Precise Object and Effect Removal with Adaptive Target-Aware Attention

— Supplementary Materials —

In this appendix, we provide additional discussions and results to complete the main paper. In Sec. A, we present further implementation details on data augmentation during training and other extended applications. In Sec. B, we provide a more detailed introduction to the proposed OBER dataset, including statistics and some examples for demonstration. In Sec. C, we report additional results, such as further ablations, a user study, experiments with user strokes, results for object insertion and movement, and additional comparisons on in-the-wild images. In addition, we provide an online [interactive demo] on Hugging Face, enabling users to remove arbitrary objects through simple clicks.

A. Implementation Details

A.1. Training Augmentations

During training, we apply on-the-fly random cropping to enable the network to learn object removal across varying object sizes. We also apply color augmentation and random flipping to enhance model robustness. To improve the model's robustness to the estimated or user-provided coarse mask, several previous methods [4, 12, 13] have introduced mask augmentation techniques. In line with these approaches, we also apply dilation and erosion to the input mask during training. Different from previous practice, our method employs an *object-aware* dilation and erosion strategy, where the dilation and erosion kernel size is adaptively determined based on the size of the object. As demonstrated in the qualitative results in Sec. C.6, our method effectively handles coarse user-drawn masks by implicitly completing and refining them, showcasing strong robustness to imprecise inputs.

A.2. More Details for Extended Applications

ObjectClear can be flexibly extended to various applications. Visual results of object insertion and movement are shown in Fig. I, where our models generate realistic visual effects. In this section, we provide the implementation details of object insertion and object movement.

Object Insertion. The insertion network also leverages the OBER dataset for training and adopts the same architecture as ObjectClear, as illustrated in Fig.3 of the main paper, which receives the input tuple of $< z_t, I_{in}, M_o, c >$ and supervised by I_{GT} (for output image) and M_{fg} (for Adaptive Target-Aware Attention). However, I_{in}, I_{GT}, and c are constructed in a *reverse manner* compared with the removal network. Specifically, the ground-truth image I_{GT} corresponds to the original image containing the object along with its associated effects, while I_{in} is obtained by simply copying and pasting the object onto the background without any effects. To generate I_{in}, we first extract the object from I_{GT} using the object mask M_o only, and then paste it into the corresponding background image. In addition to the image pair, the input text c is also reversed to "*insert the instance of*". Notably, the Adaptive Target-Aware Attention map is still supervised by M_{fg}. Here, M_{fg} refers to the mask that covers both the object and its *generated* effects (*e.g.*, shadows or reflections).

During inference, we obtain the input image I_{in} by pasting an object foreground (w/o effects) onto the background image, and we feed it together with its corresponding object mask M_o as the input pair. The network then generates the output image where both the object and its generated effects are harmoniously inserted. Since the insertion network also integrates the Adaptive Target-Aware Attention, we further apply the Attention-Guided Fusion to preserve background fidelity while synthesizing realistic object effects.

Object Movement. To enable object movement, we combine our object removal (ObjectClear) and insertion models into a two-stage framework. Specifically, we first apply ObjectClear to remove the target object and its associated effects, producing a clean, object-free background. The object is then extracted using its provided object mask, and users can specify a new location and adjust its scale before reinsertion. With our insertion network, the object is harmonized with the new context by generating realistic effects. Without retraining any additional networks, this two-stage pipeline supports controllable object movement while maintaining visual realism and consistency.

