

Restoring Generalization in Fine-tuned Multimodal LLMs via Geometric Alignment

Lixuan He
Tsinghua University

Shikang Zheng
Shanghai Jiao Tong University

Guantao Chen
Sun Yat-sen University

Abstract

Fine-tuning Multimodal Large Language Models for downstream tasks requires balancing task adaptation with retention of broad pre-trained capabilities. In practice, full fine-tuning often improves in-distribution performance but degrades zero-shot behavior, while parameter-efficient tuning is more stable yet may leave a substantial adaptation gap. We present **Restoring Generalization Alignment (ReGA)**, a simple two-stage post-tuning procedure. ReGA first applies standard LoRA to obtain a task-adapted checkpoint, and then performs a short alignment stage that combines a low-loss path constraint with a soft anchor to the zero-shot model. This second stage encourages a nearby solution that preserves task performance while remaining closer to the pre-trained operating point. Experiments on LLaVA-1.5, Qwen2.5-VL, and InternVL-3.5 show that ReGA consistently improves the trade-off between in-distribution adaptation and broad zero-shot generalization relative to standard PEFT, regularized baselines, and merging-based baselines. The same recipe also remains effective in replay-free continual learning.

1 Introduction

Multimodal Large Language Models (MLLMs) (Liu et al., 2023; Bai et al., 2025; Chen et al., 2024b) have become strong general-purpose models for visual question answering, document understanding, grounded dialogue, and zero-shot reasoning. In practice, however, these models are rarely used in a purely zero-shot manner. They are usually adapted to downstream data through instruction tuning or parameter-efficient fine-tuning. This adaptation creates a familiar but consequential trade-off. **Full Fine-Tuning (FullFT)** often gives the strongest in-distribution (ID) performance, but can substantially degrade capabilities that were useful before tuning. **Parameter-Efficient Fine-Tuning (PEFT)** methods such as LoRA (Hu et al., 2021) are more stable and cheaper, yet they often remain behind FullFT on the target task and can still accumulate forgetting under continued adaptation (Shuttleworth et al., 2024; Luo et al., 2023; Ung et al., 2024).

A growing literature has tried to improve this trade-off from several directions. Stronger adapters and prompting schemes increase the expressive power of PEFT, while weight anchoring, weight averaging, and flatness-aware optimization aim to preserve pre-trained behavior during or after tuning (Li et al., 2018; Wortsman et al., 2022; Foret et al., 2021; Garipov et al., 2018; Mirzadeh et al., 2021). These ideas are useful, but they do not directly address the regime we focus on here: a task-adapted MLLM that already performs well on the downstream task, yet has moved away from a good zero-shot operating point. This regime is especially important for MLLMs because a single adaptation step can simultaneously affect multiple heterogeneous abilities, including OCR, document reasoning, visual grounding, and hallucination behavior. As a result, over-specialization in MLLMs is often broader than a drop on a single benchmark.

This observation motivates a simple question: after ordinary PEFT has learned the task, can we cheaply move the model to a better local solution that retains task performance while recovering more of the model’s broad pre-trained behavior? We answer this question with **Restoring Generalization Alignment (ReGA)**, a two-stage post-tuning procedure. ReGA

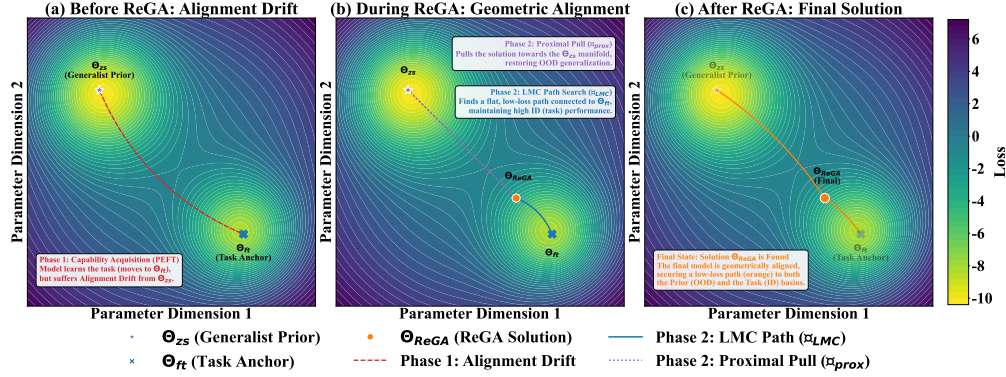


Figure 1: **Overview of ReGA.** (a) Standard PEFT produces a task-adapted checkpoint Θ_{ft} from the zero-shot model Θ_{zs} . (b) ReGA adds a short second phase that optimizes a nearby solution using two constraints: a low-loss path term that keeps the solution connected to the task anchor, and a soft proximal pull toward the zero-shot model. (c) The final checkpoint improves the trade-off between task adaptation and broad generalization.

first runs standard LoRA to obtain a *task anchor*. It then performs a short alignment phase with two complementary objectives: (i) a low-loss path constraint that keeps the updated solution connected to the task anchor, and (ii) a soft pull toward the zero-shot model. In the LoRA parameterization, the second term reduces to an adapter-space anchoring penalty. The contribution of ReGA is therefore not a new regularizer by itself, but a simple post-tuning recipe that decouples task acquisition from generalization repair.

Our focus on MLLMs is empirical. The method itself is general and does not depend on a vision-specific architecture. We study MLLMs because they provide a stringent testbed for the stability–plasticity trade-off: unlike many unimodal settings, degradation can appear across several capabilities at once, making it easier to evaluate whether a method truly improves the ID/OOD frontier rather than only one task metric. Across LLaVA-1.5, Qwen2.5-VL, and InternVL-3.5, we show that ReGA consistently improves this frontier against stronger regularized, merged, and compute-matched baselines, while remaining replay-free and lightweight. We further study the same idea in replay-free continual learning, where the tension between adaptation and retention becomes even sharper.

Takeaways

1. We study MLLM post-tuning as a trade-off between downstream adaptation and retention of broad zero-shot behavior, and analyze this trade-off through the notions of adaptation gap and alignment drift.
2. We propose **ReGA**, a simple two-stage post-tuning procedure that combines a low-loss path constraint with zero-shot anchoring in the adapter space.
3. ReGA improves the ID/OOD trade-off across three MLLM families and remains effective in replay-free continual learning with modest additional training cost.

2 Related Work

2.1 Parameter-Efficient Adaptation for LLMs and MLLMs

Parameter-efficient fine-tuning has become the standard adaptation interface for foundation models, beginning with low-rank and quantized updates (Hu et al., 2021; Dettmers et al., 2023) and extending to studies of the gap between LoRA and FullFT together with stronger adapter parameterizations, such as dynamic allocation and low-interference updates (Shuttleworth et al., 2024; Liao et al., 2025; Zhang et al., 2025b). In multimodal systems, this line has expanded from standard visual instruction tuning (Liu et al., 2023; Bai et al., 2025; Chen et al., 2024b) to continual and multi-task settings, where recent methods organize adaptation

through hierarchical decoupling, branch routing and curriculum mixtures (Zhang et al., 2025a; Ge et al., 2025; Che et al., 2025; Chen et al., 2025).

2.2 Loss Landscape Geometry, Flatness, and Model Merging

A parallel literature studies adaptation through parameter-space geometry. Linear mode connectivity and related flatness analyses showed that many trained solutions can be connected by low-loss paths (Garipov et al., 2018; Draxler et al., 2018; Frankle et al., 2020; Mirzadeh et al., 2021), which in turn motivated weight averaging and interpolation-based robust fine-tuning (Izmailov et al., 2019; Foret et al., 2021; Wortsman et al., 2022). These ideas later informed task arithmetic and sparse merging methods (Ilharco et al., 2023; Yadav et al., 2023), and recent work extended merging from one-shot fusion to aligned or continual regimes in vision-language and foundation models, including aligned model merging for VLM continual learning and adapter merging for continual adaptation (Sokar et al., 2025; Qiu et al., 2025; Qiao & Mahdavi, 2025; Shu et al., 2025; Song & Zheng, 2026).

2.3 Continual Learning and Forgetting in Foundation Models

Continual learning has traditionally combined replay, regularization, and parameter isolation (Rebuffi et al., 2017; Kirkpatrick et al., 2017; Li & Hoiem, 2017; Parisi et al., 2019). Recent work revisited these themes for foundation models by analyzing the behavioral difference between PEFT and FullFT, measuring catastrophic forgetting in LLMs, and introducing dedicated VLM/MLLM benchmarks and surveys (Shuttleworth et al., 2024; Haque, 2025; Zheng et al., 2025; Chen et al., 2024a; Zhao et al., 2025; Liu et al., 2025). Alongside these analyses, replay-free PEFT methods continue to develop around constrained full fine-tuning, shared or orthogonal low-rank subspaces, conflict-aware subspace alignment, and single-adapter continual composition (Nayak et al., 2025; Ling et al., 2025; Kaushik et al., 2026; Fa et al., 2026; Zhou et al., 2025; Cao & Wu, 2025), making parameter interference and subspace compatibility central themes in recent continual adaptation research.

3 Methodology

We study post-tuning of a Multimodal Large Language Model (MLLM) under parameter-efficient adaptation. Let ψ denote the frozen pre-trained backbone, and let ϕ denote the effective additive update induced by the trainable adapter parameters across all adapted layers. For clarity, we write the adapted model as

$$\Theta(\phi) = \psi + \phi, \quad (1)$$

where $\Theta(\phi)$ denotes the effective model after applying the adapter update to the backbone. Throughout, norms of ϕ refer to the Euclidean norm of the vectorized effective update, equivalently the sum of squared Frobenius norms over adapted weight matrices. Under standard LoRA initialization, the induced update is $\phi = 0$, so the zero-shot model is

$$\Theta_{zs} = \Theta(0) = \psi. \quad (2)$$

Given a downstream training set $\mathcal{D}_{task}$, standard LoRA fine-tuning produces a task-adapted checkpoint

$$\Theta_{ft} = \Theta(\phi_{ft}) = \psi + \phi_{ft}, \quad (3)$$

which we refer to as the *task anchor*. Our goal is to start from this task anchor and search for a nearby solution that preserves the acquired task capability while retaining more of the broad zero-shot behavior of the pre-trained model.

