EraseLoRA: MLLM-Driven Foreground Exclusion and Background Subtype Aggregation for Dataset-Free Object Removal

Sanghyun Jo[1,2⋆†], Donghwan Lee[2⋆], Eunji Jung[2⋆], Seong Je Oh[2], and Kyungsu Kim[2†]

[1] OGQ, Seoul, Korea
[2] Seoul National University, Seoul, Korea

Abstract. Object removal must prevent the masked target from reappearing and reconstruct the occluded background with structural and contextual fidelity, rather than merely filling a hole plausibly. Recent dataset-free approaches manipulate the diffusion model's internal self-attention to prevent it from referencing the masked region, yet they fail in two critical ways: (i) they treat the masked region as the sole foreground, misinterpreting non-target objects as background and regenerating them, and (ii) they apply uniform attention constraints without distinguishing diverse background subtypes, leading to textural blurring and structural misalignment. Both failures stem from the absence of explicit background-aware reasoning. We propose EraseLoRA, a dataset-free framework that replaces attention surgery with background-aware reasoning and test-time adaptation. The first stage, Background-aware Foreground Exclusion (BFE), leverages a multimodal large-language model to separate target foreground, non-target foregrounds, and clean background from a single image–mask pair. The second stage, Background-aware Reconstruction with Subtype Aggregation (BRSA), performs test-time optimization that treats inferred background subtypes as complementary pieces, enforcing their consistent integration through reconstruction and alignment objectives without explicit attention intervention. As a model-agnostic plug-in applicable to diverse diffusion backbones, EraseLoRA reconstructs backgrounds at least 23% more faithful to the original scene than previous dataset-free methods while nearly halving unwanted foreground re-generation, and surpasses all dataset-driven approaches in both aspects despite requiring no training data. Code is available at `https://shjo-april.github.io/EraseLoRA`.

Keywords: object removal · multimodal large-language model · test-time adaptation · dataset-free · attention

⋆ Equal contribution.
† Corresponding authors: `shjo.april@gmail.com`, `kyskim@snu.ac.kr`

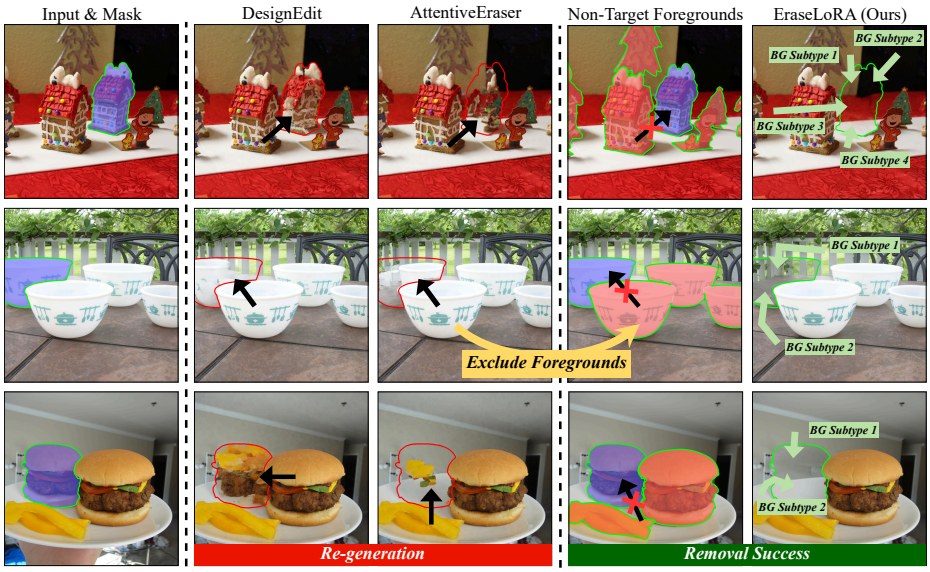


Fig. 1: Qualitative comparison with prior dataset-free methods. Previous state-of-the-art approaches [13,34] treat only the masked region as foreground, misinterpreting non-target objects as background and regenerating them. EraseLoRA identifies and excludes non-target foregrounds and reconstructs the masked region using various background subtypes, enabling faithful object removal.

1 Introduction

Image inpainting methods based on GANs [10, 21, 35] and text-to-image diffusion models [7, 11, 26, 30] can synthesize visually plausible content in missing regions, yet they primarily aim to generate realistic textures rather than restore the underlying background structure, often hallucinating new objects instead of faithfully reconstructing what lies behind the removed target.

Object removal, in contrast, requires both eliminating the target and recovering the occluded background by transferring clean background cues into the masked region with structural consistency. Recent dataset-free diffusion methods [13,34] attempt to achieve this by redirecting or blocking self-attention within the masked region so that the model focuses on unmasked context. While effective in simple cases, these approaches share two inherent limitations. First, they treat the masked region as the only foreground and often misinterpret non-target foregrounds outside the mask as background, causing unintended regeneration of objects (see Fig. 1). This reflects a lack of background-aware reasoning. Second, constraining attention uniformly without distinguishing distinct background subtypes compromises local detail and fails to coherently integrate multiple background cues, leading to blurred textures and unnatural boundaries between disparate subtypes (see Fig. 2; further analysis in Appendix B.2).

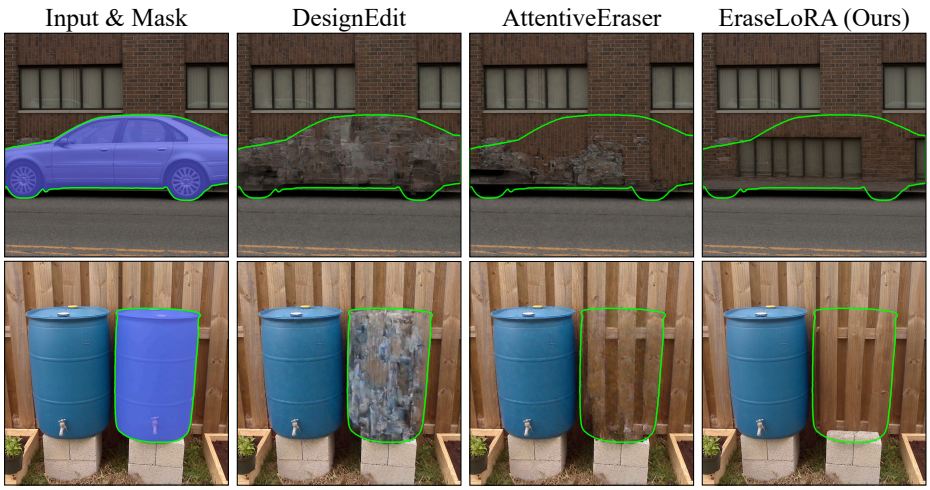


Fig. 2: Artifacts from attention manipulation. Recent dataset-free methods [13, 34] directly modify attention inside the mask without identifying background cues, leading to blurred or distorted textures, whereas EraseLoRA aggregates background subtypes without attention blocking and preserves sharp, coherent structures.

We introduce EraseLoRA, a dataset-free framework that addresses these issues through background-aware reasoning and test-time adaptation. The first stage, Background-aware Foreground Exclusion (BFE), leverages a multimodal large-language model (MLLM) [2, 51] to produce clean background cues by separating target foreground, non-target foregrounds, and background from a single image–mask pair. The second stage, Background-aware Reconstruction with Subtype Aggregation (BRSA), performs test-time optimization with Low-Rank Adaptation (LoRA) [12] to inject these cues into the masked area and aggregate multiple inferred background subtypes into a coherent reconstruction without a dataset-level removal prior or explicit attention blocking. EraseLoRA is validated as a plug-in across diverse pretrained diffusion backbones [7, 19] and standard object-removal benchmarks [18,31], demonstrating consistent gains over dataset-free baselines and competitive performance with dataset-driven methods. By combining the reasoning capability of MLLMs with the generative fidelity of diffusion models, EraseLoRA establishes an extensible formulation of dataset-free object removal that requires no additional data or retraining.

Our key contributions are as follows:

- We identify a fundamental failure mode in object removal: non-target foregrounds are frequently misinterpreted as background, causing their unintended regeneration across recent dataset-free methods.
- We propose EraseLoRA, a background-aware, dataset-free object-removal framework that combines MLLM-guided separation of target and non-target foregrounds from background with a multi-background-aware test-time adap-

Table 1: Conceptual comparison of EraseLoRA (Ours) with previous approaches for object removal.

Properties	[ECCV'24] PowerPaint [54]	[CVPR'25] EntityErasure [53]	[CVPR'25] SmartEraser [14]	[CVPR'26] ObjectClear [47]	[AAAI'25] DesignEdit [13]	[AAAI'25] AttentiveEraser [34]	**EraseLoRA**
Dataset-free object removal	✗	✗	✗	✗	✓	✓	✓
Identifies non-target foregrounds with backgrounds	✗	✗	✗	✗	✗	✗	✓
Leverages multiple background subtypes	✗	✗	✗	✗	✗	✗	✓
Model-agnostic applicability	✗	✗	✗	✗	✓	✓	✓

tation scheme, preventing foreground regeneration while maintaining contextual coherence.
- We provide three-label ground-truth annotations that distinguish target foreground, non-target foreground, and background, along with two evaluation metrics designed for unpaired object-removal settings.
- EraseLoRA improves background fidelity by at least 23% over previous dataset-free methods while nearly halving unwanted foreground re-generation, and retains these gains when diffusion backbones and MLLMs are each replaced by alternatives.

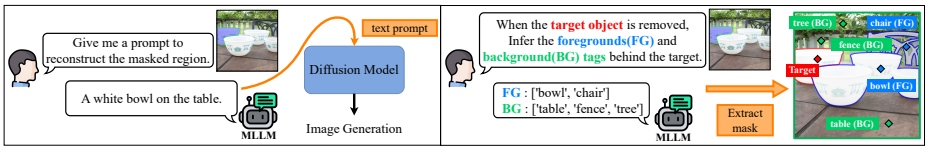
2 Related Work

2.1 Image Inpainting with Generative Models

Image inpainting aims to complete missing regions using the visible context. Early approaches [32, 49, 55] based on GANs have been surpassed by diffusion-based methods [41, 43], which produce stable, detailed, and high-fidelity completions. Building on text-to-image diffusion backbones [26, 30], existing methods [25, 41] fine-tune these backbones on paired inpainting datasets so that they can exploit text prompts while learning to fill masked regions. However, these approaches are still optimized for context-consistent completion and tend to hallucinate plausible new objects rather than preserving the original scene content.

2.2 Diffusion Models for Object Removal

Object removal is a specialized form of inpainting that must not only erase the masked target but also restore the occluded background with structural and contextual fidelity while preserving target-unrelated regions. Existing methods fall into two categories: dataset-driven approaches [6, 14, 24, 54], which learn removal priors from additional, often paired before/after data, and dataset-free approaches [5, 13, 34], which operate directly on pretrained text-to-image diffusion models [7, 26, 30] without additional data. However, dataset-driven methods inherit a static training distribution and do not explicitly distinguish non-target foregrounds from background, which can leave object traces or perturb target-unrelated regions in complex scenes. Moreover, constructing paired examples



(a) Previous MLLM's Use in Image Generation (b) Our MLLM's Use for Object Removal

Fig. 3: Background-aware reasoning power of MLLM. Unlike prior works [15, 27, 38, 50] employ MLLMs for visual reasoning over the visible scene, we first leverage MLLMs to infer background cues behind the masked target.

is expensive and often unrealistic, since most pairs must be synthesized or extracted from video frames. These limitations motivate dataset-free methods that control the diffusion process at inference time. Recent extensions address object effects such as shadows and reflections [9, 40, 47]. While related, they pursue a broader removal target and can distort target-unrelated regions, which object removal should preserve. We focus on removal within the given mask, prioritizing faithful restoration and preservation of target-unrelated regions.

Recent state-of-the-art dataset-free approaches [13, 34] redirect or suppress self-attention within the masked region to guide the model toward unmasked context. While this attention manipulation reduces unwanted regeneration to some extent, it introduces two fundamental limitations. First, these methods remain background-agnostic: by treating only the masked area as foreground, they often misinterpret non-target foregrounds as background and regenerate them, as shown in Fig. 1. Second, blocking or altering attention disrupts fine textures and prevents consistent integration of multiple background cues, leading to blurry or structurally inconsistent results, as illustrated in Fig. 2.

To address these limitations, we introduce a background-aware, dataset-free framework that leverages visual reasoning from an external model to identify and exclude non-target foregrounds and to produce clean background cues for reconstruction. We further aggregate multiple inferred background subtypes coherently through test-time adaptation, enabling structurally consistent background restoration without a dataset-level removal prior or explicit attention blocking. Our method is plug-and-play and model-agnostic, applicable across diverse diffusion backbones. Tab. 1 summarizes this design, highlighting how EraseLoRA differs from previous state-of-the-art methods [13, 14, 34, 53].

