Spectral Rank Calibration for Continual LoRA Merging in Multimodal Large Language Models

Chenrui Wu[1,2] **Haishuai Wang**[1*] **Jiajun Bu**[1] **Jiangchuan Liu**[2]
[1] Zhejiang Key Lab of Accessible Perception and Intelligent Systems,
College of Computer Science, Zhejiang University
[2] School of Computing Science, Simon Fraser University
{chenruiwu,haishuai.wang,bjj}@zju.edu.cn
jcliu@sfu.ca

Abstract

Model merging enables the integration of multiple expert models with different capabilities into a unified model. In practical deployments, new expert models are often continuously updated and arriving. Meanwhile, due to the training and communication overhead of large language models (LLMs) and multimodal large language models (MLLMs), Low-Rank Adaptation (LoRA) is widely used to adapt large models to specific domains due to its efficiency. This scenario brings a new challenge: continual LoRA merging of deployed MLLMs to load new task capabilities. Naive merging faces catastrophic forgetting and parameter conflicts. We first observe the distinct features of the LoRA direction and layers in MLLM LoRA tuning. Based on this, we propose our Spectral Rank Calibration Merging (SRC-Merging), a data-free and train-free continual LoRA merging method tailored for MLLMs. SRC-Merging regards the deployed LoRA as a compressed spectral memory and performs calibrated rank-level merging for each incoming adapter. It aims to preserve accumulated multimodal knowledge while avoiding over-suppression of new vision-language task updates during continual merging. Extensive experiments on multiple multimodal tasks have demonstrated that our method significantly outperforms state-of-the-art merging methods.

1 Introduction

In recent years, significant progress has been achieved in Multimodal Large Language Models (MLLMs) [1, 2, 3, 4], which integrate visual encoders with large language models (LLMs) and have demonstrated strong performance across a wide range of multimodal tasks, including autonomous driving [5], visual reasoning [6], visual question answering [7] and vision language action [8].

While such foundation models offer powerful generalization capabilities, targeted fine-tuning or post-training is still necessary for specific tasks. However, traditional full fine-tuning requires large computing costs and memory occupation. Therefore, model merging [9, 10, 11, 12, 13, 14, 15, 16] emerges as an important technique for combining multiple task and domain fine-tuned models into a unified single model, reducing both retraining and deployment costs.

Existing model merging methods primarily focus on conventional models [9, 17] or LLMs [12, 18, 19], but merging in multimodal tasks has not been fully explored, as it differs from single-modal language tasks. Recent efforts have explored this aspect to some extent. AdaMMS [20] aims to fuse model-heterogeneous MLLMs. UQ-Merge [21] evaluates uncertainty to rank the candidate MLLMs by vision and language inputs. Regarding full-parameter aggregation in MLLMs, existing methods

* Corresponding Author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

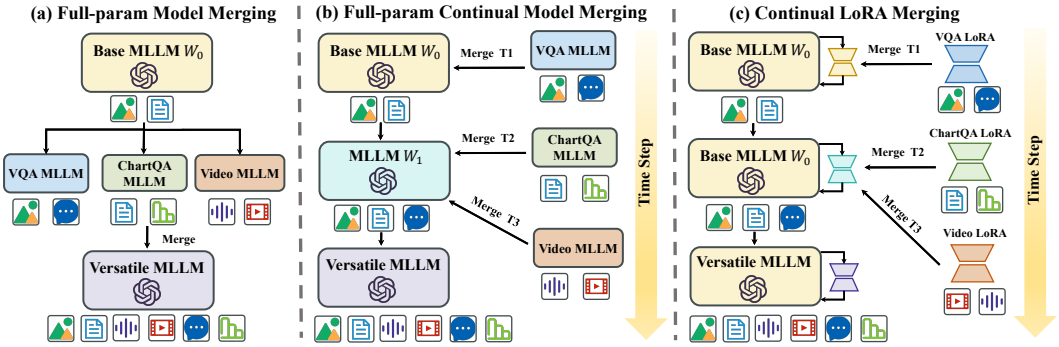


Figure 1: **Illustration of continual LoRA merging.** (a) shows the general model merging with the whole model, requiring all expert models to be prepared. It faces the overhead of memory or extra computing on multiple large models, which is not flexible. (b) represents a continual way to merge expert models. New knowledge can meet new needs over time by aggregating new models. The communication overhead poses a challenge. (c) The continual LoRA merging provides a flexible way for large models to inject new knowledge.

still require significant resource consumption. As a more efficient and low-cost method, parameter-efficient fine-tuning (PEFT) [22, 23], represented by Low-Rank Adaptation (LoRA) [24], has also attracted attention for its application in merging large models [25, 26, 27, 28, 29]. OptMerge [30] propose Adam optimizer, low-rank approximation truncate to merge LoRA fine-tuned MLLM. RobustMerge [31] investigates directions of singular values of LoRA, and merges LoRA with pruning, rescaling, and normalization for MLLM.

However, vanilla LoRA merging methods are tailored for general LLMs, and some [28, 32, 25] need to control the training process beyond typical model merging with existing checkpoints. In real-world practice, a deployed MLLM will also encounter newly fine-tuned LoRA appearing sequentially, as users will gradually raise demands for new functional tasks over time [33, 34, 35, 36, 37]. We describe this scenario in Figure 1, while severe parameter interference and conflicts between tasks, as well as catastrophic forgetting of previously learned tasks, can all negatively impact the performance of continual LoRA merging. Thus, we pose a key challenge that:

How to absorb streamed new LoRA capability for a deployed MLLM?

To address this challenge, we examine how new LoRA updates interact with the deployed LoRA. Within the deployed LoRA's subspace, we observe that new updates can both rescale existing directions and form new combinations of them, with the balance varying across tasks. This motivates a merging strategy that treats these components separately while maintaining a fixed rank.

Therefore, we propose Spectral Rank Calibration Merging (SRC-Merging), a data-free and train-free continual LoRA merging framework for MLLMs. SRC-Merging treats the deployed LoRA as a compressed spectral memory and performs calibrated rank-level merging for each incoming adapter. Specifically, our method first decomposes the deployed LoRA into canonical spectral rank atoms and projects the incoming LoRA into this spectral space to characterize how new updates interact with accumulated knowledge. It then applies structured spectral calibration to separate different types of incoming information, including calibration signals for existing directions and task-specific components that cannot be captured by simple same-direction alignment. Finally, a dual-view spectral recomposition strategy adaptively balances the incoming task information with the calibrated historical update from both adapter-intrinsic and temporal-aware perspectives. Through this design, SRC-Merging aims to preserve accumulated multimodal knowledge while avoiding over-suppression of new knowledge during continual fusion. We summarize our contributions as follows:

- We study the problem of continual LoRA merging for deployed MLLMs, where task-specific LoRA adapters arrive sequentially.
- We propose SRC-Merging, a novel continual LoRA merging framework that calibrates the deployed LoRA subspace and recomposes incoming task information through dual-view spectral weighting.