Before introducing ReGA, we first summarize the empirical behavior of standard post-training in Fig. 2. Across both model families, full fine-tuning achieves stronger task adaptation on the target distribution, whereas LoRA is typically more stable on broad zero-shot benchmarks but still drifts away from the pre-trained operating point as training proceeds.

This pattern suggests that ordinary PEFT finds a useful task solution, but not necessarily one with the best balance between task specialization and retained generalization. ReGA is

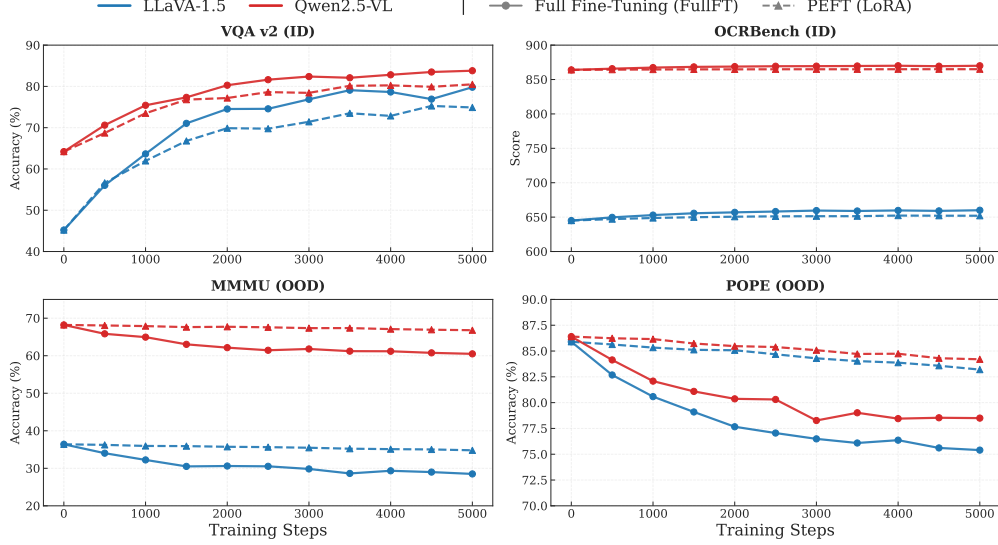


Figure 2: **Adaptation dynamics under standard post-training.** Training trajectories of LLaVA-1.5 and Qwen2.5-VL on the mixed VQA/OCR training set. **Top row:** Full fine-tuning reaches higher target-task performance than LoRA. **Bottom row:** Full fine-tuning degrades strict zero-shot retention more rapidly, while LoRA is more stable but still moves away from the zero-shot operating point over training.

designed as a short second-stage optimization that starts from this task-adapted checkpoint and searches for a nearby solution with a better trade-off.

3.1 ReGA Objective

ReGA uses a short second-stage optimization after ordinary LoRA training. The second stage optimizes the underlying adapter parameters through their induced effective update ϕ with the following objective:

$$\mathcal{L}_{\text{ReGA}}(\phi) = \mathcal{L}_{\text{task}}(\phi) + \beta \mathcal{L}_{\text{LMC}}(\phi) + \lambda \mathcal{R}_{\text{prox}}(\phi), \quad (4)$$

where $\mathcal{L}_{\text{task}}(\phi)$ is shorthand for the standard autoregressive training loss of $\Theta(\phi)$ on $\mathcal{D}_{\text{task}}$, and $\beta, \lambda \geq 0$ control the two second-stage regularizers.

Low-loss path term. A solution that remains connected to the task anchor through low-loss interpolations is less likely to move to a brittle region that preserves one metric at the cost of others. We therefore define a linear interpolation in the effective adapter-update space:

$$\Theta_{\alpha}(\phi) = \psi + (1 - \alpha)\phi_{ft} + \alpha\phi, \quad \alpha \sim \mathcal{U}[0, 1]. \quad (5)$$

The interpolated update can be evaluated functionally and need not be represented as a single rank- r adapter. The path term is the expected task loss along this segment:

$$\mathcal{L}_{\text{LMC}}(\phi) = \mathbb{E}_{\alpha \sim \mathcal{U}[0, 1]} [\mathcal{L}_{\text{task}}(\Theta_{\alpha}(\phi))]. \quad (6)$$

Minimizing Eq. (6) encourages the complete segment between the task anchor and the updated solution to remain at low task loss, rather than improving only the endpoint while introducing a loss barrier between them.

Zero-shot anchoring term. The second ingredient is a soft pull toward the zero-shot model:

$$\mathcal{R}_{\text{prox}}(\phi) = \|\Theta(\phi) - \Theta_{\text{zs}}\|_2^2. \quad (7)$$

Because ϕ is defined as the effective additive update relative to the frozen backbone, Eq. (7) reduces exactly to

$$\mathcal{R}_{\text{prox}}(\phi) = \|\phi\|_2^2. \quad (8)$$

Thus, the proximal term is an L_2 penalty in effective adapter-update space. Its role in ReGA is not to introduce a different regularizer, but to make the zero-shot checkpoint an explicit anchor during the second-stage search. The two terms in Eq. (4) therefore play complementary roles: $\mathcal{L}_{\text{LMC}}$ preserves a low-loss connection to the task anchor, while $\mathcal{R}_{\text{prox}}$ biases the search toward the pre-trained operating point.

This complementarity also admits a local interpretation. Let $d = \phi - \phi_{ft}$ and let H_{ft} denote the Hessian of the task loss at the Phase-1 anchor. When ϕ_{ft} is close to a stationary point, a second-order expansion gives

$$\mathcal{L}_{\text{LMC}}(\phi) = \mathcal{L}_{\text{task}}(\phi_{ft}) + \frac{1}{6}d^\top H_{ft}d + \mathcal{O}(\|d\|_2^3).$$

The path term therefore locally penalizes movement along task-sensitive, high-curvature directions, while the proximal term supplies the direction toward the zero-shot operating point. This interpretation is local and does not constitute a general guarantee of improved zero-shot performance.

3.2 Optimization

ReGA is implemented in two phases.

Phase 1: task adaptation. We first run standard LoRA fine-tuning on $\mathcal{D}_{\text{task}}$ and obtain the task anchor ϕ_{ft} .

Phase 2: geometric alignment. We then initialize the second stage from the task anchor, i.e., $\phi \leftarrow \phi_{ft}$, and optimize Eq. (4) while keeping the backbone ψ frozen. In practice, the expectation in Eq. (6) is estimated with K Monte Carlo samples:

$$\hat{\mathcal{L}}_{\text{LMC}}(\phi) = \frac{1}{K} \sum_{k=1}^K \mathcal{L}_{\text{task}}(\Theta_{\alpha_k}(\phi)), \quad \alpha_k \sim \mathcal{U}[0, 1]. \quad (9)$$

Writing the gradient with respect to the effective update ϕ , we obtain

$$\nabla_{\phi} \mathcal{L}_{\text{ReGA}} = \nabla_{\phi} \mathcal{L}_{\text{task}}(\phi) + \frac{\beta}{K} \sum_{k=1}^K \alpha_k \nabla_{\Theta} \mathcal{L}_{\text{task}}(\Theta_{\alpha_k}(\phi)) + 2\lambda\phi. \quad (10)$$

The factor α_k appears because $\Theta_{\alpha}(\phi)$ depends linearly on the current effective update. In implementation, gradients with respect to the underlying LoRA factors are obtained automatically through the parameterization that induces ϕ . Each Phase-2 step evaluates one endpoint loss and K path losses, so its forward-evaluation cost scales linearly with $1 + K$ and the number of second-stage steps. In the experiments, we therefore report forward-pass-equivalent controls together with measured wall-clock and memory costs.

3.3 Replay-Free Continual Learning Instantiation

The same formulation applies to replay-free continual learning. Suppose training proceeds sequentially over tasks $\{\mathcal{D}_1, \dots, \mathcal{D}_T\}$. After finishing task $t - 1$, we retain the previous checkpoint

$$\Theta_{t-1} = \psi + \phi_{t-1}. \quad (11)$$

For the new task $\mathcal{D}_t$, standard LoRA first yields a task anchor ϕ_t^{ft} . ReGA then optimizes

$$\mathcal{L}_{\text{ReGA}}^{(t)}(\phi) = \mathcal{L}_t(\phi) + \beta \mathbb{E}_{\alpha \sim \mathcal{U}[0, 1]} \left[\mathcal{L}_t(\psi + (1 - \alpha)\phi_t^{ft} + \alpha\phi) \right] + \lambda \|\phi - \phi_{t-1}\|_2^2. \quad (12)$$

Relative to the single-task setting, the zero-shot anchor is replaced by the previous checkpoint. The proximal term therefore encourages retention of previously acquired behavior, while the path term keeps the new solution connected through low task loss to the task anchor learned on the current task. This yields a replay-free update rule with the same optimization structure as Eq. (4).

Table 1: Main single-task results on three MLLM families. ReGA gives the strongest overall target-performance/generalization trade-off. Best results are in **bold**; second-best are underlined.

