ACE-LoRA: Adaptive Orthogonal Decoupling for Continual Image Editing

Anonymous authors
Paper under double-blind review

Abstract

We study continual image editing, which aims to progressively expand the capability boundary of diffusion models while preserving previously acquired editing skills. In this paper, we propose ACE-LoRA, an adaptive regularization framework that captures state-dependent task interference through an online interference vector. To enable scalable continual learning, we introduce a rank-invariant historical information compression strategy that consolidates accumulated LoRA updates into a fixed-rank representation while preserving principal task-relevant subspaces. Moreover, we present CIE-Bench, a benchmark comprising six practical image editing tasks and a domain-aware evaluation protocol. Experiments show that ACE-LoRA not only outperforms existing baselines in instruction fidelity and visual realism, but also demonstrates the strongest robustness over long task sequences.

1 Introduction

Continually acquiring new capabilities like human is the central goal of AI, which is also termed as continual learning Parisi et al. (2019); Rolnick et al. (2019); Wang et al. (2022). In this paper, we research on continual learning for image editing, wondering how to enable diffusion models Ho et al. (2020); Dhariwal & Nichol (2021); Rombach et al. (2022) to continually learn across multiple tasks while keeping original abilities. Previous works Xie et al. (2025); Jia et al. (2025) commonly use task-specific LoRA finetuning to expand image editing capabilities of diffusion models; however, separation of task knowledge limits generalization to unseen editing tasks at inference time and hinders knowledge transfer when learning subsequent tasks.

Despite extensive studies of continual learning (CL) in image classification Li & Hoiem (2017); Kirkpatrick et al. (2017) and multimodal large language model (MLLM) instruction tuning Wang et al. (2023); Guo et al. (2025), continual learning for image editing remains a non-trivial problem. Existing methods Wang et al. (2023); Liang & Li (2024); Chen et al. (2025) typically rely on parameter-update constraints computed *offline* before training each task, capturing task-sensitive directions aggregated over the task distribution. However, such *static* constraints can overlook timestep-specific interference, as conflicting directions at different timesteps may cancel out when aggregated at the task level. Prior studies Park et al. (2024); Ma et al. (2025) have shown that diffusion denoisers are timestep-conditioned functions that assume distinct roles across the denoising trajectory. Consequently, the directions that interfere with previously learned knowledge can vary substantially across denoising states, making a single task-level constraint insufficient to characterize such interference.

In this paper, we propose ACE-LoRA, an adaptive regularization framework for continual image editing. ACE-LoRA constructs an **interference vector** from the real-time responses of historical models during training, which is defined as the *gradients of previous parameters under current data*. By enforcing orthogonality between interference vector and current-task updates, the resulting *online* constraint captures state-dependent task interference, effectively mitigating catastrophic forgetting. However, orthogonalizing against accumulated historical interference vectors incurs linearly growing computation cost. We further introduce a rank-invariant historical information compression strategy to enable scalable continual learning. We leverage singular value decomposition (SVD) to compress all historical task parameters into a single fixed-rank representation.

Moreover, there is still a lack of well-established benchmark for continual image editing. To bridge this gap, we introduce CIE-Bench, the first comprehensive benchmark for continual image editing. CIE-Bench consists of 6 sub-tasks (including ERP Outpainting, Refocus, Relighting, Text Editing,

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

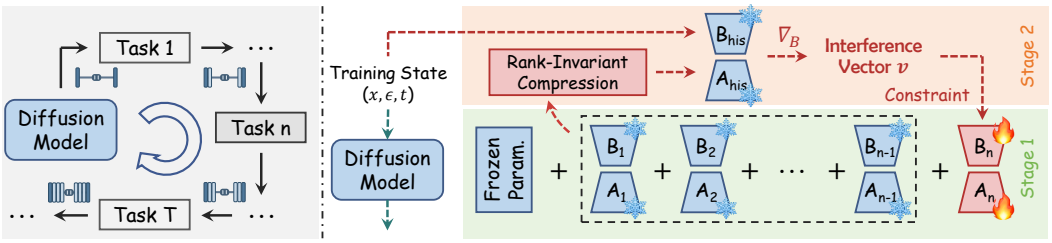


Figure 1: Continual image editing (*left*) and ACE-LoRA (*right*). In the context of continual image editing, diffusion models progressively acquire new capabilities while aiming to retain previously learned ones. We propose ACE-LoRA, which constructs an *online* interference vector from *gradients of previous parameters under current data* to capture state-dependent task interference. Moreover, we propose a two-stage finetuning strategy that first adapts without constraints and then refines with regularization, maintaining effective balance of plasticity and stability.

Virtual Try-On and Causal Reasoning) and is built upon three key principles: 1) **Diversity**: enable comprehensive evaluation across diverse image editing scenarios; 2) **Difficulty**: ensure that tasks are sufficiently challenging to expose model limitations yet effectively learnable; 3) **Practicality**: ensure alignment with real-world deployment. In addition, we develop a domain-specific evaluation protocol, mitigating hallucination issues in general-purpose evaluation metrics Ye et al. (2025).

Extensive experiments demonstrate that ACE-LoRA effectively balances plasticity and stability. ACE-LoRA not only outperforms existing baselines in terms of instruction fidelity (4.38 vs. 3.88) and visual realism (4.49 vs. 3.99), but also demonstrates the strongest robustness over long task sequences (8.87 vs. 7.88 in *Last* overall score), outperforming even vanilla data rehearsal.

2 Methodology

In this section, we present ACE-LoRA. We first formulate continual image editing (Sec. 2.1), then describe our adaptive orthogonalization (Sec. 2.2), rank-invariant historical information compression (Sec. 2.3), and a two-stage fine-tuning scheme (Sec. 2.4) for balancing plasticity and stability.

2.1 Problem Setup

Continual Image Editing. We consider the setup that new tasks arrive sequentially. Each task n is associated with a dataset $\mathcal{D}_n$ and each training example $\boldsymbol{x} = (\boldsymbol{I}^s, c, \boldsymbol{I}^t) \in \mathcal{D}_n$. Here, $\boldsymbol{I}^s$ and $\boldsymbol{I}^t$ are the source and target images, and c is the editing instruction. At each training step, a noise sample $\boldsymbol{\epsilon} \sim p_n(\boldsymbol{\epsilon})$ and a diffusion timestep $t \sim p_n(t)$ are additionally sampled. We denote the corresponding diffusion loss by $\ell_n(\boldsymbol{x}, \boldsymbol{\epsilon}, t; \boldsymbol{\theta})$. After encountering n tasks, the continual learning objective is

$$\min_{\boldsymbol{\theta}} \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}_i, \boldsymbol{\epsilon}\sim p_i(\boldsymbol{\epsilon}),\, t\sim p_i(t)} \left[\ell_i(\boldsymbol{x}, \boldsymbol{\epsilon}, t; \boldsymbol{\theta})\right], \tag{1}$$

Because only $\mathcal{D}_n$ is available during training on task n, the model must learn the current editing capability while preserving those acquired from previous tasks.

LoRA-based Sequential Adaptation. Following existing works Wang et al. (2023); Liang & Li (2024), we employ an incremental LoRA Hu et al. (2022) funetuning framework. Let $\boldsymbol{\theta}_0 \in \mathbb{R}^{d\times k}$ denote the pretrained parameters, we represent the increment introduced by task n using LoRA, with $\boldsymbol{B}_n \in \mathbb{R}^{d\times r}$, $\boldsymbol{A}_n \in \mathbb{R}^{r\times k}$. Current increment, cumulative increment, and effective parameters are

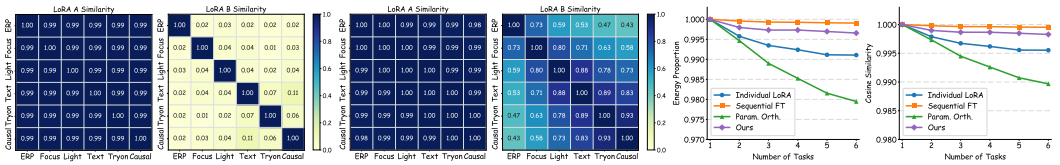
$$\boldsymbol{\delta}_n = \boldsymbol{B}_n\boldsymbol{A}_n, \quad \boldsymbol{\Delta}_n = \boldsymbol{\Delta}_{n-1} + \boldsymbol{\delta}_n, \quad \boldsymbol{\Delta}_0 = \boldsymbol{0}, \tag{2}$$

$$\boldsymbol{\theta}_n = \boldsymbol{\theta}_0 + \boldsymbol{\Delta}_n = \boldsymbol{\theta}_0 + \boldsymbol{\Delta}_{n-1} + \boldsymbol{B}_n\boldsymbol{A}_n. \tag{3}$$

When learning task n, $\boldsymbol{\theta}_0$ and $\boldsymbol{\Delta}_{n-1}$ remain fixed, ; only $\boldsymbol{A}_n$ and $\boldsymbol{B}_n$ are optimized.