- Extensive experiments on multiple multimodal tasks show that SRC-Merging improves continual LoRA merging for MLLMs by better balancing historical knowledge preservation and new task absorption.

2 Related Work

2.1 MLLM Model Merging

Recently, several studies have explored MLLM model merging, though their focuses vary. UQ-Merge [21] leverages unlabeled data to quantify the uncertainty of MLLMs, and then iteratively merges models in order. OptMerge [30] improves the merged model by optimizing a loss function defined on interactions among task vectors, while using low-rank approximations to eliminate redundant noise. They tailor the method for the full model and LoRA, respectively. AdaMMS [20] considers heterogeneous MLLM merging by an unsupervised hyperparameter selection method. RobustMerge [31] focuses on the LoRA merging by pruning, rescaling, and normalization.

2.2 Continual Model Merging

Sequence model availability is a common scenario in the real world as well as general continual data learning. TIME [33] applies a structured parameter selection to mitigate interference and forgetting. MINGLE [34] focuses on a test-time continual scenario with incoming data and models. They utilize a MoE architecture with zero-space-constraint gating and an adaptive relaxation strategy to handle test-time perturbations. DOP [35] employs dual projections to balance prior and new knowledge, and multi-gradient descent to get a Pareto-optimal solution. OPCM [36] adopts orthogonal projection of the weight matrix and an adaptive scaling mechanism to handle the arrived models. NUFILT [37] projects new task vectors to the old null space, then trains a LoRA to supply new tasks, and merges these updates back into the backbone.

2.3 Parameter-Efficient Model Merging

LoRA [24] has emerged as a parameter-efficient technique for adapting LLMs to downstream tasks. Recently, model merging has been extended to LoRA-based adaptations. OSRM [25] focuses on data space conflicts and orthogonally projects the LoRA $\mathbf{A}$ matrix for initialization. KnOTS [26] applies SVD to jointly align task updates from multiple LoRA models into a shared subspace. LoRA-LEGO [27] treats each rank of LoRA as an ability block, separates them, and clusters to form a new LoRA. NSC [32] exploits the null-space compression property of the LoRA down-projection matrix as an optimization signal for weighting task merges. SLAO [28] maintains a single shared LoRA across all tasks by sequentially merging new task updates, which is a continual training method. K-Merge [29] continually merges each incoming LoRA with its most similar stored K adapter under a fixed storage budget.

3 Preliminaries and Problem Definition

3.1 Low-Rank Adaptation

Low-Rank Adaptation (LoRA) [24] is a parameter-efficient fine-tuning method that adapts a pre-trained model by restricting parameter updates to a low-rank subspace. Given a frozen weight matrix $W_0 \in \mathbb{R}^{m\times n}$, LoRA parameterizes the update as a low-rank decomposition

$$\Delta W = \mathbf{B}\mathbf{A}, \tag{1}$$

where $\mathbf{B} \in \mathbb{R}^{m\times r}$ and $\mathbf{A} \in \mathbb{R}^{r\times n}$ are learnable matrices with rank $r \ll \min(m, n)$. During fine-tuning, the original weight W_0 remains frozen, and only the low-rank factors $\mathbf{B}$ and $\mathbf{A}$ are optimized. The adapted weight is therefore given by

$$W = W_0 + \Delta W. \tag{2}$$

3.2 Problem Definition

We consider the problem of continual LoRA merging, where task-specific LoRA adapters arrive sequentially without access to the original training data. Let $W_0 \in \mathbb{R}^{m\times n}$ denote a frozen weight

matrix of a pre-trained base model. For clarity, we present the formulation for a single weight matrix, while the same formulation applies independently to all LoRA-augmented layers in an MLLM.

For each task $\mathcal{T}_t$, the corresponding LoRA adapter is denoted by $(\mathbf{B}_t, \mathbf{A}_t)$ and produces the task-specific update

$$\Delta W_t = \mathbf{B}_t \mathbf{A}_t, \tag{3}$$

where $\mathbf{B}_t \in \mathbb{R}^{m \times r}$ and $\mathbf{A}_t \in \mathbb{R}^{r \times n}$, with $r \ll \min(m, n)$.

At each step t, the task adapter $(\mathbf{B}_t, \mathbf{A}_t)$ becomes available. Previously received adapters are not stored or revisited, and the original training data for previous tasks are unavailable. Instead, the incoming adapter is merged only with the current deployed LoRA from the previous step. We denote the accumulated adapter after merging the first t adapters by $(\bar{\mathbf{B}}_t, \bar{\mathbf{A}}_t)$, which produces the accumulated update:

$$\Delta \bar{W}_t = \bar{\mathbf{B}}_t \bar{\mathbf{A}}_t. \tag{4}$$

Under the strict streaming setting, the client has access only to the deployed LoRA and the incoming LoRA:

$$\left\{(\bar{\mathbf{B}}_{t-1}, \bar{\mathbf{A}}_{t-1}), (\mathbf{B}_t, \mathbf{A}_t)\right\}, \tag{5}$$

but not to previous LoRA $\{(\mathbf{B}_i, \mathbf{A}_i)\}_{i<t}$ or their original training data.

The objective at step t is to compute the merged LoRA $(\bar{\mathbf{B}}_t, \bar{\mathbf{A}}_t)$ by combining the deployed LoRA $(\bar{\mathbf{B}}_{t-1}, \bar{\mathbf{A}}_{t-1})$ and the current incoming LoRA $(\mathbf{B}_t, \mathbf{A}_t)$ in an online manner. The resulting merged LoRA is constrained to have the same LoRA rank r:

$$\Delta \bar{W}_t = \bar{\mathbf{B}}_t \bar{\mathbf{A}}_t, \tag{6}$$

where $\bar{\mathbf{B}}_t \in \mathbb{R}^{m \times r}$ and $\bar{\mathbf{A}}_t \in \mathbb{R}^{r \times n}$. Throughout the continual merging process, the LoRA rank r is fixed for both incoming and deployed adapters. This fixed-rank constraint prevents the adapter size from growing with the number of tasks.

Accordingly, the effective model weight after step t is defined as:

$$\bar{W}_t = W_0 + \Delta \bar{W}_t, \tag{7}$$

where the base weight W_0 remains fixed throughout the process.