Model	Method	Target / Task-Related		Generalization / Retention					
		VQA v2	OCRBench	MMMU	MME	DocVQA	InfoVQA	TextVQA	POPE
InternVL-3.5-8B	Zero-Shot	65.4	840	73.4	<u>2380.6</u>	<u>92.3</u>	<u>79.1</u>	78.2	88.7
	FullFT	84.5	848	65.2	1950.4	85.4	70.2	81.5	80.2
	LoRA	81.2	841	71.8	2310.5	90.1	77.5	79.4	86.5
	LoRA+WD	79.8	838	72.5	2360.2	91.0	78.5	78.0	87.8
	L2-SP	82.1	843	68.5	2100.2	88.6	74.3	80.1	83.4
	Wise-FT	78.5	835	72.1	2350.1	91.5	78.2	77.8	87.9
	TIES	76.2	820	70.5	2280.8	89.2	75.4	76.5	85.1
	ReGA	<u>84.2</u>	<u>845</u>	<u>73.1</u>	2382.4	92.5	79.4	<u>80.8</u>	<u>88.5</u>
Qwen2.5-VL-7B	Zero-Shot	64.2	864	68.2	2347.0	95.7	82.6	84.9	86.4
	FullFT	83.8	870	60.5	1920.5	88.2	74.5	86.2	78.5
	LoRA	80.5	865	66.8	2295.4	93.5	80.8	85.1	84.2
	LoRA+WD	78.2	862	67.5	2330.1	95.0	81.9	84.6	85.9
	L2-SP	81.4	867	63.2	2050.8	91.4	77.6	85.5	81.6
	Wise-FT	77.6	860	67.1	2315.2	94.8	81.5	84.5	85.8
	TIES	75.4	852	65.8	2250.6	92.6	79.2	83.8	83.5
	ReGA	<u>83.5</u>	<u>868</u>	<u>67.9</u>	<u>2345.8</u>	96.0	82.9	<u>85.8</u>	<u>86.2</u>
LLaVA-1.5-7B	Zero-Shot	45.2	645	36.4	1510.7	<u>68.2</u>	<u>45.1</u>	58.2	85.9
	FullFT	79.8	660	28.5	1250.2	55.4	38.5	62.5	75.4
	LoRA	74.9	652	34.8	1480.5	65.1	43.2	60.1	83.2
	LoRA+WD	73.5	649	35.9	<u>1500.2</u>	67.5	44.3	59.2	85.1
	L2-SP	76.5	653	31.2	1320.8	60.8	40.6	61.2	79.5
	Wise-FT	72.4	648	35.6	1495.3	66.5	44.5	59.5	84.8
	TIES	70.8	640	33.9	1450.6	64.2	42.8	58.4	82.6
	ReGA	<u>79.1</u>	<u>658</u>	<u>36.1</u>	1493.4	68.5	45.4	<u>61.8</u>	<u>85.7</u>

4 Experiments

4.1 Experimental Setup

We evaluate three representative MLLMs: **LLaVA-1.5-7B**, **Qwen2.5-VL-7B-Instruct**, and **InternVL-3.5-8B** (Chen et al., 2024b). All single-task experiments use a mixed training set of approximately 650k samples constructed from **VQA v2** and **OCR-VQA**. We report two target or task-related metrics, **VQA v2** and **OCRBench**, together with six held-out generalization metrics: **MMMU**, **MME**, **DocVQA**, **InfoVQA**, **TextVQA**, and **POPE** (Fu et al., 2025; Mathew et al., 2021b;a; Singh et al., 2019). Because DocVQA, InfoVQA, and TextVQA share substantial OCR and document structure with OCR-VQA, we interpret them as related-task transfer, while MMMU, MME, and POPE form a stricter zero-shot retention set. This evaluation covers perception, OCR, document understanding, and reasoning-oriented benchmarks, and is intended to measure both task adaptation and retained broad capability. We compare against **FullFT**, **LoRA** (Hu et al., 2021), **LoRA+WD**, **L2-SP**, **Wise-FT** (Wortsman et al., 2022), and **TIES** (Yadav et al., 2023). Unless otherwise stated, LoRA uses rank $r = 64$, $\alpha = 16$, dropout 0.05, and all linear layers as target modules. Additional comparisons with MaTS, multi-seed runs, and strict zero-shot retention statistics are provided in Appendix C.9, Appendix C.7, and Appendix C.8, respectively.

4.2 Main Results

Table 1 reports the main single-task results. Across all three model families, ReGA consistently improves the target-performance/generalization trade-off relative to standard LoRA and regularized or merged baselines. Compared with FullFT, ReGA reaches similar target-task performance while retaining substantially stronger zero-shot behavior. The gains are especially clear on benchmarks that depend on broad visual-language alignment rather than only narrow task fitting. For example, on **Qwen2.5-VL-7B**, ReGA attains **96.0** on **DocVQA**, exceeding both Zero-Shot (95.7) and LoRA (93.5), while remaining close to FullFT on VQA v2.

4.3 Matched-Budget Ablation

A central concern is whether the Phase-2 gain comes from the objective itself or from additional optimization. We therefore report two controls on LLaVA-1.5-7B: **forward-**

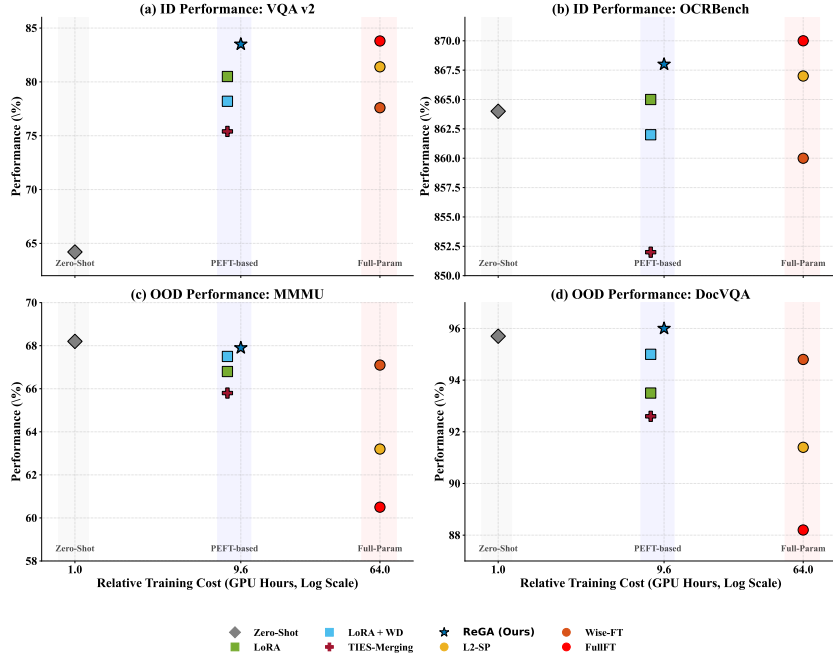


Figure 3: **Performance versus relative training cost on Qwen2.5-VL-7B.** We plot two target or task-related benchmarks (VQA v2 and OCRBench) and two held-out generalization benchmarks (MMMU and DocVQA) against relative training cost on a log scale. ReGA remains in the low-cost PEFT regime while approaching the target-task performance of FullFT and retaining stronger held-out behavior than other PEFT-based or merged baselines.

Table 2: Forward-pass-equivalent and best-of-sweep controls on LLaVA-1.5-7B. ReGA remains stronger than longer-training and stronger-sweep baselines under the same 2.2-FPE accounting.

Method	Budget / Setting	VQA v2	OCRBench	MMMU	POPE
LoRA	1.0 epoch	74.9	652	34.8	83.2
LoRA	2.2 FPE	77.4	655	33.4	81.6
LoRA+WD	1.0 epoch	73.5	649	35.9	85.1
LoRA+WD	2.2 FPE	76.9	655	35.6	84.7
L2-SP	best single setting	76.5	653	31.2	79.5
L2-SP	2.2 FPE (sweep)	77.6	657	33.1	81.2
LoRA+WD	2.2 FPE (sweep)	77.8	657	36.0	85.5
ReGA	2.2 FPE (1.0 + 0.2)	79.1	658	36.1	85.7

pass-equivalent (FPE) reruns, which equalize the number of forward evaluations at the dataset-epoch level, and **best-of-sweep controls**, which tune LoRA+WD and L2-SP over the same regularization range rather than using a single operating point. With $K = 5$, each Phase-2 step evaluates one endpoint loss and five path losses, so the complete ReGA schedule corresponds to $1.0 + 0.2(1 + 5) = 2.2$ FPE. Table 2 shows that longer training partially narrows the gap for standard baselines, but does not recover the same target-performance/retention balance as ReGA. LoRA trained to 2.2 FPE improves target-task performance, but its retention metrics drop further, while LoRA+WD is more stable but remains behind ReGA on both VQA v2 and POPE.

To summarize the cost–performance trade-off more compactly, we additionally compare methods in the plane of downstream performance versus relative training cost. Rather than listing memory and time separately, this view highlights whether a method stays in the low-cost PEFT regime while approaching the task quality of substantially more expensive full-parameter updates.

Table 4: Continual Learning performance on the MLLM-CL benchmark using LLaVA-1.5-7B. For each task, *Last* denotes the accuracy after the complete sequence (RS $\rightarrow$ Med $\rightarrow$ AD $\rightarrow$ Sci $\rightarrow$ Fin), while *Avg* denotes the average accuracy over sequential evaluations. The final column is the mean of the five task-level *Avg* values. Best replay-free results are in **bold**.

Method	RS (%)		Med (%)		AD (%)		Sci (%)		Fin (%)		Mean Avg $\uparrow$
	Last $\uparrow$	Avg $\uparrow$	Last $\uparrow$	Avg $\uparrow$	Last $\uparrow$	Avg $\uparrow$	Last $\uparrow$	Avg $\uparrow$	Last $\uparrow$	Avg $\uparrow$	
Zero-shot	32.29	-	28.28	-	15.59	-	35.55	-	62.56	-	-
<i>w/ Replay Buffer (20 samples/task)</i>											
LoRA	29.57	80.87	29.19	58.60	7.09	38.95	19.55	36.41	63.60	36.78	50.32
O-LoRA (Wang et al., 2023)	76.21	80.13	51.34	70.23	36.50	61.35	42.64	53.34	90.20	59.38	64.89
L2P (Wang et al., 2022)	75.21	80.09	38.50	68.64	32.31	54.79	41.05	48.68	88.05	55.02	61.44
HiDe-LLaVA (Guo et al., 2025)	75.36	81.51	39.23	62.37	37.17	49.37	45.02	50.61	81.89	55.73	59.92
MR-LoRA (Zhao et al., 2025)	79.87	80.82	62.71	72.19	51.89	65.41	52.48	62.52	89.69	67.31	69.65
<i>w/o Replay Buffer (Replay-Free)</i>											
LoRA	26.75	80.72	25.76	59.68	0.79	40.51	18.69	18.64	70.44	28.49	45.61
O-LoRA (Wang et al., 2023)	62.68	80.22	35.17	67.56	16.93	51.51	34.44	44.28	92.16	48.28	58.37
ModalPrompt (Zeng et al., 2024)	65.99	80.11	37.35	59.66	23.27	46.86	37.61	42.97	87.60	50.36	55.99
HiDe-LLaVA (Guo et al., 2025)	41.17	80.91	30.33	65.47	18.73	39.78	37.08	32.92	92.21	43.90	52.60
ReGA (Ours)	70.15	81.05	55.30	70.10	45.20	60.30	48.10	55.20	88.20	59.10	65.15

4.4 Representation Stability

We track the cosine similarity between the last-layer visual-token hidden states of each adapted model and those of the zero-shot checkpoint on MMMU. As shown in Fig. 4, FullFT moves away from the zero-shot representation much faster than adapter-based tuning, while standard LoRA still exhibits a steady decline over training. ReGA changes this trend during the short second stage and recovers a substantial fraction of the lost similarity.