2.2 Adaptive Orthogonalization for Continual Image Editing

Existing continual learning methods Wang et al. (2023); Liang & Li (2024); Chen et al. (2025) typically construct parameter-update constraints *offline* before training on a new task. Such constraints



(a) LoRA similarities between tasks under individual FT

(b) LoRA similarities between tasks under sequential FT

(c) Energy proportion of top-r singular values

(d) Reconstructed–original weights cosine similarity

Figure 2: Task-wise LoRA similarity and historical information compression. **(a,b)** compare the similarities of LoRA learned through individual and sequential finetuning. **(c,d)** evaluate the spectral energy retained and the reconstruction cosine similarity under different training schemes.

summarize task-level sensitive directions rather than responding to individual training state. Formally, let $\xi = (\boldsymbol{x}, \boldsymbol{\epsilon}, t)$ denote the training state, where $\boldsymbol{x}$, $\boldsymbol{\epsilon}$, and t are the current data sample, noise, and timestep, respectively. Let $\boldsymbol{r}(\xi)$ be a state-dependent constraint signal and let $\boldsymbol{u}$ denote current parameters to be constrained. An offline method summarizes the constraint signal as

$$\mathcal{L}_{\text{off}}(\boldsymbol{u}) = \langle \boldsymbol{u}, \bar{\boldsymbol{r}} \rangle^2 = \boldsymbol{u}^\top \bar{\boldsymbol{r}} \bar{\boldsymbol{r}}^\top \boldsymbol{u}. \tag{4}$$

In contrast, our adaptive formulation penalizes alignment with the signal observed at each state:

$$\mathcal{L}_{\text{adp}}(\boldsymbol{u}) = \mathbb{E}_\xi \left[\langle \boldsymbol{u}, \boldsymbol{r}(\xi) \rangle^2 \right] = \boldsymbol{u}^\top \mathbb{E}_\xi \left[\boldsymbol{r}(\xi) \boldsymbol{r}(\xi)^\top \right] \boldsymbol{u}. \tag{5}$$

The covariance decomposition makes the difference explicit:

$$\mathcal{L}_{\text{adp}}(\boldsymbol{u}) = \mathcal{L}_{\text{off}}(\boldsymbol{u}) + \boldsymbol{u}^\top \operatorname{Cov}_\xi[\boldsymbol{r}(\xi)] \boldsymbol{u} = \mathcal{L}_{\text{off}}(\boldsymbol{u}) + \operatorname{Var}_\xi \left[\boldsymbol{u}^\top \boldsymbol{r}(\xi) \right]. \tag{6}$$

The additional term measures how strongly the orthogonality constraint is violated across individual states. In diffusion models, different timesteps correspond to distinct stages and functions of the denoising process Park et al. (2024); Ma et al. (2025), causing the corresponding update directions to vary or even conflict across timesteps. Averaging state-dependent signals (*i.e.*, replacing them with task-level expectation) can therefore cancel their directional differences and yield a small offline loss, even when the current update violates the orthogonality constraint at individual states. By directly penalizing the instantaneous alignment, adaptive orthogonalization preserves this state-dependent information and encourages the current update to remain orthogonal across individual training states.

Gradient-Based Interference Signal. We instantiate the state-dependent signal using the *gradients of historical parameters on current task*. Specifically, for each previous task $i < n$, we compute

$$\boldsymbol{v}_{i,n}(\xi) = \nabla_{\boldsymbol{B}_i} \ell_n(\xi; \boldsymbol{\theta}_0 + \sum_{j=1}^{n-1} \boldsymbol{B}_j \boldsymbol{A}_j). \tag{7}$$

Unlike *offline* constraints, which are independent of current training state, $\boldsymbol{v}_{i,n}(\xi)$ measures the local sensitivity of the current-task objective to each historical parameter under current state. It therefore identifies the historical directions that are directly involved in the current optimization and are most susceptible to interference.

We construct this signal in the LoRA-B parameter space. Prior analyses Tian et al. (2024); Hayou et al. (2024) suggest that LoRA-A captures more task-shared representations, whereas LoRA-B is more closely associated with task-specific adaptation. Fig. 2(a,b) exhibits the same pattern: LoRA-A remains relatively similar across tasks, while LoRA-B varies more strongly. Constraining LoRA-A could consequently restrict useful shared knowledge. We instead penalize alignment between the current LoRA-B matrix and the historical interference vectors:

$$\mathcal{L}_{\text{orth}}^{\text{full}}(\xi) = \sum_{i=1}^{n-1} \left| \langle \boldsymbol{v}_{i,n}(\xi), \boldsymbol{B}_n \rangle \right|, \tag{8}$$

In practice, the historical gradients are recomputed online for the sampled data, noise, and timestep. This design allows ACE-LoRA to suppress state-dependent interference while retaining update directions that are not sensitive to previously learned parameters.

2.3 Rank-Invariant Historical Information Compression

While adaptive orthogonalization effectively preserves previous learned knowledge, dynamically computing gradients across an ever-expanding sequence of historical task adapters inevitably leads

to a linear escalation in computational overhead. Drawing inspiration from LoRA merging Tang et al. (2025); Stoica et al. (2024), we introduce a Rank-Invariant Historical Information Compression strategy. Specifically, for task n, we aggregate the LoRA modules across all historical tasks and then deconstruct the merged LoRA weights with singular value decomposition (SVD):

$$\boldsymbol{\Delta}_{n-1} = \sum_{i=1}^{n-1} \boldsymbol{B}_i \boldsymbol{A}_i = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top, \tag{9}$$

By retaining only the top-r largest singular values, we thereby factorize the matrix into a new pair of LoRA weights $\widetilde{\boldsymbol{\Delta}}_{n-1} = \boldsymbol{U}\boldsymbol{\Sigma}_{[:r]}\boldsymbol{V}^T = \boldsymbol{B}_{\text{his}}\boldsymbol{A}_{\text{his}}$, where $\boldsymbol{B}_{\text{his}} = \boldsymbol{U}\boldsymbol{\Sigma}_{[:r]}^{1/2}, \boldsymbol{A}_{\text{his}} = \boldsymbol{\Sigma}_{[:r]}^{1/2}\boldsymbol{V}^\top$. Theoretically, truncated SVD guarantees an optimal rank-r approximation. Since downstream finetuning on limited data naturally induces inherently low-rank weight updates, critical task-adaptive knowledge is heavily concentrated within the principal subspaces spanned by the leading singular values. Then the orthogonalization loss is formulated as:

$$\mathcal{L}_{\text{orth}}(\boldsymbol{\theta}_n; \xi) = \left| \left\langle \nabla_{\boldsymbol{B}_{\text{his}}} \ell_n \left(\xi; \boldsymbol{\theta}_0 + \boldsymbol{B}_{\text{his}}\boldsymbol{A}_{\text{his}}\right), \boldsymbol{B}_n \right\rangle \right|. \tag{10}$$

Fig. 4 shows that historical information compression prevents per-task training cost from growing linearly with the number of tasks. Furthermore, We analyze the energy proportion of top-r singular values under different training strategies as the number of tasks increases. Under ACE-LoRA, this proportion remains high and stable as tasks accumulate (Fig. 2(c)), while the original and reconstructed weights exhibit near-unit cosine similarity (Fig. 2(d)). Together, these results indicate that compression improves scalability with negligible information loss.

2.4 Two-Stage Fine-tuning Scheme

Although orthogonalization protects historical capabilities, imposing constraints throughout optimization can limit model's plasticity. We therefore separate adaptation and stabilization into two stages. The first stage allows the model to approach a task-specific solution without regularization; the second starts from that solution and refines it under constraints that limit interference with previous tasks.

Let $\{\xi_k\}_{k=1}^{K_n}$ denote the training states sampled for task n. Initializing from the previous task's first-stage weights, we optimize

$$\mathcal{L}_n^{(1)} = \frac{1}{K_n} \sum_{k=1}^{K_n} \ell_n \left(\xi_k; \boldsymbol{\theta}_n^{(1)}\right). \tag{11}$$

We then initialize $\boldsymbol{\theta}_n^{(2)}$ from $\boldsymbol{\theta}_n^{(1)}$ and minimize

$$\mathcal{L}_n^{(2)} = \frac{1}{K_n} \sum_{k=1}^{K_n} \left[\ell_n \left(\xi_k; \boldsymbol{\theta}_n^{(2)}\right) + \lambda_{\text{o}} \mathcal{L}_{\text{orth}}(\boldsymbol{\theta}_n^{(2)}; \xi_k) + \lambda_{\text{n}} \mathcal{L}_{\text{norm}}(\boldsymbol{\theta}_n^{(2)}) \right], \tag{12}$$

where $\mathcal{L}_{\text{norm}}(\boldsymbol{\theta}_n) = \|\boldsymbol{\delta}_n\|^2$ limits the magnitude of $\boldsymbol{\delta}_n$, and λ_{o} and λ_{n} control the two regularization terms. By introducing constraints only after initial adaptation, the scheme preserves flexibility in learning the new editing capability while improving stability against interference with historical tasks.

3 CIE-Bench: A Benchmark of Continual Learning for Image Editing

Aiming to systematically evaluate the ability of diffusion models to continually acquire new capabilities while retaining previously learned ones, we introduce CIE-Bench, the first benchmark for continual image editing. CIE-Bench is designed to bridge a critical gap in existing benchmarks and datasets, which largely emphasize static single-task generalization while overlooking the challenges of dynamic and lifelong adaptation. As shown in Fig. 3, CIE-Bench consists of three high-level categories and six sub-tasks, spanning diverse editing granularities from low-level visual adjustments to high-level semantic and compositional reasoning. The statistics are summarized in Appendix. B.2.

