks, substantially reducing the number of trainable parameters compared with a dense orthogonal transformation. Following OFT, each block of $\boldsymbol{S}_t$ is parameterized via the Cayley transform of a learnable skew-symmetric matrix, which guarantees its orthogonality throughout optimization. Since both $\boldsymbol{S}_t$ and $\boldsymbol{P}_t$ are orthogonal, $\bar{\boldsymbol{S}}_t$ is also orthogonal. The plasticity-aware permutation $\boldsymbol{P}_t$ rearranges the coordinate assignment across orthogonal blocks according to the current task, thereby relaxing the restriction imposed by a fixed block partition. The construction of $\boldsymbol{P}_t$ is detailed in Sec. 4.2.

To obtain $\boldsymbol{U}_t$ in practice, we follow prior null-space-based continual learning methods (Wang et al., 2025) and estimate the null space from features collected on the preceding tasks. The eigenvectors associated with low-energy directions of the previous-task feature covariance matrix, determined by a threshold τ, are retained to form the orthonormal basis $\boldsymbol{U}_t$. By restricting weight updates to this estimated null space, NOFT helps limit changes in layer responses to previous-task features and mitigate interference with previously learned knowledge. Further details on the construction of $\boldsymbol{U}_t$, including the maintenance of previous-task features and the thresholding criterion, are provided in Appendix A.2.

4.2 Plasticity-Aware Permutation for Orthogonal Transformations

While the block-diagonal parameterization of $\boldsymbol{S}_t$ is parameter-efficient, its fixed diagonal blocks may not align with the entries that are most relevant for adapting to a new task, thereby limiting plasticity. We therefore introduce a plasticity-aware permutation matrix $\boldsymbol{P}_t$ to reassign the coordinates occupied by the learnable blocks based on the task-specific gradients of $\boldsymbol{S}_t$.

We denote by Ω the set of coordinates corresponding to the diagonal blocks, and compute the gradient matrix $\boldsymbol{G}_t$ with respect to $\boldsymbol{S}_t$ of the current-task objective, where larger gradient magnitudes indicate greater sensitivity of the t-th task objective to the corresponding entries. The derivation of $\boldsymbol{G}_t$ is provided in Appendix B.1. For a candidate permutation $\boldsymbol{P} \in \mathcal{P}_m$, let $\Omega_{\boldsymbol{P}}$ denote the coordinate set obtained by mapping Ω under $\boldsymbol{P}$. We then select the permutation whose resulting coordinate set has the largest sum of squared gradient magnitudes:

$$\boldsymbol{P}_t = \arg\max_{\boldsymbol{P}\in\mathcal{P}_m} \sum_{(i,j)\in\Omega_{\boldsymbol{P}}} (\boldsymbol{G}_t)_{ij}^2, \tag{6}$$

where $\mathcal{P}_m$ denotes the set of $m \times m$ permutation matrices.

Directly optimizing over $\mathcal{P}_m$ is computationally prohibitive, as it contains $m!$ permutation matrices. We therefore exploit the block structure of Ω to reformulate the permutation problem as a coordinate-grouping problem. Specifically, a candidate permutation $\boldsymbol{P}$ partitions the m coordinates into b disjoint

groups $\{\mathcal{B}_1, \ldots, \mathcal{B}_b\}$, each of size r, where coordinates within the same group are assigned to the same diagonal block. Accordingly, the objective in Eq. 6 can be written as

$$\sum_{(i,j)\in\Omega_{\boldsymbol{P}}} (\boldsymbol{G}_t)_{ij}^2 = \sum_{k=1}^{b} \sum_{i,j\in\mathcal{B}_k} (\boldsymbol{G}_t)_{ij}^2. \tag{7}$$

Thus, selecting $\boldsymbol{P}_t$ amounts to grouping the coordinates such that large squared gradient magnitudes are preferentially retained within the same group.

To construct the coordinate groups efficiently, we adopt a greedy approximation. Let $\mathcal{R}$ denote the set of coordinates not yet assigned to a completed group. For each new group, $\mathcal{B}$ is initialized with the pair of coordinates in $\mathcal{R}$ having the largest pairwise gradient contribution. Starting from this pair, $\mathcal{B}$ is expanded one coordinate at a time. At each step, the remaining coordinate with the largest total squared gradient contribution to the coordinates already in $\mathcal{B}$ is selected:

$$v^* = \arg\max_{v\in\mathcal{R}\setminus\mathcal{B}} \sum_{u\in\mathcal{B}} \left[(\boldsymbol{G}_t)_{uv}^2 + (\boldsymbol{G}_t)_{vu}^2\right]. \tag{8}$$

Once $|\mathcal{B}| = r$, the completed group is removed from $\mathcal{R}$, and the same procedure is repeated until b disjoint groups are obtained. We then construct $\boldsymbol{P}_t$ from these groups, with each group corresponding to one diagonal block. Further details of the greedy construction are provided in Appendix B.3.

The permutation $\boldsymbol{P}_t$ is determined before optimizing the learnable blocks and remains fixed throughout adaptation to task t. With the coordinate assignment specified by $\boldsymbol{P}_t$, only the block matrices $\{\boldsymbol{S}_{t,k}\}_{k=1}^{b}$ are optimized, without increasing the number of trainable parameters. In practice, the gradient matrix $\boldsymbol{G}_t$ used to determine $\boldsymbol{P}_t$ is estimated once at task initialization using a fraction ρ of the current-task training samples. The overall training procedure of NOFT is summarized in Algorithm 1 in Appendix A.3.

The computational overhead of NOFT consists of one-time task initialization and iterative optimization. During initialization, truncated SVD constructs the m-dimensional null-space basis $\boldsymbol{U}_t$ with a complexity of $\mathcal{O}(d^2m)$, where $m \ll d$. The orthogonality-constrained gradient is estimated from a sampled subset of the current task, and obtaining an approximate solution for $\boldsymbol{P}_t$ has a worst-case time complexity of $\mathcal{O}(m^3)$. During optimization, applying the block-diagonal $\boldsymbol{S}_t$ with the projections through $\boldsymbol{U}_t$ incurs a per-step transformation cost of $\mathcal{O}(dm+mr)$. The practical efficiency comparison is reported in Table 4.

5 EXPERIMENT

Backbone MLLMs and Datasets. Our experiments are conducted on two MLLMs, LLaVA-v1.5-7B (Liu et al., 2023) with CLIP ViT-L/14-336 (Radford et al., 2021) and Qwen3-VL-8B (Bai et al., 2025). We evaluate NOFT on two multimodal continual instruction tuning benchmarks, UCIT (Guo et al., 2025a) and TriGap (Xie et al., 2026). UCIT contains six tasks with relatively limited domain shifts and comparable task scales, while TriGap includes ten tasks with larger domain shifts and stronger scale imbalance. Detailed benchmark information is provided in Appendix C.3.

Compared Methods. We compare NOFT with standard baselines, including Zero-shot, FT-LoRA (Hu et al., 2022), Replay-LoRA (Hu et al., 2022), and OFT (Qiu et al., 2023). We also compare with representative MCIT methods, including MoE-LoRA (Chen et al., 2024), HiDe-LLaVA (Guo et al., 2025a), ModalPrompt (Zeng et al., 2025), CL-MoE (Huai et al., 2025), DISCO (Guo et al., 2025b), KeepLoRA (Luo et al., 2026), and SAME (Xie et al., 2026).

Training Details. All experiments of NOFT are conducted on 4 NVIDIA RTX A6000 (48GB) GPUs with BF16 precision. We optimize NOFT with the standard autoregressive cross-entropy loss. The model is trained for one epoch on each task, with per-device batch sizes of 6 for UCIT and 8 for TriGap, corresponding to global batch sizes of 24 and 32, respectively. The visual encoder remains frozen throughout training, while the MLP modules in all LLM layers are fine-tuned. AdamW is used as the optimizer with a cosine learning-rate decay schedule. LLaVA-v1.5-7B uses a learning rate of 2×10^{-4} and $b = 32$ blocks, whereas Qwen3-VL-8B uses 1×10^{-4} and $b = 64$ blocks. The gradient used for permutation selection is estimated from $\rho = 5\%$ of the current-task training samples, while the null-space threshold is set to $\tau = 0.2$ for constructing $\boldsymbol{U}_t$.

Table 1: Continual Instruction Tuning on the TriGap benchmark. **Avg.** denotes the average final performance across all tasks. The best results are highlighted in **bold**, while the second-best results are underlined.