2.3 MLLMs for Visual Reasoning in Image Editing

Multimodal large-language models (MLLMs) [1, 2, 22, 51] have gained traction in vision-language tasks due to their ability to interpret visual scenes and reason about object relations. Recent editing methods [15, 38] leverage this capability to extract semantic descriptions, generate editing instructions, or guide global scene manipulation, while inpainting works [8, 42, 50] use MLLMs to analyze the visible context and propose content to fill masked regions. However, these

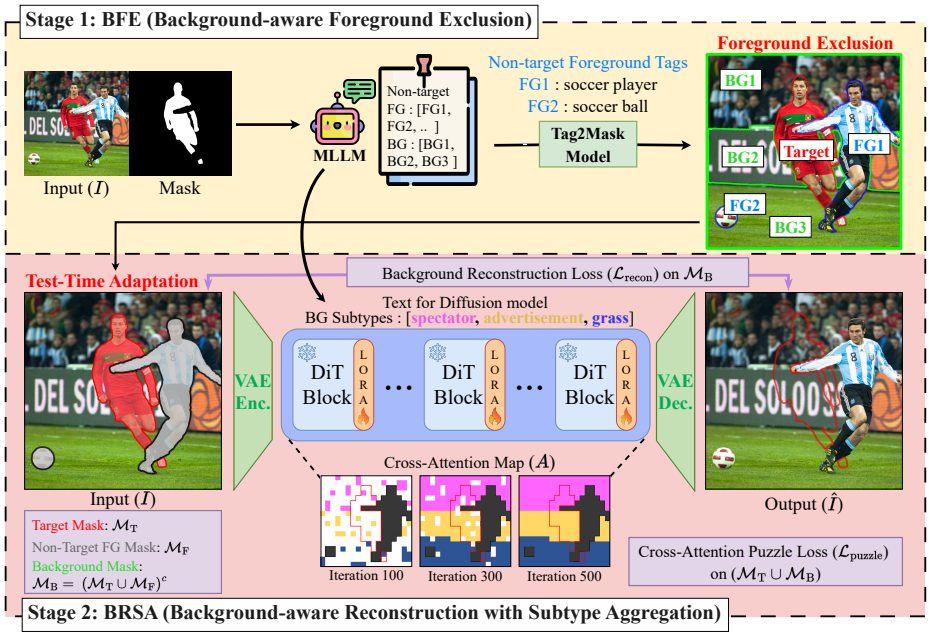


Fig. 4: Overview of EraseLoRA. BFE (Sec. 3.1) separates target foreground, non-target foregrounds, and background from a single image-mask pair using an MLLM [51] and Tag2Mask models [23,28]. After producing clean background cues, BRSA (Sec. 3.2) performs test-time adaptation [37] with reconstruction and alignment objectives, coherently integrating background subtypes into the masked region.

approaches rely on visible context and aim to generate new objects, rather than inferring what lies behind a removed target.

We leverage MLLMs in a fundamentally different role: as background-aware reasoners for object removal. Rather than generating new foreground content, we use MLLMs to infer the occluded background behind the target and to identify non-target foregrounds that cause unintended regeneration, producing clean background cues that are not directly visible in the input image (see Fig. 3).

3 Method

Our proposed EraseLoRA is a dataset-free object removal framework that leverages MLLM-guided background reasoning and test-time adaptation to achieve coherent background reconstruction. It consists of two stages: Background-aware Foreground Exclusion (BFE; Sec. 3.1) and Background-aware Reconstruction with Subtype Aggregation (BRSA; Sec. 3.2). The overall pipeline is illustrated in Fig. 4, and we provide diffusion and attention preliminaries in Appendix A.

Fig. 5: Identification of non-target foregrounds. Prior methods [5, 34] treat the entire unmasked region as background, which causes regeneration of non-target foregrounds. In contrast, EraseLoRA explicitly identifies non-target foregrounds within the unmasked region and excludes them, producing clean background.

3.1 Background-aware Foreground Exclusion

The first stage, BFE, prevents unintended object regeneration by explicitly excluding non-target foregrounds from reference regions and extracting clean background cues for contextually coherent reconstruction. Given an input image I and a target mask M_T, we leverage the background-aware reasoning of MLLMs [2,51] to partition target foreground, non-target foregrounds, and background. The MLLM first identifies all semantic tags in the image and classifies the masked object as the target foreground, visible objects that may cause regeneration as non-target foreground tags $\mathcal{F}$, and occluded objects or scene components behind the target as background subtype tags $\mathcal{B}$. For each non-target foreground tag in $\mathcal{F}$, we use Tag2Mask models (*e.g.*, Grounding DINO [23] and SAM2 [28]) to localize its corresponding region. The union of these localized regions defines the non-target foreground mask; the remaining pixels outside both the target and non-target foreground regions are treated as clean background.

We define three binary masks M_T, M_F, and M_B over the latent spatial domain $\Omega = \{1, \ldots, h \times w\}$. Following the VAE architecture of the employed diffusion backbone (*e.g.*, [7]), we set $h{=}H/8$ and $w{=}W/8$ for an input of size $H \times W$:

$$\Omega = M_T \cup M_F \cup M_B, \tag{1}$$

where M_T, M_F, and M_B denote the target, non-target foreground, and clean background masks in the latent space, respectively. This partition explicitly distinguishes non-target foreground objects from the unmasked background region, allowing us to isolate distractors that previous dataset-free methods [13, 34] mistakenly treat as background (see Fig. 5), which in turn yields cleaner background supervision for subsequent reconstruction.

3.2 Background-aware Reconstruction with Subtype Aggregation

Based on the clean background cues obtained in BFE, BRSA performs test-time optimization with Low-Rank Adaptation (LoRA) [12] to effectively aggregate

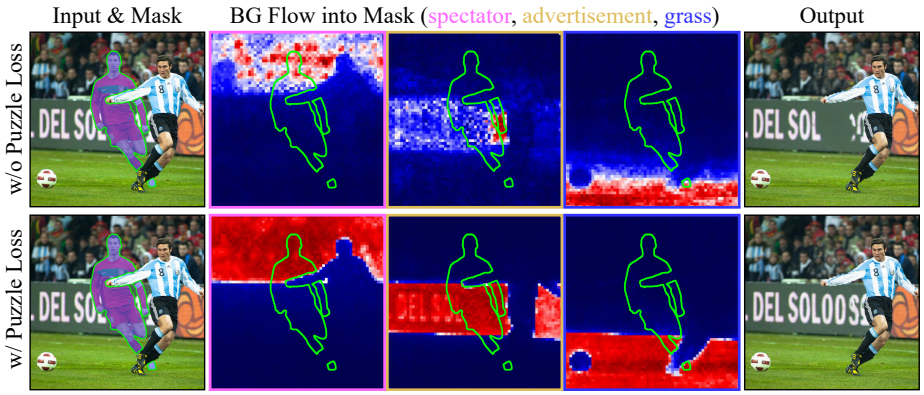


Fig. 6: Effect of the background puzzle loss. We visualize how each background subtype (spectator, advertisement, grass) is represented inside the mask. The background puzzle loss ensures structurally coherent integration of background subtypes within the mask, unlike the weak integration without it.

multiple background subtypes and reconstruct the masked region with structural and contextual consistency. This adaptation provides image-specific background fidelity in dataset-free setting without artifacts from attention manipulation. To achieve this, BRSA jointly optimizes two complementary objectives: the Background Reconstruction Loss ($\mathcal{L}_{\text{recon}}$) and the Background Puzzle Loss ($\mathcal{L}_{\text{puzzle}}$). The overall objective is formulated as $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{recon}} + \lambda \mathcal{L}_{\text{puzzle}}$, where λ=0.2 balances background reconstruction fidelity and subtype aggregation. Together, these losses guide how background information is integrated into the masked region, softly regulating attention flow without the hard attention-blocking used in prior methods [13, 34].

Background Reconstruction Loss. To preserve regions that are confidently identified as clean background by BFE (Sec. 3.1), we impose a reconstruction loss only on M_B. Let $z = \text{Enc}(I)$ be the latent representation of the input image I, and $\hat{z}$ be the reconstructed latent after denoising. The background reconstruction loss is defined as

$$\mathcal{L}_{\text{recon}} = \frac{1}{|M_B|} \sum_{p \in M_B} \left\| \hat{z}[p] - z[p] \right\|_2^2, \tag{2}$$

where p denotes spatial indices in the latent feature map. By anchoring $\hat{z}$ to z on these background locations, $\mathcal{L}_{\text{recon}}$ enforces fidelity to the original background and promotes globally coherent reconstruction.

Background Puzzle Loss. While the reconstruction loss $\mathcal{L}_{\text{recon}}$ preserves high-fidelity background appearance, it does not explicitly control how different background subtypes are filled and integrated into the masked area, often leading to structurally inconsistent or partially missing patterns (*e.g.*, misaligned background context in Fig. 6). To address this, we introduce a background puzzle loss that treats each background subtype as a distinct puzzle piece that must con-

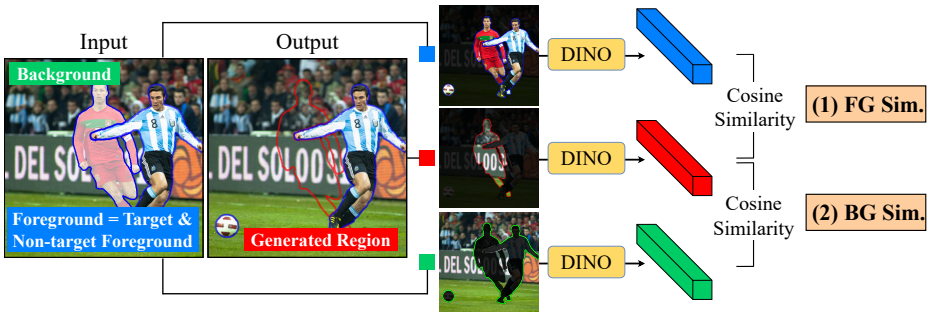


Fig. 7: Illustration of evaluation metrics (*i.e.*, BG Sim. and FG Sim.) for unpaired object removal.

tribute coherently to reconstructing the masked region. This loss enforces that background attention is concentrated only on valid regions (target foreground or clean background), preventing attention distraction toward non-target foregrounds and improving spatial consistency:

$$\mathcal{L}_{\text{puzzle}} = 1 - \text{Dice}\big(A^{\text{dom}}, \mathbf{1}_{M_T \cup M_B}\big)\,, \tag{3}$$

where $\text{Dice}(\cdot,\cdot)$ computes the soft spatial overlap between the continuous attention map and the binary valid-region indicator (see Appendix B for details). $A^{\text{dom}}[p] = \max_{b\in\mathcal{B}} A_b[p]$ selects the strongest attention response at each location p across background subtype tags $b \in \mathcal{B}$ inferred by BFE (Sec. 3.1), ensuring that every position is accounted for by its most relevant subtype.

4 Experiments

4.1 Experimental Setup

Implementation details. We implement EraseLoRA on three text-to-image diffusion backbones [7, 19, 26]. For a fair comparison, we strictly follow the official inference configurations (scheduler, guidance scale, resolution, and the number of sampling steps) and apply EraseLoRA in a purely dataset-free, test-time manner. During test-time adaptation (TTA), we freeze all backbone parameters and optimize only the inserted LoRA adapters [12] (rank r=32) for 500 iterations with respect to the final loss defined in Sec. 3.2. This configuration is kept consistent across all backbones and benchmarks. For BFE (Sec. 3.1), we use InternVL3-78B [51] as the default MLLM and Grounded SAM2 [28, 29] for tag-to-mask conversion. Additional details are provided in Appendix B.

Benchmarks. We evaluate EraseLoRA on two benchmarks: 200 samples from OpenImages V7 [18] and 343 frames from RORD [31]. In both datasets, the original annotations do not distinguish non-target foregrounds from background, so we annotate them with three-label (target / non-target foreground /

Table 2: Quantitative comparison with previous state-of-the-art methods on test datasets [18, 31]. The best results are in **bold**.