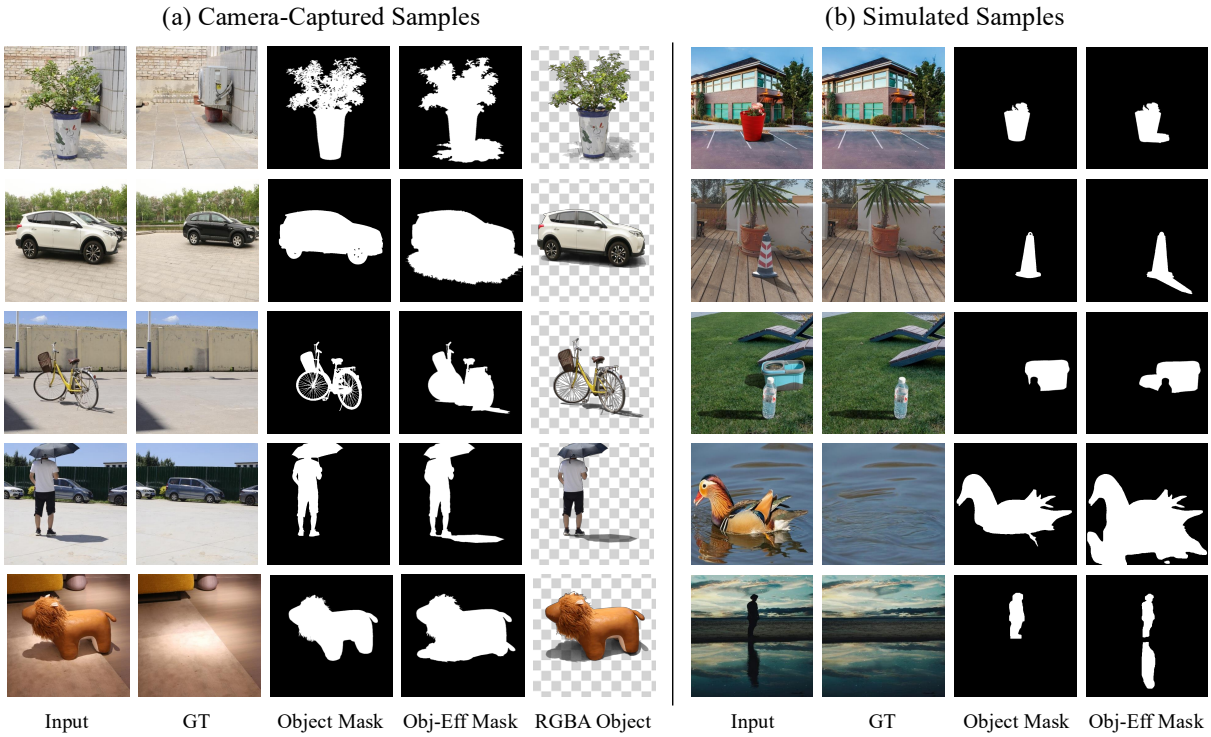


Figure A. **Samples from our OBER Dataset.** We provide some samples from our OBER dataset including both camera-captured data and simulated data. While all data from both categories are annotated with fine-grained object masks and object-effect masks, we also extract RGBA foreground objects with associated effects from camera-captured data (as discussed in Sec. 3.1 in the main paper), which could be used to construct realistic simulated samples. In the simulated samples, we not only include the shadow effects but also the reflection effects, which are rarely included in other datasets, thanks to our reflection pair simulation strategy detailed in Sec. B.

Table A. **Dataset Overview.** The OBER dataset consists of camera-captured and simulated training data, as well as two testing sets: OBER-Test (w/ ground truth) and OBER-Wild (w/o ground truth). All data subsets are annotated with object masks and object-effect masks.

Properties	**Training Set**		**Testing Set**	
	Camera-Captured	Simulated	OBER-Test	OBER-Wild
#Image Pairs	2,715	10,000	163	302
w/ Ground Truth	✓	✓	✓	×
w/ Object Mask	✓	✓	✓	✓
w/ Object-Effect Mask	✓	✓	✓	✓

B. More Details of OBER Dataset

Overview. Table A provides an overview of our OBER dataset [link]. The training set consists of 2,715 camera-captured image pairs and 10,000 simulated pairs, all annotated with object masks and object-effect masks. For evaluation, we provide two test subsets: OBER-Test (includes 163 pairs with ground truth) and OBER-Wild (includes 302 pairs without ground truth), where object masks and object-effect masks are also available. Figure A presents samples from our OBER dataset, including camera-captured and simulated data with annotations such as object masks, object-effect masks, and RGBA foregrounds.

Reflection Pair Simulation. Thanks to the strong priors of generative models, we observed that a model trained with only limited indoor mirror-reflection data can still generalize to removing simple outdoor reflection effects (*e.g.*, water reflections), which is consistent with the findings in ObjectDrop [13]. However, the model often fails in more challenging cases, particularly when the object and its reflection are spatially separated or when the reflection is heavily distorted by water ripples. To address this limitation, we adopt a human-in-the-loop strategy to collect paired reflection data. Specifically, we first use our trained model to perform inference on 200 real-world reflection images, and then manually select 50 high-quality results (see examples

Table B. **Comparison of OBER Dataset with Existing Datasets.** $*$ indicates that the dataset is not publicly available.