4.5 Replay-Free Continual Learning

We further evaluate ReGA on MLLM-CL (Zhao et al., 2025), where LLaVA-1.5-7B is trained sequentially on Remote Sensing, Medical, Autonomous Driving, Science, and Finance.

4.6 Ablations and Robustness

We further isolate the contribution of the path term, the sampling approximation, the adapter budget, and the adapter parameterization. Table 3 shows that the path term is beneficial beyond proximal anchoring alone: moving from $\beta = 0$ to $\beta = 1.0$ improves both target-task and retention metrics. The same table also shows that increasing the number of interpolation samples improves the Monte Carlo approximation up to about $K = 5$, after which the gains saturate. We additionally compare a diagonal-Fisher-weighted proximal variant as a stronger anisotropic regularizer. Its gain over a scalar proximal term is small, and both remain behind the full ReGA objective. Table 5 shows that the improvement is stable across LoRA ranks and also transfers to stronger PEFT variants such as DoRA (Liu

Table 3: Component, proximal-form, and sampling ablations.

β	VQA v2	OCRBench	MMMU	POPE
0.0 (proximal only)	77.8	653	35.8	85.2
0.5	79.0	657	36.0	85.5
1.0	79.1	658	36.1	85.7
2.0	78.9	656	36.1	85.6
Proximal Form	VQA v2	OCRBench	MMMU	POPE
Scalar L_2 only	77.8	653	35.8	85.2
Diagonal Fisher only	79.0	656	35.9	85.4
Scalar L_2 + LMC (ReGA)	79.1	658	36.1	85.7
K	VQA v2	OCRBench	MMMU	POPE
1	78.5	648	35.2	84.8
3	78.9	654	35.8	85.3
5	79.1	658	36.1	85.7
10	79.1	659	36.1	85.8

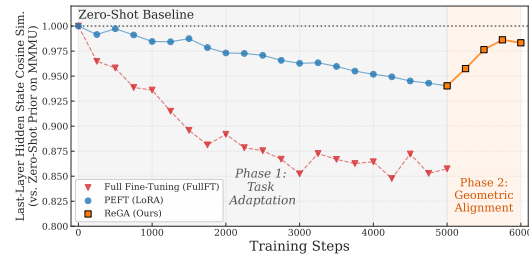


Figure 4: **Representation stability relative to the zero-shot anchor.** ReGA increases the similarity during its short second stage, indicating that the final checkpoint remains closer to the pre-trained representation while preserving the task gains obtained in Phase 1.

Table 6: Additional robustness checks on LLaVA-1.5-7B. ReGA remains effective under narrower supervision and more restrictive adapter placement, with the largest gain in the default mixed-data, all-linear setting.

Category	Setting	VQA v2	OCRBench	MMMU	POPE
Data	VQA v2 only + LoRA	77.3	606	34.3	82.7
	VQA v2 only + ReGA	79.6	611	35.1	84.0
	OCR-VQA only + LoRA	57.8	686	33.5	83.9
	OCR-VQA only + ReGA	58.4	695	34.1	84.8
Modules	attention-only LoRA	73.6	646	34.3	82.9
	attention-only ReGA	75.7	650	35.0	84.0
	MLP-only LoRA	72.7	643	33.9	82.5
	MLP-only ReGA	74.8	647	34.7	83.6
	all-linear LoRA	74.9	652	34.8	83.2
	all-linear ReGA	79.1	658	36.1	85.7

et al., 2024) and AdaLoRA (Zhang et al., 2023). Cross-model sensitivity and stronger-proximal-pull results are provided in Appendix C.2 and Appendix C.3.

4.7 Interpolation-Path Diagnostics

We additionally examine the linear interpolation path between the zero-shot anchor and each final task-adapted checkpoint. This prior-to-endpoint path differs from the task-anchor-to-candidate path optimized during Phase 2, and is used only as an auxiliary diagnostic of how adaptation changes the loss and error profiles between the pretrained and final operating points. We evaluate this interpolation for both standard LoRA and ReGA. As shown in Fig. 5, the difference between standard LoRA and ReGA is not limited to the endpoint metrics.

Table 5: Robustness across adapter rank and PEFT parameterization on LLaVA-1.5-7B. Left: LoRA rank sweep. Right: applying ReGA on top of stronger PEFT baselines.

Rank	Method	VQA	OCR	MMMU	POPE	Method	VQA	OCR	MMMU	POPE
16	LoRA	77.2	641	35.0	83.8	DoRA	77.8	645	35.1	84.5
16	ReGA	79.3	649	35.9	85.6	DoRA + ReGA	80.2	661	36.4	86.0
64	LoRA	74.9	652	34.8	83.2	AdaLoRA	76.2	638	34.9	83.9
64	ReGA	79.1	658	36.1	85.7	AdaLoRA + ReGA	79.5	654	36.0	85.6
128	LoRA	74.5	655	34.5	82.9					
128	ReGA	78.8	660	36.2	85.8					

4.8 Additional Robustness Checks

To make the evaluation less dependent on a single training recipe, we additionally test **dataset composition** and **module scope**. The first asks whether the proximal pull remains useful when the supervision becomes narrower; the second asks whether the method depends on using all linear layers. Table 6 shows that ReGA still improves the target-performance/retention trade-off under single-source supervision and restricted-module adaptation, although the gain is smaller than in the mixed-data, all-linear setting. This is consistent with the role of the second stage as a local correction around a strong task anchor: when the Phase-1 anchor is weaker or less expressive, the room for improvement is reduced but does not disappear.

4.9 Optimization Behavior

We finally give a qualitative view of the optimization path. As shown in Fig. 6, the first stage accounts for the main movement toward a task-adapted solution, while the second stage performs a shorter local adjustment around that checkpoint. Although this two-dimensional projection is only illustrative, it is consistent with the role of ReGA in practice: the second stage does not replace task learning, but refines an existing task solution toward a parameter region that better preserves the pre-trained operating point.

5 Conclusion and Discussion

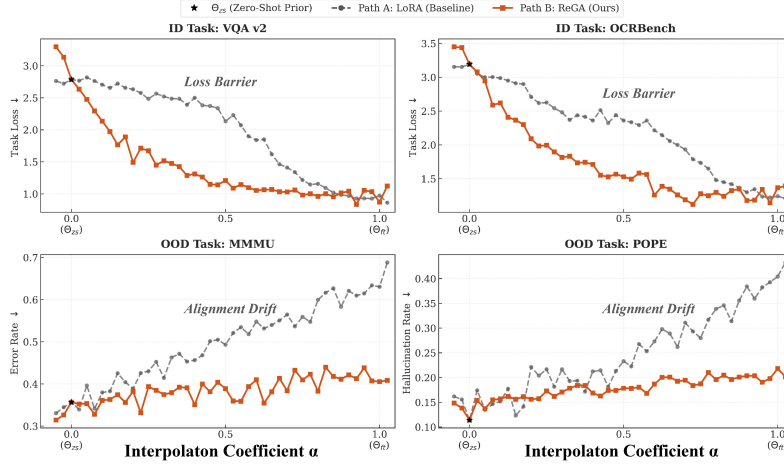


Figure 5: Auxiliary zero-shot-to-endpoint interpolation diagnostics. **Top row:** on target-task benchmarks, standard LoRA exhibits a more pronounced loss barrier, whereas ReGA produces a lower-loss interpolation path. **Bottom row:** on held-out benchmarks, the error increase along the path is smaller for ReGA than for standard LoRA.

We studied MLLM post-tuning as a trade-off between downstream adaptation and retention of broad zero-shot capability. The proposed ReGA framework addresses this trade-off with a simple two-stage recipe: standard PEFT first acquires the task, and a short second stage then improves the operating point of the adapted checkpoint through path-based alignment and zero-shot anchoring. Across three MLLM families, ReGA consistently strengthened the ID/OOD frontier relative to standard LoRA, regularized PEFT, and merging-based baselines, while remaining replay-free and lightweight. The same idea also transferred effectively to replay-free continual learning, where the tension between plasticity and retention is more severe.

At the same time, the method should be viewed as a local post-tuning correction rather than a universal solution to forgetting or generalization. Its effectiveness depends on the existence of a task-adapted checkpoint that is already strong and lies within a region where a better trade-off can still be reached by a short geometric adjustment. When the downstream objective is highly specialized, strongly shifts away from pre-training, or requires capabilities that are poorly aligned with the zero-shot model, the proximal pull may provide only a limited benefit and can even restrict further specialization. In addition, the Phase-2 objective introduces extra compute through path sampling, and the current formulation uses a simple isotropic anchoring term instead of a more structured importance-aware prior. These limitations suggest several natural directions for future work, including stronger anchor selection strategies, more adaptive geometric objectives, and extensions beyond adapter-based post-tuning.

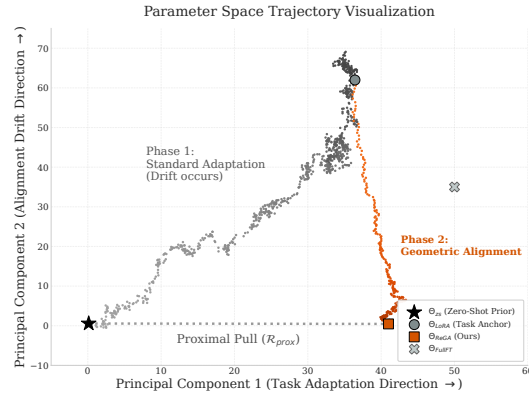


Figure 6: **Adapter trajectory visualization during two-stage training.** We project the adapter optimization path of LLaVA-1.5-7B into two principal components. Phase 1 moves the model toward a strong task-adapted solution. Phase 2 makes a smaller update around this checkpoint and shifts the parameters toward a region that is closer to the zero-shot anchor while maintaining task performance.