Methods	PMCVQA	DocVQA	ChartQA	IconQA	InfoVQA	ArxivQA	Roadside	ChemVQA	FloodNetVQA	CLEVR	Avg.(↑)
LLaVA-v1.5-7B											
Zero-shot	35.40	12.68	9.36	19.27	5.06	53.77	7.40	5.30	47.41	20.37	21.60
FT-LoRA (Hu et al., 2022)	34.20	23.32	9.84	37.07	23.53	83.83	7.00	12.70	80.31	60.27	37.21
Replay-LoRA (Hu et al., 2022)	33.70	33.95	14.00	46.67	28.97	75.57	9.40	15.90	73.81	58.80	39.08
OFT (Qiu et al., 2023)	37.33	12.76	4.20	32.67	14.56	87.50	8.20	13.83	79.54	49.77	34.04
MoE-LoRA (Chen et al., 2024)	40.07	36.90	11.84	51.47	36.82	88.97	7.83	17.17	80.31	61.23	43.26
HiDe-LLaVA (Guo et al., 2025a)	37.00	33.20	10.52	41.97	24.09	79.20	7.73	11.17	57.39	23.00	32.53
ModalPrompt (Zeng et al., 2025)	38.23	38.23	11.92	44.73	37.37	84.47	10.13	12.43	71.52	52.50	40.15
CL-MoE (Huai et al., 2025)	36.93	36.39	12.32	50.40	36.06	91.80	7.73	18.13	78.94	61.43	43.01
DISCO (Guo et al., 2025b)	42.21	43.75	18.13	63.44	38.50	91.72	**10.93**	22.49	**80.59**	56.40	46.82
KeepLoRA (Luo et al., 2026)	41.67	41.16	13.92	49.90	39.40	89.00	8.97	17.97	79.76	53.00	43.48
SAME (Xie et al., 2026)	42.03	43.29	18.08	63.47	**40.03**	**92.47**	10.20	19.63	80.41	56.60	46.62
NOFT (ours)	**44.50**	**43.81**	**18.72**	**66.63**	38.91	92.30	10.60	**23.30**	80.47	**61.47**	**48.07**
Qwen3-VL-8B											
Zero-shot	40.83	7.45	75.84	45.53	2.20	93.97	10.70	14.23	49.65	95.43	43.58
OFT (Qiu et al., 2023)	28.70	50.44	36.00	47.67	31.95	78.30	6.23	13.40	68.09	84.43	44.52
DISCO (Guo et al., 2025b)	58.77	94.34	77.86	92.70	77.45	**95.27**	12.70	29.90	78.99	98.53	71.65
KeepLoRA (Luo et al., 2026)	57.40	93.59	76.92	91.50	77.03	94.83	12.07	21.27	77.89	98.00	70.05
SAME (Xie et al., 2026)	58.13	**94.69**	**78.16**	94.80	77.14	94.27	10.63	20.40	56.54	**98.97**	68.37
NOFT (ours)	**59.00**	94.64	**78.16**	**95.07**	**77.78**	93.33	**15.67**	**33.73**	**81.18**	98.83	**72.74**

Evaluation Metrics. Following Xie et al. (2026), let $\mathcal{A}_{s,t}$ denote the performance on the s-th task after sequential training up to the t-th task, where T is the total number of tasks. We report the *Average Final Performance* and *Average Immediate Performance*, defined as

$$\bar{\mathcal{A}}_{\text{final}} = \frac{1}{T}\sum_{s=1}^{T}\mathcal{A}_{s,T}, \qquad \bar{\mathcal{A}}_{\text{imm}} = \frac{1}{T}\sum_{s=1}^{T}\mathcal{A}_{s,s}.$$

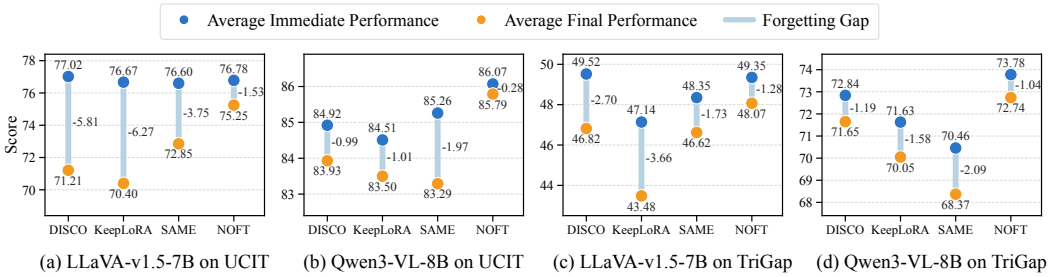
Here, $\bar{\mathcal{A}}_{\text{final}}$ measures the average performance over all tasks after completing the training of the entire task sequence, while $\bar{\mathcal{A}}_{\text{imm}}$ measures the immediate performance after each task was trained and averaged over all tasks, as a measure for forgetting analysis.

Benchmark Comparison. Tables 1 and 2 summarize the average final performance after sequentially learning all tasks on TriGap and UCIT, respectively. On LLaVA-v1.5-7B, NOFT achieves the best overall performance on both benchmarks, outperforming the strongest competing baselines by 1.25% on TriGap and 2.40% on UCIT. Compared with vanilla OFT, which only preserves weight geometry through orthogonal transformations, NOFT obtains substantial improvements of 14.03% and 9.11%, respectively. These results indicate that orthogonality alone is insufficient for continual instruction tuning, while incorporating the null-space constraint and plasticity-aware permutation provides more effective adaptation. The advantage of NOFT is also consistently observed with Qwen3-VL-8B. It improves over the strongest competing baselines by 1.09% on TriGap and 1.86% on UCIT, while achieving gains of 28.22% and 19.03% over vanilla OFT. The consistent improvements across two backbones and benchmarks demonstrate that NOFT maintains a favorable balance between stability and plasticity under different continual instruction tuning settings. Qualitative comparisons on TriGap after learning all ten tasks using LLaVA-v1.5-7B are provided in Appendix C.4.

Forgetting Analysis. To further investigate the stability of different methods during continual instruction tuning, we compare the average immediate performance after learning each task with the average final performance after completing the entire task sequence, as shown in Fig. 2. The gap between these two measurements reflects the performance degradation after subsequent task adaptation. Across both benchmarks and backbones, NOFT consistently achieves the smallest average forgetting gap among the methods included in this analysis. For example, on UCIT with

Table 2: Continual Instruction Tuning performance on the UCIT benchmark. **Avg.** denotes the average final performance across all tasks. The best results are highlighted in **bold**, while the second-best results are underlined.

Methods	ImageNet-R	ArxivQA	Vizcap	IconQA	CLEVR	Flickr30k	Avg.(↑)
LLaVA-v1.5-7B							
Zero-shot	18.88	52.62	38.75	21.25	21.12	41.44	32.34
FT-LoRA (Hu et al., 2022)	29.33	55.30	45.51	26.13	13.07	58.07	37.90
Replay-LoRA (Hu et al., 2022)	76.93	87.07	54.31	56.43	36.40	55.94	61.18
OFT (Qiu et al., 2023)	77.67	78.30	46.32	73.57	63.33	57.66	66.14
MoE-LoRA (Chen et al., 2024)	58.43	77.57	44.83	68.90	56.73	**58.27**	60.79
HiDe-LLaVA (Guo et al., 2025a)	87.62	91.12	42.68	57.62	31.00	50.41	60.08
ModalPrompt (Zeng et al., 2025)	80.50	90.62	**60.13**	63.50	55.75	57.09	67.93
CL-MoE (Huai et al., 2025)	64.12	78.38	44.83	62.00	50.75	58.06	59.69
DISCO (Guo et al., 2025b)	89.13	**94.32**	51.52	72.18	63.45	56.65	71.21
KeepLoRA (Luo et al., 2026)	81.30	91.40	46.04	75.40	71.10	57.15	70.40
SAME (Xie et al., 2026)	86.23	92.57	48.41	79.97	72.17	57.75	72.85
NOFT (ours)	**91.03**	93.53	54.77	**80.70**	**73.80**	57.69	**75.25**
Qwen3-VL-8B							
Zero-shot	32.07	93.97	26.69	45.53	95.43	28.87	53.76
OFT (Qiu et al., 2023)	42.00	80.60	48.40	88.40	81.60	59.57	66.76
DISCO (Guo et al., 2025b)	93.13	94.60	61.31	97.43	**99.50**	57.58	83.93
KeepLoRA (Luo et al., 2026)	88.70	94.73	61.78	96.07	99.20	60.52	83.50
SAME (Xie et al., 2026)	92.80	94.80	55.17	97.60	99.07	60.27	83.29
NOFT (ours)	**94.27**	**95.67**	**66.63**	**98.33**	98.83	**60.98**	**85.79**



(a) LLaVA-v1.5-7B on UCIT (b) Qwen3-VL-8B on UCIT (c) LLaVA-v1.5-7B on TriGap (d) Qwen3-VL-8B on TriGap

Figure 2: Comparison of Average Immediate Performance (evaluated after the current task) and Average Final Performance (evaluated after the final task) on LLaVA-v1.5-7B and Qwen3-VL-8B. The vertical separation between immediate and final performance indicates the forgetting gap after continual instruction tuning. Per-task forgetting results are further reported in Appendix C.2.

LLaVA-v1.5-7B, NOFT yields a forgetting gap of 1.53%, compared with 3.75% for SAME. Similar trends are observed in the other settings, where NOFT also achieves the smallest forgetting gap, reaching 0.28% on UCIT with Qwen3-VL-8B, 1.28% on TriGap with LLaVA-v1.5-7B, and 1.04% on TriGap with Qwen3-VL-8B. Notably, NOFT maintains comparable immediate performance while achieving higher final performance after sequential adaptation, indicating that its improvement comes from reducing task interference rather than sacrificing adaptation ability. These results demonstrate that NOFT effectively protects previously learned knowledge, leading to a better stability-plasticity trade-off in MCIT. Additional results under different task orders are provided in Appendix C.1.