	Description	RORD [9]	MULAN [11]	DESOBAv2 [7]	Counterfactual* [13]	Video4Removal* [12]	OBER (ours)
Tasks	Object Removal	×	✓	×	✓	✓	✓
	Effect Removal	×	×	✓	✓	✓	✓
	Object-Effect Removal	✓	×	×	✓	✓	✓
Annotations	Object Mask	×	✓	✓	✓	✓	✓
	Effect Mask	×	×	✓	×	×	✓
	Object-Effect Mask	✓(Coarse)	×	✓	×	×	✓
	RGBA Objects	×	×	×	×	×	✓
	Multi Objects	✓	✓	×	✓	✓	✓
	Camera-Captured GT	✓	×	×	✓	✓	✓

in Fig. A(b)), which are then added as an important supplement to the training data. We found that even a small amount of high-quality reflection pairs can significantly improve the model's generalization ability across diverse reflection scenarios.

Comparison with Existing Datasets. We compare our OBER dataset with existing datasets, including those focused on shadow removal (DESOBA-v2 [7]) and object removal (RORD [9], MULAN [11], Counterfactual Dataset [13], Video4Removal [12]). As summarized in Table B, unlike prior datasets, OBER provides all three types of mask annotations as well as RGBA object foregrounds, enabling a wide range of tasks, including object removal, effect removal, and joint object–effect removal.

DESOBA-v2 [7] targets only effect removal and thus does not handle the objects. In contrast, MULAN [11] focuses solely on object removal without associated effects such as shadows or reflections. Moreover, the ground-truth images in both DESOBA-v2 and MULAN are produced by existing inpainting models, which may limit their visual realism. RORD [9] is among the first to provide object–effect masks, however, these masks are *very coarse* and can not separately annotate objects and their effects. Consequently, it cannot support independent removal of either the object or the effect. The Counterfactual Dataset [13] and Video4Removal Dataset [12] are designed for object–effect removal tasks, the same with ours, but they are not publicly accessible. Furthermore, these datasets provide only object masks, lacking effect masks and RGBA foregrounds, which limits their scalability and diverse usage. In contrast, our OBER dataset provides richer annotations, including precise and separate masks for objects and their effects, together with RGBA foregrounds, enabling more flexible and fine-grained removal tasks. Notably, the effect masks serve as crucial supervision for learning accurate object–effect removal while preserving background fidelity. OBER is a hybrid dataset comprising high-quality captured and realistic synthetic data, covering diverse and complex scenarios such as multi-object occlusions and indoor/outdoor reflections. **We will release our OBER dataset publicly**, which we believe will significantly benefit future research in this field.

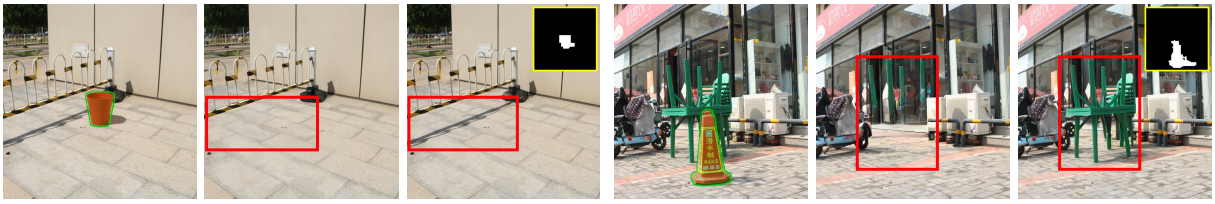


Figure B. **Effectiveness of Adaptive Target-Aware Attention (ATA).** It could be observed that when training without ATA (w/o $\mathcal{L}_{mask}$), the model struggles to accurately remove the target object and its effects, leading to mistakenly erasing unrelated background object (*right*) or effects (*left*). In contrast, when introducing TAT trained with the mask supervision $\mathcal{L}_{mask}$, the attention maps (shown in yellow boxes) can accurately localize the removal regions, leading to more precise and complete removal results.

C. More Results

C.1. Results for Ablation Study

Effectiveness of Adaptive Target-Aware Attention. Thanks to the OBER dataset's rich annotations, Adaptive Target-Aware Attention (ATA) leverages the object-effect mask and its supervision loss $\mathcal{L}_{mask}$. It guides cross-attention layers to focus on the object and its associated effects while preserving background textures, which enables decoupled optimization of object

Figure C. **Effectiveness of Simulated Data.** Since our simulation data includes multi-object compositions, training with such data enables the model to accurately remove the target object and its associated effects while preserving unrelated object effects (*left*). In addition, adding the reflection data pairs during training greatly enhances the model capability of removing reflections, even in challenging cases (*right*).