	OpenImages V7			RORD		
	BG Sim.(↑)	FG Sim.(↓)	BG Pres.(↑)	BG Sim.(↑)	FG Sim.(↓)	BG Pres.(↑)
Dataset-Free Approaches:						
SD3.5-M [7]	0.605	0.286	**0.934**	0.582	0.319	0.907
+ AttentiveEraser [34]	0.559	0.276	0.931	0.541	0.302	0.901
+ DesignEdit [13]	0.600	0.255	0.933	0.597	0.273	**0.908**
+ **EraseLoRA**	**0.743**	**0.151**	0.924	**0.779**	**0.138**	0.901
Dataset-Driven Approaches:						
SDXL-Inpainting [26]	0.677	0.212	0.742	0.645	0.234	0.720
PowerPaint [54]	0.669	0.217	0.719	0.729	0.176	0.687
CLIPAway [6]	0.656	0.223	0.713	0.744	0.156	0.705
SmartEraser [14]	0.709	0.185	0.727	0.768	0.148	0.672
EntityErasure [53]	0.679	0.204	0.728	0.766	0.175	0.716

background) ground-truth masks to capture distractor regions. These refined annotations will be released and form the basis of the evaluation metrics described below. Additional results on RemovalBench [40] are provided in Appendix C.

Evaluation metrics. For unpaired object removal, we use DINO similarity [4], following recent image generation and editing works [20, 45]. Foreground Similarity (FG Sim.) and Background Similarity (BG Sim.) measure, respectively, how much the reconstructed region stays similar to the foreground (lower is better) and how well it aligns with the background (higher is better; see Fig. 7). We additionally report Background Preservation (BG Pres.), computed via SSIM [39] on unmasked regions, following the evaluation protocol of recent editing methods [17, 52], to assess fidelity outside the masked area.

4.2 Comparison with State-of-the-art Approaches

Table 2 shows the quantitative results on two benchmarks [18, 31]. Compared to the baseline [7], EraseLoRA substantially improves BG Sim., from 0.605 to 0.743 on OpenImages V7 [18] and from 0.582 to 0.779 on RORD [31] (absolute gains of +0.14 and +0.20, corresponding to roughly 23% and 34% relative improvements). At the same time, FG Sim. is almost halved (0.286→0.151 on OpenImages V7, 0.319→0.138 on RORD), indicating that EraseLoRA suppresses foreground re-generation while filling the mask with background-consistent content. EraseLoRA also outperforms dataset-driven methods [6, 14, 26, 53, 54]: it attains the highest BG Sim. and the lowest FG Sim. on both benchmarks [18, 31], while maintaining background preservation around 0.90, which is about 0.18 higher than all five dataset-driven methods. This indicates that EraseLoRA predominantly modifies only the masked region and leaves the unmasked background almost unchanged, unlike dataset-driven models that often perturb surrounding content. Qualitative comparisons in Fig. 8 reflect the same trend: EraseLoRA removes the target object cleanly while preserving sharp background details and

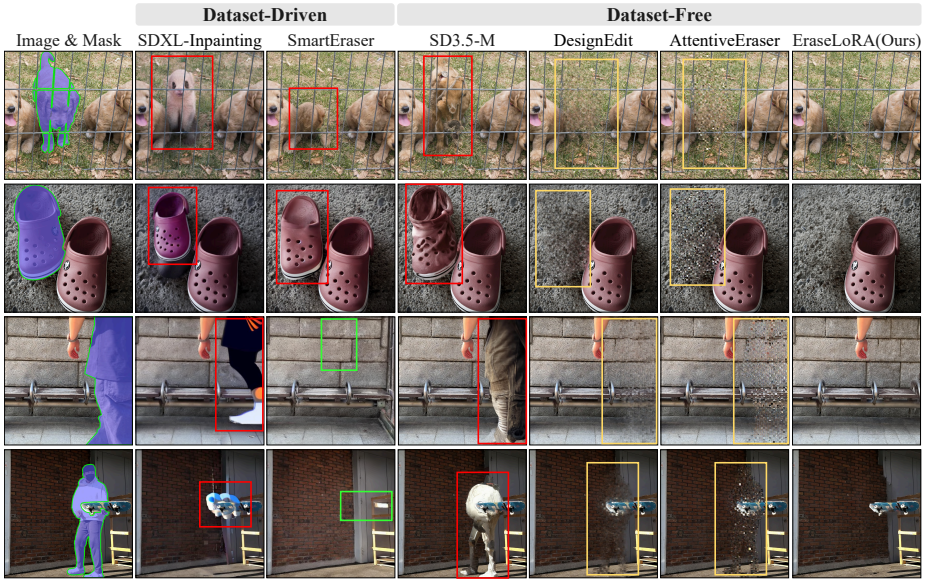


Fig. 8: Qualitative comparison on OpenImages V7 [18] and RORD [31]. Without any paired data, EraseLoRA avoids unwanted background changes (green), copying nearby objects (red), and residual foreground artifacts (yellow), achieving cleaner object removal and more faithful backgrounds than both dataset-driven and dataset-free methods [7, 13, 14, 26, 34].

avoiding unwanted edits in unmasked regions. Additional quantitative and qualitative results are provided in Appendix C and F.

4.3 Discussion

To understand EraseLoRA's performance gains, we conduct component-wise ablations on OpenImages V7 [18], measuring the effect of each stage and loss term.

Effect of BFE with prior dataset-free methods. To test whether our foreground exclusion module (BFE; Sec. 3.1) generalizes beyond EraseLoRA, we plug it into prior dataset-free object-removal methods [13, 34] without modifying their inference pipelines. By explicitly excluding non-target foregrounds from their reference regions, background similarity improves by up to 6.6%, while foreground similarity decreases by 8.6% (Tab. 3, left). As shown in the left panel of Figure 9, non-target foregrounds that were previously regenerated are successfully suppressed once BFE provides clean background references, confirming that BFE alleviates the re-generation limitation of existing dataset-free approaches [13, 34] in a fully model-agnostic manner.

Effect of losses in BRSA. To determine the most effective objective for BRSA (Sec. 3.2), we examine the roles of the two complementary losses through a loss-combination study (see Tab. 3, right). The background reconstruction loss

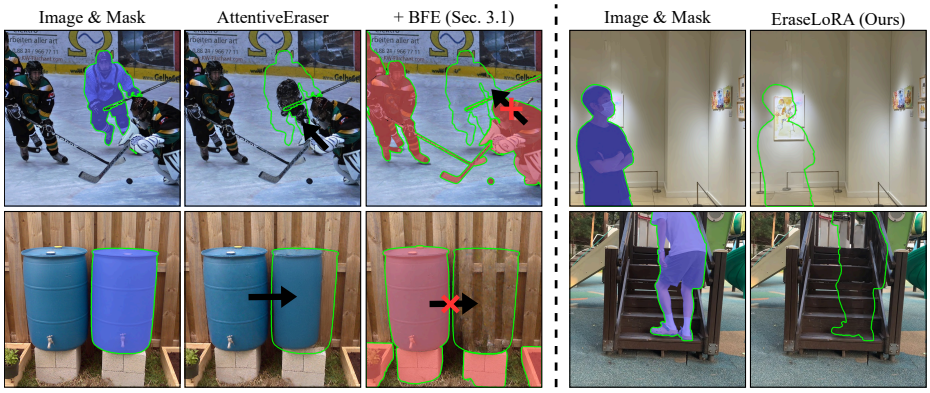


Fig. 9: (Left) Effect of BFE (Sec. 3.1) on AttentiveEraser [34]. Red masks denote excluded non-target foregrounds. (Right) Examples of EraseLoRA in occlusion cases. EraseLoRA reconstructs occluded content (*e.g.*, a painting, stairs) as background from surrounding context.

Table 3: (Left) Effect of non-target foreground exclusion (BFE; Sec. 3.1) in previous state-of-the-art dataset-free methods [13, 34] and (Right) loss component in BRSA (Sec. 3.2).

Method	Metrics	
	BG Sim.(↑)	FG Sim.(↓)
AttentiveEraser [34]	0.559	0.276
+ BFE (Ours; Sec. 3.1)	**0.596**	**0.252**
DesignEdit [13]	0.600	0.255
+ BFE (Ours; Sec. 3.1)	**0.603**	**0.251**

Method	Loss Components		Metrics	
	$\mathcal{L}_{\text{recon}}$	$\mathcal{L}_{\text{puzzle}}$	BG Sim.(↑)	FG Sim.(↓)
SD3.5-M [7]	✗	✗	0.605	0.286
+ EraseLoRA	✓	✗	0.736	0.158
+ EraseLoRA	✗	✓	0.561	0.278
+ EraseLoRA	✓	✓	**0.743**	**0.151**

($\mathcal{L}_{\text{recon}}$; Eq. (2)) preserves structural background consistency, improving BG Sim. from 0.605 to 0.736 (+21.7%), but can still leave faint foreground traces. In contrast, the background puzzle loss ($\mathcal{L}_{\text{puzzle}}$; Eq. (3)) suppresses foreground artifacts via background subtype aggregation, but alone lacks explicit background anchoring and fails to capture fine background structure and global patterns, often leading to visually inconsistent completions (see Fig. 10). Together (*i.e.*, EraseLoRA), the two losses achieve the best results, yielding coherent and detail-preserving background restoration while overcoming the limitations observed when either loss is used alone.

Flexibility. To evaluate the general applicability of EraseLoRA, we apply our method to different text-to-image diffusion backbones [7, 19, 26, 30], MLLMs of varying scales [2, 22, 48, 51], and Tag2Mask models [3, 16, 29, 36]. EraseLoRA demonstrates generalization ability by yielding reliable gains in both BG Sim. and FG Sim. regardless of the underlying diffusion backbones (see Tab. 7 and Fig. 15). Detailed quantitative and qualitative analyses are provided in Appendix D.1.

Fig. 10: Visualization of loss components in BRSA (Sec. 3.2). $\mathcal{L}_{\text{recon}}$ preserves background structure and $\mathcal{L}_{\text{puzzle}}$ completes the masked region; using both (*i.e.*, EraseLoRA) produces the most coherent and detailed background.

Table 4: Applicability across different (Left) MLLMs and (Right) Tag2Mask models.

Method	MLLM	Metrics	
		BG Sim.(↑)	FG Sim.(↓)
SD3.5-M [7]	N/A	0.605	0.286
+ EraseLoRA	LLaVA-7B [22]	0.728	0.164
+ EraseLoRA	LLaVA-7B+MARINE [48]	0.727	0.161
+ EraseLoRA	Qwen2.5-VL-72B [2]	0.726	0.165
+ EraseLoRA	InternVL3-78B [51]	**0.743**	**0.151**

Method	Tag2Mask Model	Metrics	
		BG Sim.(↑)	FG Sim.(↓)
SD3.5-M [7]	N/A	0.605	0.286
+ EraseLoRA	Seg4Diff [16]	0.666	0.205
+ EraseLoRA	YOLOE [36]	0.709	0.177
+ EraseLoRA	SAM3 [3]	0.708	0.177
+ EraseLoRA	G.SAM2 [29]	**0.743**	**0.151**

We also evaluate EraseLoRA across MLLMs of various scales [2, 22, 48, 51] from 7B to 78B parameters, including a hallucination-mitigated variant [48]. Across all these models, EraseLoRA provides consistent gains (Tab. 4, left), confirming that our framework reliably leverages the background-aware reasoning ability of MLLMs rather than depending on a specific architecture. Notably, even lightweight 7B MLLMs yield substantial improvements comparable to much larger models [2], achieving gains of up to 20.3% in BG Sim. and 42.7% in FG Sim., which indicates that EraseLoRA remains highly effective in resource-constrained settings without requiring a large MLLM. Despite their effectiveness, we adopt a large-scale MLLM [51] as our default configuration in the main experiments, as it provides the strongest background-aware reasoning and achieves the best overall performance.

We further validate EraseLoRA's effectiveness across different Tag2Mask models [3,16,29,36] in BFE (Sec. 3.1). Across all Tag2Mask variants, EraseLoRA

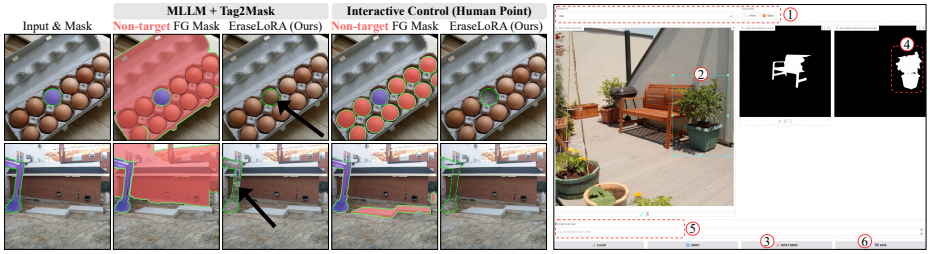


Fig. 11: Interactive Control. (Right) Our interactive interface allows users to generate customized non-target foreground masks for BFE (Sec. 3.1) based on manual points or bounding boxes and background tags. (Left) These human-guided non-target foreground exclusion and background tag selection effectively correct challenging failure cases of EraseLoRA.

consistently improves background reconstruction and foreground suppression, yielding at least 10.0% gains in BG Sim. and 28.3% reductions in FG Sim. over the SD3.5-M [7] baseline (see Tab. 4, right). Grounded SAM2 achieves the best performance, improving BG Sim. by up to 22.8% and reducing FG Sim. by up to 47.2%, resulting in the cleanest and most faithful background reconstruction. Additional qualitative results and detailed analysis regarding the impact of MLLM hallucinations are provided in Appendix D.1 and D.3.