The goal of continual LoRA merging is to update $\Delta \bar{W}_t$ such that the model preserves knowledge encoded in previously merged adapters while effectively incorporating information from the current task adapter, under a strict streaming setting where only the previous accumulated adapter and the current task adapter are accessible at each step.

4 Methodology

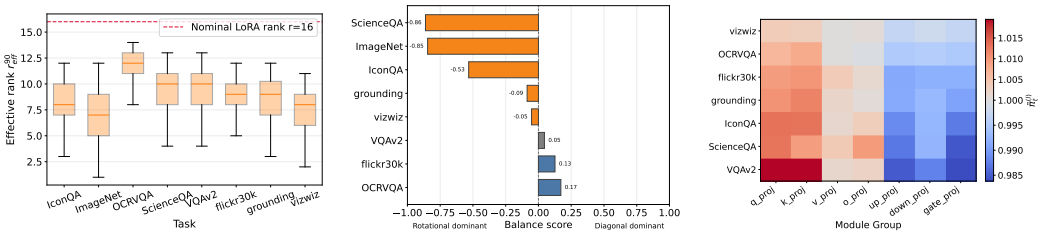
4.1 Observation and Motivation

Many works have improved the LoRA tuning of MLLM in training perspectives [38, 39, 40]. We revisit the fine-tuned LoRA for MLLM [31] and try to investigate its features under a continual merging perspective.

In Figure 2(a), we analyzes the per-block 90% effective rank r_{eff}^{90} of task LoRA adapters. For each LoRA block, we compute the update matrix $\Delta W = \mathbf{BA}$ and obtain its singular values $\{\sigma_i\}_{i=1}^{r}$. The cumulative spectral energy captured by the top-k directions is defined as $E(k) = \sum_{i=1}^{k} \sigma_i^2 / \sum_{i=1}^{r} \sigma_i^2$, and the 90% effective rank is $r_{\text{eff}}^{90} = \min\{k : E(k) \geq 0.9\}$. Although all adapters use the same nominal rank $r = 16$, most LoRA blocks reach 90% energy with fewer directions, indicating substantial spectral redundancy. The task-wise variation of r_{eff}^{90} further suggests that different vision-language tasks require different spectral capacities.

Takeaway 1: *The nominal rank of LoRA is often greater than its effective rank.*

Figure 2(b) examines how a new LoRA interacts with the deployed LoRA. For the overlapped subspace, we summarize each task by a balance score, where positive values indicate that the new update mostly follows existing directions, negative values indicate stronger reorganization among



(a) Effective Rank Distribution (b) Task Subspace (c) Module Plasticity

Figure 2: Observations on fine-tuned LoRA from MLLM in a rank view, merging interactions with deployed LoRA, and layer-wise impacts

existing directions, and values near zero indicate mixed behavior. The figure shows clear task-dependent variation: some tasks are alignment-dominant, some are rotation-dominant, and several lie near the mixed region. This suggests that the overlapped LoRA space is not a uniform shared component; new tasks may reinforce existing abilities and recombine them in different proportions.

Takeaway 2: *General and specialized knowledge are mixed in the LoRA subspace.*

Figure 2(c) analyzes how strongly different MLLM modules need to absorb new task information. For each incoming task and each LoRA module, we project the incoming update onto the spectral subspace of the deployed LoRA and measure the relative energy that is not captured by simple same-direction alignment. This gives a module-wise new-information ratio. To compare modules within the same task, we normalize this ratio by the task-wise module average. The heatmap shows clear heterogeneity across modules: attention projections such as q_proj and k_proj often exhibit higher relative new-information ratios, while MLP-related modules such as up_proj, $down_proj$, and $gate_proj$ are generally lower. This suggests that different MLLM modules absorb incoming vision-language task information to different degrees.

Takeaway 3: *Different modules of MLLM have different levels of plasticity.*

4.2 Overview

In this work, we propose SRC-Merging, a Spectral Rank Calibration framework for continual LoRA merging in MLLM. SRC-Merging consists of three components: Spectral Memory Decomposition, Structured Spectral Calibration, and Dual-view Spectral Recomposition. First, we decompose the deployed LoRA into canonical spectral rank atoms and project the incoming LoRA onto this spectral memory to characterize their interaction. Then, we calibrate different spectral components of the incoming update. Finally, we combine the calibrated historical update and the extracted task-specific component using adapter-intrinsic plasticity and temporal-aware weighting, and project the result back to the original LoRA rank. By explicitly calibrating spectral directions and adaptively balancing residual and rotational task information, SRC-Merging achieves more stable continual merging for MLLMs, preserving old multimodal knowledge while improving the absorption of new knowledge.

4.3 Spectral Memory Decomposition

Previous studies [27, 41, 16] have found that the rank dimension in LLM has functional modularity. To identify general knowledge, specialized knowledge, and redundant or low-energy directions, we can decompose LoRA based on rank. At the t-th merging step, we have the deployed LoRA $\Delta\bar{W}_{t-1}$ and the incoming LoRA ΔW_t. Since the LoRA factorization $\Delta W = \mathbf{BA}$ is not unique, we decompose the deployed LoRA $\Delta\bar{W}_{t-1}$ into spectral rank atoms to get canonical rank representations:

$$\Delta\bar{W}_{t-1} = \bar{\mathbf{U}}_{t-1}\bar{\mathbf{\Sigma}}_{t-1}\bar{\mathbf{V}}_{t-1}^{\top}. \tag{8}$$

Here, $\bar{\mathbf{U}}_{t-1}$ and $\bar{\mathbf{V}}_{t-1}$ denote the left and right singular subspaces of the deployed LoRA. $\bar{\mathbf{\Sigma}}_{t-1}$ contains the corresponding singular values. We regard these singular directions as the spectral memory of previously merged tasks. We then align the incoming LoRA with this spectral memory by projecting ΔW_t onto singular subspaces $\bar{\mathbf{U}}_{t-1}$ and $\bar{\mathbf{V}}_{t-1}$:

$$\mathbf{C}_t = \bar{\mathbf{U}}_{t-1}^{\top}\Delta W_t\bar{\mathbf{V}}_{t-1}, \tag{9}$$

where $\mathbf{C}_t$ means the impact of new LoRA in the old basis's spectral directions. We decompose it into a diagonal component and an off-diagonal component:

$$\mathbf{D}_t = \mathrm{Diag}\left(\mathrm{diag}(\mathbf{C}_t)\right), \tag{10}$$

$$\mathbf{O}_t = \mathbf{C}_t - \mathbf{D}_t. \tag{11}$$

where the diagonal part $\mathbf{D}_t$ measures how the incoming task strengthens or weakens existing spectral directions, while the off-diagonal part $\mathbf{O}_t$ captures the recomposition among accumulated directions. This alignment step separates in-memory calibration signals from potential task-specific recomposition signals. For example, if a certain direction originally corresponds to visual classification ability, and the new task also requires this ability, then the new LoRA will enhance this direction.