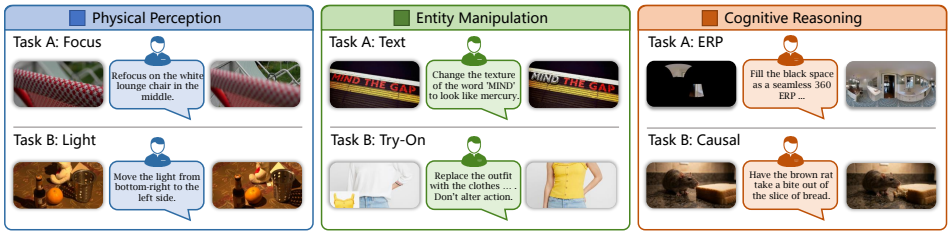


Figure 3: Task categories and representative source image–instruction–target image examples in CIE-Bench. The six tasks span Physical Perception (Refocus and Relighting), Entity Manipulation (Text Editing and Virtual Try-On), and Cognitive Reasoning (ERP Outpainting and Causal Reasoning).

Table 1: Effectiveness of Adaptive Orthogonalization. We compare two constraint strategies, Adaptive (ours) and Static (transferred from OLoRA), and report *Last* and *Imd.* metrics for each task.

Orthogonal Setting		ERP	Focus	Light	Text	Tryon	Causal	**Avg.**	**BWT**
Static	Imd.	9.0403	**8.5877**	8.0246	9.2295	**9.5362**	9.1702	8.9314	-0.5358
	Last	8.3548	6.6794	7.7049	9.2541	9.2101	9.1702	8.3956	
Adaptive (ours)	Imd.	**9.0645**	8.3588	**9.1066**	**9.7869**	9.4855	**9.4298**	**9.2054**	**-0.1702**
	Last	**8.4274**	**7.9466**	**8.1885**	**9.5738**	**9.6449**	**9.4298**	**8.8685**	

Task Selection To construct a representative evaluation suite, the tasks of CIE-Bench are curated based on three key criteria: *comprehensive multi-domain diversity*, *demonstrable adaptation plasticity*, and *substantial practical utility* for real-world scenarios. Critically, the selected tasks must probe the intrinsic capability bottlenecks of existing foundational models while exhibiting robust learnability under few-shot finetuning regimes. This ensures that each incremental learning step represents a meaningful expansion of capability boundaries, accurately simulating the demands of scalable generative systems.

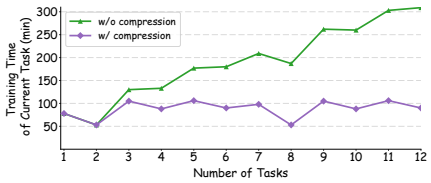


Figure 4: Per-task cost under w/ and w/o historical compression on a 12-task sequence (repeating CIE-Bench tasks).

Guided by the aforementioned principles, we carefully curate a sequence of six representative tasks encompassing Physical Perception, Entity Manipulation, and Cognitive Reasoning. This composition ensures that the benchmark thoroughly tests the model's capacity to decouple and retain complex, non-overlapping data distributions. The final benchmark evaluates sequential learning across the following distinct image editing tasks:

- **ERP Outpainting:** Reconstruct the complete panoramic image from a heavily masked input.
- **Refocus:** Change the focal plane or depth of field of the image.
- **Relighting:** Change the direction of the light source in the image.
- **Text Editing:** Locally edit textual content while preserving consistency.
- **Virtual Try-On:** Synthesize target garments onto a human.
- **Causal Reasoning:** Depict the plausible visual outcome of a specified action or event.

Data curation. CIE-Bench is constructed from high-fidelity image pairs carefully curated for continual image editing tasks. The raw data is collected from a heterogeneous mixture of existing domain-specific datasets, internet-sourced imagery, and outputs generated by state-of-the-art specialized editing models (See in Appendix. B.3), ensuring broad coverage of diverse editing scenarios.

All image pairs are standardized through a unified preprocessing pipeline to maintain consistent resolution, visual quality, and formatting across domains. To guarantee reliable optimization targets, all candidate pairs undergo a rigorous multi-stage manual filtering process. We retain only high-quality instances where the target edit is accurately executed while task-irrelevant regions and structural content remain faithfully preserved. Samples containing artifacts, semantic inconsistencies, excessive distortions, or unintended background modifications are systematically removed.

Evaluation protocol. To accurately evaluate the performance of image editing, we develop a customized and reproducible evaluation framework tailored for CIE-Bench. We adopt one of the strongest open-weight MLLMs Qwen3.5-plus Yang et al. (2025a) (open-source version: Qwen3.5-397B-A17B) as our base evaluator, driven by task-specific evaluation prompts formulated with explicit domain priors. Following Ye et al. (2025), our pipeline introduces a dual-dimensional assessment

Table 2: Quantitative comparison (*Overall Score*) of different methods on CIE-Bench based on Flux2-Klein-9B. We compare ACE-LoRA with representative continual learning methods (including architecture-based and regularization-based methods) and model merging techniques on CIE-Bench. Best and second-best results are marked in **bold** and underlined, respectively

	Method	Replay	ERP	Focus	Light	Text	Virtual Tryon	Causal	Avg.
	Zero-shot	-	7.0645	4.4656	2.9590	5.8852	4.6014	7.1140	5.3483
	MTFT	-	8.9677	9.0076	9.0656	9.7951	9.8768	9.9035	9.4361
Avg	Sequential FT	-	7.4556	6.5947	7.8836	8.4722	9.3365	9.2807	8.1705
Avg	Data Rehearsal	✓	8.2873	8.0657	7.4854	8.3029	9.2933	9.4825	8.4862
Avg	▽ LoRA Merging								
Avg	TIES Yadav et al. (2023)	✗	8.0804	5.4322	5.9881	7.3099	6.0261	8.0263	6.8105
Avg	DARE Yu et al. (2024b)	✗	8.1475	5.7232	4.7378	7.2525	6.0285	8.1053	6.6658
Avg	▽ Architecture-based								
Avg	CL-MoE Huai et al. (2025)	✗	7.7829	7.4330	6.1226	8.5043	8.7794	8.7456	7.8946
Avg	MoELoRA Chen et al. (2024)	✗	7.2554	6.2904	5.7618	**9.2377**	9.1826	**9.5526**	7.8801
Avg	HiDe-LLaVA Guo et al. (2025)	✗	8.0622	4.4349	6.5079	9.1664	2.2547	7.5175	6.3239
Avg	SMoLoRA Wang et al. (2025)	✗	8.0889	7.2518	6.7972	8.1194	7.8651	9.1316	7.8757
Avg	▽ Regularization-based								
Avg	O-LoRA Wang et al. (2023)	✗	8.1142	6.7604	6.1898	7.7523	8.8202	8.7018	7.7231
Avg	RegLoRA Chen et al. (2025)	✗	8.1298	6.9167	7.5932	7.8457	9.0191	8.7632	8.0446
Avg	KeepLoRA Luo et al. (2026)	✗	8.3749	6.1147	7.4302	7.4588	8.5423	8.1842	7.6842
Avg	**ACE-LoRA (ours)**	✗	**8.6344**	**7.9585**	**8.1472**	9.0635	**9.9501**	9.4298	**8.8639**
Last	Sequential FT	-	7.3548	5.8091	4.5901	8.3442	9.2826	9.2807	7.4436
Last	Data Rehearsal	✓	7.9435	8.7405	7.5492	8.9180	9.2826	9.4825	8.6527
Last	▽ LoRA Merging								
Last	TIES Yadav et al. (2023)	✗	8.0887	5.5420	4.5410	7.2705	6.0507	8.0263	6.5865
Last	DARE Yu et al. (2024b)	✗	8.0403	5.2538	4.0902	7.1230	6.1232	8.1053	6.4560
Last	▽ Architecture-based								
Last	CL-MoE Huai et al. (2025)	✗	8.0081	6.1374	4.6148	8.0984	9.2899	8.7456	7.4823
Last	MoELoRA Chen et al. (2024)	✗	7.7258	5.1374	4.3361	7.8279	9.1014	**9.5526**	7.2802
Last	HiDe-LLaVA Guo et al. (2025)	✗	8.4194	4.6947	5.7787	8.3443	2.2826	7.5175	6.1729
Last	SMoLoRA Wang et al. (2025)	✗	7.7177	6.8702	6.6885	7.9180	8.0290	9.1316	7.7258
Last	▽ Regularization-based								
Last	O-LoRA Wang et al. (2023)	✗	8.3710	6.2901	6.2705	7.5082	9.0797	8.7018	7.7035
Last	RegLoRA Chen et al. (2025)	✗	8.0323	6.5725	7.3852	7.4262	9.0725	8.7632	7.8753
Last	KeepLoRA Luo et al. (2026)	✗	8.1048	4.3511	6.4344	7.7377	8.8116	8.1842	7.2707
Last	**ACE-LoRA (ours)**	✗	**8.4274**	**7.9466**	**8.1885**	**9.5738**	**9.6449**	9.4298	**8.8685**

strategy, which includes *Instruction Following* (IF, 1-5 ratings) and *Perceptual Naturalness* (PN, 1-5ratings). IF quantifies adherence to target concepts and domain constraints, whereas PN penalizes unnatural artifacts and unintended alterations in non-target regions. We define the Overall Score as the sum of IF and PN, to characterize the overall editing performance. Together, they offer a granular analysis of model performance aligned with real-world standards.