MLLM-guided removal under occlusion. Occlusion poses a unique challenge for object removal: content normally regarded as foreground may be hidden behind the target, yet must be recovered as background during inpainting. By leveraging the MLLM's scene-level understanding, EraseLoRA correctly identifies such occluded elements and reconstructs them with semantically coherent completions (see Fig. 9, right).

Interactive Control. While EraseLoRA effectively removes the target automatically via MLLM's background-aware reasoning [51], we offer an interactive variant where users provide minimal guidance via an interactive interface to enhance user control and computational efficiency. This mode allows users to bypass MLLM overhead or correct potential imperfect predictions of its reasoning. To support these needs, our interactive variant follows a streamlined workflow (Fig. 11, right): (i) selecting a specific sample and the edit mode (*e.g.*, Point or BBox prompts), (ii) providing visual guidance via points or bounding boxes on the input image, (iii) clicking the apply button to (iv) update and refine the precise non-target foreground mask by observing the extracted results, (v) generating descriptive background tags based on the input image, and (vi) clicking the save button to store the manual results for BRSA (Sec. 3.2).

Limitations. While EraseLoRA achieves strong performance in object removal, several limitations remain. First, the use of large-scale MLLMs such as InternVL3-78B [51] and test-time optimization for fixed 500 iterations with LoRA adapters introduces additional computational overhead. Specifically, the full optimization process takes approximately three minutes per test image on

the default backbone, SD3.5-M [7]. Although no paired data are required, iterative optimization and MLLM queries increase latency and memory usage at inference time. However, this cost can be significantly mitigated by (i) employing lightweight MLLMs (*e.g.*, 7B scale), which show comparable quality as shown in Tab. 4, and (ii) applying an early stopping strategy. We observe that employing an early stopping strategy can reduce the average optimization cost to approximately 141 iterations, achieving a more than 3.5× speedup compared to the fixed 500-iterations baseline while preserving over 96.5% of the background similarity (see Appendix E.1 for more details). Second, the MLLM-guided background definition can be imperfect in complex scenes with subtle semantic boundaries or heavy occlusion, which may lead to incomplete or inaccurate removal. A more detailed analysis is provided in Appendix D.3.

5 Conclusion

In this paper, we introduce EraseLoRA, a dataset-free object removal framework that leverages MLLM-guided background-aware reasoning and test-time adaptation to enable faithful background restoration. Building on the failure modes identified in existing dataset-free methods, EraseLoRA explicitly separates target, non-target foreground, and background, filters out distractors to prevent their regeneration, and integrates multiple background subtypes. As a plug-and-play and model-agnostic module, EraseLoRA consistently produces cleaner and more coherent background reconstructions across diffusion backbones and benchmarks without requiring any paired data. An interesting future direction is video object removal, where shared background context across frames can amortize the per-image optimization cost.

Acknowledgements

This work was partly supported by the KHIDI grant funded by the Korean government (MOHW) [No.RS-2025-02307233], the NRF or IITP grants funded by the Korean government (MSIT) [No.RS-2026-25472075, No.RS-2025-02305581, No.RS-2025-25442338 (AI Star Fellowship-SNU), and No.RS-2021II211343 (SNU AI)], the ITIP grant funded by the Korean government (MOTIR) [No.RS-2026-25549946], the Advanced GPU Utilization and AI Computing Infrastructure Enhancement User Support Programs funded by the Korean government (MSIT) [No.05-26-04-0094], the Research grant from SNU, and the Strategic Hub grant for International Research Collaboration of SNU. Kyungsu Kim is affiliated with the School of Transdisciplinary Innovations, Department of Biomedical Science, Interdisciplinary Program in Artificial Intelligence (IPAI), Medical Research Center, and AI Institute at SNU.

References

1. Bai, S., Cai, Y., Chen, R., Chen, K., Chen, X., Cheng, Z., Deng, L., Ding, W., Gao, C., Ge, C., Ge, W., Guo, Z., Huang, Q., Huang, J., Huang, F., Hui, B., Jiang, S., Li, Z., Li, M., Li, M., Li, K., Lin, Z., Lin, J., Liu, X., et al.: Qwen3-vl technical report (2025), https://arxiv.org/abs/2511.21631
2. Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., Dang, K., Wang, P., Wang, S., Tang, J., et al.: Qwen2.5-VL technical report (2025)
3. Carion, N., Gustafson, L., Hu, Y.T., Debnath, S., Hu, R., Coll-Vinent, D.S., Ryali, C., Alwala, K.V., Khedr, H., Huang, A., Lei, J., Ma, T., Guo, B., Kalla, A., Marks, M., Greer, J., Wang, M., Sun, P., Rädle, R., Afouras, T., Mavroudi, E., Xu, K., Wu, T.H., Zhou, Y., Momeni, L., Hazra, R., Ding, S., Vaze, S., Porcher, F., Li, F., Li, S., Kamath, A., Cheng, H.K., Dollar, P., Ravi, N., Saenko, K., Zhang, P., Feichtenhofer, C.: SAM 3: Segment Anything with Concepts. In: ICLR (2026)
4. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: ICCV. pp. 9650–9660 (2021)
5. Chen, Z., Wang, W., Yang, Z., Yuan, Z., Chen, H., Shen, C.: Freecompose: Generic zero-shot image composition with diffusion prior. In: ECCV. pp. 70–87 (2024). https://doi.org/10.1007/978-3-031-72643-9_5
6. Ekin, Y., Yildirim, A.B., Caglar, E.E., Erdem, A., Erdem, E., Dundar, A.: Clipaway: Harmonizing focused embeddings for removing objects via diffusion models. In: NeurIPS. vol. 37, pp. 17572–17601 (2024). https://doi.org/10.52202/079017-0559
7. Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., Podell, D., Dockhorn, T., English, Z., Rombach, R.: Scaling rectified flow transformers for high-resolution image synthesis. In: ICML. vol. 235, pp. 12606–12633 (2024)
8. Fanelli, N., Vessio, G., Castellano, G.: I dream my painting: Connecting mllms and diffusion models via prompt generation for text-guided multi-mask inpainting. In: IEEE WACV. pp. 6073–6082 (2025)
9. Fu, Y., Zheng, Y., Dai, Z., Ding, H.: EffectErase: Joint video object removal and insertion for high-quality effect erasing. In: CVPR. pp. 2005–2014 (2026)
10. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: NeurIPS. vol. 27 (2014)
11. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: NeurIPS. vol. 33, pp. 6840–6851 (2020)
12. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: LoRA: Low-rank adaptation of large language models. In: ICLR (2022)
13. Jia, Y., Cheng, A., Yuan, Y., Wang, C., Li, J., Jia, H., Zhang, S.: Designedit: Unify spatial-aware image editing via training-free inpainting with a multi-layered latent diffusion framework. In: AAAI. vol. 39, pp. 3958–3966 (2025). https://doi.org/10.1609/aaai.v39i4.32414
14. Jiang, L., Wang, Z., Bao, J., Zhou, W., Chen, D., Shi, L., Chen, D., Li, H.: Smarteraser: Remove anything from images using masked-region guidance. In: CVPR. pp. 24452–24462 (2025)
15. Kim, B.S., Kim, J., Ye, J.C.: Chain-of-zoom: Extreme super-resolution via scale autoregression and preference alignment. In: NeurIPS. vol. 38 (2025)
16. Kim, C., Shin, H., Hong, E., Yoon, H., Arnab, A., Seo, P.H., Hong, S., Kim, S.: Seg4diff: Unveiling open-vocabulary segmentation in text-to-image diffusion transformers. In: NeurIPS. vol. 38 (2025)

17. Kim, J., Lee, Z., Cho, D., Jo, S., Jung, Y., Kim, K., Yang, E.: Early timestep zero-shot candidate selection for instruction-guided image editing. In: ICCV. pp. 18844–18854 (2025)
18. Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Malloci, M., Kolesnikov, A., Duerig, T., Ferrari, V.: The open images dataset V4: Unified image classification, object detection, and visual relationship detection at scale. IJCV **128**, 1956–1981 (2020). https://doi.org/10.1007/s11263-020-01316-z
19. Labs, B.F.: Flux. https://github.com/black-forest-labs/flux (2024)
20. Li, P., Nie, Q., Chen, Y., Jiang, X., Wu, K., Lin, Y., Liu, Y., Peng, J., Wang, C., Zheng, F.: Tuning-free image customization with image and text guidance. In: ECCV. pp. 233–250 (2024). https://doi.org/10.1007/978-3-031-73116-7_14
21. Li, W., Lin, Z., Zhou, K., Qi, L., Wang, Y., Jia, J.: Mat: Mask-aware transformer for large hole image inpainting. In: CVPR. pp. 10758–10768 (2022)
22. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. In: NeurIPS. vol. 36, pp. 34892–34916 (2023)
23. Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Jiang, Q., Li, C., Yang, J., Su, H., Zhu, J., Zhang, L.: Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection. In: ECCV. pp. 38–55 (2024). https://doi.org/10.1007/978-3-031-72970-6_3
24. Liu, Y., Zhou, H., Cui, B., Shang, W., Lin, R.: Erase diffusion: Empowering object removal through calibrating diffusion pathways. In: CVPR. pp. 2418–2427 (2025)
25. Manukyan, H., Sargsyan, A., Atanyan, B., Wang, Z., Navasardyan, S., Shi, H.: Hd-painter: High-resolution and prompt-faithful text-guided image inpainting with diffusion models. In: ICLR. pp. 96301–96330 (2025)
26. Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., Rombach, R.: SDXL: Improving latent diffusion models for high-resolution image synthesis. In: ICLR. pp. 1862–1874 (2024)
27. Qu, L., Li, H., Wang, W., Liu, X., Li, J., Nie, L., Chua, T.S.: Silmm: Self-improving large multimodal models for compositional text-to-image generation. In: CVPR. pp. 18497–18508 (2025)
28. Ravi, N., Gabeur, V., Hu, Y.T., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L., Mintun, E., Pan, J., Alwala, K.V., Carion, N., Wu, C.Y., Girshick, R., Dollár, P., Feichtenhofer, C.: SAM 2: Segment anything in images and videos. In: ICLR (2025)
29. Ren, T., Liu, S., Zeng, A., Lin, J., Li, K., Cao, H., Chen, J., Huang, X., Chen, Y., Yan, F., Zeng, Z., Zhang, H., Li, F., Yang, J., Li, H., Jiang, Q., Zhang, L.: Grounded sam: Assembling open-world models for diverse visual tasks (2024)
30. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR. pp. 10684–10695 (2022)
31. Sagong, M.C., Yeo, Y.J., Jung, S.W., Ko, S.J.: RORD: A real-world object removal dataset. In: BMVC. p. 542 (2022)
32. Sargsyan, A., Navasardyan, S., Xu, X., Shi, H.: Mi-gan: A simple baseline for image inpainting on mobile devices. In: ICCV. pp. 7335–7345 (2023)
33. Siméoni, O., Vo, H.V., Seitzer, M., Baldassarre, F., Oquab, M., Jose, C., Khalidov, V., Szafraniec, M., Yi, S., Ramamonjisoa, M., Massa, F., Haziza, D., Wehrstedt, L., Wang, J., Darcet, T., Moutakanni, T., Sentana, L., Roberts, C., Vedaldi, A., Tolan, J., Brandt, J., Couprie, C., Mairal, J., Jégou, H., Labatut, P., Bojanowski, P.: Dinov3 (2025), https://arxiv.org/abs/2508.10104