Ethics Statement

This work studies post-tuning and continual adaptation of multimodal language models using public benchmarks and standard open-model training pipelines. The method does not involve private user data, biometric identification, or human-subject interaction. As with other fine-tuning methods, ReGA can inherit biases, errors, or unsafe behaviors from the underlying pre-trained models, and it should not be interpreted as a safety guarantee. In particular, the method may be less reliable under severe domain shift or in high-stakes settings, where additional task-specific evaluation, monitoring, and human oversight remain necessary.

References

- Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Jun Tang, Shijie Wang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.
- Kefan Cao and Shuaicheng Wu. Orthogonal low-rank adaptation in lie groups for continual learning of large language models. *arXiv preprint arXiv:2509.06100*, 2025.
- Chang Che, Ziqi Wang, Pengwan Yang, Qi Wang, Hui Ma, and Zenglin Shi. LoRA in LoRA: Towards parameter-efficient architecture expansion for continual visual instruction tuning. *arXiv preprint arXiv:2508.06202*, 2025.
- Cheng Chen, Junchen Zhu, Xu Luo, Hengtao Shen, Lianli Gao, and Jingkuan Song. Coin: A benchmark of continual instruction tuning for multimodal large language model, 2024a. URL <https://arxiv.org/abs/2403.08350>.
- Jinpeng Chen, Runmin Cong, Yuzhi Zhao, Hongzheng Yang, Guangneng Hu, Horace Ho Shing Ip, and Sam Kwong. SEFE: Superficial and essential forgetting eliminator for multimodal continual instruction tuning. *arXiv preprint arXiv:2505.02486*, 2025.
- Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks, 2024b. URL <https://arxiv.org/abs/2312.14238>.
- Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms, 2023. URL <https://arxiv.org/abs/2305.14314>.
- Felix Draxler, Kambis Veschgini, Manfred Salmhofer, and Fred Hamprecht. Essentially no barriers in neural network energy landscape. In *International conference on machine learning*, pp. 1309–1318. PMLR, 2018.
- Zhan Fa, Yue Duan, Jian Zhang, Lei Qi, Wanqi Yang, and Yinghuan Shi. Decomposing and composing: Towards efficient vision-language continual learning via rank-1 expert pool in a single LoRA. *arXiv preprint arXiv:2601.22828*, 2026.
- Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2021.
- Jonathan Frankle, Gintare Karolina Dziugaite, Daniel M Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *International conference on machine learning*, pp. 3259–3269. PMLR, 2020.
- Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, Rongrong Ji, Caifeng Shan, and Ran He. Mme: A comprehensive evaluation benchmark for multimodal large language models, 2025. URL <https://arxiv.org/abs/2306.13394>.

- Timur Garipov, Pavel Izmailov, Dmitrii Podoprikin, Dmitry P Vetrov, and Andrew G Wilson. Loss surfaces, mode connectivity, and fast ensembling of dnns. In *Advances in Neural Information Processing Systems*, volume 31, 2018.
- Chendi Ge, Xin Wang, Zeyang Zhang, Hong Chen, Jiawei Fan, Longtao Huang, Hui Xue, and Wenwu Zhu. Dynamic mixture of curriculum LoRA experts for continual multimodal instruction tuning. *arXiv preprint arXiv:2506.11672*, 2025.
- 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. *arXiv preprint arXiv:2503.12941*, 2025.
- Naimul Haque. Catastrophic forgetting in LLMs: A comparative analysis across language tasks. *arXiv preprint arXiv:2504.01241*, 2025.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.
- Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *International Conference on Learning Representations*, 2023.
- Pavel Izmailov, Dmitrii Podoprikin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization, 2019. URL <https://arxiv.org/abs/1803.05407>.
- Prakhar Kaushik, Ankit Vaidya, Shravan Chaudhari, Rama Chellappa, and Alan Yuille. Shared LoRA subspaces for almost strict continual learning. *arXiv preprint arXiv:2602.06043*, 2026.
- James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.
- Xuhong Li, Yves Grandvalet, and Franck Davoine. Explicit inductive bias for transfer learning with convolutional networks, 2018. URL <https://arxiv.org/abs/1802.01483>.
- Zhizhong Li and Derek Hoiem. Learning without forgetting, 2017. URL <https://arxiv.org/abs/1606.09282>.
- Xiaoxuan Liao, Chihang Wang, Shicheng Zhou, Jiacheng Hu, Hongye Zheng, and Jia Gao. Dynamic adaptation of LoRA fine-tuning for efficient and task-specific optimization of large language models. *arXiv preprint arXiv:2501.14859*, 2025.
- Shimou Ling, Liang Zhang, Jiangwei Zhao, Lili Pan, and Hongliang Li. LoRA-based continual learning with constraints on critical parameter changes. *arXiv preprint arXiv:2504.13407*, 2025.
- Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 69960–69976, 2023.
- Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. Dora: Weight-decomposed low-rank adaptation, 2024. URL <https://arxiv.org/abs/2402.09353>.
- Yuyang Liu, Qiuhe Hong, Linlan Huang, Alexandra Gomez-Villa, Dipam Goswami, Xialei Liu, Joost van de Weijer, and Yonghong Tian. Continual learning for VLMs: A survey and taxonomy beyond forgetting. *arXiv preprint arXiv:2508.04227*, 2025.
- Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *arXiv preprint arXiv:2308.08747*, 2023.

- Minesh Mathew, Viraj Bagal, Rubèn Pérez Tito, Dimosthenis Karatzas, Ernest Valveny, and C. V. Jawahar. Infographicvqa, 2021a. URL <https://arxiv.org/abs/2104.12756>.
- Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. Docvqa: A dataset for vqa on document images, 2021b. URL <https://arxiv.org/abs/2007.00398>.
- Seyed Iman Mirzadeh, Mehrdad Farajtabar, Dilan Gorur, Razvan Pascanu, and Hassan Ghasemzadeh. Linear mode connectivity in multitask and continual learning. In *International Conference on Learning Representations*, 2021.
- Nikhil Shivakumar Nayak, Krishnateja Killamsetty, Ligong Han, Abhishek Bhandwaldar, Prateek Chanda, Kai Xu, Hao Wang, Aldo Pareja, Oleg Silkin, Mustafa Eyceoz, and Akash Srivastava. Sculpting subspaces: Constrained full fine-tuning in LLMs for continual learning. *arXiv preprint arXiv:2504.07097*, 2025.
- German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural networks*, 113:54–71, 2019.
- Fuli Qiao and Mehrdad Mahdavi. Merge before forget: A single LoRA continual learning via continual merging. *arXiv preprint arXiv:2512.23017*, 2025.
- 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 stability, promoting plasticity. *arXiv preprint arXiv:2509.21413*, 2025.
- Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2001–2010, 2017.
- Yuan-Chen Shu, Zhiwei Lin, and Yongtao Wang. RegCL: Continual adaptation of segment anything model via model merging. *arXiv preprint arXiv:2507.12297*, 2025.
- Reece Shuttleworth, Jacob Andreas, Antonio Torralba, and Pratyusha Sharma. Lora vs full fine-tuning: An illusion of equivalence. *arXiv preprint arXiv:2410.21228*, 2024.
- Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read, 2019. URL <https://arxiv.org/abs/1904.08920>.
- Ghada Sokar, Gintare Karolina Dziugaite, Anurag Arnab, Ahmet Iscen, Pablo Samuel Castro, and Cordelia Schmid. Continual learning in vision-language models via aligned model merging. *arXiv preprint arXiv:2506.03189*, 2025.
- Mingyang Song and Mao Zheng. Model merging in the era of large language models: Methods, applications, and future directions. *arXiv preprint arXiv:2603.09938*, 2026.
- Megan Ung, Alicia Sun, Samuel J. Bell, Bhaktipriya Radharapu, Levent Sagun, and Adina Williams. Chained tuning leads to biased forgetting. *arXiv preprint arXiv:2412.16469*, 2024.
- Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuanjing Huang. Orthogonal subspace learning for language model continual learning, 2023. URL <https://arxiv.org/abs/2310.14152>.
- Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *CVPR*, 2022.
- Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, et al. Robust fine-tuning of zero-shot models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7959–7971, 2022.

- Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. Ties-merging: Resolving interference when merging models. In *Advances in Neural Information Processing Systems*, volume 36, pp. 7093–7115, 2023.
- Fanhu Zeng, Fei Zhu, Haiyang Guo, Xu-Yao Zhang, and Cheng-Lin Liu. Modalprompt: Dual-modality guided prompt for continual learning of large multimodal models. *arXiv preprint arXiv:2410.05849*, 2024.
- 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. *arXiv preprint arXiv:2506.02041*, 2025a.
- Juzheng Zhang, Jiacheng You, Ashwinee Panda, and Tom Goldstein. LoRI: Reducing cross-task interference in multi-task low-rank adaptation. *arXiv preprint arXiv:2504.07448*, 2025b.
- Qingru Zhang, Minshuo Chen, Alexander Bukharin, Nikos Karampatziakis, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adalora: Adaptive budget allocation for parameter-efficient fine-tuning, 2023. URL <https://arxiv.org/abs/2303.10512>.
- Hongbo Zhao, Fei Zhu, Haiyang Guo, Meng Wang, Rundong Wang, Gaofeng Meng, and Zhaoxiang Zhang. Mllm-cl: Continual learning for multimodal large language models, 2025. URL <https://arxiv.org/abs/2506.05453>.
- Junhao Zheng, Xidi Cai, Shengjie Qiu, and Qianli Ma. Spurious forgetting in continual learning of language models. *arXiv preprint arXiv:2501.13453*, 2025.
- Yueer Zhou, Yichen Wu, and Ying Wei. Resolving conflicts in lifelong learning via aligning updates in subspaces. *arXiv preprint arXiv:2512.08960*, 2025.