5.1 Results and Model Analysis

Ablation Study. To disentangle the contribution of each design in NOFT, we conduct ablation studies on UCIT using both LLaVA-v1.5-7B and Qwen3-VL-8B, with the results summarized in Table 3. First, *w/o* $\boldsymbol{U}_t$ removes the null-space constraint and leads to performance drops of 9.47% and 14.36%. To further determine whether this benefit comes from the specific subspace identified from previous-task features, *w/ Rand.* $\boldsymbol{U}_t$ replaces $\boldsymbol{U}_t$ with a random orthonormal basis of the same size. This variant remains 7.40% and 9.76% below NOFT, showing that the null space is more effective than an arbitrary subspace. We further examine the complementary role of orthogonality through *w/o Orth.* $\boldsymbol{R}_t$, which removes the orthogonality constraint while retaining the null-space constraint.

The resulting drops of 2.49% and 3.27% indicate that restricting updates to the null space alone does not capture the full benefit of NOFT, supporting the joint use of the null-space and orthogonality constraints for maintaining stability. To evaluate the effect of plasticity-aware permutation, *w/o* $\boldsymbol{P}_t$ removes the permutation mechanism and causes drops of 0.64% and 1.01%. Moreover, *w/ Rand.* $\boldsymbol{P}_t$ replaces $\boldsymbol{P}_t$ with a randomly sampled permutation matrix and results in performance drops of 0.51% and 0.74%. These results indicate that the improvement is achieved through the gradient-guided permutation strategy rather than arbitrary coordinate reorganization.

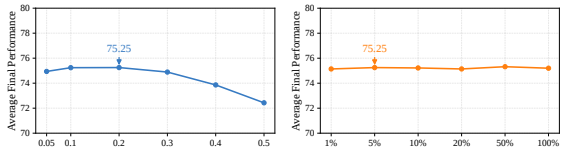
Table 3: Ablation study on UCIT. Δ denotes the change relative to NOFT.

	LLaVA-v1.5-7B		Qwen3-VL-8B	
Variant	Avg.	Δ	Avg.	Δ
OFT	66.14	-9.11	66.76	-19.03
w/o Orth. $\boldsymbol{R}_t$	72.76	-2.49	82.52	-3.27
w/o $\boldsymbol{U}_t$	65.78	-9.47	71.43	-14.36
w/o $\boldsymbol{P}_t$	74.61	-0.64	84.78	-1.01
w/ Rand. $\boldsymbol{U}_t$	67.85 $\pm$ 0.86	-7.40	76.03 $\pm$ 1.12	-9.76
w/ Rand. $\boldsymbol{P}_t$	74.74 $\pm$ 0.24	-0.51	85.05 $\pm$ 0.31	-0.74
NOFT	**75.25**	0.00	**85.79**	0.00

Rand. variants report the mean over three runs with different seeds.

Impact of Null-Space Threshold τ. To examine the sensitivity of NOFT to the approximation of the null space, we vary the threshold τ used to select singular vectors for constructing $\boldsymbol{U}_t$. The results are reported in Fig. 3a. Performance remains relatively stable across the tested values of τ from 0.05 to 0.3, while increasing τ to 0.5 leads to a 2.8% drop relative to the default $\tau = 0.2$. A larger τ includes directions associated with larger singular values, which deviate further from the exact null space and may therefore weaken the preservation of previous-task responses.

Impact of Gradient Sampling Ratio ρ. To evaluate how much current-task data is required to determine the plasticity-aware permutation, we vary the sampling ratio ρ used for gradient estimation. The results are shown in Fig. 3b. NOFT remains stable across a broad range of sampling ratios, and using only 5% of the current-task samples achieves performance comparable to substantially larger subsets. This indicates that the gradient information needed for trainable block reorganization can be estimated from a small fraction of training data, reducing the additional cost of permutation selection.



(a) Null-space threshold τ. (b) Gradient sampling ratio ρ.

Figure 3: Hyperparameter ablation of NOFT on UCIT using LLaVA-v1.5-7B. The arrows indicate our default hyperparameter settings ($\tau = 0.2$ and $\rho = 5\%$).

Table 4: Efficiency comparison on UCIT with LLaVA-v1.5-7B. † Reproduced on 4 NVIDIA PRO 6000 (96GB) GPUs.

Method	Param. (M)	Mem. (GB)	Time (h)
SAME	73.28	46.2	13.5
KeepLoRA†	64.50	73.8	7.4
NOFT	70.25	42.2	15.4

Computational Efficiency Comparison. To assess the practical efficiency of NOFT, we compare it with two recent MCIT methods, SAME and KeepLoRA, in terms of trainable parameters, runtime, and peak memory. The results are reported in Table 4. All three methods have comparable trainable parameter counts. Under the same A6000 setup, NOFT requires 15.4 h compared with 13.5 h for SAME, with the additional overhead mainly arising from the one-time null-space construction and plasticity-aware permutation selection at the beginning of each task. KeepLoRA requires 73.8 GB of peak memory and is therefore evaluated on 4 PRO 6000 (96GB) GPUs. Since this hardware configuration differs from the A6000 setup, its runtime is not directly comparable. NOFT achieves the lowest peak memory usage of 42.2 GB among the three methods.

6 Conclusion

We propose Null-space Orthogonal Fine-Tuning (NOFT), a parameter-efficient approach for multimodal continual instruction tuning that extends orthogonal adaptation to improve the stability–plasticity trade-off. By constraining task-specific updates to the null space of previous-task features while maintaining orthogonality, NOFT reduces interference with previously acquired knowledge and preserves the geometric properties of model weights. To overcome the limited flexibility of the fixed block-diagonal structure, NOFT introduces plasticity-aware permutation. It adapts the coordinate assignment of trainable blocks based on current-task gradients, enabling more effective use of the fixed parameter budget. Extensive experiments on two MLLM backbones and two continual instruction tuning benchmarks demonstrate that NOFT consistently improves continual adaptation and achieves a better stability-plasticity trade-off. Future work may extend NOFT to larger-scale multimodal foundation models and richer modalities beyond vision-language settings.

AI USE STATEMENT

In this work, we used generative AI tools to assist with language editing, including improving the clarity, grammar, and presentation of author-written text. We also used these tools to assist with the presentation of experimental results, such as refining figure and table layouts and generating or adjusting visualization code. Generative AI was not used to design the proposed method or to generate experimental results. All reported numerical results were obtained from our experiments and verified against the original experimental outputs. All AI-assisted text, code, and visualizations were reviewed and verified by the authors. The authors made the final decisions regarding the scientific claims, interpretations, and presentation, and take full responsibility for the content of this work.

REPRODUCIBILITY STATEMENT

To ensure the reproducibility of our findings, we provide the main training configurations and implementation details in Sec. 5. The construction of null-space basis and implementation details of plasticity-aware permutation are provided in Appendices A.2 and B, respectively. The complete training procedure of NOFT is provided in Appendix A.3. The source code will be released soon.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Chandramohan Bhuma, Vallabhaneni Madhava Rao, N. Srinivasa Rao, and Sumanth Kumar Panguluri. Chemvqa-2k: A visual question answering dataset for molecular understanding, 2025.

Cheng Chen, Junchen Zhu, Xu Luo, Heng Tao Shen, Jingkuan Song, and Lianli Gao. Coin: A benchmark of continual instruction tuning for multimodel large language models. *Advances in Neural Information Processing Systems*, 37:57817–57840, 2024.

Haoran Chen, Ping Wang, Zihan Zhou, Xu Zhang, Zuxuan Wu, and Yu-Gang Jiang. Achieving more with less: Additive prompt tuning for rehearsal-free class-incremental learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 340–349, 2025a.

Jinpeng Chen, Runmin Cong, Yuzhi Zhao, Hongzheng Yang, Guangneng Hu, Horace Ip, and Sam Kwong. SEFE: Superficial and essential forgetting eliminator for multimodal continual instruction tuning. In *International Conference on Machine Learning*, volume 267, pp. 7982–8001, 2025b.

Hao Fu, Hanbin Zhao, Jiahua Dong, Henghui Ding, Chao Zhang, and Hui Qian. Iap: Improving continual learning of vision-language models via instance-aware prompting. *IEEE Transactions on Image Processing*, 35:717–731, 2026.

Mikhail Gorbunov, Nikolay Yudin, Vera Soboleva, Aibek Alanov, Alexey Naumov, and Maxim Rakhuba. Group and shuffle: Efficient structured orthogonal parametrization. *Advances in Neural Information Processing Systems*, 37:68713–68739, 2024.

Runwei Guan, Rongsheng Hu, Shangshu Chen, Ningyuan Xiao, Xue Xia, Jiayang Liu, Beibei Chen, Ziren Tang, Ningwei Ouyang, Shaofeng Liang, et al. Roadscenevqa: Benchmarking visual question answering in roadside perception systems for intelligent transportation system. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 4366–4375, 2026.