34. Sun, W., Dong, X.M., Cui, B., Tang, J.: Attentive eraser: Unleashing diffusion model's object removal potential via self-attention redirection guidance. In: AAAI. vol. 39, pp. 20734–20742 (2025). https://doi.org/10.1609/aaai.v39i19.34285
35. Suvorov, R., Logacheva, E., Mashikhin, A., Remizova, A., Ashukha, A., Silvestrov, A., Kong, N., Goka, H., Park, K., Lempitsky, V.: Resolution-robust large mask inpainting with fourier convolutions. In: IEEE WACV. pp. 2149–2159 (2022)
36. Wang, A., Liu, L., Chen, H., Lin, Z., Han, J., Ding, G.: Yoloe: Real-time seeing anything. In: ICCV. pp. 24591–24602 (2025)
37. Wang, D., Shelhamer, E., Liu, S., Olshausen, B., Darrell, T.: Tent: Fully test-time adaptation by entropy minimization. In: ICLR (2021)
38. Wang, Z., Li, A., Li, Z., Liu, X.: Genartist: Multimodal llm as an agent for unified image generation and editing. In: NeurIPS. vol. 37, pp. 128374–128395 (2024)
39. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE TIP **13**(4), 600–612 (2004). https://doi.org/10.1109/TIP.2003.819861
40. Wei, R., Yin, Z., Zhang, S., Zhou, L., Wang, X., Ban, C., Cao, T., Sun, H., He, Z., Liang, K., Ma, Z.: OmniEraser: Remove objects and their effects in images with paired video-frame data (2025)
41. Xie, S., Zhang, Z., Lin, Z., Hinz, T., Zhang, K.: Smartbrush: Text and shape guided object inpainting with diffusion model. In: CVPR. pp. 22428–22437 (2023)
42. Xie, T., Ma, R., Wang, Q., Ye, X., Liu, F., Tai, Y., Zhang, Z., Wang, L., Yi, Z.: Anywhere: A multi-agent framework for user-guided, reliable, and diverse foreground-conditioned image generation. In: AAAI. vol. 39, pp. 7410–7418 (2025). https://doi.org/10.1609/aaai.v39i7.32797
43. Yang, B., Gu, S., Zhang, B., Zhang, T., Chen, X., Sun, X., Chen, D., Wen, F.: Paint by example: Exemplar-based image editing with diffusion models. In: CVPR. pp. 18381–18391 (2023)
44. Yu, Y., Zeng, Z., Zheng, H., Luo, J.: Omnipaint: Mastering object-oriented editing via disentangled insertion-removal inpainting. In: ICCV. pp. 17324–17334 (2025)
45. Zhang, K., Mo, L., Chen, W., Sun, H., Su, Y.: Magicbrush: a manually annotated dataset for instruction-guided image editing. In: NeurIPS. vol. 36, pp. 31428–31449 (2023)
46. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR. pp. 586–595 (2018)
47. Zhao, J., Wang, Z., Yang, P., Zhou, S.: Precise object and effect removal with adaptive target-aware attention. In: CVPR. pp. 19370–19379 (2026)
48. Zhao, L., Deng, Y., Zhang, W., Gu, Q.: Mitigating object hallucination in large vision-language models via image-grounded guidance. In: ICML. vol. 267, pp. 77461–77486 (2025)
49. Zhao, S., Cui, J., Sheng, Y., Dong, Y., Liang, X., Chang, E.I., Xu, Y.: Large scale image completion via co-modulated generative adversarial networks. In: ICLR (2021)
50. Zhou, J., Li, J., Xu, Z., Li, H., Cheng, Y., Hong, F.T., Lin, Q., Lu, Q., Liang, X.: Fireedit: Fine-grained instruction-based image editing via region-aware vision language model. In: CVPR. pp. 13093–13103 (2025)
51. Zhu, J., Wang, W., Chen, Z., Liu, Z., Ye, S., Gu, L., Tian, H., Duan, Y., Su, W., Shao, J., et al.: InternVL3: Exploring advanced training and test-time recipes for open-source multimodal models (2025)
52. Zhu, T., Zhang, S., Shao, J., Tang, Y.: Kv-edit: Training-free image editing for precise background preservation. In: ICCV. pp. 16607–16617 (2025)

53. Zhu, Y., Zhang, Q., Wang, Y., Nie, Y., Zheng, W.S.: Entityerasure: Erasing entity cleanly via amodal entity segmentation and completion. In: CVPR. pp. 28274–28283 (2025)
54. Zhuang, J., Zeng, Y., Liu, W., Yuan, C., Chen, K.: A task is worth one word: Learning with task prompts for high-quality versatile image inpainting. In: ECCV. pp. 195–211 (2024)
55. Zuo, Z., Zhao, L., Li, A., Wang, Z., Zhang, Z., Chen, J., Xing, W., Lu, D.: Generative image inpainting with segmentation confusion adversarial training and contrastive learning. In: AAAI. vol. 37, pp. 3888–3896 (2023). https://doi.org/10.1609/aaai.v37i3.25502

A Preliminaries

Diffusion Models. Diffusion models generate images by learning a reverse denoising process that gradually transforms noise into image [7, 26]. In a typical text-to-image implementation, an input image I is first mapped into a latent representation $z_0 = \mathrm{Enc}(I)$ through a VAE encoder. A denoising network ϵ_θ then iteratively predicts and removes the noise from z_t, optionally conditioned on a text embedding c. Finally, the denoised latent $\hat{z}_0$ is decoded back into the image space by a VAE decoder, yielding $\hat{I} = \mathrm{Dec}(\hat{z}_0)$. This framework allows controllable generation by conditioning on text prompts or other external signals.

Attention Mechanisms in Diffusion Models. In latent diffusion models [30], attention blocks regulate how information flows during denoising. There are two types of attention: self-attention captures dependencies among latent tokens, while cross-attention establishes interactions between latent tokens and external condition tokens such as text prompts. For a latent $z_t \in \mathbb{R}^{HW \times d}$ at step t, self-attention computes a weight matrix $A^{\text{self}} = \mathrm{softmax}(QK^\top/\sqrt{d}) \in [0,1]^{HW \times HW}$, where each row sums to 1, which represents how strongly one latent token attends to all other tokens. Cross-attention follows the same principle but aligns latent queries with condition tokens $C \in \mathbb{R}^{L \times d}$, yielding $A^{\text{cross}} \in [0,1]^{HW \times L}$. Here each row of A^{cross} sums to 1, corresponding to $\sum_{i=1}^{L} A_i^{\text{cross}}[p] = 1$ for each spatial index p, where $A_i^{\text{cross}} \in [0,1]^{HW}$ denotes the cross-attention matrix for condition token i. This indicates how strongly each latent token relates to the L condition tokens. In practice, the way attention is computed has evolved with the design of modern diffusion backbones. Earlier text-to-image diffusion models [26, 30] compute attention directly at the latent level on feature maps. However, recent powerful text-to-image diffusion models [7, 19] group multiple latent tokens into larger patch tokens (*latent patchify*) and compute attention at the patch level to enable more efficient and scalable processing of high-resolution images.

B Details of EraseLoRA

B.1 Implementation Details

For reproducibility, we provide the implementation details and hyperparameters of EraseLoRA. During test-time adaptation, only the LoRA adapters [12] inserted into the attention blocks are updated, while all backbone parameters remain frozen. Optimization is performed using the test-time adaptation objective defined in Sec. 3.2. All baseline methods [6, 7, 13, 14, 19, 26, 34, 40, 53, 54] are reproduced using official implementations when available, or re-implemented following the descriptions in their respective papers.

Dice Score. We use a soft Dice coefficient between a continuous attention map X and a binary region indicator Y:

$$\mathrm{Dice}(X, Y) = \frac{2\sum_p X[p]Y[p]}{\sum_p X[p] + \sum_p Y[p] + \epsilon},$$

where ϵ is a small constant added for numerical stability.

Normalized Cross-Attention. For each background subtype tag $b \in \mathcal{B}$, we compute a raw spatial attention map $\tilde{A}_b$ from the cross-attention values of the diffusion model. Let $\mathcal{T}_b$ denote the set of text token indices corresponding to tag b. For each spatial index p, we average the cross-attention responses over attention layers, heads, and the text tokens in $\mathcal{T}_b$:

$$\tilde{A}_b[p] = \frac{1}{|\mathcal{L}|\,|\mathcal{H}|\,|\mathcal{T}_b|} \sum_{\ell \in \mathcal{L}} \sum_{h \in \mathcal{H}} \sum_{i \in \mathcal{T}_b} A^{\text{cross}}_{\ell,h}[p, i],$$

where $\mathcal{L}$ and $\mathcal{H}$ denote the sets of attention layers and heads, respectively, and $A^{\text{cross}}_{\ell,h}[p, i]$ denotes the cross-attention value between spatial index p and text token i.

We then apply tag-wise normalization across background subtype tags:

$$A_b[p] = \frac{\exp(\tau \tilde{A}_b[p])}{\sum_{b' \in \mathcal{B}} \exp(\tau \tilde{A}_{b'}[p])},$$

where τ controls the sharpness of subtype assignment, and we set $\tau = 100$ in all experiments.

Finally, the dominant subtype aggregation map used in $\mathcal{L}_{\text{puzzle}}$ is defined as

$$A^{\text{dom}}[p] = \max_{b \in \mathcal{B}} A_b[p].$$

This map records the strongest normalized response among background subtype tags at each spatial location. Matching A^{dom} to $\mathbf{1}_{M_T \cup M_B}$ promotes subtype aggregation on the target and clean background regions while suppressing responses on the non-target foreground mask M_F.

Configurations.

- **Hardware.** All experiments are conducted on NVIDIA RTX PRO 6000 Blackwell Server Edition GPUs with 96GB VRAM using mixed-precision (FP16).
- **Diffusion Backbone choices.** SD3.5-M [7] is used as the default text-to-image diffusion backbone based on its powerful text-image alignment ability. To validate flexibility across diverse modern diffusion backbones, we additionally test our method on several different diffusion architectures, including SD1.5 [30], SDXL [26], and FLUX.1 [19] (see Tab. 7).
- **MLLM choices.** InternVL3-78B [51] is used in BFE (Sec. 3.1) as the default MLLM due to its strong background-aware reasoning. To validate flexibility across different MLLMs, we additionally test representative models of various scales, including LLaVA-7B [22] and Qwen2.5-VL-72B [2]. We further include a hallucination-mitigated model, LLaVA-7B w/ MARINE [48], to examine the impact of object hallucination on removal quality (see Tab. 4, left). For all MLLM variants, we consistently apply the same full prompt as illustrated in Fig. 12 to ensure a fair comparison of their background-aware reasoning capabilities.

You will **receive TWO images** in this exact order:
[0] Original RGB image
[1] Binary MASK image (WHITE = region to remove, BLACK = keep)

CRITICAL: Focus on Image [1] (the MASK) to identify what needs to be removed. The WHITE area in the mask shows EXACTLY what object is being removed.

CONTEXT:
This is for text-guided inpainting using Stable Diffusion. Your tags will be directly used as text prompts to guide the inpainting model.

ANALYSIS PROCESS (Internal thinking - do NOT write these steps in output):

STEP 1 - IDENTIFY THE **MASKED OBJECT**:
Look at Image [1]. What object is covered by the WHITE region?
Compare the white mask shape with objects in Image [0]. The WHITE region shows EXACTLY what is **being removed**.

STEP 2 - WHAT SHOULD **FILL THE EMPTY SPACE**?
After removing the masked object, what should Stable Diffusion generate?
Look at what's VISIBLE around the white boundary in Image [0]. Include ALL elements that **should appear** (multiple tags OK):

A. GROUND/FLOOR (what the object sits on):
Examples: "grass", "green grass", "asphalt road", "wooden floor", "sand"
B. BACKGROUND SCENERY (what's behind/around):
Examples: "blue sky", "white clouds", "trees", "buildings", "ocean", "mountains"
C. CONTINUATION CHECK:
Does the white mask cover ONLY PART of a person/vehicle/animal?
If YES and the object clearly extends beyond the mask - Include for continuation. If the mask covers the ENTIRE object - DO NOT include in background.

List all background elements (near to far).

STEP 3 - WHAT SHOULD **NOT APPEAR**?
What other complete objects are visible OUTSIDE the white mask region? These objects should **NOT be generated in the inpainted area**.

STEP 4 - COMPILE FINAL TAGS:
- **target_object_tags**: The object covered by WHITE mask (STEP 1)
- **background_tags**: Elements to generate (STEP 2)
- **foreground_tags**: Objects to exclude (STEP 3)

IMPORTANT: If target and background contain the SAME tag:
- Is it a continuation case? Keep in both
- Is it the entire object? Remove from background

TAGGING RULES FOR STABLE DIFFUSION:
- Use natural language: "blue airplane", "green grass", "blue sky"
- Be specific: "vintage biplane", "small airplane", "seaplane"
- Multiple words OK: "cloudy sky", "grass field", "airplane hangar"
- Common SD vocabulary: terms diffusion models understand
- NO duplication: same tag shouldn't be in multiple lists (unless continuation)
- NO vague terms: "object", "thing", "background"

OUTPUT:
After thinking through all steps internally, output ONLY this JSON (no other text):
{
"target_object_tags": ["..."], "background_tags": ["...", "..."], "foreground_tags": ["..."]
}

Fig. 12: Full MLLM prompt used in BFE (Sec. 3.1) for background-aware reasoning and tag extraction from a single image–mask pair. The prompt guides a step-by-step reasoning that accounts for occlusions (orange line) to extract key cues: target objects (red line), background subtypes for reconstruction (green line), and non-target foregrounds to be excluded (blue line).