4.4 Structured Spectral Calibration

Orthogonal projection has been widely used in continual learning [42, 43] and continual merging [37, 35, 36]. Despite simply projecting new and old subspaces, as previous work does. We aim to calibrate different information parts of the incoming LoRA with spectral decompositions. Here, we obtain three components: diagonal calibration, orthogonal residual, and rotational residual.

Diagonal calibration. This section represents the adjustment of the intensity of the old spectral directions by the new task. we adopt the diagonal signal $\mathbf{D}_t$ to calibrate the old singular values:

$$\bar{\mathbf{\Sigma}}'_{t-1} = \bar{\mathbf{\Sigma}}_{t-1} + \eta_t \mathbf{D}_t, \tag{12}$$

where the coefficient η_t controls the strength of in-memory calibration. Because deployed LoRA is a compressed representation of historical knowledge, directly updating the old spectral basis for a new task could disrupt the information structure of previous tasks. Therefore, we strengthen or weaken the old directions, but avoid easy rotation.

Orthogonal residual. This direction represents information that the old spectral space cannot explain in the new LoRA. These directions may correspond to capabilities specific to new tasks, such as new visual distributions and new cross-modal alignment requirements. Formally, the full projection of the incoming update onto the accumulated spectral memory is

$$\Delta W_t^{prj} = \bar{\mathbf{U}}_{t-1} \mathbf{C}_t \bar{\mathbf{V}}_{t-1}^{\top}. \tag{13}$$

The orthogonal residual, as the remaining component outside the deployed spectral memory, is defined as:

$$\Delta W_t^{\mathrm{ort}} = \Delta W_t - \Delta W_t^{prj}. \tag{14}$$

Rotational residual. The off-diagonal component $\mathbf{O}_t$ in Eq. 11 represents how the incoming LoRA re-composes existing spectral directions. We map it back to the parameter space as the rotational residual:

$$\Delta W_t^{\mathrm{rot}} = \bar{\mathbf{U}}_{t-1} \mathbf{O}_t \bar{\mathbf{V}}_{t-1}^{\top}. \tag{15}$$

This direction is not new orthogonal directions (introducing new knowledge), but reorganizing existing capabilities. For example, OCR-VQA needs to recombine visual text recognition and language response, while VQAv2 might need to combine visual understanding and language generation.

We combine these two residual terms as the task-specific component of the incoming LoRA:

$$\Delta W_t^{\mathrm{new}} = \Delta W_t^{\mathrm{ort}} + \Delta W_t^{\mathrm{rot}}. \tag{16}$$

Compared with using only the orthogonal residual, this formulation keeps both the new directions outside the deployed spectral memory and the re-composition signals inside the deployed spectral memory. Therefore, the incoming LoRA is decomposed into two complementary roles: $\mathbf{D}_t$ calibrates existing spectral ranks, while $\Delta W_t^{\mathrm{new}}$ provides task-specific information to be injected during recomposition.

4.5 Dual-view Spectral Recomposition

After rank calibration, we obtain the calibrated historical spectrum $\bar{\mathbf{\Sigma}}'_{t-1}$ and the task-specific component $\Delta W_t^{\mathrm{new}}$. The final step is to recompose them into a single rank-r LoRA. We adopt two aspects of weight to recompose the LoRA.

Adapter-intrinsic plasticity. We first compute the strength of the current task injection as how much information in ΔW_t cannot be represented by diagonal calibration alone. With Frobenius norm, we obtain:

$$\pi_t = \frac{\|\Delta W_t^{\text{ort}}\|_F + \|\Delta W_t^{\text{rot}}\|_F}{\|\Delta W_t\|_F + \epsilon}, \tag{17}$$

where a larger π_t indicates that the incoming LoRA contains more task-specific residual or recomposition information, and therefore requires stronger plasticity.

Temporal-aware weight. A time factor is used to ensure stability during the continuous merging process and balance the old and new LoRA.

$$\eta_t = \max\left(\frac{1}{\sqrt{t+1}}, \eta_{\min}\right), \tag{18}$$

where $\eta_{\min}$ is a lower bound to add new knowledge. Then we obtain the overall coefficient as:

$$\alpha_t^{\text{new}} = \eta_t \left(\alpha_{\min} + (1 - \alpha_{\min})\pi_t\right), \tag{19}$$

where the lower bound $\alpha_{\min}$ ensures that the incoming task still receives a minimum injection strength even when most of its update overlaps with the deployed spectral memory.

We reconstruct the calibrated historical update as:

$$\Delta W_t^{\text{old}} = \bar{\mathbf{U}}_{t-1} \bar{\mathbf{\Sigma}}'_{t-1} \bar{\mathbf{V}}_{t-1}^{\top}. \tag{20}$$

Then we combine the calibrated historical update and the task-specific component through a normalized recomposition:

$$\Delta W_t^{\text{cal}} = \frac{\Delta W_t^{\text{old}} + \alpha_t^{\text{new}} \Delta W_t^{\text{new}}}{1 + \alpha_t^{\text{new}}}. \tag{21}$$

This recomposition preserves the calibrated historical spectral memory while injecting the residual-rotation information from the incoming LoRA. Finally, since the recomposed update ΔW_t^{cal} may exceed the original LoRA rank, we project it back as:

$$\Delta \bar{W}_t = \text{SVD}_r\left(\Delta W_t^{\text{cal}}\right), \tag{22}$$

where $\text{SVD}_r(\cdot)$ denotes the best rank-r approximation obtained by truncated singular value decomposition. The obtained $\Delta \bar{W}_t$ as well as its corresponding $\mathbf{B}$ and $\mathbf{A}$ are deployed.

5 Experiment

5.1 Experimental setup

Model and Datasets. We evaluate our approach on the pretrained LLaVA-v1.5-7B [2], with Vicuna-v1.5-7B [44] as the LLM backbone, CLIP-ViT-L/14-336 [45] as the vision encoder, following [21, 31]. We adopt eight common multimodal tasks: ScienceQA [46], VizWiz [47], ImageNet [48], VQAv2 [49], IconQA [50], Flickr30k [51], Grounding (REC-COCO) [52], OCR-VQA [53], covering various types of alignment, OCR, perception, and reasoning QA tasks. We also consider the vision-language model (VLM), CLIP-ViT-B-32 [54]. We evaluate it on eight vision tasks: SUN397 [55], Cars [56], RESISC45 [57] EuroSAT [58], SVHN [59], GTSRB [60], MNIST [61], DTD [62].