Haiyang Guo, Fanhu Zeng, Ziwei Xiang, Fei Zhu, Da-Han Wang, Xu-Yao Zhang, and Cheng-Lin Liu. HiDe-LLaVA: Hierarchical decoupling for continual instruction tuning of multimodal large language model. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, pp. 13572–13586, 2025a.

Haiyang Guo, Fanhu Zeng, Fei Zhu, Wenzhuo Liu, Da-Han Wang, Jian Xu, Xu-Yao Zhang, and Cheng-Lin Liu. Federated continual instruction tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1325–1335, 2025b.

Danna Gurari, Qing Li, Abigale J. Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P. Bigham. Vizwiz grand challenge: Answering visual questions from blind people. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Jinghan He, Haiyun Guo, Kuan Zhu, Ming Tang, and Jinqiao Wang. Continual instruction tuning for large multimodal models. *IEEE Transactions on Image Processing*, 35:2699–2713, 2026.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8340–8349, 2021.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Tianyu Huai, Jie Zhou, Xingjiao Wu, Qin Chen, Qingchun Bai, Ze Zhou, and Liang He. Cl-moe: Enhancing multimodal large language model with dual momentum mixture-of-experts for continual visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19608–19617, 2025.

Shibo Jie, Yehui Tang, Kai Han, Yitong Li, Duyu Tang, Zhi-Hong Deng, and Yunhe Wang. Mixture of lookup experts. In *International Conference on Machine Learning*, volume 267, pp. 27929–27940, 2025.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Lei Li, Yuqi Wang, Runxin Xu, Peiyi Wang, Xiachong Feng, Lingpeng Kong, and Qi Liu. Multimodal ArXiv: A dataset for improving scientific comprehension of large vision-language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pp. 14369–14387, 2024.

Adam Dahlgren Lindström and Savitha Sam Abraham. Clevr-math: A dataset for compositional language, visual and mathematical reasoning. *arXiv preprint arXiv:2208.05358*, 2022.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in Neural Information Processing Systems*, 36:34892–34916, 2023.

Weiyang Liu, Rongmei Lin, Zhen Liu, James M. Rehg, Liam Paull, Li Xiong, Le Song, and Adrian Weller. Orthogonal over-parameterized training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7251–7260, 2021.

Weiyang Liu, Zeju Qiu, Yao Feng, Yuliang Xiu, Yuxuan Xue, Longhui Yu, Haiwen Feng, Zhen Liu, Juyeon Heo, Songyou Peng, Yandong Wen, Michael J. Black, Adrian Weller, and Bernhard Schölkopf. Parameter-efficient orthogonal finetuning via butterfly factorization. In *International Conference on Learning Representations*, 2024.

Pan Lu, Liang Qiu, Jiaqi Chen, Tony Xia, Yizhou Zhao, Wei Zhang, Zhou Yu, Xiaodan Liang, and Song-Chun Zhu. Iconqa: A new benchmark for abstract diagram understanding and visual language reasoning. In *Advances in Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021.

Mao-Lin Luo, Zi-Hao Zhou, Yi-Lin Zhang, Yuanyu Wan, Min-Ling Zhang, and Tong Wei. KeeploRA: Continual learning with residual gradient adaptation. In *International Conference on Learning Representations*, 2026.

Xinyu Ma, Xu Chu, Zhibang Yang, Yang Lin, Xin Gao, and Junfeng Zhao. Parameter efficient quasi-orthogonal fine-tuning via givens rotation. In *International Conference on Machine Learning*, volume 235, pp. 33686–33729, 2024.

Xu Ma, Yitian Zhang, and Yun Fu. Towards unified multimodal large language models: A survey. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 37212–37230, 2026.

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. ChartQA: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the Association for Computational Linguistics*, pp. 2263–2279, 2022.

Minesh Mathew, Dimosthenis Karatzas, and C.V. Jawahar. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 2200–2209, 2021.

Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and C.V. Jawahar. Infographicvqa. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 1697–1706, 2022.

Bryan A. Plummer, Liwei Wang, Chris M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015.

Zeju Qiu, Weiyang Liu, Haiwen Feng, Yuxuan Xue, Yao Feng, Zhen Liu, Dan Zhang, Adrian Weller, and Bernhard Schölkopf. Controlling text-to-image diffusion by orthogonal finetuning. *Advances in Neural Information Processing Systems*, 36:79320–79362, 2023.

Zeju Qiu, Simon Buchholz, Tim Xiao, Maximilian Dax, Bernhard Schölkopf, and Weiyang Liu. Reparameterized llm training via orthogonal equivalence transformation. *Advances in Neural Information Processing Systems*, 38:140775–140821, 2025a.

Zeju Qiu, Weiyang Liu, Adrian Weller, and Bernhard Schölkopf. Orthogonal finetuning made scalable. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 31946–31963, 2025b.

Zeju Qiu, Lixin Liu, Adrian Weller, Han Shi, and Weiyang Liu. POET-x: Memory-efficient LLM training by scaling orthogonal transformation. In *International Conference on Machine Learning*, 2026.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, volume 139, pp. 8748–8763, 2021.

Maryam Rahnemoonfar, Tashnim Chowdhury, Argho Sarkar, Debvrat Varshney, Masoud Yari, and Robin Roberson Murphy. Floodnet: A high resolution aerial imagery dataset for post flood scene understanding. *IEEE Access*, 9:89644–89654, 2021.

Shipeng Wang, Xiaorong Li, Jian Sun, and Zongben Xu. Training networks in null space of feature covariance with self-supervision for incremental learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(4):2563–2580, 2025.

Fei Wu, Jia Hu, Geyong Min, and Shiqiang Wang. Efficient orthogonal fine-tuning with principal subspace adaptation. In *International Conference on Learning Representations*, 2026.

Zhen-Hao Xie, Jun-Tao Tang, Yu-Cheng Shi, Han-Jia Ye, De-Chuan Zhan, and Da-Wei Zhou. SAME: Stabilized mixture-of-experts for multimodal continual instruction tuning. In *International Conference on Machine Learning*, 2026.

Dianzhi Yu, Xinni Zhang, Yankai Chen, Aiwei Liu, Yifei Zhang, Philip S. Yu, and Irwin King. Recent advances of multimodal continual learning: A comprehensive survey. *IEEE Transactions on Neural Networks and Learning Systems*, 37(9):4014–4034, 2026.

Yahan Yu, Duzhen Zhang, Yong Ren, Xuanle Zhao, Xiuyi Chen, and Chenhui Chu. Progressive LoRA for multimodal continual instruction tuning. In *Findings of the Association for Computational Linguistics*, pp. 2779–2796, 2025.

Fanhu Zeng, Fei Zhu, Haiyang Guo, Xu-Yao Zhang, and Cheng-Lin Liu. ModalPrompt: Towards efficient multimodal continual instruction tuning with dual-modality guided prompt. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 12126–12141, 2025.

Xiaoman Zhang, Chaoyi Wu, Ziheng Zhao, Weixiong Lin, Ya Zhang, Yanfeng Wang, and Weidi Xie. Pmc-vqa: Visual instruction tuning for medical visual question answering. *arXiv preprint arXiv:2305.10415*, 2023.

Hengyuan Zhao, Ziqin Wang, Qixin Sun, Kaiyou Song, Yilin Li, Xiaolin Hu, Qingpei Guo, and Si Liu. Llava-cmoe: Towards continual mixture of experts for large vision-language models. *arXiv preprint arXiv:2503.21227*, 2025.

Da-Wei Zhou, Yuanhan Zhang, Yan Wang, Jingyi Ning, Han-Jia Ye, De-Chuan Zhan, and Ziwei Liu. Learning without forgetting for vision-language models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(6):4489–4504, 2025.

A Details of Null-Space-Constrained Orthogonal Adaptation

A.1 Detailed Proof of Proposition 1

Proof. Recall that $\boldsymbol{U}_t \in \mathbb{R}^{d\times m}$ is an orthonormal basis that spans the null space of previous-task features, satisfying

$$\boldsymbol{X}_{<t}\boldsymbol{U}_t = \mathbf{0}, \qquad \boldsymbol{U}_t^\top \boldsymbol{U}_t = \boldsymbol{I}_m.$$

For simplicity, define the orthogonal projectors

$$\mathbf{P}_{\mathcal{N}} = \boldsymbol{U}_t\boldsymbol{U}_t^\top, \qquad \boldsymbol{Q}_t = \boldsymbol{I}_d - \mathbf{P}_{\mathcal{N}}. \tag{9}$$

Then $\mathbf{P}_{\mathcal{N}}$ projects onto $\mathrm{col}(\boldsymbol{U}_t)$ and $\boldsymbol{Q}_t$ projects onto its orthogonal complement. Since $\boldsymbol{U}_t$ has orthonormal columns, we have $\mathbf{P}_{\mathcal{N}}^2 = \mathbf{P}_{\mathcal{N}}$, $\boldsymbol{Q}_t^2 = \boldsymbol{Q}_t$, and $\boldsymbol{Q}_t\boldsymbol{U}_t = \mathbf{0}$.