4 Experiments

4.1 Experimental Setup

Baseline. We compare ACE-LoRA with representative continual learning methods (including architecture-based and regularization-based methods) and model merging techniques on CIE-Bench. Consistent with prior work, we further include **Zero-Shot** (inference with the pretrained model), **Multi-Task Fine-Tuning** (MTFT, jointly finetune on all tasks), and **Sequential Fine-Tuning** (sequentially fine-tunes a single LoRA across all tasks), that are commonly regarded as the empirical lower bound, upper bound, and baseline for continual learning methods, respectively. We also report the performance of Data Rehearsal (retain a subset of samples from previous tasks and replay them during current-task training) as a reference.

Evaluation Metrics. Following prior works Chen et al. (2024); Guo et al. (2025), we evaluate continual learning performance using *Last* and *Avg*. *Last* measures the performance on all tasks after the completion of sequential fine-tuning, while *Avg* reports the average performance across tasks during sequential fine-tuning process. In some experiments, we report the immediate results(*Imd.*) and Backward Transfer (BWT). The former refers to the performance evaluated after training on the current task, while the latter quantifies the average degree of forgetting on previously learned

Table 3: Quantitative comparison (*Overall Score*) of different methods on CIE-Bench based on Qwen-Image-Edit-2509. We compare ACE-LoRA with representative continual learning methods (including architecture-based and regularization-based methods) and model merging techniques on CIE-Bench. Best and second-best results are marked in bold and underlined, respectively.

	Method	Replay	ERP	Focus	Light	Text	Virtual Tryon	Causal	Avg.
	Zero-shot	-	6.7545	5.0856	3.8390	5.6452	5.1114	6.7340	5.5283
	MTFT	-	9.0277	8.9176	9.1756	9.7551	9.8942	9.8335	9.4340
Avg	Sequential FT	-	7.8756	7.1047	8.1736	8.2122	9.1465	9.2143	8.2878
	Data Rehearsal	✓	8.0073	8.4057	7.2754	8.5729	9.1333	9.5725	8.4945
	▽ LoRA Merging								
	TIES Yadav et al. (2023)	✗	7.8904	5.0922	5.4781	7.4499	5.7461	7.8063	6.5772
	DARE Yu et al. (2024b)	✗	8.2675	5.4332	4.4078	7.0425	5.9785	7.8553	6.4975
	▽ Architecture-based								
	HiDe-LLaVA Guo et al. (2025)	✗	7.7522	4.1049	6.0379	<u>9.2136</u>	2.4647	7.1175	6.1151
	SMoLoRA Wang et al. (2025)	✗	**8.9274**	7.8538	<u>8.5462</u>	8.4398	9.0274	<u>9.2456</u>	8.6734
	▽ Regularization-based								
	RegLoRA Chen et al. (2025)	✗	8.4362	<u>8.2741</u>	8.0438	7.9315	<u>9.2417</u>	8.8803	8.4679
	KeepLoRA Luo et al. (2026)	✗	8.1349	6.2647	7.1402	7.2788	8.2723	7.9742	7.5109
	ACE-LoRA (ours)	✗	<u>8.8136</u>	**8.7412**	**9.0273**	**9.3148**	**9.8371**	**9.6234**	**9.2262**
Last	Sequential FT	-	7.7248	6.0991	5.0001	8.0342	9.0426	9.2143	7.5192
	Data Rehearsal	✓	8.0735	8.4305	7.7692	8.6780	9.0026	9.5725	8.5877
	▽ LoRA Merging								
	TIES Yadav et al. (2023)	✗	7.8687	5.2320	4.2710	7.3805	5.8107	7.8063	6.3949
	DARE Yu et al. (2024b)	✗	8.1803	4.9938	3.8602	6.9330	6.2532	7.8553	6.3460
	▽ Architecture-based								
	HiDe-LLaVA Guo et al. (2025)	✗	**8.8094**	4.4847	5.4987	8.1043	2.0926	7.1175	6.0179
	SMoLoRA Wang et al. (2025)	✗	8.3102	7.8089	<u>8.3714</u>	<u>8.3489</u>	8.9583	<u>9.2456</u>	8.5072
	▽ Regularization-based								
	RegLoRA Chen et al. (2025)	✗	8.2514	<u>8.1235</u>	7.9512	7.6523	<u>9.1681</u>	8.8803	8.3361
	KeepLoRA Luo et al. (2026)	✗	7.9248	4.5611	6.6744	7.4677	8.5016	7.9742	7.1840
	ACE-LoRA (ours)	✗	<u>8.7513</u>	**8.3867**	**8.8742**	**9.6115**	**9.7708**	**9.6234**	**9.1696**

Table 4: Ablation studies of Two-Stage Finetuning Scheme. We compare our two-stage fine-tuning scheme with applying constraints throughout training (All Constraint), gradually increasing their weights (Increasing), and reversing the two stages (Reverse).

Setting	ERP	Focus	Light	Text	Tryon	Causal	**Avg.**
All constraint	8.3031	6.2474	7.2650	7.5948	8.3097	9.2257	7.8243
Reverse	7.5023	6.0381	4.5215	8.4356	9.3102	9.2614	7.5115
Increasing	8.0872	7.6772	7.7914	9.4787	8.7963	**9.4928**	8.5539
Ours	**8.4274**	**7.9466**	**8.1885**	**9.5738**	**9.6449**	9.4298	**8.8685**

tasks during the acquisition of new tasks. For each task, we follow our evaluation pipeline to assess instruction following (IF), perceptual naturalness (PN), and overall score (OS = IF + PN).

Implementation Details. See in Appendix. C

4.2 Main Results

4.2.1 Quantitative Evaluation

We compare ACE-LoRA with existing baselines on CIE-Bench. As shown in Tab. 2 and Tab. 3, ACE-LoRA achieves the highest *Avg* and *Last* Overall Scores among sequential methods on both Flux2-Klein-9B Labs (2025) (8.86/8.87) and Qwen-Image-Edit-2509 Wu et al. (2025) (9.23/9.17). These gains extend to both components of the Overall Score: Tab. 10 (in Appendix) shows that ACE-LoRA obtains the highest average *Last* scores for instruction following (4.38) and perceptual naturalness (4.49), indicating improvements in both adherence to editing instructions and visual realism. Furthermore, despite using no replay data, ACE-LoRA exceeds Data Rehearsal in the final average score on both backbones (8.87 vs. 8.65 and 9.17 vs. 8.59, respectively). Its final performance on Text Editing and Virtual Try-On even approaches that of Multi-Task Finetuning.

4.2.2 Qualitative Analysis

Fig. 5 qualitatively compares different methods on CIE-Bench. ACE-LoRA clearly outperforms existing CL baselines in both Instruction Following and Perceptual Naturalness on historical tasks: on Focus, Light, and Text, it faithfully follows the editing prompts and stays close to the ground

Table 5: Effectiveness of different strategies for history information compression (Imd./Last/BWT). We compare our SVD-based compression strategy with random sampling (randomly sampling a single historical task at each training step) and summation (summing LoRA-B over historical tasks)

Method		ERP	Focus	Light	Text	Tryon	Causal	**Avg.**	**BWT**
Random	Imd.	8.8226	8.3125	8.7916	8.2750	9.2147	8.6160	8.6721	-0.2802
	Last	8.3548	7.9375	7.5416	8.7250	9.1764	8.6160	8.3919	
Summation	Imd.	8.1290	7.8281	8.6583	8.2750	9.0809	9.1785	8.5250	-0.3030
	Last	7.8065	6.3359	7.8750	9.0333	9.1029	9.1785	8.2220	
SVD	Imd.	**9.0645**	**8.3588**	**9.1066**	**9.7869**	**9.4855**	**9.4298**	**9.2054**	**-0.1702**
	Last	**8.4274**	**7.9466**	**8.1885**	**9.5738**	**9.6449**	**9.4298**	**8.8685**	