Baselines. We compare our method SRC-Merging with the following three groups of baselines:(1) General model merging: linear merge [9, 10], Ties-merging [11], DARE [12]. (2) Continual model merging: DOP [35], OPCM [36], NUFLIT [37]. (3)LoRA merging: LoRA-LEGO [27], KnOTS [26], RobustMerge [31], and SLAO [28]. For some baselines are not directly applicable to continual LoRA merging, we provide detailed implementations in the Appendix.

Evaluation Metrics. We evaluate continual LoRA merging using two metrics: average accuracy (ACC) and backward transfer (BWT). Let $\mathcal{T}_1, \ldots, \mathcal{T}_T$ denote the sequence of tasks, and let $a_{t,i}$ be the performance on task $\mathcal{T}_i$ after merging the incoming LoRA of task $\mathcal{T}_t$ into the deployed LoRA. After the final merging step T, ACC measures the average performance over all tasks: $\text{ACC} = \frac{1}{T}\sum_{i=1}^{T} a_{T,i}$. To quantify forgetting, we report BWT, which measures how the final deployed LoRA performs on previous tasks compared with the performance obtained immediately after each task was first merged:

Table 1: **Results in terms of accuracy (%) and backward transfer(%) of LLaVA model on eight datasets.** The best results are in **bold**, and the second-best are underlined.

Methods	ImageNet	VizWiz	OCR-VQA	Flickr30k	Grounding	IconQA	ScienceQA	VQAv2	Average ↑	BWT ↑
Zero-Shot	50.28	35.01	64.98	37.47	36.27	18.96	69.39	67.48	47.48	–
Single-LoRA	96.00	64.56	65.47	59.74	42.23	75.58	87.36	67.60	69.94	–
Linear	51.33	36.52	64.80	39.79	37.78	27.93	73.59	67.33	49.88	-16.48
Ties	50.12	37.16	<u>64.92</u>	40.92	<u>39.07</u>	23.60	71.06	67.87	49.34	-17.55
DARE	51.23	36.58	64.90	39.81	37.88	28.50	73.70	67.36	49.99	-16.38
DOP	53.82	37.90	63.59	42.93	39.57	<u>38.71</u>	<u>78.49</u>	68.36	<u>52.92</u>	-14.47
OPCM	<u>73.01</u>	23.42	56.03	14.58	23.45	20.30	58.83	60.08	41.21	-12.19
NUFILT	53.50	37.37	60.99	42.03	39.68	37.39	78.47	**68.53**	52.24	-16.32
LoRA-LEGO	51.15	36.38	64.73	39.55	37.45	27.57	73.54	67.63	49.75	-16.19
KnOTS	54.95	<u>38.00</u>	60.54	<u>44.70</u>	39.13	32.81	73.42	25.74	46.16	<u>1.69</u>
RobustMerge	51.35	36.49	64.81	40.75	37.88	30.87	73.56	67.40	50.38	-16.47
SLAO	50.65	36.65	64.70	40.12	38.12	24.66	71.51	67.98	49.29	-20.76
SRC-Merging	**82.34**	**55.63**	**65.22**	**49.41**	**40.58**	**56.03**	**82.24**	67.78	**62.40**	**3.24**

$\mathrm{BWT} = \frac{1}{T-1}\sum_{i=1}^{T-1}\left(a_{T,i} - a_{i,i}\right)$. A higher ACC indicates better overall performance of the final deployed LoRA, while a less negative BWT indicates better preservation of previously merged task knowledge during continual merging.

Implement Details. All baselines use the same task order and the same set of prepared LoRA adapters. At step t, each method only has access to the deployed adapter $(\bar{\mathbf{B}}_{t-1}, \bar{\mathbf{A}}_{t-1})$ and the incoming adapter $(\mathbf{B}_t, \mathbf{A}_t)$; previous task adapters and training data are not stored or revisited. All task adapters use LoRA rank $r = 16$, and the merged adapter is always projected back to the same rank $r = 16$. We set the temporal lower bound to $\eta_{\min} = 0.5$. The minimum task-injection strength is set to $\alpha_{\min} = 0.5$, and the numerical stability constant is $\epsilon = 10^{-8}$. All methods are evaluated using the same model checkpoint. We use greedy decoding with temperature 0, top-p 1.0, and batch size 1 for inference. We report the mean of three runs of different orders.

5.2 Main Results

The results on eight multi-modal tasks are shown in Tab. 1. We display the original MLLM performance as Zero-shot, and its fine-tuned LoRA performance as Single-LoRA for reference. Overall, SRC-Merging consistently outperforms different types of merging methods, demonstrating its effectiveness for continual LoRA merging in MLLMs. Compared with standard merging baselines, SRC-Merging improves the average accuracy, yielding gains of 12.52%, 13.06%, over Linear and Ties, respectively. Against continual merging methods, SRC-Merging outperforms DOP and NUFILT by 9.48% and 10.16%. In terms of forgetting, our methods realize stable retention.

In Table 2, CLIP vision tasks are generally easier than MLLM tasks. So the improvements of continual baselines in DOP and NUFILT are more significant. KnOTS has good retention but is unstable in DTD. SRC-Merging is close to Single-LoRA in most tasks but is still below the single-task limit, and its overall performance is reasonable.

5.3 Further Analysis

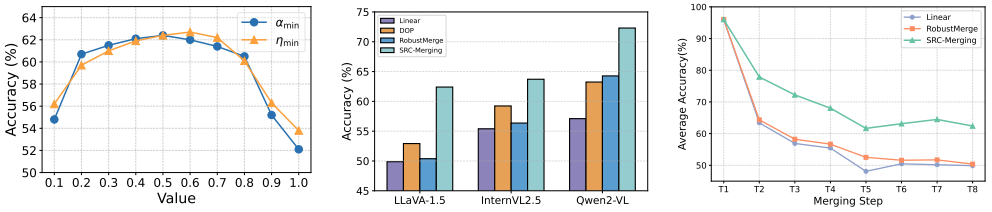
Impact of hyperparameter. Figure 3(a) show the hyperparameter sensitivity of $\alpha_{\min}$ and $\eta_{\min}$. For $\alpha_{\min}$, the accuracy first increases from 54.8% at 0.1 to the best value of 62.4% at 0.5, and then gradually decreases as the value becomes larger. This shows that too small $\alpha_{\min}$ under-suppresses the minimum injection of incoming task information, while a larger $\alpha_{\min}$ weakens the adaptive effect of adapter-intrinsic plasticity and introduces more interference. For $\eta_{\min}$, the accuracy follows a similar trend and reaches the best result of 62.7% at 0.6, indicating that a moderate temporal lower bound is beneficial for balancing historical stability and new-task absorption.