According to Eq. 3, the transformation is parameterized as

$$\boldsymbol{R}_t = \boldsymbol{I}_d + \boldsymbol{U}_t(\bar{\boldsymbol{S}}_t - \boldsymbol{I}_m)\boldsymbol{U}_t^\top, \tag{10}$$

where $\bar{\boldsymbol{S}}_t$ is an orthogonal matrix, $\bar{\boldsymbol{S}}_t^\top \bar{\boldsymbol{S}}_t = \boldsymbol{I}_m$. The above parameterization can be rewritten as

$$\boldsymbol{R}_t = \boldsymbol{Q}_t + \boldsymbol{U}_t\bar{\boldsymbol{S}}_t\boldsymbol{U}_t^\top. \tag{11}$$

Orthogonality. Using $\boldsymbol{Q}_t\boldsymbol{U}_t = \mathbf{0}$ and $\boldsymbol{U}_t^\top\boldsymbol{Q}_t = \mathbf{0}$, we obtain

$$\begin{aligned}\boldsymbol{R}_t^\top\boldsymbol{R}_t &= \left(\boldsymbol{Q}_t + \boldsymbol{U}_t\bar{\boldsymbol{S}}_t^\top\boldsymbol{U}_t^\top\right)\left(\boldsymbol{Q}_t + \boldsymbol{U}_t\bar{\boldsymbol{S}}_t\boldsymbol{U}_t^\top\right)\\ &= \boldsymbol{Q}_t + \boldsymbol{U}_t\bar{\boldsymbol{S}}_t^\top\bar{\boldsymbol{S}}_t\boldsymbol{U}_t^\top\\ &= \boldsymbol{Q}_t + \boldsymbol{U}_t\boldsymbol{U}_t^\top\\ &= \boldsymbol{I}_d.\end{aligned} \tag{12}$$

Therefore, $\boldsymbol{R}_t$ is orthogonal.

Null-space-constrained update. The induced weight update is

$$\begin{aligned}\Delta\boldsymbol{W}_t &= (\boldsymbol{R}_t - \boldsymbol{I}_d)\boldsymbol{W}_{t-1}\\ &= \boldsymbol{U}_t(\bar{\boldsymbol{S}}_t - \boldsymbol{I}_m)\boldsymbol{U}_t^\top\boldsymbol{W}_{t-1}.\end{aligned} \tag{13}$$

Since the leftmost factor is $\boldsymbol{U}_t$, the columns of $\Delta\boldsymbol{W}_t$ are contained in the subspace spanned by $\boldsymbol{U}_t$, i.e.,

$$\mathrm{col}(\Delta\boldsymbol{W}_t) \subseteq \mathrm{col}(\boldsymbol{U}_t). \tag{14}$$

Because $\boldsymbol{U}_t$ spans the null space of previous-task features, we have

$$\boldsymbol{X}_{<t}\boldsymbol{U}_t = \mathbf{0}. \tag{15}$$

Therefore,

$$\boldsymbol{X}_{<t}\Delta\boldsymbol{W}_t = \boldsymbol{X}_{<t}\boldsymbol{U}_t(\bar{\boldsymbol{S}}_t - \boldsymbol{I}_m)\boldsymbol{U}_t^\top\boldsymbol{W}_{t-1} = \mathbf{0}. \tag{16}$$

Hence, the proposed parameterization simultaneously preserves the orthogonality of $\boldsymbol{R}_t$ and constrains the induced update to the null space of previous-task features. □

A.2 Details of Null-Space Construction

Under our parameter-efficient adaptation setting, the visual encoder and the multimodal projector remain frozen, and NOFT is applied only to the FFN linear layers of the LLM backbone. The null-space construction is performed independently for each adapted FFN linear layer.

For a layer with weight matrix $\boldsymbol{W}_{t-1} \in \mathbb{R}^{d\times n}$, let $\boldsymbol{X}_v \in \mathbb{R}^{N_v\times d}$ and $\boldsymbol{X}_q \in \mathbb{R}^{N_q\times d}$ denote the input feature matrices collected from preceding tasks at visual and textual token positions, respectively. Here, N_v and N_q denote the numbers of corresponding feature vectors, and layer indices are omitted for simplicity. The previous-task feature matrix used in Sec. 4.1 can therefore be written as

$$\boldsymbol{X}_{<t} = \begin{bmatrix}\boldsymbol{X}_v\\ \boldsymbol{X}_q\end{bmatrix}. \tag{17}$$

Accordingly, the null space of $\boldsymbol{X}_{<t}$ corresponds to the common null space of the two modalities:

$$\mathcal{N}(\boldsymbol{X}_{<t}) = \mathcal{N}(\boldsymbol{X}_v) \cap \mathcal{N}(\boldsymbol{X}_q). \tag{18}$$

We define the feature covariance matrices as

$$\boldsymbol{C}_v = \frac{1}{N_v}\boldsymbol{X}_v^\top \boldsymbol{X}_v, \qquad \boldsymbol{C}_q = \frac{1}{N_q}\boldsymbol{X}_q^\top \boldsymbol{X}_q. \tag{19}$$

To estimate the common null space, we first select low-energy directions of $\boldsymbol{C}_v$ and then identify directions within this subspace that also have low energy with respect to $\boldsymbol{C}_q$. Specifically, we perform singular value decomposition (SVD) of $\boldsymbol{C}_v$:

$$\boldsymbol{C}_v = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^\top, \tag{20}$$

where the singular values in $\boldsymbol{\Lambda}$ are arranged in ascending order, $\lambda_1 \le \lambda_2 \le \cdots \le \lambda_d$.

We retain the largest set of lowest-energy directions whose cumulative energy does not exceed a fraction τ of the total:

$$m_v = \max\left\{ k \in \{0, \ldots, d\} : \frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{d} \lambda_i} \le \tau \right\}. \tag{21}$$

The corresponding singular vectors form $\boldsymbol{U}_v = \boldsymbol{U}_{[:,1:m_v]} \in \mathbb{R}^{d \times m_v}$.

We next identify the directions within $\mathrm{col}(\boldsymbol{U}_v)$ that also correspond to low-energy directions of the previous-task textual features. The projected textual covariance matrix is

$$\widetilde{\boldsymbol{C}}_q = \boldsymbol{U}_v^\top \boldsymbol{C}_q \boldsymbol{U}_v = \boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{V}^\top, \tag{22}$$

where the singular values satisfy $\sigma_1 \le \sigma_2 \le \cdots \le \sigma_{m_v}$. Applying the same energy criterion gives

$$m = \max\left\{ k \in \{0, \ldots, m_v\} : \frac{\sum_{i=1}^{k} \sigma_i}{\sum_{i=1}^{m_v} \sigma_i} \le \tau \right\}. \tag{23}$$

Let $\boldsymbol{V}_q = \boldsymbol{V}_{[:,1:m]} \in \mathbb{R}^{m_v \times m}$. The final null-space basis is

$$\boldsymbol{U}_t = \boldsymbol{U}_v \boldsymbol{V}_q, \qquad \boldsymbol{U}_t^\top \boldsymbol{U}_t = \boldsymbol{I}_m. \tag{24}$$

Layer-response perturbation. The two-stage energy criteria provide a quantitative characterization of the estimated null space. Combining the visual and textual selection criteria gives

$$\mathrm{tr}(\boldsymbol{U}_t^\top \boldsymbol{C}_a \boldsymbol{U}_t) \le \tau \ \mathrm{tr}(\boldsymbol{C}_a), \qquad a \in \{v, q\}. \tag{25}$$

For the visual modality, this follows from the first-stage criterion and $\mathrm{col}(\boldsymbol{U}_t) \subseteq \mathrm{col}(\boldsymbol{U}_v)$; for the textual modality, it follows from the second-stage criterion together with $\mathrm{tr}(\widetilde{\boldsymbol{C}}_q) \le \mathrm{tr}(\boldsymbol{C}_q)$.

Since the weight update is restricted to $\mathrm{col}(\boldsymbol{U}_t)$ and $\boldsymbol{U}_t$ has orthonormal columns, $\Delta\boldsymbol{W}_t = \boldsymbol{U}_t\boldsymbol{U}_t^\top \Delta\boldsymbol{W}_t$. The average squared change in layer responses on the collected previous-task features therefore satisfies

$$\begin{aligned} \frac{1}{N_a}\|\boldsymbol{X}_a \Delta\boldsymbol{W}_t\|_F^2 &\le \|\Delta\boldsymbol{W}_t\|_2^2 \,\mathrm{tr}(\boldsymbol{U}_t^\top \boldsymbol{C}_a \boldsymbol{U}_t) \\ &\le \tau \|\Delta\boldsymbol{W}_t\|_2^2 \,\mathrm{tr}(\boldsymbol{C}_a), \qquad a \in \{v, q\}. \end{aligned} \tag{26}$$

This bound shows how restricting weight updates to the selected low-energy subspace helps limit changes in layer responses to previous-task features from both modalities, while the actual change also depends on the weight-update magnitude. When $\boldsymbol{X}_{<t}\boldsymbol{U}_t = \boldsymbol{0}$, the layer responses on these previous-task features are preserved exactly, consistent with Sec. 4.1.