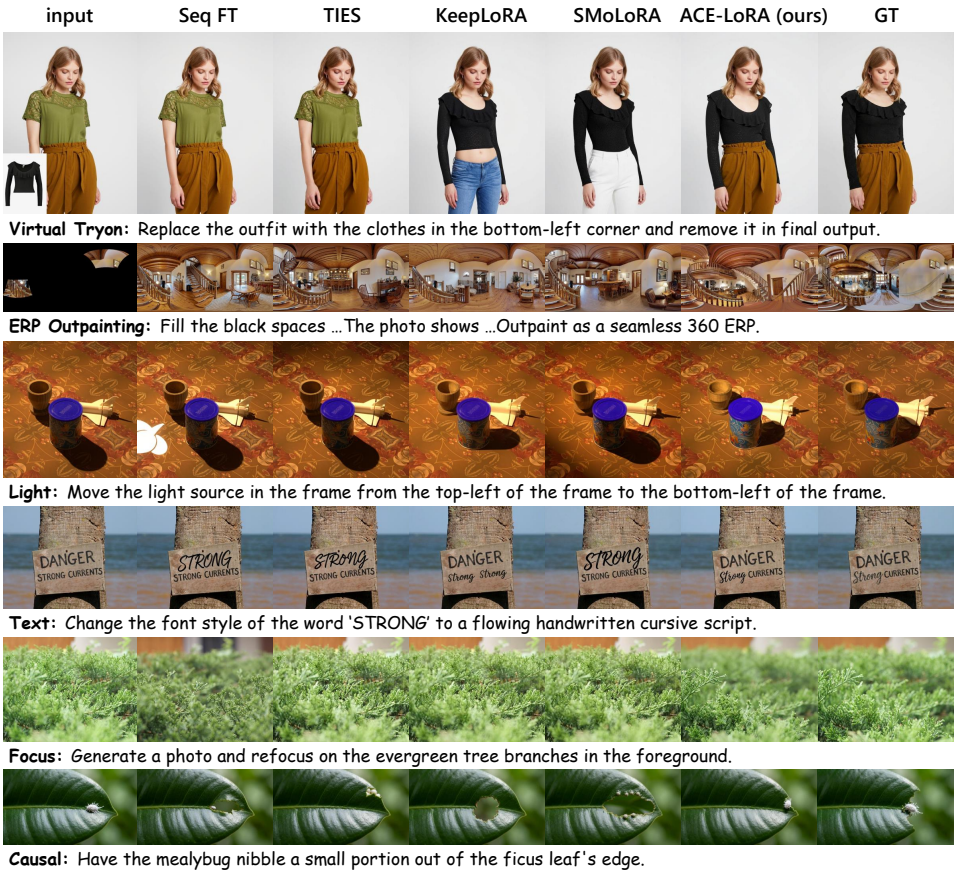


Figure 5: Qualitative comparison of visual results for different methods on CIE-Bench. For each dataset, we selected a representative example and attached the editing instruction above the image.

truth (GT), whereas baselines suffer severe semantic drift or instruction neglect. ACE-LoRA also preserves the structural constraints of past tasks while keeping images natural. For instance, in ERP Outpainting it achieves seamless left–right alignment and global continuity that baselines often break, and in Virtual Try-on it precisely restricts edits to the target garment while preserving other regions, whereas baselines introduce unintended modifications beyond the specified area.

4.3 Model Analysis

Effectiveness of Adaptive Orthogonalization. Tab. 1 compares our *adaptive* orthogonalization with *static* parameter-space orthogonal constraints. Overall, our adaptive orthogonalization consistently outperforms static orthogonalization across both *Imd.* and *Last* metrics. This is due to our state-dependent modeling of interference, which identifies and suppresses only task-conflicting directions while preserving shared optimization directions beneficial for both historical and current tasks. In contrast, static parameter-space orthogonalization enforces uniform constraints on all parameter directions, which may unnecessarily restrict shared knowledge and thus limit plasticity.

Effectiveness of two-stage finetuning scheme. Tab. 4 compares our two-stage fine-tuning scheme with applying constraints throughout training (All Constraint), gradually increasing their weights

Table 6: External validation of ACE-LoRA. We compare ACE-LoRA with representative continual learning methods on a five-task sequence constructed from existing public image-editing datasets. The table reports each method's *Last* score on every task.

Method	Denoising			Moving↑	Reflection			Relight			Rotation↑
	LPIPS↓	FID↓	CMMD↓		PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	
Zero-shot	0.26	63.76	50.02	0.15	21.77	0.81	0.15	13.77	0.66	0.37	0.44
MTFT	0.14	37.96	13.71	0.57	26.87	0.87	0.09	14.92	0.69	0.31	0.52
Sequential FT	0.28	54.46	30.32	0.38	26.89	0.83	0.11	13.96	0.68	0.34	0.50
SMoLoRA	0.19	45.77	18.75	0.41	27.18	0.86	0.09	15.38	0.71	0.30	**0.52**
RegLoRA	0.18	43.73	25.11	0.45	27.05	**0.87**	0.10	15.03	0.74	0.27	0.51
ACE-LoRA (ours)	**0.16**	**39.76**	**16.16**	**0.52**	**27.45**	**0.87**	**0.09**	**15.79**	**0.76**	**0.26**	**0.52**

Table 7: Quantitative comparison under other evaluators. We compare ACE-LoRA with the strongest baselines using GPT-5.5 and human experts as evaluators, and report the *Last* metric of *Overall Score*.

Evaluator	Methods	ERP	Focus	Light	Text	Tryon	Causal	**Avg.**
GPT-5.5	RegLoRA	9.18	5.17	7.62	7.84	9.74	8.04	7.93
	SMoLoRA	8.95	6.19	7.10	7.88	9.04	7.97	7.85
	ACE-LoRA (ours)	**9.34**	**6.68**	**8.69**	**8.43**	**9.88**	**8.11**	**8.52**
Human	RegLoRA	9.42	4.53	8.17	8.33	9.58	8.32	8.06
	SMoLoRA	9.22	5.92	7.68	8.17	9.03	8.07	8.02
	ACE-LoRA (ours)	**9.47**	**6.88**	**8.92**	**8.67**	**9.83**	**8.43**	**8.70**

(Increasing), and reversing the two stages (Reverse). Results demonstrate that our two-stage fine-tuning strategy consistently outperforms other strategy. For reversing the two stages, the model cannot effectively constrain parameter updates during the later phase of each task, resulting in significant forgetting. For All Constraint and Increasing, constraints imposed during training process restrict the parameter freedom of current task and consequently reducing plasticity. In contrast, our two-stage fine-tuning strategy starts the second-stage optimization from an unconstrained local optimum, allowing the model to explore a richer solution space.

Effectiveness of SVD-based historical information compression. Tab. 5 compares random sampling (randomly sampling a single historical task at each training step), summation (summing LoRA-A and LoRA-B over historical tasks), and SVD-based compression. Overall, SVD achieves the highest average scores in *Imd.* and *Last* as well as the least BWT, indicating both more effective mitigation of forgetting and improved plasticity. The advantage of SVD lies in its ability to preserve the principal directions of historical knowledge while filtering redundancy, enabling more accurate interference estimation and maintaining strong model plasticity.

External validation for ACE-LoRA. To further validate the effectiveness of ACE-LoRA, we construct a five-task sequence from public image editing datasets and compare ACE-LoRA with representative continual learning methods (SMoLoRA Wang et al. (2025) and RegLoRA Chen et al. (2025)), as shown in Tab. 6. ACE-LoRA achieves the highest *Last* score on every task, providing strong evidence of its effectiveness and robustness across diverse editing scenarios. Further details about the data sources and evaluation metrics for each task are provided in Appendix. C.

Extra evaluator and human study for MLLM-as-a-Judge. We further adopt closed-source MLLM (GPT-5.5 Singh et al. (2025)) and human experts (3 human experts; results are averaged) to compare ACE-LoRA with the strongest baselines, as shown in Tab. 7. Results show that, under different evaluators, ACE-LoRA achieves the highest *Last* Overall Score on every task, providing further evidence for the effectiveness of ACE-LoRA and the reliability of our evaluation protocol. Details of Human study for our MLLM-as-a-Judge are deferred to Appendix D.

More analysis. We provide more model analysis in Appendix C, including why construct interference vector from LoRA-B and sensitivity to task order and hyperparameters.

5 Conclusion

In this work, we propose ACE-LoRA, a dynamic regularization framework for continual image editing. ACE-LoRA mitigates catastrophic forgetting via adaptive orthogonality and remains scalable with rank-invariant historical information compression. We also introduce CIE-Bench, the first comprehensive benchmark for continual image editing with a standardized evaluation protocol. Extensive experiments demonstrate consistent improvements over strong baselines.

Limitations. ACE-LoRA introduces additional training constraints, increasing training time to about $2\times$ that of sequential finetuning. Like other LoRA-based CL methods, its scalability may be limited as task-specific adapters accumulate across tasks. See the Appendix. E for more details.

AI USE STATEMENT

In this work, we used generative AI tools for generating synthetic datasets, as detailed in Appendix. B.3. We have not used generative AI tools for aiding or polishing writing, retrieval and discovery, research ideation or execution, drafting sections of the paper, and proving mathematical claims. We have reviewed all AI-assisted work. We check all AI-generated data. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Alexandre Araujo, Jean Ponce, and Julien Mairal. Towards real-world focus stacking with deep learning. *arXiv preprint arXiv:2311.17846*, 2023.

Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *CVPR*, pp. 18392–18402, 2023.

Jie Cai, Kangning Yang, Zhiyuan Li, Florin-Alexandru Vasluianu, Radu Timofte, Jinlong Li, Jinglin Shen, Zibo Meng, Junyan Cao, Lu Zhao, et al. Ntire 2026 challenge on single image reflection removal in the wild: Datasets, results, and methods. *arXiv preprint arXiv:2604.10321*, 2026.

Cheng Chen, Junchen Zhu, Xu Luo, Heng T Shen, Jingkuan Song, and Lianli Gao. Coin: A benchmark of continual instruction tuning for multimodel large language models. *NeurIPS*, 2024.

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. 2025.

Aiyu Cui, Jay Mahajan, Viraj Shah, Preeti Gomathinayagam, Chang Liu, and Svetlana Lazebnik. Street tryon: Learning in-the-wild virtual try-on from unpaired person images. In *WACV*, 2025.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *NeurIPS*, 2021.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *ICML*, 2024.