Impact of foundation model. Figure 3(b) compares different merging methods across three MLLM backbones. We apply the same strategy on InternVL2.5-1B-Instruct[63] and Qwen2-VL-7B-Base [64]. SRC-Merging consistently achieves the highest accuracy on these models, showing that its advantage is not limited to a specific base model. But the performance gain in InternVL is 4.52%, which is lower than the other two models.

Merging curve analysis. In Figure 3(c), we observe the average accuracy trajectory of merged tasks for merging steps. Linear merging and RobustMerge exhibit a sharp performance drop after the early steps and remain at a relatively low level as more adapters are merged, indicating severe interference

Table 2: **Results in terms of accuracy (%) and backward transfer(%) of CLIP-ViT-B-32 model on eight datasets.** The best results are in **bold**, and the second-best are underlined.

Methods	SUN397	Cars	RESISC45	EuroSAT	SVHN	GTSRB	MNIST	DTD	Average ↑	BWT ↑
Zero-Shot	65.23	59.60	55.39	48.69	30.54	32.12	47.60	58.72	49.74	–
Single-LoRA	74.25	76.49	60.35	66.10	95.63	91.04	99.25	88.42	81.44	–
Linear	61.42	60.35	54.88	52.67	64.21	59.14	71.80	54.96	59.93	-17.82
Ties	62.18	61.06	55.72	54.80	66.95	62.41	74.36	56.67	61.77	-15.96
DARE	62.04	60.82	56.31	55.12	68.43	63.85	76.12	57.28	62.50	-15.18
DOP	64.71	64.33	57.42	59.68	78.94	72.36	88.15	62.48	68.51	-9.64
OPCM	68.55	55.24	51.80	45.92	70.26	65.19	82.43	47.36	60.84	-7.83
NUFILT	64.05	63.71	57.83	60.45	80.12	74.20	89.04	62.87	69.03	-10.72
LoRA-LEGO	61.28	59.87	55.04	53.36	65.70	60.53	72.91	55.82	60.56	-16.31
KnOTS	65.92	62.18	54.69	57.31	75.44	69.02	84.66	28.35	62.20	0.84
RobustMerge	62.37	61.24	56.02	55.76	69.38	64.77	76.95	57.90	63.05	-14.76
SLAO	60.85	59.43	54.91	53.04	63.58	58.06	70.44	55.18	59.44	-19.37
SRC-Merging	**70.82**	**70.56**	**59.48**	**64.93**	**90.74**	**84.32**	**96.38**	**68.25**	**75.69**	**6.39**



(a) Hyperparameter sensitivity (b) Different model impacts (c) Average ACC trajectory

Figure 3: Further analysis on hyperparameter, applied backbone, and the process of merging.

during continual LoRA merging. In contrast, SRC-Merging maintains a consistently higher trajectory throughout the process and degrades much more slowly

5.4 Ablation Study

We conduct an ablation study to verify the contributions of each component in Table 3. Removing Spectral Memory Decomposition leads to a large performance drop, reducing ACC from 62.40 to 50.82, which shows that directly merging LoRA updates without canonical spectral rank atoms fails to capture the structured rank-level information in deployed LoRA. Without diagonal calibration also degrades both ACC and BWT, indicating that incoming adapters should not only introduce new task information but also calibrate existing spectral directions. The orthogonal residual is especially important for task plasticity: without it, ACC drops significantly to 47.68, although BWT remains relatively high.The ablations on adapter-intrinsic plasticity and temporal-aware weighting verify the importance of the dual-view recomposition strategy.

Table 3: Ablation study of SRC-Merge.

Method	ACC	BWT
SRC-Merge	**62.40**	**3.24**
w/o Spectral Memory Decomposition	50.82	-15.94
w/o Diagonal Calibration	59.31	0.87
w/o Orthogonal Residual	47.68	0.11
w/o Rotational Residual	57.84	1.36
w/o Adapter-intrinsic Plasticity	60.73	-2.28
w/o Temporal-aware Weight	58.16	-1.72
w/o Dual-view Weighting	56.91	-4.85

6 Conclusion

In this work, we aim to tackle the problem of continual LoRA merging for MLLMs, where task-specific LoRA adapters arrive sequentially and without local training. The primary challenge lies in preserving accumulated multimodal knowledge while reliably absorbing new vision-language updates. To this end, we analyzed the spectral rank structure of LoRA updates from MLLMs. Building on the observations, we proposed SRC-Merging, a Spectral Rank Calibration framework that decomposes deployed LoRA into spectral memory, performs structured spectral calibration over incoming updates, and recomposes the merged adapter through dual-view weighting. Experiments across eight vision-language benchmarks demonstrated that SRC-Merging achieves superior average accuracy and backward transfer compared with existing full-model, LoRA, and continual merging baselines.

Acknowledge

This work was supported by the National Key Research and Development Program of China (2025YFC2708700), Zhejiang Province Major Science and Technology Plan Project (2026C01021), the Provincial Key Laboratory Program (2025E10048), and Zhejiang Province Science and Technology Plan Project (2025C01128).

References

[1] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

[2] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, pages 34892–34916, 2023.

[3] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of International Conference on Machine Learning (ICML)*, pages 19730–19742, 2023.

[4] Wanyi Chen, Zihua Zhao, Jiangchao Yao, Ya Zhang, Jiajun Bu, and Haishuai Wang. Multi-modal medical diagnosis via large-small model collaboration. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 30763–30773. IEEE, 2025.

[5] Can Cui, Yunsheng Ma, Xu Cao, Wenqian Ye, Yang Zhou, Kaizhao Liang, Jintai Chen, Juanwu Lu, Zichong Yang, Kuei-Da Liao, et al. A survey on multimodal large language models for autonomous driving. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 958–979, 2024.

[6] Yuhao Dong, Zuyan Liu, Hai-Long Sun, Jingkang Yang, Winston Hu, Yongming Rao, and Ziwei Liu. Insight-v: Exploring long-chain visual reasoning with multimodal large language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pages 9062–9072, 2025.

[7] Dexuan Xu, Yanyuan Chen, Jieyi Wang, Yue Huang, Hanpin Wang, Zhi Jin, Hongxing Wang, Weihua Yue, Jing He, Hang Li, et al. Mlevlm: Improve multi-level progressive capabilities based on multimodal large language model for medical visual question answering. In *Findings of the Association for Computational Linguistics (ACL)*, pages 4977–4997, 2024.

[8] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

[9] Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *Proceedings of International Conference on Machine Learning (ICML)*, pages 23965–23998, 2022.

[10] Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2023.

[11] Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. Ties-merging: Resolving interference when merging models. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pages 7093–7115, 2023.