Maintaining previous-task statistics. For each adapted FFN linear layer, we incrementally maintain $\boldsymbol{X}_v^\top \boldsymbol{X}_v$ and $\boldsymbol{X}_q^\top \boldsymbol{X}_q$ together with the corresponding feature counts N_v and N_q, without storing the full previous-task feature matrices. After completing task t, we use the trained model to collect layer input features from all examples in $\mathcal{D}_t$ and update these statistics separately for visual and textual modalities. Dividing the accumulated matrices by their respective counts yields $\boldsymbol{C}_v$ and $\boldsymbol{C}_q$ for subsequent tasks. Maintaining two $d \times d$ matrices and two counts requires $\mathcal{O}(d^2)$ storage per adapted layer for previous-task statistics, independent of the numbers of tasks and processed tokens.

A.3 Overall Training Procedure of NOFT

Algorithm 1 summarizes the overall training procedure of NOFT. For each task, we construct the estimated null-space basis $\boldsymbol{U}_t$ from previous-task feature statistics and use the current-task gradient at initialization to determine the plasticity-aware permutation $\boldsymbol{P}_t$. We then optimize $\boldsymbol{S}_t$ with $\boldsymbol{U}_t$ and $\boldsymbol{P}_t$ fixed and merge the resulting transformation into the model. These operations apply to each adapted FFN linear layer in the LLM backbone. For the first task, we set $\boldsymbol{U}_1 = \boldsymbol{I}_d$, where d is the layer input dimension, to perform orthogonal transformation. The previous-task feature statistics are maintained as described in Appendix A.2.

Algorithm 1 NOFT for Continual Instruction Tuning

Input: Pre-trained weights $\boldsymbol{W}_0$, task sequence $\{\mathcal{D}_t\}_{t=1}^{T}$, number of blocks b
Output: Updated weights $\boldsymbol{W}_T$

```
for task t = 1 to T do
    if t = 1 then
        Set U_t = I_d without null-space constraint
    else
        Construct U_t from previous-task feature statistics via Eq. (24)
    end if
    Initialize S_t = I_m as a block-diagonal orthogonal matrix
    Estimate the current-task gradient from a subset of D_t
    Determine P_t via Eq. (6)
    Optimize S_t on D_t with fixed U_t and P_t via Eq. (3)
    Merge the transformation by W_t = R_t W_{t-1}
    Update previous-task feature statistics from D_t using W_t
end for
```

B Details of Plasticity-aware Permutation

B.1 Derivation of the Orthogonality-Constrained Gradient

We provide the derivation of the orthogonality-constrained gradient used for task-specific permutation selection in Sec. 4.2. Before constructing the permutation, the transformation is written as

$$\boldsymbol{R}_t = \boldsymbol{I}_d + \boldsymbol{U}_t \left(\boldsymbol{S}_t - \boldsymbol{I}_m\right) \boldsymbol{U}_t^\top, \tag{27}$$

where $\boldsymbol{S}_t$ is initialized as $\boldsymbol{I}_m$.

Given the forward transformation $\boldsymbol{h} = \boldsymbol{x}\boldsymbol{R}_t\boldsymbol{W}_{t-1}$, let $\boldsymbol{G}_h = \partial\mathcal{L}/\partial\boldsymbol{h}$. The Euclidean gradient with respect to $\boldsymbol{R}_t$ is

$$\boldsymbol{G}_R = \frac{\partial\mathcal{L}}{\partial\boldsymbol{R}_t} = \boldsymbol{x}^\top \boldsymbol{G}_h \boldsymbol{W}_{t-1}^\top. \tag{28}$$

From $d\boldsymbol{R}_t = \boldsymbol{U}_t(d\boldsymbol{S}_t)\boldsymbol{U}_t^\top$, we have $d\mathcal{L} = \operatorname{tr}\left(\boldsymbol{G}_R^\top d\boldsymbol{R}_t\right) = \operatorname{tr}\left[\left(\boldsymbol{U}_t^\top \boldsymbol{G}_R \boldsymbol{U}_t\right)^\top d\boldsymbol{S}_t\right]$. Therefore,

$$\boldsymbol{G}_S = \frac{\partial\mathcal{L}}{\partial\boldsymbol{S}_t} = \boldsymbol{U}_t^\top \boldsymbol{G}_R \boldsymbol{U}_t = \boldsymbol{U}_t^\top \boldsymbol{x}^\top \boldsymbol{G}_h \boldsymbol{W}_{t-1}^\top \boldsymbol{U}_t. \tag{29}$$

Since $\boldsymbol{S}_t$ is orthogonal, $\boldsymbol{S}_t^\top \boldsymbol{S}_t = \boldsymbol{I}_m$. At task initialization, $\boldsymbol{S}_t = \boldsymbol{I}_m$. Taking the differential of the orthogonality constraint at this point gives

$$d\boldsymbol{S}_t + d\boldsymbol{S}_t^\top = \boldsymbol{0}, \tag{30}$$

showing that the admissible first-order directions are skew-symmetric. Projecting the Euclidean gradient onto this space therefore gives

$$\boldsymbol{G}_t = \operatorname{skew}\left(\boldsymbol{G}_S\right) = \frac{1}{2}\left(\boldsymbol{G}_S - \boldsymbol{G}_S^\top\right). \tag{31}$$

NOFT estimates $\boldsymbol{G}_t$ once using a sampled subset of the current-task training data $\mathcal{D}_t$ and uses it to determine the task-specific permutation $\boldsymbol{P}_t$.

Algorithm 2 Greedy Approximation for Plasticity-Aware Permutation

Input: Orthogonality-constrained gradient $\boldsymbol{G}_t$, block size r
Output: Permutation matrix $\boldsymbol{P}_t$

$\mathcal{R} \leftarrow \{1, \dots, m\}$
Initialize an empty coordinate ordering
while $\mathcal{R} \neq \emptyset$ **do**
  if $|\mathcal{R}| = r$ **then**
    $\mathcal{B} \leftarrow \mathcal{R}$
  else
    $(i, j) \leftarrow \underset{i<j,\ i,j\in\mathcal{R}}{\arg\max} \left[(\boldsymbol{G}_t)_{ij}^2 + (\boldsymbol{G}_t)_{ji}^2\right]$ ▷ *Initialize group*
    $\mathcal{B} \leftarrow \{i, j\}$
    while $|\mathcal{B}| < r$ **do**
      $v \leftarrow \underset{v\in\mathcal{R}\setminus\mathcal{B}}{\arg\max} \sum_{u\in\mathcal{B}} (\boldsymbol{G}_t)_{uv}^2$ ▷ *Expand group*
      $\mathcal{B} \leftarrow \mathcal{B} \cup \{v\}$
    end while
  end if
  Append the coordinates in $\mathcal{B}$ to the coordinate ordering
  $\mathcal{R} \leftarrow \mathcal{R} \setminus \mathcal{B}$
end while
Construct $\boldsymbol{P}_t$ from the resulting coordinate ordering
return $\boldsymbol{P}_t$

B.2 Cayley Parameterization

Following Qiu et al. (2023), we use the Cayley parameterization to maintain the orthogonality of each trainable block. Specifically, an orthogonal matrix $\boldsymbol{S} \in \mathbb{R}^{r\times r}$ is parameterized as

$$\boldsymbol{S} = (\boldsymbol{I}_r + \boldsymbol{Q})\,(\boldsymbol{I}_r - \boldsymbol{Q})^{-1}, \qquad \boldsymbol{Q}^\top = -\boldsymbol{Q}, \tag{32}$$

where $\boldsymbol{Q}$ is skew-symmetric. This parameterization guarantees $\boldsymbol{S}^\top \boldsymbol{S} = \boldsymbol{I}_r$. Initializing $\boldsymbol{Q} = \boldsymbol{0}$ gives $\boldsymbol{S} = \boldsymbol{I}_r$, such that NOFT starts each task from the current model weights.

B.3 Approximate Solution for Plasticity-Aware Permutation

We provide additional details on the approximate solution to the permutation objective in Eq. 6. Recall that the block-diagonal structure contains b blocks of size $r \times r$, with $m = br$. A permutation $\boldsymbol{P}$ determines how the m coordinates are assigned to these blocks. Equivalently, it partitions the coordinates into b disjoint groups $\{\mathcal{B}_1, \dots, \mathcal{B}_b\}$, each of size r, such that coordinates in the same group are assigned to the same diagonal block. Accordingly, Eq. 6 can be written as

$$\sum_{k=1}^{b} \sum_{i,j\in\mathcal{B}_k} (\boldsymbol{G}_t)_{ij}^2.$$

Thus, the permutation objective favors grouping coordinates with large pairwise gradient components into the same block. Since exhaustively searching over the $m!$ possible permutations is impractical, we construct these groups using a greedy approximation.