Appendix

A Detailed Implementation Settings

We provide a comprehensive overview of the experimental configuration to ensure reproducibility. Our framework is implemented using the PyTorch library and integrates with the Hugging Face Transformers and PEFT ecosystems.

A.1 Hyperparameter Configurations

Table 7 details the default hyperparameters used for both the initial Capability Acquisition (Phase 1) and the subsequent Geometric Alignment (Phase 2). Consistent with the main text, Phase 1 utilizes standard LoRA fine-tuning to establish the Task Anchor (Θ_{ft}), while Phase 2 applies the ReGA objective to search for a nearby solution with a better target-performance/generalization trade-off. We employ a substantially reduced learning rate and a constant scheduler in Phase 2 to support stable local optimization around the task anchor.

ReGA uses the same Phase-1 LoRA checkpoint as its task anchor, followed by a short Phase-2 optimization with learning rate 1×10^{-5} , $\lambda = 0.1$, $\beta = 1.0$, and $K = 5$. The reported 0.2 Phase-2 epochs are optimizer-step epochs. Because each Phase-2 step evaluates one endpoint loss and five path losses, the full schedule corresponds to $1.0 + 0.2(1 + 5) = 2.2$ forward-pass-equivalent (FPE) epochs. We therefore compare against baselines using the same 2.2-FPE accounting. For LoRA+WD and L2-SP, the regularization sweep range is $\lambda \in \{0.01, 0.03, 0.1, 0.3, 1.0\}$.

Table 7: Default Hyperparameter Configuration for LLaVA-1.5, Qwen2.5-VL, and InternVL-3.5. We distinguish between common architectural settings and phase-specific training parameters.

Configuration Category	Parameter	Value
LoRA Architecture	Rank (r)	64
	Alpha (α)	16
	Dropout	0.05
	Target Modules	All Linear Layers
	Bias	None
Optimizer (Common)	Optimizer	AdamW
	Adam β_1	0.9
	Adam β_2	0.999
	Global Batch Size	128
Phase 1: Capability Acquisition	Learning Rate	2e-4
	LR Scheduler	Cosine Decay
	Warmup Ratio	0.03
	Epochs	1.0
	Weight Decay	0.0
Phase 2: Geometric Alignment	Learning Rate	1e-5
	LR Scheduler	Constant
	Warmup Ratio	0.0
	Optimizer-Step Epochs	0.2
	Proximal Weight (λ)	0.1
	LMC Weight (β)	1.0
	LMC Samples (K)	5

B Theoretical Derivations

B.1 Gradient Derivation of the ReGA Objective

For consistency with the main text, let ϕ denote the effective additive update induced by the trainable adapter parameters across all adapted layers. The adapted model is

$$\Theta(\phi) = \psi + \phi, \quad (13)$$

where ψ is the frozen backbone. The zero-shot model is therefore

$$\Theta_{zs} = \Theta(0) = \psi, \quad (14)$$

and the Phase-1 task-adapted checkpoint is

$$\Theta_{ft} = \Theta(\phi_{ft}) = \psi + \phi_{ft}. \quad (15)$$

Throughout this section, $\|\phi\|_2^2$ denotes the squared Euclidean norm of the vectorized effective update, equivalently the sum of squared Frobenius norms over the adapted weight matrices.

The ReGA objective in the single-task setting is

$$\mathcal{L}_{\text{ReGA}}(\phi) = \mathcal{L}_{\text{task}}(\phi) + \beta \mathcal{L}_{\text{LMC}}(\phi) + \lambda \mathcal{R}_{\text{prox}}(\phi), \quad (16)$$

where

$$\mathcal{L}_{\text{LMC}}(\phi) = \mathbb{E}_{\alpha \sim \mathcal{U}[0,1]} [\mathcal{L}_{\text{task}}(\Theta_{\alpha}(\phi))], \quad (17)$$

and the interpolation path is defined between the *task anchor* and the current candidate:

$$\Theta_{\alpha}(\phi) = \psi + (1 - \alpha)\phi_{ft} + \alpha\phi. \quad (18)$$

The proximal term anchors the solution to the zero-shot model:

$$\mathcal{R}_{\text{prox}}(\phi) = \|\Theta(\phi) - \Theta_{zs}\|_2^2 = \|\phi\|_2^2. \quad (19)$$

Using a Monte Carlo approximation with K samples $\{\alpha_k\}_{k=1}^K$, we obtain

$$\hat{\mathcal{L}}_{\text{LMC}}(\phi) = \frac{1}{K} \sum_{k=1}^K \mathcal{L}_{\text{task}}(\Theta_{\alpha_k}(\phi)), \quad \alpha_k \sim \mathcal{U}[0,1]. \quad (20)$$

Since

$$\frac{\partial \Theta_{\alpha_k}(\phi)}{\partial \phi} = \alpha_k I, \quad (21)$$

the gradient with respect to the effective update is

$$\nabla_{\phi} \mathcal{L}_{\text{ReGA}} = \nabla_{\phi} \mathcal{L}_{\text{task}}(\phi) + \frac{\beta}{K} \sum_{k=1}^K \alpha_k \nabla_{\Theta} \mathcal{L}_{\text{task}}(\Theta_{\alpha_k}(\phi)) + 2\lambda \phi. \quad (22)$$

Let θ denote the underlying trainable adapter parameters and let $\phi = \phi(\theta)$ be their induced effective update. The implementation obtains gradients through the adapter parameterization by automatic differentiation:

$$\nabla_{\theta} \mathcal{L}_{\text{ReGA}} = J_{\phi}(\theta)^{\top} \nabla_{\phi} \mathcal{L}_{\text{ReGA}}, \quad (23)$$

where $J_{\phi}(\theta)$ is the Jacobian of the effective update with respect to the trainable adapter parameters.

The complementary role of the path term also admits a local interpretation. Let

$$d = \phi - \phi_{ft}, \quad (24)$$

Algorithm 1 ReGA Phase-2 Optimization

Require: Frozen backbone ψ , Phase-1 adapter parameters θ_{ft} , induced task-anchor update $\phi_{ft} = \phi(\theta_{ft})$, training data $\mathcal{D}_{task}$, proximal weight λ , LMC weight β , number of samples K , Phase-2 step budget S_2

- 1: Initialize current adapter parameters: $\theta \leftarrow \theta_{ft}$
- 2: **for** $s = 1$ to S_2 **do**
- 3: Sample minibatch $\mathcal{B} \sim \mathcal{D}_{task}$
- 4: Compute current effective update $\phi \leftarrow \phi(\theta)$
- 5: Compute endpoint task loss:

$$\mathcal{L}_{task} \leftarrow \mathcal{L}_{task}(\psi + \phi; \mathcal{B})$$
- 6: Initialize $\mathcal{L}_{LMC} \leftarrow 0$
- 7: **for** $k = 1$ to K **do**
- 8: Sample $\alpha_k \sim \mathcal{U}[0, 1]$
- 9: Form interpolated effective update:

$$\phi_{\alpha_k} \leftarrow (1 - \alpha_k)\phi_{ft} + \alpha_k\phi$$
- 10: Accumulate path loss:

$$\mathcal{L}_{LMC} \leftarrow \mathcal{L}_{LMC} + \mathcal{L}_{task}(\psi + \phi_{\alpha_k}; \mathcal{B})$$
- 11: **end for**
- 12: $\mathcal{L}_{LMC} \leftarrow \mathcal{L}_{LMC} / K$
- 13: Compute proximal term:

$$\mathcal{R}_{prox} \leftarrow \|\phi\|_2^2$$
- 14: Form total loss:

$$\mathcal{L} \leftarrow \mathcal{L}_{task} + \beta\mathcal{L}_{LMC} + \lambda\mathcal{R}_{prox}$$
- 15: Update θ with AdamW on $\nabla_{\theta}\mathcal{L}$
- 16: **end for**
- 17: **return** $\Theta(\phi(\theta)) = \psi + \phi(\theta)$

and let H_{ft} denote the Hessian of the task loss at the Phase-1 anchor. A second-order expansion along the interpolation path gives

$$\begin{aligned} \mathcal{L}_{task}(\phi_{ft} + \alpha d) &= \mathcal{L}_{task}(\phi_{ft}) + \alpha \nabla \mathcal{L}_{task}(\phi_{ft})^\top d \\ &\quad + \frac{\alpha^2}{2} d^\top H_{ft} d + \mathcal{O}(\|d\|_2^3). \end{aligned} \tag{25}$$

Taking the expectation over $\alpha \sim \mathcal{U}[0, 1]$ yields

$$\begin{aligned} \mathcal{L}_{LMC}(\phi) &= \mathcal{L}_{task}(\phi_{ft}) + \frac{1}{2} \nabla \mathcal{L}_{task}(\phi_{ft})^\top d \\ &\quad + \frac{1}{6} d^\top H_{ft} d + \mathcal{O}(\|d\|_2^3). \end{aligned} \tag{26}$$

When the task anchor is close to a stationary point, the first-order term is small. The path term therefore locally discourages movement along task-sensitive, high-curvature directions. This is a local interpretation rather than a general guarantee of improved zero-shot performance.

B.2 Algorithm Pseudocode

Algorithm 1 provides a PyTorch-style implementation of the ReGA Geometric Alignment phase. We use functional model calls to evaluate interpolated effective updates without modifying the stored task-anchor parameters.

C Additional Quantitative Experiments

C.1 Hyperparameter Sensitivity Analysis

The geometric alignment phase has two principal hyperparameters: the proximal weight λ , which controls the strength of the pull toward the zero-shot checkpoint, and the LMC weight β , which controls the low-loss path constraint.

Sensitivity to Proximal Weight (λ). We investigate the impact of λ by varying it across a logarithmic scale from 10^{-3} to 5.0 on LLaVA-1.5-7B. As visualized in Figure 7, we observe a clear target-performance/retention trade-off. Small values ($\lambda < 0.01$) provide little correction relative to standard LoRA. Conversely, excessively large values ($\lambda > 1.0$) pull the model too strongly toward the prior and degrade the acquired VQA v2 performance. The default $\lambda = 0.1$ provides a strong balance, increasing MMMU from 34.8% for standard LoRA to 36.1% while maintaining high target-task accuracy.