- **Tag2Mask choices.** Grounding DINO [23] and SAM2 [28] are used in BFE (Sec. 3.1) as the default Tag2Mask model to obtain pixel-level masks for MLLM-predicted tags. To validate flexibility across different Tag2Mask models, we additionally test three state-of-the-art segmentation models, including Seg4Diff [16], YOLOE [36] and SAM3 [3] (see Tab. 4, right).
- **LoRA details.** We test multiple ranks ($r \in \{16, 32, 64, 128\}$) for the LoRA adapters inserted into attention blocks, and set the rank to $r = 32$, providing the best results, for all experiments (see Fig. 18).
- **TTA iterations.** We compare different numbers of iterations ($\{100, 200, 300, 400, 500, 600\}$) during test-time adaptation. (see Fig. 18). While we set 500 iterations as the default for all experiments to ensure maximum quality, we also introduce an Early Stopping (E.S.) strategy to significantly improve efficiency while maintaining quality. We provide the technical details of this strategy in Sec. E.1.
- **Loss weights.** The weight of the puzzle loss $\mathcal{L}_{\text{puzzle}}$ is set to $\lambda = 0.2$ in the TTA objective.
- **Computational cost of TTA.** The VRAM usage, number of additional parameters, and optimization time of BRSA (Sec. 3.2) during test-time optimization are summarized in Tab. 7. Only the LoRA parameters are updated during optimization and the updated LoRA weights are merged into the backbone afterward [12]. As a result, EraseLoRA incurs no extra computational cost at inference time, consistent with standard text-to-image diffusion architectures [7, 19, 26, 30] (see Tab. 6, right).

Metric Details. To evaluate object-removal performance on unpaired datasets that do not have ground-truth images after removal, we use BG Sim. and FG Sim., two DINO-based cosine similarity metrics [4], along with BG Pres. (introduced in Sec. 4). Following the mask partition defined in Eq. 1, we use the target mask M_T, non-target foreground mask M_F, and background mask M_B for metric computation. We manually curate the corresponding masks to obtain reliable regions for evaluation and will release them publicly.

BG Sim. Background Similarity (BG Sim.) measures how well the reconstructed masked region aligns with the true background (higher is better). We compute it as the cosine similarity between DINOv3 [33] features extracted from the background region in the input image and the reconstructed region in the output:

$$\text{BG Sim.} = \frac{f(I[M_B]) \cdot f(\hat{I}[M_T])}{\|f(I[M_B])\| \, \|f(\hat{I}[M_T])\|} \tag{4}$$

where M_B denotes the background mask and M_T denotes the reconstructed target region.

FG Sim. Foreground Similarity (FG Sim.) measures how much the reconstructed masked region incorrectly refers to foreground content (lower is better). It is computed as the cosine similarity between DINOv3 [33] features extracted from the foreground region in the input image and the reconstructed region in

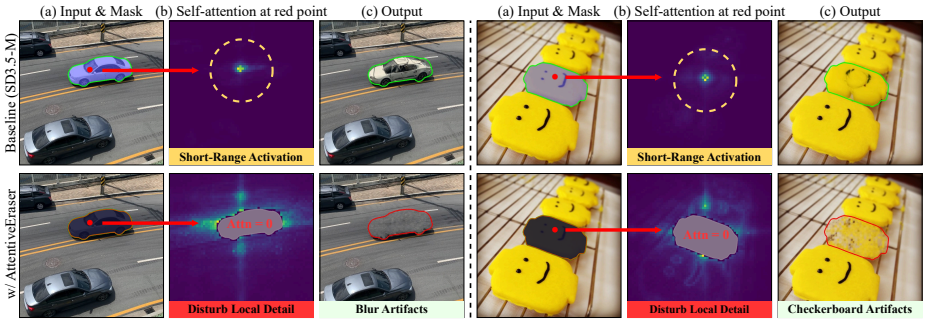


Fig. 13: Artifacts from disrupting short-range self-attention. Prior self-attention control methods [13, 34] suppress short-range activations (b), which erodes fine details and produces blur and checkerboard artifacts in the reconstructed background (c).

the output. To discourage background-inconsistent restoration, we weight this score by (1 − BG Sim.):

$$\text{FG Sim.} = (1 - \text{BG Sim.}) \cdot \frac{f(I[M_T \cup M_F]) \cdot f(\hat{I}[M_T])}{\|f(I[M_T \cup M_F])\| \, \|f(\hat{I}[M_T])\|} \tag{5}$$

where $M_T \cup M_F$ denotes the all foreground region (target and non-target).

BG Pres. Background Preservation (BG Pres.) evaluates how well the unmasked region is maintained after object removal. It is computed as SSIM [39] between the input image I and reconstructed image $\hat{I}$ over the unmasked region M_T^c, following the protocol of recent editing works [17, 52].

B.2 Design Rationale

Limitations of Attention Surgery. Previous state-of-the-art dataset-free methods [5, 13, 34] redirect or block self-attention inside the mask so that the model focuses on unmasked context. Specifically, they block any interaction within the masked region itself by setting those attention values to zero (see Fig. 13 (b)). However, we identify two fundamental limitations in these methods.

First, they are inherently unstable and often produce blur or structural artifacts (see Fig. 13 (c)). They unintentionally disrupt short-range activations of self-attention, referring to the local interactions where latent tokens mainly attend to their nearby neighbors. These activations are crucial for preserving fine-grained details, and their disruption often leads to blurred textures. Furthermore, when applied to recent text-to-image diffusion models [7,19] that compute attention at the patch level, blocking attention inside the mask amplifies instability, resulting in patch-wise artifacts such as checkerboard patterns. Second, they apply uniform attention constraints without distinguishing the diverse background

subtypes in the unmasked region. By treating the entire context as a single and undifferentiated background, they indiscriminately attend to irrelevant features, such as using sidewalk details to reconstruct a road, and fail to preserve the unique structural priors of specific patterns such as road markings or cooling racks. This leads to textural blurring, structural misalignment, and unnatural boundaries between disparate subtypes (see Fig. 2 and Fig. 13 (c)).

These failure modes stem from the absence of explicit background-aware reasoning, showing that prior self-attention control methods [5, 13, 34] fail to maintain fine-grained fidelity and produce visually inauthentic results, lacking robustness across modern text-to-image diffusion architectures [7, 19].

Effect of Test-Time Adaptation. Instead of explicitly blocking self-attention, BRSA (Sec. 3.2) provides a dataset-free mechanism for injecting BFE-inferred background cues into the diffusion model through test-time adaptation with LoRA adapters [12]. Unlike dataset-driven approaches [14, 53, 54] that rely on paired before/after removal data or a dataset-level removal prior, BRSA adapts the frozen backbone to the current image context, allowing clean background information to be incorporated into the masked-region reconstruction through the adapted weights rather than explicit attention manipulation.

This test-time adaptation is a key design choice for inducing removal behavior without additional training data while avoiding artifacts caused by attention surgery. Together with the complementary BRSA objectives (Eqs. 2 and 3), it enables background cues to be aggregated with textual and structural consistency. As a result, BRSA reconstructs the masked region using image-specific background information, reducing blur, checkerboard artifacts, and subtype-misaligned textures across diverse diffusion backbones [7, 19, 26, 30].

C Additional Quantitative Results

C.1 Experimental Setup

This section provides extended evaluation details on baselines, datasets, and metrics.

Baselines. Beyond the main comparison in Tab. 2, we provide additional evaluations of FLUX.1-Fill [19] to broaden the set of competitive baselines.

Benchmarks. We further evaluate EraseLoRA on RemovalBench [40], which provides 68 paired samples with ground-truth images after target removal. We also consider OmniPaint-Bench [44], which offers 1,300 paired samples. However, we find that 1,000 samples overlap with RORD [31], and the remaining 300 samples exhibit notable inconsistencies between the input and ground-truth images, such as mismatched colors in unmasked regions (see Fig. 14). Therefore, we exclude OmniPaint-Bench [44] from our evaluation, as its samples do not provide reliable ground-truth backgrounds for fair quantitative comparison.

Extended metrics. For unpaired object removal, we use three evaluation metrics: Foreground Similarity (FG Sim.), Background Similarity (BG Sim.),

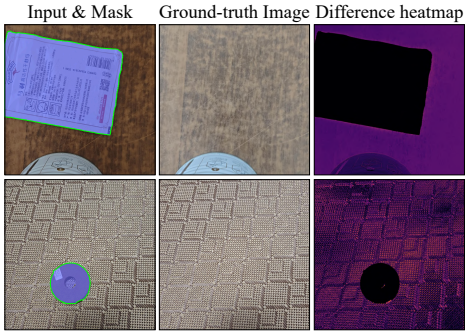


Fig. 14: Color discrepancies on paired OmniPaint-Bench [44] dataset. Before/after object removal pairs show substantial color mismatch in target-unrelated regions, as shown in the difference heatmap.

and Background Preservation (BG Pres.). For paired object removal, we further report representative fidelity metrics, PSNR, SSIM [39], and LPIPS [46], to compare predictions against ground-truth backgrounds. Following the evaluation protocol of RemovalBench [40], all paired metrics are computed on masked regions to measure pixel-level accuracy, structural consistency, and perceptual similarity (see Tab. 6, left).

Although these evaluation metrics capture different aspects of image fidelity, they are still insufficient to determine whether object removal is semantically successful. In particular, they cannot verify whether the target object truly disappears without residual foreground traces, nor whether the reconstructed background is contextually plausible.

To address this, we follow recent VLM-based evaluation protocols [34] for image inpainting and object removal and adapt them into a removal-specific metric, GPT-Metric, scored on a 0–100 scale. GPT-Metric assesses object removal from a semantic perspective via two components: (1) a removal success rate, which checks whether the target object is correctly perceived as absent without any traces, and (2) a semantic perceptual score, which evaluates the quality of the reconstructed background, including contextual consistency and hallucination artifacts. Detailed quantitative results for GPT-Metric are reported in Tab. 5.

C.2 Quantitative Analysis

Additional quantitative results. Across OpenImages V7 [18], RORD [31], and RemovalBench [40], EraseLoRA consistently surpasses all dataset-free approaches [7, 13, 34] and remains competitive against dataset-driven methods [6, 14, 19, 26, 40, 53, 54]. We observe that EraseLoRA improves background fidelity and alleviates regeneration of undesired foreground without any noise while preserving background. Table 5 and the left of Table 6 summarize the extended quantitative results across datasets.

Table 5: Extended quantitative comparison with previous state-of-the-art methods on OpenImages V7 [18] and RORD [31] dataset. The best results are in **bold**.

Method	OpenImages V7				RORD			
	BG Sim.	FG Sim.	GPT-Success	GPT-Score	BG Sim.	FG Sim.	GPT-Success	GPT-Score
Dataset-Free Approaches								
SD3.5-M [7]	0.605	0.286	12.7%	16.9	0.582	0.319	3.80%	6.87
+ AttentiveEraser AAAI'25 [34]	0.559	0.276	10.5%	31.6	0.541	0.302	2.04%	23.0
+ DesignEdit AAAI'25 [13]	0.600	0.255	24.8%	34.1	0.597	0.273	10.2%	27.6
+ EraseLoRA (Ours)	**0.743**	**0.151**	**71.0%**	**61.0**	**0.779**	**0.138**	**81.3%**	70.2
Dataset-Driven Approaches								
SDXL-Inpainting [26]	0.677	0.212	27.1%	30.9	0.645	0.234	3.83%	12.9
FLUX.1-Fill-dev [19]	0.661	0.255	30.3%	33.5	0.688	0.232	9.47%	13.8
PowerPaint ECCV'24 [54]	0.669	0.217	33.2%	34.9	0.729	0.176	34.1%	37.6
CLIPAway NeurIPS'24 [6]	0.656	0.223	33.2%	38.4	0.744	0.156	35.8%	39.9
SmartEraser CVPR'25 [14]	0.709	0.185	59.5%	57.4	0.768	0.148	75.8%	**72.5**
EntityErasure CVPR'25 [53]	0.679	0.204	50.5%	51.4	0.766	0.175	47.5%	49.7

Table 6: (Left) Quantitative comparison on paired RemovalBench [40] and (Right) inference-time computational cost comparison with previous state-of-the-art methods.