Greedy Approximation. Let $\mathcal{R}$ denote the set of coordinates not yet assigned to a completed group. To initialize a new group, we select the pair of coordinates in $\mathcal{R}$ with the largest pairwise contribution:

$$(i^*, j^*) = \underset{i<j,\ i,j\in\mathcal{R}}{\arg\max} \left[(\boldsymbol{G}_t)_{ij}^2 + (\boldsymbol{G}_t)_{ji}^2\right].$$

The selected pair forms the initial group $\mathcal{B} = \{i^*, j^*\}$. The group is then expanded one coordinate at a time. For a candidate $v \in \mathcal{R} \setminus \mathcal{B}$, assigning v to the current group adds the pairwise terms between v and the coordinates already in $\mathcal{B}$. Therefore, the next coordinate is selected according to

$$v^* = \underset{v\in\mathcal{R}\setminus\mathcal{B}}{\arg\max} \sum_{u\in\mathcal{B}} \left[(\boldsymbol{G}_t)_{uv}^2 + (\boldsymbol{G}_t)_{vu}^2\right]. \tag{33}$$

Since $\boldsymbol{G}_t$ is skew-symmetric, $(\boldsymbol{G}_t)_{uv} = -(\boldsymbol{G}_t)_{vu}$, and hence $(\boldsymbol{G}_t)^2_{uv} = (\boldsymbol{G}_t)^2_{vu}$. The objective in Eq. 33 is equivalent to

$$v^* = \arg\max_{v \in \mathcal{R} \setminus \mathcal{B}} \sum_{u \in \mathcal{B}} (\boldsymbol{G}_t)^2_{uv}. \tag{34}$$

Therefore, each greedy step selects the remaining coordinate that yields the largest immediate increase in the objective for the current group.

The expansion continues until $|\mathcal{B}| = r$. The completed group is then removed from $\mathcal{R}$, and a new group is initialized from the remaining coordinates using the same procedure. Repeating this process yields b disjoint groups of size r. These groups determine the coordinate assignment represented by $\boldsymbol{P}_t$. Specifically, the coordinates belonging to each group are placed in the positions associated with one diagonal block. The ordering within each group does not affect the objective and can therefore be chosen arbitrarily.

Algorithm 2 summarizes the complete greedy construction. The resulting $\boldsymbol{P}_t$ is computed once before optimizing the task-specific block matrices and remains fixed throughout adaptation to task t.

Computational Overhead. The approximation is performed only once at the beginning of each task. As described in Sec. 4.2, $\boldsymbol{G}_t$ is estimated from a sampled subset of the current-task data and is used to determine the task-specific permutation. The greedy procedure constructs the block assignment from this fixed gradient matrix, after which $\boldsymbol{P}_t$ remains fixed throughout subsequent optimization. Training therefore updates only the block matrices in $\boldsymbol{S}_t$ and does not repeatedly solve the discrete permutation problem. Consequently, the additional search introduced by plasticity-aware permutation is confined to task initialization and introduces no combinatorial search during subsequent training iterations.

C Additional Experimental Details

C.1 The impact of Different Task Orders

To evaluate the sensitivity of NOFT to task ordering, we further compare SAME and NOFT on UCIT under three different task sequences. The Original Order follows the sequence used in the main experiments, while Order 2 and Order 3 adopt two alternative sequences. The results are reported in Tables 5–7. NOFT maintains relatively stable performance across both individual tasks and the overall benchmark, indicating lower sensitivity to task ordering compared with SAME. These results suggest that the proposed null-space-constrained orthogonal update and plasticity-aware permutation help maintain stable adaptation under varying task sequences.

Table 5: Per-task final performance on UCIT under the Original Order: ImageNet-R → ArxivQA → Vizcap → IconQA → CLEVR → Flickr30k. The best result in each column is highlighted in **bold**.

Method	ImageNet-R	ArxivQA	Vizcap	IconQA	CLEVR	Flickr30k	Avg.
SAME (Xie et al., 2026)	86.23	92.57	48.41	79.97	72.17	**57.75**	72.85
NOFT (ours)	**91.03**	**93.53**	**54.77**	**80.70**	**73.80**	57.69	**75.25**

Table 6: Per-task final performance on UCIT under Order 2: Flickr30k → Vizcap → CLEVR → ImageNet-R → IconQA → ArxivQA. The task columns follow Original Order for comparison.

Method	ImageNet-R	ArxivQA	Vizcap	IconQA	CLEVR	Flickr30k	Avg.
SAME (Xie et al., 2026)	90.37	**93.67**	48.09	80.90	**72.73**	51.50	72.88
NOFT (ours)	**90.83**	93.00	**58.18**	**81.57**	72.03	**57.40**	**75.50**

Table 7: Per-task final performance on UCIT under Order 3: IconQA → ImageNet-R → Vizcap → CLEVR → ArxivQA → Flickr30k. The task columns follow Original Order for comparison.

Method	ImageNet-R	ArxivQA	Vizcap	IconQA	CLEVR	Flickr30k	Avg.
SAME (Xie et al., 2026)	88.20	90.03	47.09	79.30	72.00	57.18	72.30
NOFT (ours)	**90.27**	**92.80**	**56.54**	**79.97**	**75.00**	**58.28**	**75.48**

C.2 PER-TASK FORGETTING ANALYSIS

We report per-task forgetting results in Tables 8 and 9 to examine retention across individual tasks. For each task s, the forgetting score is computed as $F_s = A_{s,T} - A_{s,s}$, where $A_{s,t}$ denotes the performance on task s after learning up to task t. Higher values indicate less forgetting. The final task is excluded only from the per-task ranking discussion and the reported average includes all T tasks.

For TriGap, NOFT achieves the lowest forgetting on six of the nine remaining tasks with LLaVA-v1.5-7B and four with Qwen3-VL-8B, with consistent advantages on PMCVQA, ArxivQA, and Roadside across both backbones. With LLaVA-v1.5-7B, it also ranks second on IconQA and InfoVQA, placing among the top two methods on eight of the nine tasks. With Qwen3-VL-8B, NOFT additionally achieves the lowest forgetting on ChemVQA and ranks second on DocVQA and ChartQA. These results indicate broad task-level benefits, with the specific gains varying across backbones.

For UCIT, NOFT achieves the lowest forgetting on all five remaining tasks with LLaVA-v1.5-7B. Its forgetting gap on IconQA and CLEVR are 1.80% and 0.33%, compared with 4.32% and 3.54% for the best respective baselines. With Qwen3-VL-8B, NOFT performs best on ImageNet-R and Vizcap, while maintaining low forgetting gap of 0.47% and 0.05% on IconQA and CLEVR, respectively. Overall, these results support the effectiveness of NOFT in mitigating forgetting at the individual-task level across both benchmarks and backbones.

Table 8: Per-task forgetting results on the TriGap benchmark. Higher values indicate less forgetting. The best results are highlighted in **bold**, while the second-best results are underlined.

Methods	PMCVQA	DocVQA	ChartQA	IconQA	InfoVQA	ArxivQA	Roadside	ChemVQA	FloodNetVQA	CLEVR	Avg.(↑)
LLaVA-v1.5-7B											
DISCO (Guo et al., 2025b)	-3.32	-3.76	-2.19	-5.49	-1.95	-1.85	-3.17	-3.48	-1.83	-0.00	-2.70
KeepLoRA (Luo et al., 2026)	-2.18	-3.12	-7.48	-7.80	-4.62	-3.22	-4.88	-1.75	-1.50	-0.00	-3.66
SAME (Xie et al., 2026)	-2.70	-1.85	-2.20	**-2.06**	**-0.85**	-0.70	-4.13	**-0.97**	-1.88	-0.00	-1.73
NOFT (ours)	**-0.23**	**-1.00**	**-1.00**	-3.40	-1.04	**-0.27**	**-2.80**	-2.07	**-0.98**	-0.00	**-1.28**
Qwen3-VL-8B											
DISCO (Guo et al., 2025b)	-0.46	-0.99	-1.26	-2.10	-0.81	-0.90	-1.83	-2.40	-1.12	-0.00	-1.19
KeepLoRA (Luo et al., 2026)	-1.90	-1.19	-1.68	-2.13	**-0.52**	-0.90	-2.03	-4.56	**-0.89**	-0.00	-1.58
SAME (Xie et al., 2026)	-0.87	**-0.41**	**-0.64**	**-0.00**	-0.66	-1.53	-4.57	-10.10	-2.12	-0.00	-2.09
NOFT (ours)	**-0.03**	-0.75	-0.68	-2.35	-0.89	**-0.56**	**-0.69**	**-2.26**	-2.19	-0.00	**-1.04**

Table 9: Per-task forgetting results on the UCIT benchmark. Higher values indicate less forgetting. The best results are highlighted in **bold**, while the second-best results are underlined.

Methods	ImageNet-R	ArxivQA	Vizcap	IconQA	CLEVR	Flickr30k	Avg.(↑)
LLaVA-v1.5-7B							
DISCO (Guo et al., 2025b)	-2.37	-1.56	-10.12	-9.96	-10.86	-0.00	-5.81
KeepLoRA (Luo et al., 2026)	-8.96	-0.52	-13.72	-8.89	-5.54	-0.00	-6.27
SAME (Xie et al., 2026)	-1.59	-2.53	-10.51	-4.32	-3.54	-0.00	-3.75
NOFT (ours)	**-0.27**	**-0.10**	**-6.68**	**-1.80**	**-0.33**	-0.00	**-1.53**
Qwen3-VL-8B							
DISCO (Guo et al., 2025b)	-1.00	+0.47	-5.03	**-0.35**	**-0.03**	-0.00	-0.99
KeepLoRA (Luo et al., 2026)	-2.70	-0.17	-2.57	-0.43	-0.20	-0.00	-1.01
SAME (Xie et al., 2026)	-0.97	**+0.87**	-10.09	-0.77	-0.85	-0.00	-1.97
NOFT (ours)	**-0.45**	+0.14	**-0.87**	-0.47	-0.05	-0.00	**-0.28**

C.3 Details of Benchmarks

Table 10: Details of datasets used in the UCIT benchmark.