Mehrdad Farajtabar, Navid Azizan, Alex Mott, and Ang Li. Orthogonal gradient descent for continual learning. In *International conference on artificial intelligence and statistics*, 2020.

Haoran Feng, Dizhe Zhang, Xiangtai Li, Bo Du, and Lu Qi. Dit360: High-fidelity panoramic image generation via hybrid training. In *CVPR*, 2026.

Rui Gao and Weiwei Liu. Ddgr: Continual learning with deep diffusion-based generative replay. In *ICML*, 2023.

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.

Soufiane Hayou, Nikhil Ghosh, and Bin Yu. Lora+: Efficient low rank adaptation of large models. 2024.

Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *NeurIPS*, 2017.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 2020.

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

Tianyu Huai, Jie Zhou, Xingjiao Wu, Qin Chen, Qingchun Bai, Ze Zhou, and Liang He. Cl-moe: Enhancing multimodal large language model with dual momentum mixture-of-experts for continual visual question answering. In *CVPR*, 2025.

Zhehao Huang, Yuhang Liu, Yixin Lou, Zhengbao He, Mingzhen He, Wenxing Zhou, Tao Li, Kehan Li, Zeyi Huang, and Xiaolin Huang. T2i-conbench: Text-to-image benchmark for continual post-training. *arXiv preprint arXiv:2505.16875*, 2025.

Sadeep Jayasumana, Srikumar Ramalingam, Andreas Veit, Daniel Glasner, Ayan Chakrabarti, and Sanjiv Kumar. Rethinking fid: Towards a better evaluation metric for image generation. In *CVPR*, 2024.

Qifei Jia, Yu Liu, Yajie Chai, Xintong Yao, Qiming Lu, Yasen Zhang, Runyu Shi, Ying Huang, and Guoquan Zhang. Lego-edit: A general image editing framework with model-level bricks and mllm builder. *arXiv preprint arXiv:2509.12883*, 2025.

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*, 2017.

Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-concept customization of text-to-image diffusion. In *CVPR*, 2023.

Black Forest Labs. FLUX.2: Frontier Visual Intelligence. `https://bfl.ai/blog/flux-2`, 2025.

Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, et al. Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space. *arXiv preprint arXiv:2506.15742*, 2025.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *TPAMI*, 2017.

Yan-Shuo Liang and Wu-Jun Li. Inflora: Interference-free low-rank adaptation for continual learning. In *CVPR*, 2024.

Huiwei Lin, Baoquan Zhang, Shanshan Feng, Xutao Li, and Yunming Ye. Pcr: Proxy-based contrastive replay for online class-incremental continual learning. In *CVPR*, 2023.

Shuaiqi Liu, Weijian Peng, Wenjing Jiang, Yang Yang, Jie Zhao, and Yonggang Su. Multi-focus image fusion dataset and algorithm test in real environment. *Frontiers in Neurorobotics*, 2022.

Mao-Lin Luo, Zi-Hao Zhou, Yi-Lin Zhang, Yuanyu Wan, Tong Wei, and Min-Ling Zhang. Keeplora: Continual learning with residual gradient adaptation. 2026.

Qianli Ma, Xuefei Ning, Dongrui Liu, Li Niu, and Linfeng Zhang. Decouple-then-merge: Finetune diffusion models as multi-task learning. In *CVPR*, 2025.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021.

Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *CVPR*, 2023.

German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural networks*, 2019.

Byeongjun Park, Sangmin Woo, Hyojun Go, Jin-Young Kim, and Changick Kim. Denoising task routing for diffusion models. In *ICLR*, 2024.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. In *NeurIPS*, 2019.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

Juwon Seo, Ji-Su Kang, and Gyeong-Moon Park. Lfs-gan: Lifelong few-shot image generation. In *ICCV*, 2023.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*, 2025.

James Seale Smith, Leonid Karlinsky, Vyshnavi Gutta, Paola Cascante-Bonilla, Donghyun Kim, Assaf Arbelle, Rameswar Panda, Rogerio Feris, and Zsolt Kira. Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning. In *CVPR*, 2023.

George Stoica, Pratik Ramesh, Boglarka Ecsedi, Leshem Choshen, and Judy Hoffman. Model merging with svd to tie the knots. *arXiv preprint arXiv:2410.19735*, 2024.

Dennis Tang, Prateek Yadav, Yi-Lin Sung, Jaehong Yoon, and Mohit Bansal. Lora merging with svd: Understanding interference and preserving performance. In *ICML*, 2025.

Chunlin Tian, Zhan Shi, Zhijiang Guo, Li Li, and Chengzhong Xu. Hydralora: An asymmetric lora architecture for efficient fine-tuning. 2024.

Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuan-Jing Huang. Orthogonal subspace learning for language model continual learning. In *Findings of the Association for Computational Linguistics: EMNLP*, 2023.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 2004.

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.

Ziqi Wang, Chang Che, Qi Wang, Yangyang Li, Zenglin Shi, and Meng Wang. Smolora: Exploring and defying dual catastrophic forgetting in continual visual instruction tuning. In *CVPR*, 2025.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025.

Yicheng Xiao, Wenhu Zhang, Lin Song, Yukang Chen, Wenbo Li, Nan Jiang, Tianhe Ren, Haokun Lin, Wei Huang, Haoyang Huang, et al. Spatialedit: Benchmarking fine-grained image spatial editing. *arXiv preprint arXiv:2604.04911*, 2026.

Ming Xie, Chenjie Cao, Yunuo Cai, Xiangyang Xue, Yu-Gang Jiang, and Yanwei Fu. Anyrefill: A unified, data-efficient framework for left-prompt-guided vision tasks. *arXiv preprint arXiv:2502.11158*, 2025.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. Ties-merging: Resolving interference when merging models. 2023.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a.

Yixiong Yang, Hassan Ahmed Sial, Ramon Baldrich, and Maria Vanrell. Relighting from a single image: Datasets and deep intrinsic-based architecture. *IEEE Transactions on Multimedia*, 2025b.

Yang Ye, Xianyi He, Zongjian Li, Bin Lin, Shenghai Yuan, Zhiyuan Yan, Bohan Hou, and Li Yuan. Imgedit: A unified image editing dataset and benchmark. 2025.

Mustafa Akın Yılmaz, Ahmet Bilican, Burak Can Biner, and Ahmet Murat Tekalp. Edit2restore: Few-shot image restoration via parameter-efficient adaptation of pre-trained editing models. *arXiv preprint arXiv:2601.03391*, 2026.

Jiazuo Yu, Yunzhi Zhuge, Lu Zhang, Ping Hu, Dong Wang, Huchuan Lu, and You He. Boosting continual learning of vision-language models via mixture-of-experts adapters. In *CVPR*, 2024a.

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 *ICML*, 2024b.

Mengyao Zhai, Lei Chen, Frederick Tung, Jiawei He, Megha Nawhal, and Greg Mori. Lifelong gan: Continual learning for conditional image generation. In *ICCV*, 2019.

Bohan Zhang, Huanwei Liang, Yuhan He, Hongteng Xu, Xiaochao Qu, Luoqi Liu, Dixin Luo, and Ting Liu. Consistent feature transport for image relighting. In *ECCV*, 2026.

Juncheng Zhang, Qingmin Liao, Shaojun Liu, Haoyu Ma, Wenming Yang, and Jing-Hao Xue. Real-mff: A large realistic multi-focus image dataset with ground truth. *Pattern Recognition Letters*, 2020.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

Xingchen Zhang. Deep learning-based multi-focus image fusion: A survey and a comparative study. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

APPENDIX

A RELATED WORK

Diffusion-Based Image Editing. Large-scale diffusion models Rombach et al. (2022); Podell et al. (2023); Esser et al. (2024); Labs et al. (2025); Peebles & Xie (2023) have demonstrated remarkable success in synthesizing high-fidelity and semantically complex images from textual prompts. Leveraging these powerful generative priors for image editing, zero-shot approaches formulate editing as controlled trajectory manipulation in the generative process. Specifically, SDEdit Meng et al. (2021) performs zero-shot editing by injecting stochastic noise into input images and subsequently reversing the diffusion process conditioned on target prompts. To better preserve fine-grained structural consistency, Prompt-to-Prompt Hertz et al. (2022) replaces cross-attention maps associated with source text tokens during latent denoising steps. Supervised approaches like InstructPix2Pix Brooks et al. (2023) fine-tune diffusion models on large-scale instruction–image pairs, enabling direct image editing in a single forward pass. For efficient adaptation to new concepts, low-rank adaptation techniques such as LoRA Hu et al. (2022) constrain parameter updates within low-rank subspaces, while methods like CustomDiffusion Kumari et al. (2023) further optimize these adapters for rapid customization of target concepts. However, sequential fine-tuning of low-rank modules in dynamic editing scenarios can lead to unintended interference in the learned subspaces, resulting in progressive degradation of historical knowledge.