Method	SSIM (↑)	PSNR (↑)	LPIPS (↓)
Dataset-Free Approaches			
SD3.5-M [7]	0.772	22.3	0.185
+ AttentiveEraser [34]	0.780	24.5	0.181
+ DesignEdit [13]	0.782	24.9	0.168
+ EraseLoRA (Ours)	**0.786**	**25.1**	**0.163**
Dataset-Driven Approaches			
SDXL-Inpainting [26]	0.726	20.7	0.430
FLUX.1-Fill-dev [19]	0.757	21.6	0.212
PowerPaint [54]	0.751	22.9	0.213
CLIPAway [6]	0.722	22.5	0.198
SmartEraser [14]	0.744	24.2	0.168
OmniEraser [40]	0.699	23.9	0.253
EntityErasure [53]	0.723	22.4	0.208

Method	Params.	VRAM	Latency
Dataset-Free Approaches			
SD3.5-M [7]	2,243 M	21.9 GB	4 s
+ AttentiveEraser [34]	2,243M	43.2 GB	15 s
+ DesignEdit [13]	2,243M	43.2 GB	9 s
+ EraseLoRA (Ours)	2,243 M	30.9 GB	4 s
Dataset-Driven Approaches			
SDXL-Inpainting [26]	2,568 M	8.3 GB	5 s
FLUX.1-Fill-dev [19]	11,902 M	38.3 GB	15 s
PowerPaint [54]	1,952 M	4.7GB	2 s
CLIPAway [6]	1,390 M	11.3 GB	2 s
SmartEraser [14]	1,494 M	9.7 GB	2 s
OmniEraser [40]	16,961 M	35.1 GB	6 s
EntityErasure [53]	2,607 M	13.6 GB	3 s

Inference efficiency. We compare computational efficiency during inference across recent object removal methods [6, 13, 14, 19, 26, 34, 40, 53, 54]. Although EraseLoRA requires additional computation costs during BRSA (Sec. 3.2) due to LoRA adapters (see Tab. 7), EraseLoRA incurs no extra cost at inference by merging LoRA weights into the frozen diffusion backbone's weights [12]. (see Tab. 6, right). While dataset-driven models may offer lower inference cost, EraseLoRA achieves comparable or lower overhead while delivering substantially higher foreground suppression and background fidelity. Therefore, even when a computational gap exists, the quality gains make the trade-off clearly advantageous.

D Details of Ablation Study

D.1 Flexibility

EraseLoRA is designed as a model-agnostic framework that can be plugged into different components of the object-removal pipeline. In the following, we examine

Table 7: EraseLoRA on different diffusion backbones with TTA cost.

Method	Metrics		TTA Cost		
	BG Sim.(↑)	FG Sim.(↓)	Param.	VRAM	Opt. Time
SD1.5 [30]	0.596	0.271	858 M	3.39 GB	-
+ EraseLoRA	**0.702**	**0.176**	+6.38 M	+0.68 GB	117 s
SDXL [26]	0.608	0.297	2,573 M	11.5 GB	-
+ EraseLoRA	**0.730**	**0.186**	+46.5 M	+5.86 GB	250 s
SD3.5-M [7]	0.605	0.286	2,243 M	21.9 GB	-
+ EraseLoRA	**0.743**	**0.151**	+23.9 M	+9.0 GB	191 s
FLUX.1 [19]	0.588	0.205	11,902 M	38.5 GB	-
+ EraseLoRA	**0.760**	**0.146**	+52.3 M	+20.9 GB	495 s

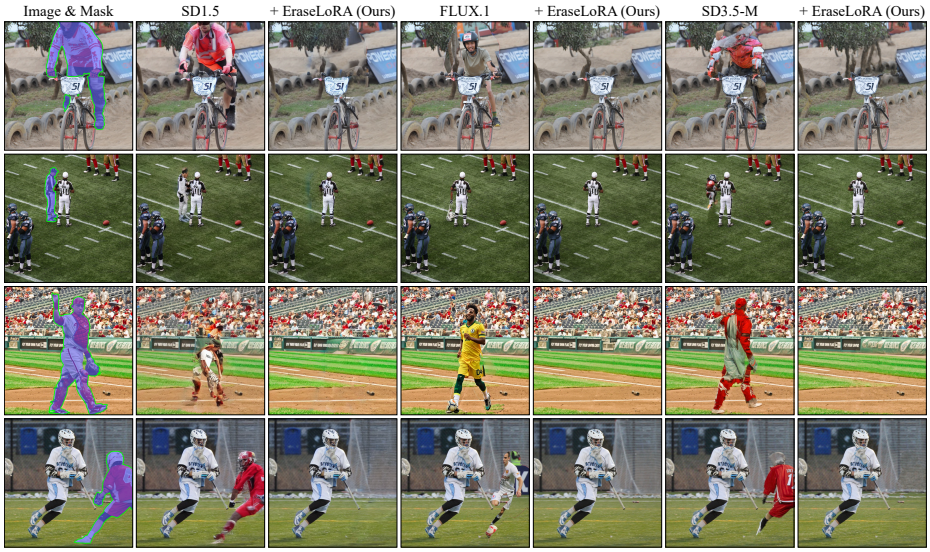


Fig. 15: Qualitative results across diffusion architectures. Consistent clean background restoration on SD1.5, FLUX.1, and SD3.5-M [7, 19, 30].

its flexibility by varying (i) the underlying diffusion backbone, (ii) the MLLM used for background-aware reasoning, and (iii) the Tag2Mask pipeline, and show that EraseLoRA yields consistent improvements across these choices.

Diffusion architectures. We evaluate EraseLoRA on four representative text-to-image diffusion backbones, including SD1.5 [30], SDXL [26], SD3.5-M [7], and FLUX.1 [19], thereby demonstrating that EraseLoRA robustly reconstructs foreground-free background across diverse architectures. EraseLoRA consistently improves the performance, showing at least a 17.8% increase in BG similarity and a 28.8% decrease in FG similarity, across all employed backbones (see Tab. 7). Notably, this improvement is more pronounced in text-to-image diffusion backbones with superior text-to-image alignment power (*e.g.*, SD3.5-M [7] and FLUX.1 [19]), as they provide more precise guidance for identifying and reconstructing background. This backbone-agnostic behavior is also clearly observed

Fig. 16: Qualitative results across MLLMs. Clean background reconstruction and strong foreground suppression are consistently achieved across diverse MLLMs [2, 22, 48,51]. MLLMs with strong background-aware reasoning [2,51] exhibit superior removal quality (blue overlays) by providing accurate background cues (green boxes), whereas smaller MLLMs [22,48] often fail to remove the object (red overlays) due to inaccurate background cues (*e.g.*, dog or person).

in qualitative results, where it stably removes target objects without foreground traces or noise, while preserving fine details and global background coherence (see Fig. 15).

MLLMs. EraseLoRA yields noticeable improvements in different MLLMs [2, 22, 48, 51], even when using lightweight models (see Tab. 4). This tendency is also qualitatively confirmed, where EraseLoRA suppresses object generation and restores the background coherently guided by MLLM-driven background cues (see Fig. 16, left). While we test MARINE [48], a hallucination-mitigated model, to examine the impact on removal quality, we observed that background-aware reasoning power, the ability to accurately identify and classify background subtypes among diverse candidates, is a more critical factor. Large MLLMs (*e.g.*, Qwen2.5-VL-72B [2] and InternVL3-78B [51]) successfully remove the target object and reconstruct background based on precisely inferred background subtypes, unlike MARINE [48] which fails to remove objects effectively due to inaccurate background cues (see Fig. 16, last row).

Tag2Mask models. We further validate that the proposed framework remains effective with different Tag2Mask models in BFE (Sec. 3.1), including

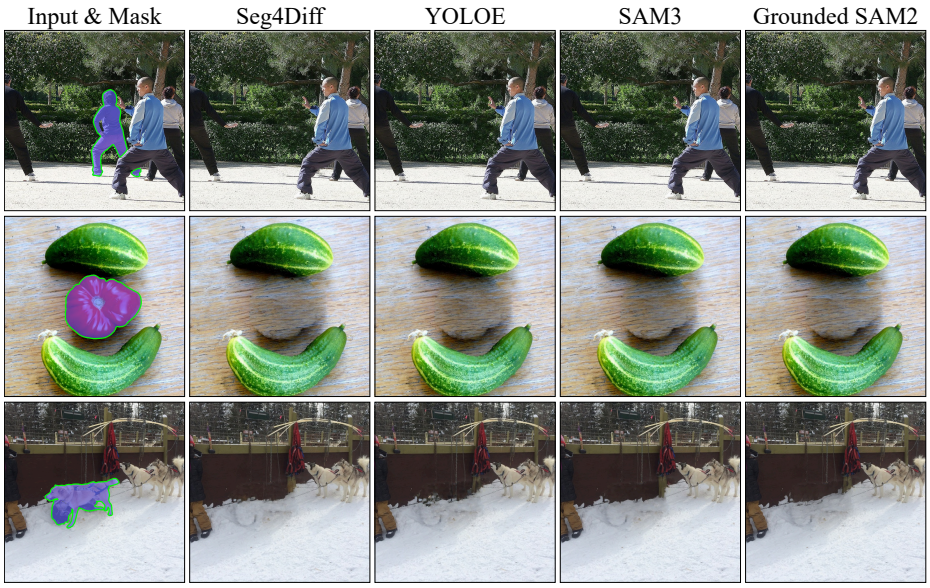


Fig. 17: Qualitative results across Tag2Mask models. Different Tag2Mask models [3, 16, 23, 28, 36] reliably localize non-target foreground regions, enabling complete background reconstruction without foreground traces.

Seg4Diff [16], YOLOE [36], SAM3 [3] and Grounded SAM2 (Grounding DINO [23] and SAM2 [28]). Across all Tag2Mask variants, EraseLoRA consistently improves background reconstruction and foreground suppression, yielding at least 10.0% gains in BG Sim. and 28.3% reductions in FG Sim. over the SD3.5-M [7] baseline (see Tab. 4, right). Notably, Grounded SAM2 achieves the best performance, improving BG Sim. by up to 22.8% and reducing FG Sim. by up to 47.2%, resulting in the cleanest and most faithful background reconstruction (see Fig. 17).

These results show that EraseLoRA is model-agnostic, supporting plug-and-play object removal without depending on specific external modules.

D.2 Adaptation Capacity

We vary two key factors in test-time optimization (BRSA; Sec. 3.2) on SD3.5-M [7] and FLUX.1 [19]: (1) the LoRA rank [12], which controls the learnable capacity of adapters, and (2) the number of test-time adaptation [37] iterations, which determines how long the model adapts to background cues.

LoRA rank. From experiments with ranks {16, 32, 64, 128}, rank 32 yields the best trade-off, achieving the strongest foreground suppression and the most consistent background reconstruction (see Fig. 18). Larger ranks such as 64 or 128 offer no meaningful gains while incurring higher optimization cost, so we adopt rank 32 as the default configuration.

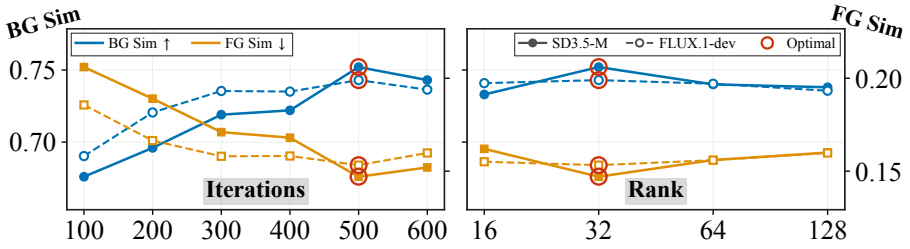


Fig. 18: Effect of test-time optimization [37] capacity. Varying LoRA [12] rank and number of iterations shows that 500 iterations and LoRA rank 32 achieve the best performance on SD3.5-M [7] and FLUX.1 [19].

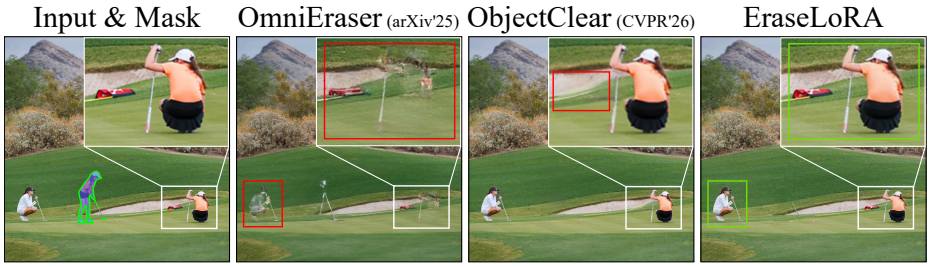


Fig. 19: Trade-off in effect-aware removal. Effect-aware methods [40, 47] can target object-induced effects, but often distort target-unrelated regions (red), whereas EraseLoRA preserves them (green).