Dataset	Train	Test	Domain Description
ImageNet-R (Hendrycks et al., 2021)	24000	3000	Object recognition with artistic renditions
ArxivQA (Li et al., 2024)	40000	3000	Academic paper figure understanding
VizWiz (Gurari et al., 2018)	40000	3000	Visual assistance for visually impaired
IconQA (Lu et al., 2021)	30000	3000	Icon comprehension
CLEVR-Math (Lindström & Abraham, 2022)	40000	3000	Mathematical reasoning on synthetic scenes
Flickr30k (Plummer et al., 2015)	40000	3000	Image captioning for real-world photos

Table 11: Details of datasets used in the TriGap benchmark.

Dataset	Train	Test	Domain Description
PMCVQA (Zhang et al., 2023)	40,000	3,000	Medical image analysis and diagnosis
DocVQA (Mathew et al., 2021)	30,000	5,349	Document understanding and text extraction
ChartQA (Masry et al., 2022)	25,000	2,500	Chart and graph reasoning
IconQA (Lu et al., 2021)	10,000	3,000	Icon comprehension
InfographicVQA (Mathew et al., 2022)	20,000	2,801	Infographic information extraction
ArxivQA (Li et al., 2024)	10,000	3,000	Academic paper figure analysis
Roadside (Guan et al., 2026)	30,023	3,000	Autonomous driving scene understanding
ChemVQA (Bhuma et al., 2025)	15,392	3,000	Molecular structure analysis
FloodNetVQA (Rahnemoonfar et al., 2021)	5,898	1,833	Disaster scene assessment
CLEVR (Johnson et al., 2017)	10,000	3,000	Reasoning over synthetic scenes

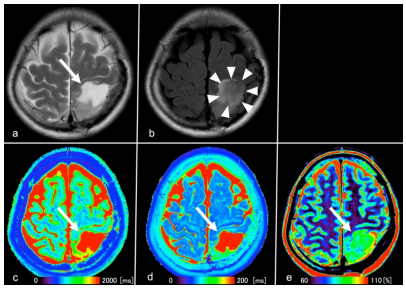
C.4 Qualitative comparison on TriGap Benchmark

Training order: *PMCVQA* → *DocVQA* → *ChartQA* → *IconQA* → *InfoVQA* → *ArxivQA* → *Roadside* → *ChemVQA* → *FloodNetVQA* → *CLEVR*
Evaluation: After learning all tasks.

Task 1: PMCVQA

User: What do the T1 and T2 relaxation time and proton density (PD) maps from SyMRI show in the tumor?

A. T1 relaxation time prolongation and decreased proton density
B. T1 relaxation time prolongation and increased proton density
C. T2 relaxation time prolongation and decreased proton density
D. T2 relaxation time prolongation and increased proton density

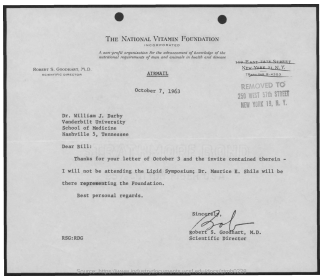


Ground Truth: B

DISCO: B ✓
KeepLoRA: D ✗
SAME: A ✗
NOFT (ours): B ✓

Task 2: DocVQA

User: Who will be attending the Lipid symposium, representing the national vitamin foundation?

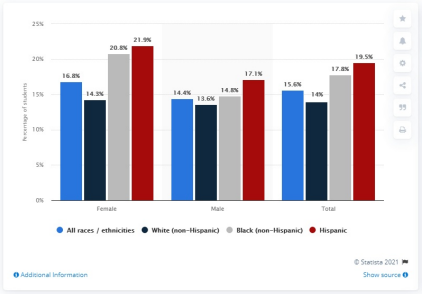


Ground Truth: Dr. Maurice E. Shils

DISCO: Dr. William J. Darby ✗
KeepLoRA: Dr. William Shank ✗
SAME: Dr. William S. Bills ✗
NOFT (ours): Dr. Maurice E. Shils ✓

Figure 4: Qualitative comparison on TriGap using LLaVA-v1.5-7B after learning all tasks (Part 1).

Task 3: ChartQA



User: Which gender had the highest obesity rates between 2016 and 2017?

Ground Truth: Female

DISCO: female ✓
KeepLoRA: black ✗
SAME: Hispanic ✗
NOFT (ours): Female ✓

Task 4: IconQA

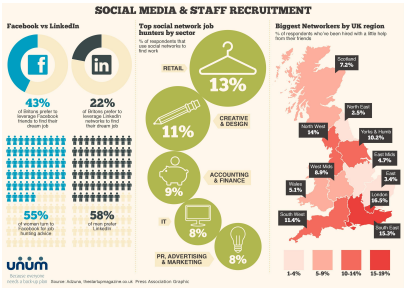


User: What time is shown? Answer by typing a time word, not a number. It is (_) past eight.

Ground Truth: half

DISCO: 6 ✗
KeepLoRA: quarter ✗
SAME: half ✓
NOFT (ours): half ✓

Task 5: InfoVQA

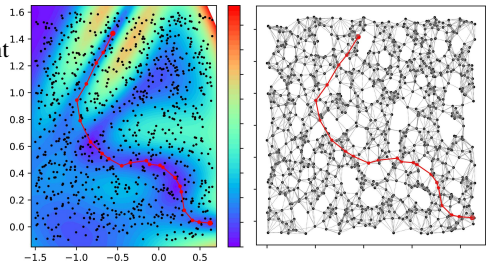


User: What percentage of Britons do not prefer to leverage Facebook friends to find their dream job according to the survey?

Ground Truth: 57%

DISCO: 20 ✗
KeepLoRA: 50% ✗
SAME: 15% ✗
NOFT (ours): 55% ✗

Task 6: ArxivQA



User: How can the distribution of the black dots in the left figure be described?

A. Randomly distributed without any clear pattern.
B. Concentrated along the red line, indicating a critical point
C. Uniformly distributed throughout the space.
D. Showing a gradient of concentration from left to right.

Rationale: The black dots in the left figure are concentrated along the red line, which ...

Ground Truth: B

DISCO: B ✓
KeepLoRA: B ✓
SAME: B ✓
NOFT (ours): B ✓

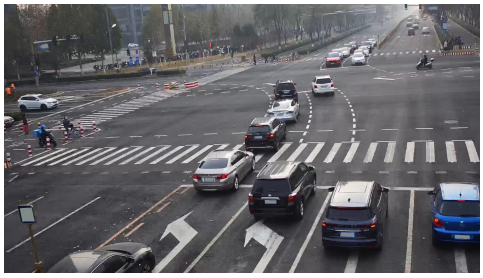
Figure 5: Qualitative comparison on TriGap using LLaVA-v1.5-7B after learning all tasks (Part 2).

Task 7: Roadside

User: Which lane is the blue car driving in?

Ground Truth: Rightmost lane

DISCO: rightmost lane ✓
KeepLoRA: Right lane ✗
SAME: rightmost lane ✓
NOFT (ours): rightmost lane ✓

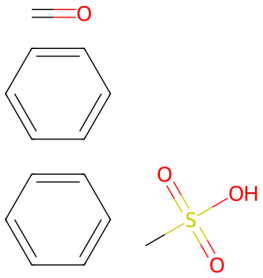


Task 8: ChemVQA

User: How many total oxygen atoms are present?

Ground Truth: 4

DISCO: 4 ✓
KeepLoRA: 2 ✗
SAME: 3 ✗
NOFT (ours): 4 ✓



Task 9: FloodNetVQA

User: How many buildings are flooded in this image?

Ground Truth: 14

DISCO: 10 ✗
KeepLoRA: 10 ✗
SAME: 14 ✓
NOFT (ours): 14 ✓



Task 10: CLEVR

User: Subtract all small yellow rubber cylinders. Subtract all large red metallic things. How many objects are left?

Ground Truth: 8

DISCO: 8 ✓
KeepLoRA: 7 ✗
SAME: 7 ✗
NOFT (ours): 8 ✓

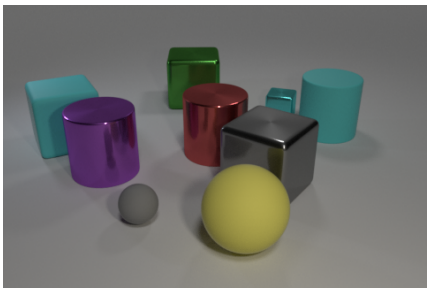


Figure 6: Qualitative comparison on TriGap using LLaVA-v1.5-7B after learning all tasks (Part 3).