Continual Learning in Foundation Models. The central objective of continual learning (CL) is to mitigate catastrophic forgetting in sequential task learning. Current CL methods can be broadly categorized into architecture-, rehearsal-, and regularization-based approaches. Architecture-based methods preserve task-specific knowledge by allocating dedicated model components. For instance, CODA-Prompt Smith et al. (2023) maintains an expandable pool of learnable prompts, while MoE-Adapters Yu et al. (2024a) route representations through specialized expert modules conditioned on the input task. Rehearsal-based methods alleviate forgetting by retaining or approximating past data distributions. For example, DDGR Gao & Liu (2023) leverages pre-trained diffusion models to synthesize representative historical samples, while memory-efficient approaches such as PCR Lin et al. (2023) maintain compact exemplar sets and employ proxy-based contrastive learning to preserve past knowledge. Regularization-based methods constrain parameter updates to reduce interference with previously learned tasks. OGD Farajtabar et al. (2020) enforces orthogonality between current updates and the gradient subspace of prior tasks. Similarly, O-LoRA Wang et al. (2023) mitigates interference by learning task-specific low-rank adapters under orthogonality constraints.

Continual Learning for Generative Models. Preserving generative capabilities under sequential task shifts requires carefully balancing plasticity and stability between historical knowledge and new adaptations. Lifelong GAN Zhai et al. (2019) addresses this issue by applying knowledge distillation on intermediate representations of a frozen generator to maintain consistency. In low-data regimes, LFS-GAN Seo et al. (2023) restricts parameter updates to normalization layers while preserving source-domain fidelity through contrastive patch-level objectives. In diffusion-based generative models, Null-text Inversion Mokady et al. (2023) avoids direct parameter updates by optimizing unconditional text embeddings, enabling faithful reconstruction of real images without modifying core network weights. To systematically evaluate sequential adaptation in text-to-image models, T2I-ConBench Huang et al. (2025) introduces metrics for quantifying semantic degradation across consecutively fine-tuned concepts. Despite these advances, continual learning for image editing remains underexplored. To address this gap, we propose ACE-LoRA, a regularization-based framework tailored for continual image editing, and introduce the first comprehensive benchmark along with a standardized evaluation protocol for assessing continual image editing performance.

B MORE DETAILS OF CIE-BENCH

B.1 MORE EXAMPLES

We present more examples from CIE-Bench in Fig. 6. For each task, we provide three examples, each consisting of the original image (**left**), the edited image (**right**), and the editing instruction displayed above the images.

Table 8: Statistics of CIE-Bench. We report the number of image pairs in the training and testing splits of each task.

	ERP	Focus	Light	Text	Tryon	Causal	All Data
All Data (# pairs)	427	413	409	361	425	337	2372
Training Data (# pairs)	300	300	300	259	294	239	1692
Testing Data (# pairs)	127	113	109	102	131	98	680

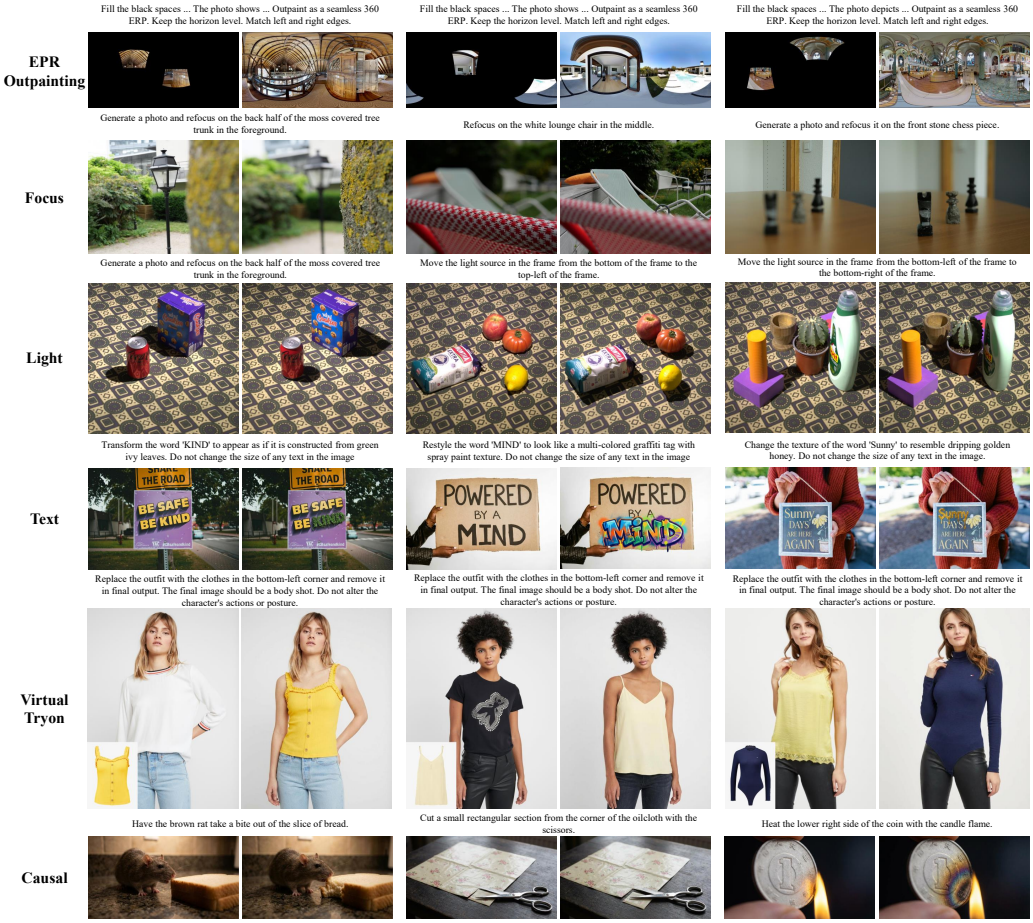


Figure 6: Visualization of CIE-Bench, which consists of six sub-tasks: ERP Outpainting, Refocus, Relighting, Text Editing, Virtual Try-on, and Causal Reasoning

B.2 Statics

Tab. 8 summarizes the statistics of CIE-Bench. Each task contains ∼300 training image pairs and ∼100 test image pairs, yielding 1692 training pairs and 680 test pairs in total.

B.3 Data curation

CIE-Bench is constructed from high-fidelity image pairs carefully curated for continual image editing tasks. The raw data is collected from a heterogeneous mixture of existing domain-specific datasets, internet-sourced imagery, and outputs generated by state-of-the-art specialized editing models, ensuring broad coverage of diverse editing scenarios.

We detail the data sources for each task below:

- ERP Outpainting: Matterport3D_Polished Feng et al. (2026).
- Refocus: MFIF Zhang (2021), HBU-CVMDSP Liu et al. (2022), RealMFF Zhang et al. (2020) and FocusStacking Araujo et al. (2023).

Table 9: Ablation studies of interference vector construction. We compare the results of constructing the interference vector using gradients with respect to LoRA-A, LoRA-B, or the full LoRA-AB.

Setting	ERP	Focus	Light	Text	Tryon	Causal	**Avg.**
LoRA-A	7.5191	5.7956	5.0348	8.0524	7.8574	9.3295	7.2648
LoRA-AB	**8.6435**	6.9906	**8.5732**	9.4605	9.4068	8.7774	8.6420
LoRA-B (ours)	8.4274	**7.9466**	8.1885	**9.5738**	**9.6449**	**9.4298**	**8.8685**



Figure 7: Representative examples comparison between our evaluation metrics and ImgEdit-Judge Ye et al. (2025). Each row shows the editing instruction, input and edited images, and the outputs of both evaluators. These examples illustrate that our protocol provides more fine-grained, instruction-aware assessments of visual details, while being less prone to factual errors than ImgEdit-Judge.

- Relighting: RSR Yang et al. (2025b).
- Text Editing: Source images are public images collected from the Internet, and target images are generated by the closed-source image editing model (Google Nano Banana Pro).
- Virtual Try-On: Street TryOn Cui et al. (2025).
- Causal Reasoning: Both the source and target images are generated by the closed-source image editing model (Google Nano Banana Pro).

C More Analysis

Implementation Details. We adopt Flux2-Klein-9B Labs (2025) and Qwen-Image-Edit-2509 Wu et al. (2025) as the base model. All experiments were conducted on 4 NVIDIA H20 or A800 GPUs. We optimize with AdamW using a learning rate of 1.0×10^{-4}, $\beta_1 = 0.9$, $\beta_2 = 0.99$, and a weight decay of 1.0×10^{-2}. We default use a batch size of 4 and a LoRA rank of 32. In our two-stage fine-tuning scheme, the first and second stages run for 10 and 4 epochs, respectively. During the second stage, we set the orthogonality weight to $\lambda_o = 0.02$ (0.01 for Causal Reasoning) and the norm regularization weight to $\lambda_n = 0.005$.

Why construct interference vector from LoRA-B. Tab. 9 compares the results of constructing interference vector using gradients with respect to LoRA-A, LoRA-B, or the full LoRA-AB parameters. LoRA-B achieves the highest average score, surpassing LoRA-AB and LoRA-A. Results demonstrate that LoRA-B primarily governs task-specific adaptation during fine-tuning, while LoRA-A captures shared representations across tasks. Furthermore, while the interference vector constructed from LoRA-AB can also alleviate forgetting, it introduces substantially higher computational overhead. Therefore, constructing interference constraints based on LoRA-B provides a more effective and efficient solution for continual learning in LoRA-based fine-tuning frameworks.