TTA iterations. Although longer optimization generally improves reconstruction performance, the marginal gains diminish relative to the additional time cost. Hence, we adopt 500 iterations as a practical balance between quality and efficiency (see Fig. 18).

D.3 Discussion

Multiple objects removal. When the mask contains multiple objects to erase, EraseLoRA removes all targets jointly and reconstructs each region with coherent background subtypes. Because the adaptation operates per background rather than per instance, its performance remains stable regardless of the type or number of masked objects, requiring no modification to the framework (see Fig. 21 (b)).

Trade-off in effect-aware removal. Effect-aware methods [40, 47] explicitly target object-induced effects such as shadows or reflections, but may distort target-unrelated regions (see Fig. 19). In contrast, EraseLoRA prioritizes preserving such regions by editing only the masked area. When object-induced effects should also be removed, they can be handled by expanding the mask or using interactive control (see Fig. 11).

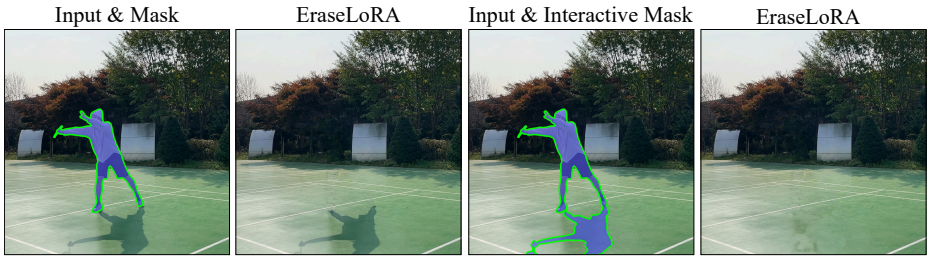


Fig. 20: Failure case and practical solution. While physical effects (*e.g.*, shadows) outside the initial mask can leave subtle traces (left), using an interactive mask that encompasses these effects ensures complete object removal (right).

Misclassification of background subtypes. While our method utilizes MLLM's background-aware reasoning capabilities, the removal quality may be affected if foreground and background tags are misclassified. In such cases, EraseLoRA follows incorrect cues and regenerates residual object traces. This issue can be alleviated by employing larger MLLMs [2,51] with stronger background-aware reasoning or through interactive control (see last row of Fig. 16 and Fig. 11).

E Limitations and Future Works

E.1 Limitations

Presence of object effects. EraseLoRA removes the target object and synthesizes plausible background texture, but does not explicitly handle physical effects caused by the object outside the mask (*e.g.*, shadows, lighting distortion, reflections). Thus, subtle traces may remain if these effects are not included in the mask. A practical solution is to use an interactive mask that encompasses both the object and its effects, which helps achieve more seamless removal (see Fig. 20).

Additional computational overhead. EraseLoRA requires background-aware reasoning from MLLMs [2, 22, 51] and test-time optimization [37] with LoRA adapters [12], introducing extra computation compared to training-free object removal methods [7,13,34]. Since the optimization is performed per background context, the cost scales with the number of different backgrounds encountered. However, this overhead can be effectively mitigated through several efficiency-oriented strategies. By selectively utilizing MLLMs of various scales, such as using LLaVA-7B [22] instead of the default InternVL3-78B [51], the parameter scale is reduced by approximately 11× while maintaining over 98.0% and 91.4% of the performance in BG Sim. and FG Sim., respectively (see Tab. 4, left).

Furthermore, we adopt an early stopping (E.S.) rule based on a standard elbow-based criterion to determine the stopping point automatically for compu-

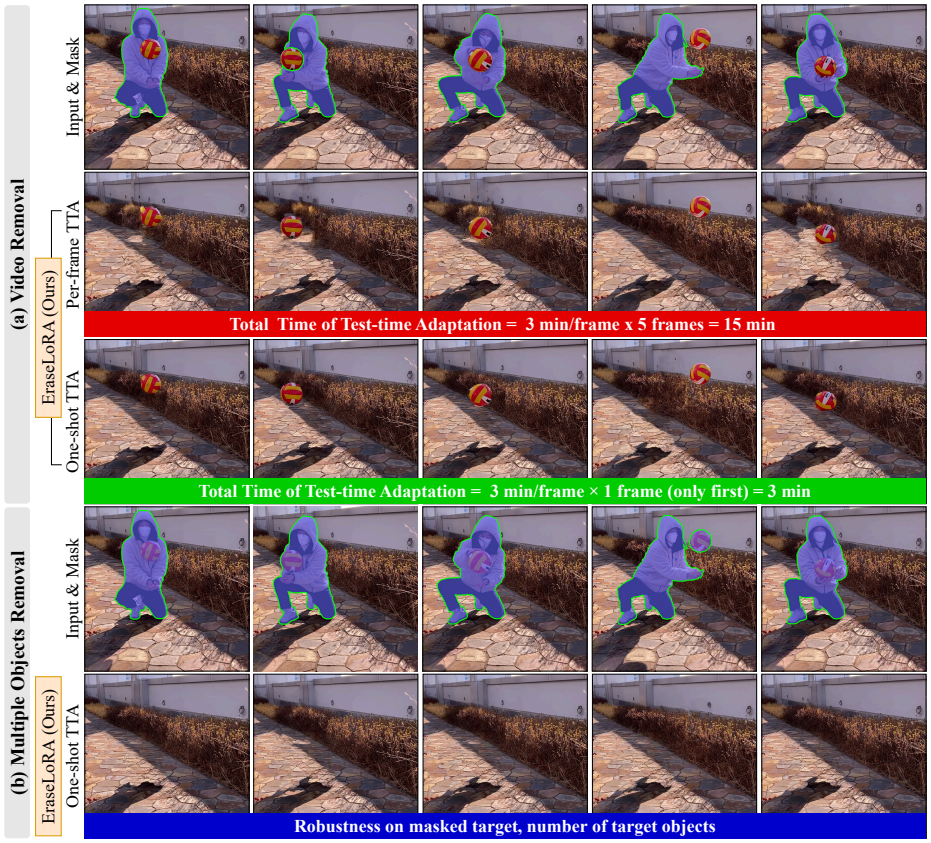


Fig. 21: Efficiency of video extension. For video frames sharing similar background context, one-shot optimization on a single frame can be reused across the sequence, achieving performance comparable to per-frame optimization while reducing adaptation cost by the number of frames. Moreover, EraseLoRA remains robust for multiple object removal without additional optimization.

tational efficiency, as the optimization exhibits a clear diminishing-return pattern. For each sample, we track the optimization objective $\mathcal{L}_{\text{total}}$ and identify the elbow of its smoothed trajectory, which marks the transition from rapid improvement to a near-plateau regime. Importantly, this criterion is not tailored to our method; rather, it is a conventional curve-based stopping rule applied directly to the loss dynamics. Using this criterion, the average optimization length is reduced from 500 to approximately 140 iterations, while preserving more than 96.5% of the final BG Sim. relative to the full optimization budget. While E.S. rule offers a favorable efficiency-performance trade-off, we do not adopt it in the main experiments because our primary goal in this paper is to maximize final restoration quality under a fixed optimization budget. In practice, early stopping introduces a small but non-negligible performance drop compared with the

full 500-iteration optimization. Nevertheless, we believe it provides a practical option for follow-up studies and for deployment scenarios where computational overhead is a more critical concern.

E.2 Future Works

Although EraseLoRA incurs additional computational overhead, its optimization is performed per background rather than per image. In video sequences where many frames share similar backgrounds, this allows the optimization cost to be reused across frames, making video object removal a promising next step.

To validate this potential, we apply test-time optimization [37] only to a single representative frame and reuse the adapted model for the remaining frames. As shown in Fig. 21, the outputs remain comparable to per-frame optimization, where the model is independently adapted for every frame. This demonstrates that leveraging shared background context makes video object removal an efficient extension of EraseLoRA (see Fig. 21 (a)).

F Additional qualitative results

This section provides extended qualitative comparisons between EraseLoRA and various baseline methods [6, 7, 13, 14, 19, 26, 34, 40, 53, 54]. All results were generated using the same experimental setup, including the baselines and benchmarks [18, 31, 40] detailed in Sec. C.1.

EraseLoRA clearly removes target objects without leaving semantic traces and reconstructs the background with artifact-free textures. In contrast, previous dataset-free methods [7, 13, 34] tend to hallucinate foreground-like patterns or overly smooth textures. Visual comparisons are shown in Fig. 22 and Fig. 23.

Fig. 22: Additional qualitative comparison with dataset-driven and dataset-free methods [6, 7, 13, 14, 19, 26, 34, 40, 53, 54] on OpenImages V7 [18] dataset.

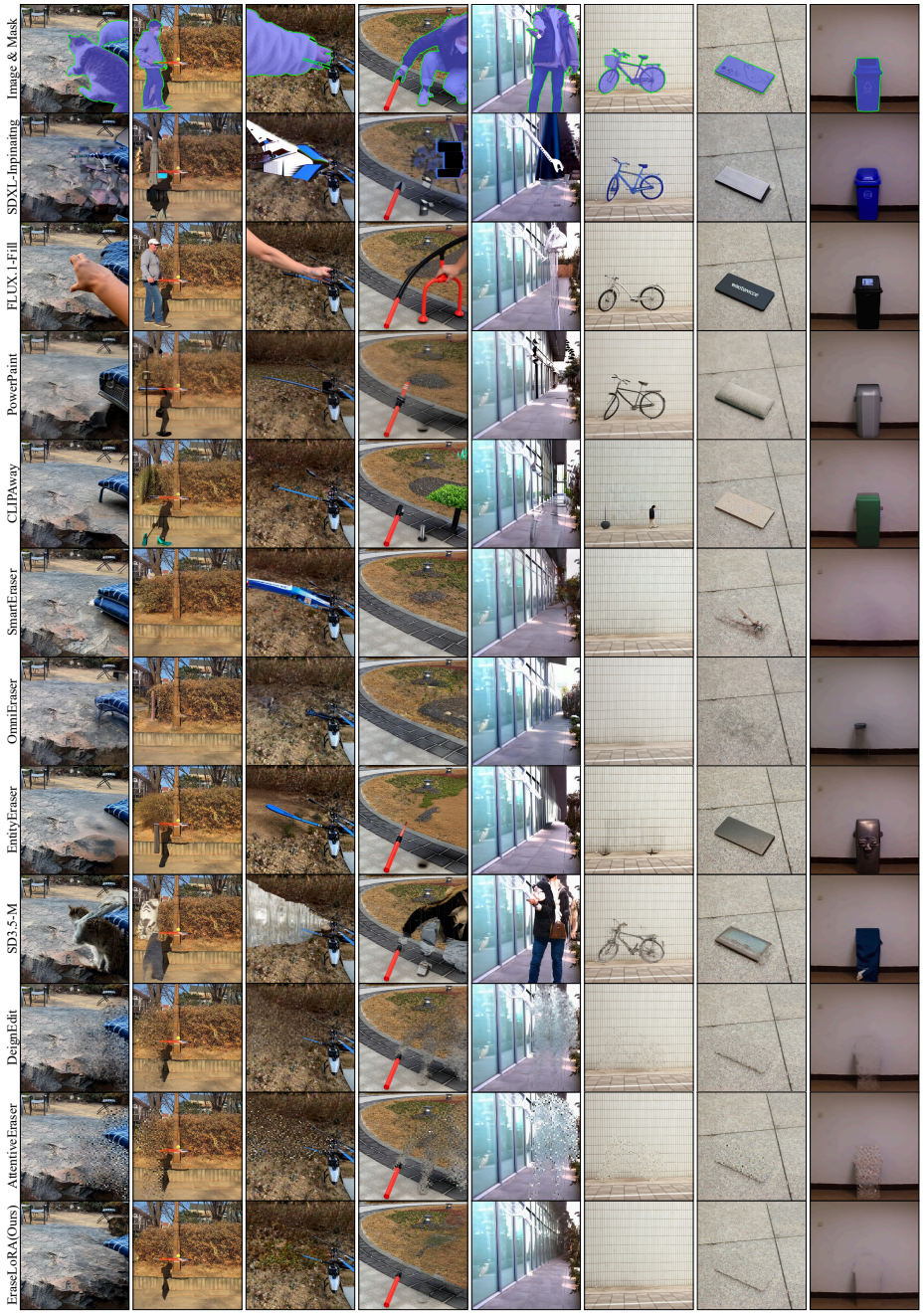


Fig. 23: Additional qualitative comparison with dataset-driven and dataset-free methods [6, 7, 13, 14, 19, 26, 34, 40, 53, 54] on RORD [31] and RemovalBench [40] datasets.