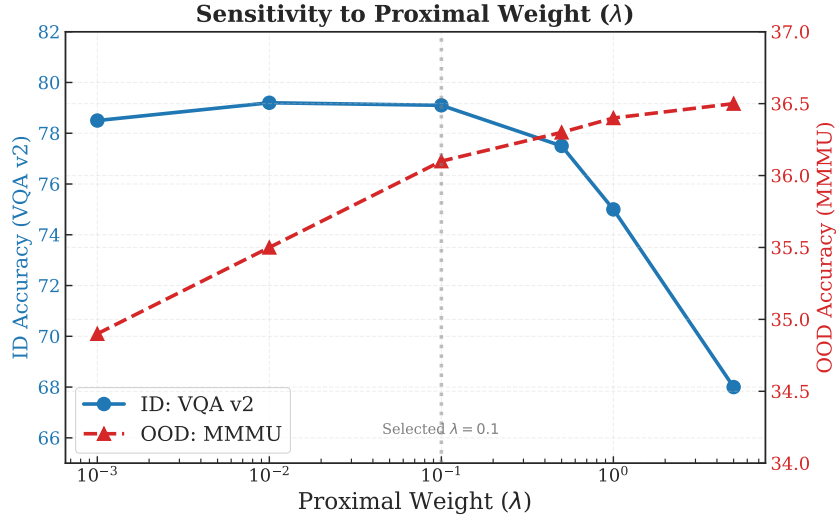


Figure 7: **Impact of Proximal Weight (λ).** We track VQA v2 target-task performance and MMMU zero-shot retention on LLaVA-1.5. The default $\lambda = 0.1$ provides a favorable balance between retaining the Phase-1 task capability and moving toward the zero-shot operating point.

Sensitivity to LMC Weight (β). Table 8 presents the ablation of the low-loss path term. When $\beta = 0$, the objective reduces to proximal-only optimization. Incorporating the path term further improves the evaluated target-performance/retention operating point by constraining the path from the task anchor to the updated solution. The results are relatively stable for $\beta \in [0.5, 2.0]$.

Table 8: Ablation of LMC Weight β on LLaVA-1.5. The default $\beta = 1.0$ provides the strongest overall operating point across the reported metrics.

LMC Weight (β)	Target / Task-Related		Retention	
	VQA v2	OCRBench	MMMU	POPE
0.0 (Proximal Only)	77.8	653	35.8	85.2
0.5	79.0	657	36.0	85.5
1.0 (Default)	79.1	658	36.1	85.7
2.0	78.9	656	36.1	85.6

C.2 Cross-Model Hyperparameter Sensitivity

We additionally vary one coefficient at a time around the default $(\lambda, \beta) = (0.1, 1.0)$ on all three model families. This is a local sensitivity analysis rather than a complete two-dimensional sweep.

Table 9: Cross-model sensitivity to λ and β . The default is $\lambda = 0.1$, $\beta = 1.0$, and $K = 5$.

Model	λ	β	VQA v2	MMMU	POPE
LLaVA-1.5	0.03	1.0	78.7	35.6	85.1
	0.10	0.0	77.8	35.8	85.2
	0.10	0.5	79.0	36.0	85.5
	0.10	1.0	79.1	36.1	85.7
	0.10	2.0	78.9	36.1	85.6
	0.30	1.0	78.6	36.3	86.1
Qwen2.5-VL	0.03	1.0	82.9	67.6	85.7
	0.10	0.0	82.4	67.7	85.9
	0.10	0.5	83.3	67.8	86.1
	0.10	1.0	83.5	67.9	86.2
	0.10	2.0	83.2	67.9	86.2
	0.30	1.0	83.0	68.0	86.4
InternVL-3.5	0.03	1.0	83.7	72.7	88.1
	0.10	0.0	82.9	72.8	88.3
	0.10	0.5	84.0	73.0	88.4
	0.10	1.0	84.2	73.1	88.5
	0.10	2.0	84.0	73.1	88.4
	0.30	1.0	83.8	73.2	88.6

Across all three backbones, β is stable within the tested range $[0.5, 2.0]$. Increasing λ shifts the operating point toward stronger retention with a small reduction in VQA v2. We therefore use $\lambda = 0.1$, $\beta = 1.0$, and $K = 5$ as the default. When retention is prioritized, λ can be moderately increased; when target-task degradation is observed, it should be reduced.

C.3 Stress Test under Stronger Proximal Pulls

To isolate the role of the path term when the pull toward the zero-shot checkpoint is strong, we compare proximal-only optimization and ReGA at $\lambda \in \{0.3, 1.0\}$. The update-norm ratio is computed using the induced effective LoRA updates rather than the raw low-rank factors. We also report $B_{\max}$, the maximum validation cross-entropy loss barrier along the interpolation path from the Phase-1 task anchor to the Phase-2 solution.

At the default $\lambda = 0.1$, the incremental improvement over proximal-only optimization is moderate. Under stronger proximal pulls, the complementary role of the path term becomes

Table 10: Stress test under stronger proximal pulls on LLaVA-1.5-7B. ReGA preserves substantially more target-task performance while maintaining similar retention.

λ	Method	VQA v2	OCRBench	MMMU	POPE	Norm Ratio	$B_{\max}$
0.30	Proximal only	75.7	651	36.2	85.9	0.74	0.231
	ReGA	78.6	657	36.3	86.1	0.77	0.058
1.00	Proximal only	69.5	642	36.5	86.2	0.49	0.386
	ReGA	76.3	653	36.4	86.3	0.57	0.081

clearer: ReGA preserves substantially more target-task performance while maintaining similar MMMU and POPE scores. This stress test characterizes the evaluated operating points but does not establish dominance over every possible proximal-only coefficient.

C.4 Scalability Analysis: Impact of LoRA Rank

To verify that ReGA’s efficacy is not limited to a specific parameter budget, we evaluate performance across varying LoRA ranks $r \in \{8, 16, 64, 128\}$. As shown in Table 11, ReGA consistently outperforms the corresponding LoRA baseline across all tested ranks. The gains are not restricted to low-capacity adapters and remain substantial at ranks 64 and 128.

Table 11: Scalability across LoRA Ranks (r) on LLaVA-1.5-7B. ReGA consistently improves the target-performance/retention trade-off across adapter capacities.

Rank (r)	Method	Target / Task-Related		Retention	
		VQA v2	OCRBench	MMMU	POPE
8	LoRA	76.5	635	35.2	84.1
	ReGA	79.4 (+2.9%)	642	35.8	85.5
16	LoRA	77.2	641	35.0	83.8
	ReGA	79.3 (+2.1%)	649	35.9	85.6
64	LoRA	74.9	652	34.8	83.2
	ReGA	79.1 (+4.2%)	658	36.1	85.7
128	LoRA	74.5	655	34.5	82.9
	ReGA	78.8 (+4.3%)	660	36.2	85.8

C.5 Applicability to Advanced PEFT Mechanisms

To test whether ReGA is limited to standard LoRA, we extend the evaluation to two stronger PEFT variants: **DoRA** (Weight-Decomposed LoRA), which separates magnitude and direction updates, and **AdaLoRA** (Adaptive LoRA), which dynamically allocates the rank budget based on importance scores.

As presented in Table 12, ReGA improves both baselines across all reported metrics. When applied to DoRA, ReGA enhances VQA v2 accuracy by 2.4 points and matches the zero-shot MMMU score of 36.4 in this experiment. These results show that the Phase-2 recipe transfers to the additional PEFT parameterizations evaluated here.

C.6 Ablation on Monte Carlo Sampling Steps

The low-loss path objective relies on a Monte Carlo approximation of the path expectation, where K denotes the number of stochastic samples drawn along the interpolation path. We analyze the trade-off between approximation quality and computational overhead by varying $K \in \{1, 3, 5, 10\}$.

Table 12: ReGA across different PEFT architectures on LLaVA-1.5-7B. ReGA provides consistent gains in target-task performance and retention when integrated with DoRA and AdaLoRA.

Method	Target / Task-Related		Retention	
	VQA v2	OCRBench	MMMU	POPE
<i>DoRA Baseline</i>	77.8	645	35.1	84.5
DoRA + ReGA	80.2 (+2.4%)	661	36.4	86.0
<i>AdaLoRA Baseline</i>	76.2	638	34.9	83.9
AdaLoRA + ReGA	79.5 (+3.3%)	654	36.0	85.6

Table 13 summarizes the results. A single sample provides a less stable approximation and gives lower performance. Increasing K to 3 and 5 improves the evaluated operating point, while increasing K from 5 to 10 provides negligible marginal gains but increases Phase-2 cost. We therefore adopt $K = 5$ as the practical default.

Table 13: Impact of Sampling Steps (K) on Performance and Efficiency. Training time is reported relative to the $K = 1$ setting in Phase 2. $K = 5$ provides the best practical trade-off.

Samples (K)	Relative Time	Target / Task-Related		Retention	
		VQA v2	OCRBench	MMMU	POPE
1	1.00x	78.5	648	35.2	84.8
3	1.35x	78.9	654	35.8	85.3
5 (Default)	1.68x	79.1	658	36.1	85.7
10	2.55x	79.1	659	36.1	85.8

C.7 Seed Sensitivity

The main tables report the canonical training seed used throughout the study. To estimate run-to-run variation, we repeat representative settings using three training seeds. Table 14 reports the mean and standard deviation across these runs.

Table 14: Three-seed results on representative LLaVA-1.5-7B and Qwen2.5-VL-7B settings. Values are mean $\pm$ standard deviation across training seeds.

Model	Method	VQA v2	OCRBench	MMMU	POPE
LLaVA-1.5	LoRA	74.8 $\pm$ 0.3	651.7 $\pm$ 1.5	34.7 $\pm$ 0.2	83.1 $\pm$ 0.3
	LoRA+WD	73.6 $\pm$ 0.4	649.3 $\pm$ 1.2	35.8 $\pm$ 0.2	85.0 $\pm$ 0.2
	Proximal only	77.7 $\pm$ 0.3	653.2 $\pm$ 1.1	35.7 $\pm$ 0.2	85.2 $\pm$ 0.2
	ReGA	79.0$\pm$0.2	657.6$\pm$0.8	36.0$\pm$0.1	85.7$\pm$0.1
Qwen2.5-VL	LoRA	80.4 $\pm$ 0.2	865.0 $\pm$ 0.6	66.8 $\pm$ 0.2	84.2 $\pm$ 0.2
	ReGA	83.4$\pm$0.1	868.2$\pm$0.7	67.9$\pm$0.2	86.1$\pm$0.1

The VQA v2, OCRBench, and POPE gains are larger than the observed seed variation. The smaller MMMU differences should be interpreted as retention improvements within the broader multi-metric pattern rather than as the sole source of ReGA’s benefit.