[12] Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *Proceedings of International Conference on Machine Learning (ICML)*, 2024.

[13] Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. Adamerging: Adaptive model merging for multi-task learning. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2024.

[14] Runxi Cheng, Feng Xiong, Yongxian Wei, Wanyun Zhu, and Chun Yuan. Whoever started the interference should end it: Guiding data-free model merging via task vectors. In *Proceedings of International Conference on Machine Learning (ICML)*, 2025.

[15] Michael S Matena and Colin A Raffel. Merging models with fisher-weighted averaging. *Advances in Neural Information Processing Systems*, 35:17703–17716, 2022.

[16] Yayuan Li, Ze Peng, Jian Zhang, Jintao Guo, Yue Duan, and Yinghuan Shi. When shared knowledge hurts: Spectral over-accumulation in model merging. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 69355–69375, 2026.

[17] Brendan McMahan, Eider Moore, and et al. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017.

[18] Wenlong Deng, Yize Zhao, Vala Vakilian, Minghui Chen, Xiaoxiao Li, and Christos Thrampoulidis. Dare the extreme: Revisiting delta-parameter pruning for fine-tuned models. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2025.

[19] Yuxuan Yao, LIU Shuqi, Zehua Liu, Qintong Li, LIU Mingyang, Xiongwei Han, Zhijiang Guo, Han Wu, and Linqi Song. Activation-guided consensus merging for large language models. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[20] Yiyang Du, Xiaochen Wang, Chi Chen, Jiabo Ye, Yiru Wang, Peng Li, Ming Yan, Ji Zhang, Fei Huang, Zhifang Sui, et al. Adamms: Model merging for heterogeneous multimodal large language models with unsupervised coefficient optimization. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9413–9422, 2025.

[21] Huaizhi Qu, Xinyu Zhao, Jie Peng, Kwonjoon Lee, Behzad Dariush, and Tianlong Chen. `UQ-Merge`: Uncertainty guided multimodal large language model merging. In *Findings of the Association for Computational Linguistics (ACL)*, pages 1401–1417, 2025.

[22] Chenrui Wu, Haishuai Wang, Xiang Zhang, Chengqi Zhang, and Jiajun Bu. Efficient personalized adaptation for physiological signal foundation model. In *Forty-Second International Conference on Machine Learning*, 2025.

[23] Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, et al. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature machine intelligence*, 5(3):220–235, 2023.

[24] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2022.

[25] Haobo Zhang and Jiayu Zhou. Unraveling lora interference: Orthogonal subspaces for robust model merging. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 26459–26472, 2025.

[26] George Stoica, Pratik Ramesh, Boglarka Ecsedi, Leshem Choshen, and Judy Hoffman. Model merging with svd to tie the knots. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2025.

[27] Ziyu Zhao, Tao Shen, Didi Zhu, Zexi Li, Jing Su, Xuwu Wang, Kun Kuang, and Fei Wu. Merging loras like playing lego: Pushing the modularity of lora to extremes through rank-wise clustering. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2025.

[28] Fuli Qiao and Mehrdad Mahdavi. Merge before forget: A single lora continual learning via continual merging. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2026.

[29] Donald Shenaj, Ondrej Bohdal, Taha Ceritli, Mete Ozay, Pietro Zanuttigh, and Umberto Michieli. K-merge: Online continual merging of adapters for on-device large language models. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2026.

[30] Yongxian Wei, Runxi Cheng, Weike Jin, Enneng Yang, Li Shen, Lu Hou, Sinan Du, Chun Yuan, Xiaochun Cao, and Dacheng Tao. Optmerge: Unifying multimodal llm capabilities and modalities via model merging. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2025.

[31] Fanhu Zeng, Haiyang Guo, Fei Zhu, Li Shen, and Hao Tang. Robustmerge: Parameter-efficient model merging for mllms with direction robustness. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[32] Wonyoung Lee, Wooseong Jeong, and Kuk-Jin Yoon. Label-free cross-task lora merging with null-space compression. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

[33] Sebastian Dziadzio, Vishaal Udandarao, Karsten Roth, Ameya Prabhu, Zeynep Akata, Samuel Albanie, and Matthias Bethge. How to merge your multimodal models over time? In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20479–20491, 2025.

[34] Zihuan Qiu, Yi Xu, Chiyuan He, Fanman Meng, Linfeng Xu, Qingbo Wu, and Hongliang Li. Mingle: Mixture of null-space gated low-rank experts for test-time continual model merging. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[35] Enneng Yang, Anke Tang, Li Shen, Guibing Guo, Xingwei Wang, Xiaochun Cao, and Jie Zhang. Continual model merging without data: Dual projections for balancing stability and plasticity. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[36] Anke Tang, Enneng Yang, Li Shen, Yong Luo, Han Hu, Lefei Zhang, Bo Du, and Dacheng Tao. Merging on the fly without retraining: A sequential approach to scalable continual model merging. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[37] Zihuan Qiu, Lei Wang, Yang Cao, Runtong Zhang, Bing Su, Yi Xu, Fanman Meng, Linfeng Xu, Qingbo Wu, and Hongliang Li. Null-space filtering for data-free continual model merging: Preserving transparency, promoting fidelity. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2026.

[38] Jian Liang, Wenke Huang, Guancheng Wan, Qu Yang, and Mang Ye. Lorasculpt: Sculpting lora for harmonizing general and specialized knowledge in multimodal large language models. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 26170–26180, 2025.

[39] Wenke Huang, Jian Liang, Guancheng Wan, Didi Zhu, He Li, Jiawei Shao, Mang Ye, Bo Du, and Dacheng Tao. Be confident: Uncovering overfitting in mllm multi-task tuning. In *Proceedings of International Conference on Machine Learning (ICML)*, 2025.

[40] Duzhen Zhang, Yong Ren, Zhong-Zhi Li, Yahan Yu, Jiahua Dong, Chenxing Li, Zhilong Ji, and Jinfeng Bai. Enhancing multimodal continual instruction tuning with branchlora. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5743–5756, 2025.

[41] Ziyu Zhao, Yixiao Zhou, Xin Yu, Zhi Zhang, Didi Zhu, Tao Shen, Zexi Li, Jinluan Yang, Xuwu Wang, Jing Su, et al. Each rank could be an expert: Single-ranked mixture of experts lora for multi-task learning. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1998–2009, 2026.

[42] Mehrdad Farajtabar, Navid Azizan, Alex Mott, and Ang Li. Orthogonal gradient descent for continual learning. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 3762–3773, 2020.

[43] Gobinda Saha, Isha Garg, and Kaushik Roy. Gradient projection memory for continual learning. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2022.