Table 10: Quantitative comparison (*IF&PN*) of different methods on CIE-Bench. We compare ACE-LoRA with representative continual learning methods and model merging techniques on CIE-Bench. Best and second-best results are marked in bold and underlined, respectively. In the table, we report the *Last* metric of different methods on each dataset, organized by *IF*/*PN*.

Method	Replay	ERP	Focus	Light	Text	Virtual Tryon	Causal	Avg.
Zero-shot	-	3.19/3.87	2.79/1.68	1.19/1.77	3.24/2.65	1.17/3.43	3.17/3.95	2.46/2.89
MTFT	-	3.99/4.98	4.56/4.44	4.56/4.51	4.90/4.89	4.97/4.91	4.92/4.98	4.65/4.78
Sequential FT	-	3.23/4.13	2.72/3.09	2.23/2.36	4.30/4.04	4.72/4.56	4.45/4.83	3.61/3.84
Data Rehearsal	✓	3.09/4.85	4.56/4.18	3.91/3.64	4.46/4.46	4.78/4.50	4.68/4.80	4.25/4.40
▽ LoRA Merging								
TIES Yadav et al. (2023)	✗	3.23/4.85	2.44/3.11	1.88/2.66	3.75/3.52	2.25/3.80	3.67/4.36	2.87/3.72
DARE Yu et al. (2024b)	✗	3.20/4.84	2.42/2.83	1.65/2.44	3.63/3.49	2.25/3.88	3.68/4.43	2.80/3.65
▽ Architecture-based								
CL-MoE Huai et al. (2025)	✗	3.25/4.76	2.81/3.33	2.04/2.57	4.11/3.99	4.78/4.51	4.22/4.53	3.53/3.95
MoELoRA Chen et al. (2024)	✗	3.02/4.70	2.31/2.83	1.98/2.36	4.02/3.80	4.67/4.43	**4.70**/4.85	3.45/3.83
HiDe-LLaVA Guo et al. (2025)	✗	**3.53**/4.89	1.71/2.98	2.56/3.22	4.07/4.27	1.01/1.28	3.51/4.01	2.73/3.44
SMoLoRA Wang et al. (2025)	✗	2.89/4.83	3.58/3.29	3.43/3.26	4.06/3.86	4.36/3.67	4.49/4.64	3.80/3.93
▽ Regularization-based								
O-LoRA Wang et al. (2023)	✗	3.42/4.95	3.07/3.22	3.20/3.07	3.87/3.64	4.70/4.38	4.16/4.54	3.74/3.97
RegLoRA Chen et al. (2025)	✗	3.13/4.90	3.56/3.01	3.87/3.52	3.81/3.61	4.78/4.29	4.15/4.61	3.88/3.99
KeepLoRA Luo et al. (2026)	✗	3.23/4.87	1.88/2.47	3.29/3.15	4.04/3.70	4.57/4.24	3.99/4.19	3.50/3.77
ACE-LoRA (ours)	✗	3.43/**5.00**	**4.31/3.63**	**4.26/3.93**	**4.80/4.77**	**4.93/4.72**	4.56/**4.87**	**4.38/4.49**

Table 11: Results of adopting different task orders. In the table, we report the *Last* metric of each task under different task orders. We use the initials to represent different tasks; see Sec. 4.3 for details.

Task Order	ERP	Focus	Light	Text	Tryon	Causal	**Avg.**
CVLEFT	8.0806	7.7656	7.9583	9.6825	9.3676	8.9464	8.6335
FVELTC	8.1452	7.5234	8.0081	9.2899	9.1618	9.0797	8.5347
EFLTVC	8.4274	7.9466	8.1885	9.5738	9.6449	9.4298	8.8685

Sensitivity to task order. We evaluate ACE-LoRA under different task sequences (Tab. 11) and observe stable performance across orders (Avg: 8.53–8.87), indicating low sensitivity to task permutations.

Sensitivity to hyperparameters. To assess the sensitivity of ACE-LoRA to hyperparameter choices, we evaluate different settings, as shown in Tab. 12. We set $\lambda_o = 0.02, \lambda_n = 0.005$ as the default configuration; each column reports the *Last* score obtained by varying one hyperparameter while keeping the other fixed. ACE-LoRA maintains high and stable performance within $\lambda_o \in [0.005, 0.05], \lambda_n \in [0.01, 0.1]$, indicating low sensitivity to these choices and strong robustness to their variation. This further supports the practicality of ACE-LoRA.

External validation for ACE-LoRA. To further validate the effectiveness of ACE-LoRA, we conduct an external evaluation using public image editing datasets. Specifically, we construct a five-task sequence from multiple datasets and compare ACE-LoRA with representative continual learning methods. We provide the data sources and evaluation metrics for each task in detail here; the analysis of the results is presented in the main text.

- Denoising: We randomly sampled 160 pairs from Edit2Restore Yılmaz et al. (2026) train set for training and 100 pairs from Edit2Restore test set for testing. We used LPIPS Zhang et al. (2018), FID Heusel et al. (2017) and CMMD Jayasumana et al. (2024) as the evaluation metric, consistent with Edit2Restore.
- Moving: We randomly sampled 200 pairs from SpatialEdit500K Xiao et al. (2026) object moving subset for training and 100 pairs from SpatialEdit-Bench Xiao et al. (2026) object moving subset for testing. We used *Moving Score* as the evaluation metric, consistent with SpatialEdit-Bench.

Table 12: Ablation studies of sensitivity to hyperparameters. We conducted separate experiments to evaluate λ_o and λ_n.

Hyperparam.	λ_o					λ_n				
	0.005	0.01	**0.02**	0.03	0.05	0.001	0.003	**0.005**	0.007	0.01
ACE-LoRA	8.5517	8.7642	8.8685	8.6104	8.7213	8.5668	8.6907	8.8685	8.7529	8.4723

Table 13: Alignment ratio of MLLM evaluator with human preferences in absolute scores.

Score	ERP	Focus	Light	Text	Tryon	Causal	**Avg.**
Instruction Following	0.4762	0.6571	0.8190	0.8000	0.9143	0.7619	0.7381
Perceptual Naturalness	0.8667	0.7143	0.9476	0.8762	0.6476	0.8095	0.8103
Overall Score	0.6476	0.6095	0.8667	0.7048	0.6667	0.7333	0.7048

Table 14: Alignment ratio of MLLM evaluator with human preferences in relative rankings.

Score	ERP	Focus	Light	Text	Tryon	Causal	**Avg.**
Instruction Following	0.6800	0.7708	0.9286	0.9811	0.9190	0.6667	0.8244
Perceptual Naturalness	0.9600	0.7778	0.9655	0.9333	0.5625	0.8889	0.8480
Overall Score	0.8293	0.7126	0.9508	0.9200	0.8621	0.7021	0.8295

- Reflection : We randomly sampled 300 pairs from OpenRR Cai et al. (2026) train set for training and 100 pairs from OpenRR val set for testing. We used PSNR, SSIM Wang et al. (2004) and LPIPS Zhang et al. (2018) as the evaluation metric, consistent with OpenRR.
- Relight: We randomly sampled 300/100 pairs from CFT Zhang et al. (2026) test set for training/testing (no overlap). We used PSNR, SSIM Wang et al. (2004) and LPIPS Zhang et al. (2018) as the evaluation metric, consistent with CFT.
- Rotation : We randomly sampled 200 pairs from SpatialEdit500K Xiao et al. (2026) object rotation subset for training and 100 pairs from SpatialEdit-Bench Xiao et al. (2026) object rotation subset for testing. We used *Rotation Score* as the evaluation metric, consistent with SpatialEdit-Bench.

D Human Study for MLLM-as-a-Judge

To validate the effectiveness of the MLLM-as-a-Judge framework, we conduct a human study from both absolute and relative perspectives. We select 90 editing samples, with 15 samples for each task, covering diverse editing quality levels.

For absolute evaluation, following Ye et al. (2025), we compare the scores assigned by the MLLM evaluator with those given by human experts for each edited sample. A match is counted when the absolute score difference is within 1.

For relative evaluation, we assess pairwise ranking agreement, i.e., whether the MLLM evaluator and human experts agree on which sample in each pair exhibits higher editing quality. This setting is motivated by potential discrepancies in absolute score calibration, particularly for partially successful edits.

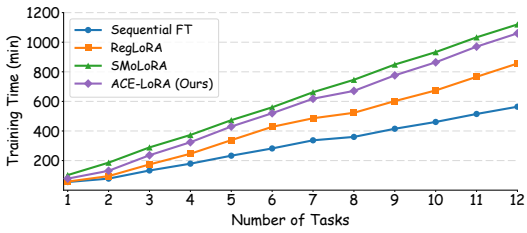


Figure 8: Comparison of training cost under different number of tasks for Sequential FT, existing CL baselines (RegLoRA and SMoLoRA) and ACE-LoRA (ours)

E Limitations

ACE-LoRA introduces additional training constraints, increasing training time to about $2\times$ that of sequential finetuning. Fig. 8 compares the total training cost of various methods under different numbers of tasks. The results show that the overall training cost of ACE-LoRA is roughly twice that of Sequential Fine-tuning. Furthermore, Like other LoRA-based CL methods, its scalability may be limited as task-specific adapters accumulate across tasks.