Figure D. **Effectiveness of Spatially-Varying Denoising Strength (SVDS).** $\mathrm{DS} = 0.99$ often leads to incomplete removal or hallucinated objects, while $\mathrm{DS} = 1.0$ causes noticeable color inconsistency (shown after AGF, where the background is from the input image and the object/affected areas are from the removal results). In contrast, SVDS achieves complete object removal with consistent background colors.

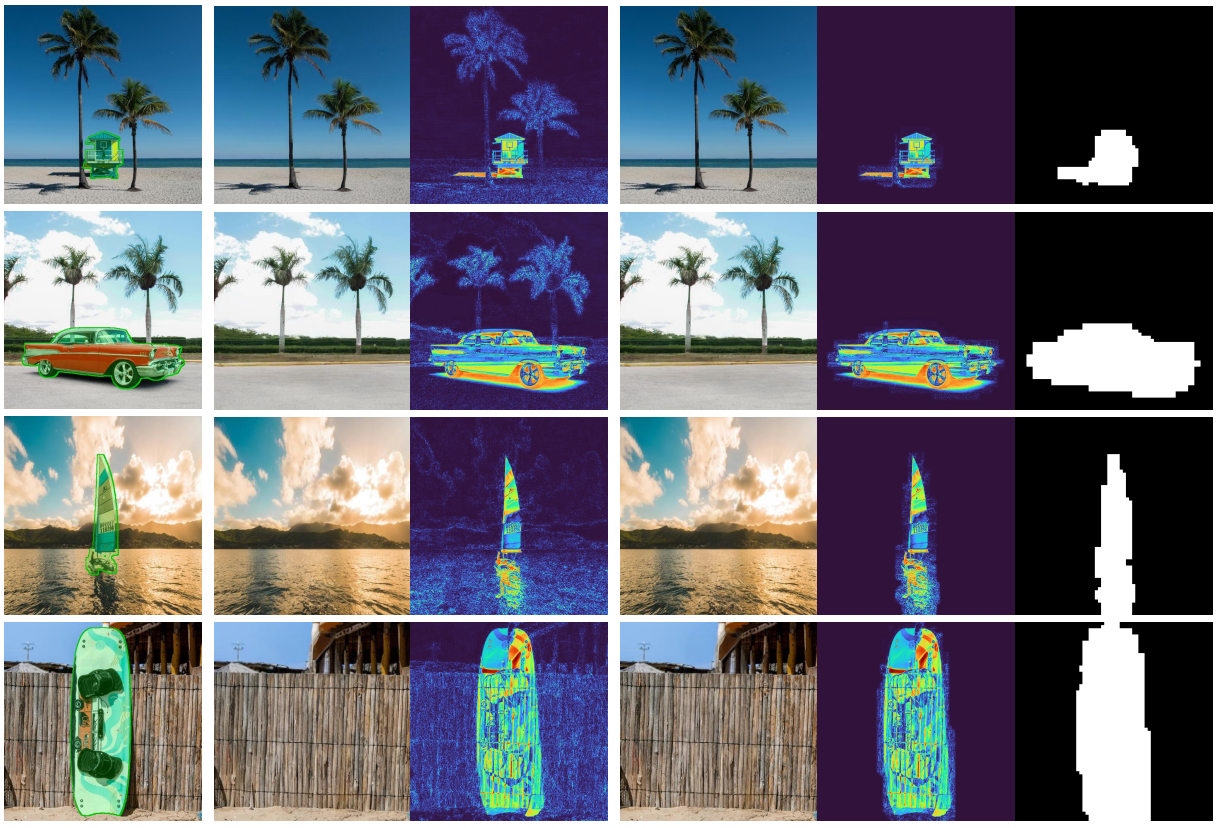


Figure E. **Effectiveness of Attention-Guided Fusion (AG Fusion).** We visualize background detail changes by computing difference maps between the generated image and input. Ideally, these maps should exhibit large values only within the object removal regions, covering the object and its effects, while remaining minimal in the unrelated background areas. As observed, without AG Fusion, noticeable differences appear in the background. In contrast, with the AG Fusion strategy, these undesired background discrepancies are largely eliminated.