[44] Wei-Lin Chiang, Zhuohan Li, Ziqing Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2(3):6, 2023.

[45] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *Proceedings of International Conference on Machine Learning (ICML)*, pages 8748–8763, 2021.

[46] Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 2507–2521, 2022.

[47] 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 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3608–3617, 2018.

[48] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009.

[49] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6904–6913, 2017.

[50] 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 *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[51] 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*, pages 2641–2649, 2015.

[52] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. Referitgame: Referring to objects in photographs of natural scenes. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, pages 787–798, 2014.

[53] Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. Ocr-vqa: Visual question answering by reading text in images. In *2019 international conference on document analysis and recognition (ICDAR)*, pages 947–952, 2019.

[54] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *Proceedings of International Conference on Machine Learning (ICML)*, pages 8748–8763, 2021.

[55] Jianxiong Xiao, Krista A Ehinger, James Hays, Antonio Torralba, and Aude Oliva. Sun database: Exploring a large collection of scene categories. *International Journal of Computer Vision*, 119(1):3–22, 2016.

[56] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) workshops*, pages 554–561, 2013.

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

[58] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019.

[59] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin Wu, Andrew Y Ng, et al. Reading digits in natural images with unsupervised feature learning. In *NeurIPS workshop on deep learning and unsupervised feature learning*, volume 2011, page 4, 2011.

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

[61] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 2002.

[62] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3606–3613, 2014.

[63] Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024.

[64] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

A Appendix

A.1 Limitations

SRC-Merging keeps the merged LoRA at a fixed rank throughout the continual merging process for fairness and stabilization. While this design prevents storage and deployment costs from growing with the number of tasks, it may also limit the capacity of the merged adapter when the task stream becomes longer or more diverse. The method relies on SVD calculation. While the LoRA rank is small and the cost is much lower than full-model training or merging, repeated SVD-based operations over many LoRA-augmented layers may still introduce non-negligible overhead in extremely large models or high-frequency online merging scenarios. Compared to test-time continual merging [34], the adaptability of our method needs to be improved when the domain distribution changes.

A.2 Experimental Settings

A.2.1 Datasets and Tasks

ScienceQA [46]: is a multimodal question-answering benchmark covering science topics from elementary and high school curricula. It includes questions with text, images, and explanations, emphasizing scientific reasoning and knowledge understanding.

VizWiz [47]: is a visual question-answering dataset built from images taken by visually impaired users. Its questions are typically open-ended and contain real-world visual and linguistic noise.

ImageNet [48]: we adopt ILSVRC 2012 here, which is a large-scale image classification benchmark built from the ImageNet dataset. It contains over one million natural images across 1,000 object categories and is widely used to evaluate visual recognition models.

VQAv2 [49]: is a visual question-answering benchmark built from real-world images and natural language questions. It balances answer priors by pairing similar images with different answers, emphasizing visual grounding and robust multimodal reasoning.

IconQA [50]: is a diagram-based question-answering benchmark for evaluating visual reasoning on abstract and textbook-style images, with a focus on symbolic understanding and multi-step reasoning.

Flickr30k [51]:is a benchmark for image–text retrieval and captioning, containing 31K real-world images, each paired with multiple human-written captions.

Grounding (REC-COCO) [52]: is a referring expression comprehension benchmark based on COCO images. It requires models to localize the target object described by a natural language expression, emphasizing language-guided visual grounding.

OCR-VQA [53]: is a visual question-answering benchmark focused on images containing text, especially book covers. It requires models to read and understand scene text in images to answer natural language questions.

A.2.2 Baselines

For a fair comparison, all baselines are adapted to the same streaming LoRA setting. We made some necessary modifications, which may cause differences with their original results, as our scenario is not the same as vanilla full-param model merging, multiple LoRA merging, and continual full-param model merging.

Linear merge [9, 10]: the most common merging methods that directly average model parameters. In our LoRA context, linear average [9] equals Task Arithmetic [10], as LoRA itself is a task vector.

Ties-merging [11]: resolves parameter conflicts by trimming small-magnitude updates and electing a consistent sign. We adapt it to LoRA by applying trimming and sign election to the merged LoRA weights.

DARE [12]: randomly drops delta parameters and rescales the remaining updates. We adapt it by applying the drop-and-rescale operation to LoRA weights. We use the DARE+average form with a drop rate of 0.1, where LLMs with larger size parameters adopt larger drop rates.

DOP [35]: aims to balance stability and plasticity by using dual projections to separate old-knowledge-preserving and new-task-adaptive components. We transfer its operations to full models to LoRA.

OPCM [36]: removes components that interfere with the previous merged model through orthogonal projection. It works for the whole model, and we adopt it to LoRA.

NUFLIT [37]: preserves prior knowledge by filtering new task vectors through the null space while promoting task plasticity. We adopt its operations to full models to LoRA, along with its own trained LoRA.

LoRA-LEGO [27]: treats each LoRA rank as a modular ability block, clusters them, and reuses them. We apply it to the two LoRA merging condition.

KnOTS [26]: aligns LoRA updates through SVD and merges them with sparse sign-consistent aggregation. We adapt it by applying pairwise SVD-based alignment between the deployed and incoming LoRA

RobustMerge [31]: improves MLLM LoRA merging by pruning unstable parameters and rescaling LoRA factors. We adapt it by applying magnitude masking, scale estimation from LoRA-A, and weighted reduction to the deployed and incoming LoRA weights at each step.

SLAO [28]: maintains a single continual LoRA by asymmetrically updating LoRA factors during training. We use it to checkpoint-only merging by replacing or orthogonalizing the incoming LoRA-A factor while temporally interpolating the LoRA-B factor with the deployed adapter.

A.2.3 Implementation Details

All multimodal task experiments were conducted on a single NVIDIA RTX 5090 GPU. For all baseline methods, we generally adopt the hyperparameters in their original work, except for the drop rate, like DARE. For the experiments of LLaVA, we adopt the checkpoints in MM-MergeBench [31]. For other models, each task is trained separately to obtain a LoRA adapter and used to continual merging. Specifically, LoRA adapters are inserted into the linear layers of the foundation blocks, and all models are trained for one epoch.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: They are properly claimed in the abstract and introduction.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations existing in this paper in the Appendix A.1.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: This work is not about theory.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide the main implementation details.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The datasets and models are sourced from HuggingFace or the public.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide the experimental setting in Section 5 and appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Our main experimental results include the average of three runs.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We describe the computational resources in Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: We do not have societal impacts.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This work does not release any models or data with known risks of misuse.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cited all models and data.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: No new assets are released in this work.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: No human subjects or crowdsourcing experiments are involved.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: No research involving human subjects is performed.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: [N/A]

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.