C.8 Strict Zero-Shot Retention

Because DocVQA, InfoVQA, and TextVQA share OCR and document structure with the OCR-VQA training set, we treat them as related-task transfer rather than strict distribution-

shift evaluations. We define strict zero-shot retention using MMMU, MME, and POPE:

$$R_{\text{strict}}(m) = \frac{100}{3} \sum_{b \in \{\text{MMMU}, \text{MME}, \text{POPE}\}} \frac{s_{m,b}}{s_{\text{ZS},b}}, \quad (27)$$

where $s_{m,b}$ is the score of method m on benchmark b , and $s_{\text{ZS},b}$ is the corresponding zero-shot score.

Table 15: Target-task gap and strict zero-shot retention. ΔVQA is measured relative to FullFT within each backbone. Retention is the macro-average percentage of the zero-shot score retained on MMMU, MME, and POPE.

Model	Method	$\Delta\text{VQA to FullFT}$	Strict Retention (%)
InternVL-3.5	FullFT	0.0	87.1
	LoRA+WD	-4.7	99.0
	ReGA	-0.3	99.8
Qwen2.5-VL	FullFT	0.0	87.1
	LoRA+WD	-5.6	99.2
	ReGA	-0.3	99.8
LLaVA-1.5	FullFT	0.0	82.9
	LoRA+WD	-6.3	99.0
	ReGA	-0.7	99.3

Across the three backbones, ReGA remains within 0.3–0.7 VQA points of FullFT while retaining 99.3–99.8% of the zero-shot score on the strict retention set.

C.9 Comparison with a Stronger Merging Baseline

We additionally compare against MaTS on the common five-metric subset. MaTS improves substantially over TIES, but retains a larger target-task gap than ReGA.

Table 16: Comparison with MaTS on the common five-metric subset.

Model	Method	VQA v2	OCRBench	MMMU	MME	POPE
InternVL-3.5	TIES	76.2	820	70.5	2280.8	85.1
	MaTS	79.4	838	72.7	2364.1	88.0
	ReGA	84.2	845	73.1	2382.4	88.5
Qwen2.5-VL	TIES	75.4	852	65.8	2250.6	83.5
	MaTS	79.0	863	67.6	2332.5	85.8
	ReGA	83.5	868	67.9	2345.8	86.2
LLaVA-1.5	TIES	70.8	640	33.9	1450.6	82.6
	MaTS	73.3	649	35.4	1498.7	85.0
	ReGA	79.1	658	36.1	1493.4	85.7

C.10 Replay-Free Continual Learning on Two Backbones

The detailed main-table results report both final and trajectory-averaged metrics for LLaVA-1.5. To compare the two backbones in a uniform compact format, Table 17 reports the final accuracy on each task after the complete sequence RS→Med→AD→Sci→Fin. Mean Final is the arithmetic mean of the five final task accuracies.

Table 17: Replay-free continual learning on two backbones. All entries are final task accuracies after the complete sequence; no replay buffer is used.

Backbone	Method	RS	Med	AD	Sci	Fin	Mean Final
LLaVA-1.5	LoRA	26.8	25.8	0.8	18.7	70.4	28.5
	O-LoRA	62.7	35.2	16.9	34.4	92.2	48.3
	SEFE	66.7	50.9	39.6	44.8	89.4	58.3
	ReGA	70.2	55.3	45.2	48.1	88.2	61.4
Qwen2.5-VL	LoRA	39.8	35.4	17.2	30.8	77.6	40.2
	O-LoRA	65.1	47.8	34.5	43.2	90.1	56.1
	SEFE	68.5	53.1	42.7	48.6	91.6	60.9
	ReGA	73.4	59.6	49.1	54.3	90.8	65.4

ReGA improves Mean Final over SEFE by 3.1 points on LLaVA-1.5 and 4.5 points on Qwen2.5-VL under the same replay-free task order.

D Computational Cost Analysis

D.1 Training Efficiency and Resource Overhead

We provide a breakdown of the computational resources required by ReGA compared with the baseline methods. Table 18 reports peak VRAM usage and total wall-clock training time for fine-tuning LLaVA-1.5-7B on the composite dataset of 650k samples.

Memory Efficiency. ReGA maintains the memory-efficient profile of PEFT methods. Unlike Full Fine-Tuning, which requires gradients and optimizer states for the complete 7B parameter set, ReGA only trains the lightweight adapters. The measured Phase-2 memory overhead includes the frozen task-anchor update, functional interpolation, additional path evaluations, activation or computation-graph storage, and allocator effects. In our implementation, peak VRAM increases from 14.2 GB for standard LoRA to 16.1 GB for ReGA.

Time Efficiency. Phase 1 is identical to standard LoRA training for 1.0 epoch. Phase 2 uses 0.2 optimizer-step epochs and evaluates one endpoint loss together with $K = 5$ path losses per step, corresponding to a total budget of 2.2 FPE. Because Phase 2 is short, the measured end-to-end training time is 3.2 hours, or approximately 1.28 times that of standard LoRA. This additional cost improves the target-performance/retention trade-off while remaining below the memory requirement of full-parameter training.

Table 18: Computational Cost Breakdown on LLaVA-1.5-7B ($8 \times A100$). Training time includes both Phase 1 and Phase 2 for ReGA.

Method	Precision	Peak VRAM	Total Training Time	Time / LoRA
Full Fine-Tuning	BF16	78.5 GB	3.8 hours	1.52x
L2-SP	BF16	78.5 GB	3.9 hours	1.56x
LoRA ($r = 64$)	BF16	14.2 GB	2.5 hours	1.00x
DoRA	BF16	14.5 GB	2.8 hours	1.12x
ReGA (Ours)	BF16	16.1 GB	3.2 hours	1.28x

E Limitations and Anticipated Failure Modes

Figure 8 illustrates a potential failure mode in which directions toward the zero-shot checkpoint conflict with directions that preserve low task loss around the Phase-1 anchor. When the target task differs substantially from pre-training, there may be no nearby solution that simultaneously retains the specialized task capability and moves meaningfully toward the pretrained operating point. In this case, a strong proximal pull can produce a suboptimal compromise.

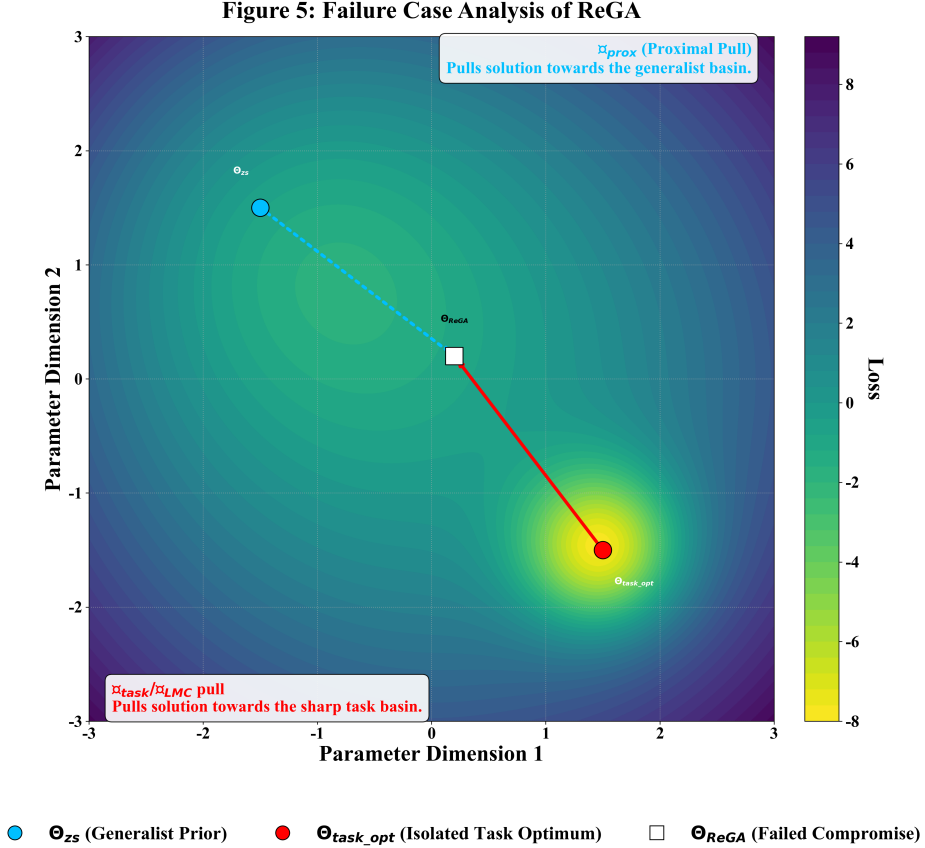


Figure 8: Schematic illustration of a potential failure mode. When the target-task solution is poorly compatible with the zero-shot operating point, the proximal direction may conflict with the task-preserving directions available around the Phase-1 anchor.

Conflict with Incorrect Priors. ReGA assumes that the zero-shot checkpoint provides a useful generalization anchor. If the pre-trained model contains a systematic misconception or bias that the adapted model has corrected, the proximal pull may partially restore that error. For example, a task-adapted model may learn a fine-grained distinction in low-resolution OCR images that the zero-shot checkpoint cannot resolve. Pulling too strongly toward the prior can then degrade performance on these specialized instances. Confidence-aware or parameter-wise anchoring is a natural direction for future work.

Extreme Domain Shift. When the target data are poorly represented in the visual backbone’s pre-training distribution, the zero-shot checkpoint may not provide a useful anchor. In such cases, the proximal term can interfere with learning the novel features required by the target task. ReGA should therefore be viewed as a local post-tuning correction rather than a universal solution to forgetting under arbitrary domain shift.