Table C. **Quantitative comparison on OmniPaint-Bench and OmniEraser-Bench.** The best and second performances are marked in red and orange, respectively.

Methods	OmniPaint-Bench				OmniEraser-Bench			
	PSNR↑	LPIPS↓	DINO↓	CLIP↓	PSNR↑	LPIPS↓	DINO↓	CLIP↓
SDXL-INP [8]	20.51	0.1924	0.0929	0.1100	21.26	0.1968	0.0938	0.0926
PowerPaint [15]	21.55	0.1882	0.0874	0.0990	22.21	0.2001	0.0911	0.0907
BrushNet [5]	19.00	0.2733	0.1909	0.1446	20.58	0.2098	0.1182	0.1098
DesignEdit [3]	24.32	0.1763	0.0769	0.0717	23.87	0.2168	0.1020	0.0755
CLIPAway [2]	20.07	0.1985	0.0929	0.1129	20.78	0.2035	0.0934	0.0956
FreeCompose [1]	22.25	0.1631	0.0723	0.0770	22.60	0.1782	0.0733	0.0688
Attentive Eraser [10]	24.17	0.1523	0.0501	0.0590	24.77	0.1538	0.0463	0.0388
RORem [6]	23.92	0.1462	0.0478	0.0576	23.70	0.1746	0.0532	0.0416
OmniEraser [12]	23.02	0.2034	0.0564	0.0640	23.83	0.1766	0.0460	0.0481
OmniPaint [14]	25.56	0.1001	0.0249	0.0367	25.67	0.1069	0.0213	0.0182
ObjectClear (Ours)	26.35	0.0950	0.0244	0.0334	27.90	0.0942	0.0230	0.0142

removal and background reconstruction. As shown in Fig. B, ATA adaptively identifies object-effect regions to be removed, as reflected in the attention mask (shown in yellow boxes). This leads to more accurate and complete removal of objects and their shadow effects, without mistakenly erasing unrelated background content.

Effectiveness of Simulated Data. To balance realism and scalability, in addition to camera-captured data, we augment OBER with a simulation pipeline. Our simulation data is generated by compositing the RGBA foreground objects (extracted from camera-captured data) onto diverse backgrounds. In particular, the simulation of multi-object compositions leads to notable improvements in object removal robustness under mutual occlusions, as shown in Fig. C (*left*). Furthermore, our simulated reflection pairs greatly improve the model capability of removing reflections, even in challenging cases (Fig. C, *right*).

Effectiveness of Attention-Guided Fusion. The attention map supervised by $\mathcal{L}_{mask}$ supports the Attention-Guided Fusion (AG Fusion) strategy during inference. It helps to blend the generated image with the original input via a copy-and-paste operation, where pixels within the object-effect region are taken from the generated result, and the rest are preserved from the original image. Such practice effectively reduces undesired background detail changes caused by VAE reconstruction errors and the diffusion process, thereby greatly preserving the background fidelity. In Fig. E, we visualize the background detail changes by showing the difference maps between the generated image and the corresponding input, where a clear improvement on background preservation could be observed.

Effectiveness of Spatially-Varying Denoising Strength. In diffusion-based image editing, the initial latent for denoising is typically obtained by adding noise to the input image latent. The denoising strength (DS), $\mathrm{DS} \in [0, 1]$, controls the noise level: a larger value injects more noise, thereby pushing the initial noisy latent closer to the pure-noise prior. When $\mathrm{DS} = 1.0$, the diffusion process starts entirely from noise, discarding information from the input. In this paper, we propose Spatial-Varying Denoising Strength (SVDS), which applies $\mathrm{DS} = 1.0$ within the masked object region and $\mathrm{DS} = 0.99$ (an empirical setting commonly adopted by previous methods [8]) outside in the unmasked background, ensuring complete object removal while maintaining color consistency. As shown in Fig. D, setting $\mathrm{DS} = 0.99$ often leads to incomplete removal or hallucinated objects, whereas $\mathrm{DS} = 1.0$ results in noticeable color inconsistency. To highlight this inconsistency, the results of $\mathrm{DS} = 1.0$ in Fig. D are obtained after the Attention-Guided Fusion (AGF) operation, where the background is taken from the original input image, while the object and affected areas are taken from the removal results. In contrast, our method with SVDS achieves superior performance in both object removal and preservation of background color consistency.

C.2. Comparisons on Additional Benchmarks

To further evaluate the robustness and generalization ability of our method, we conduct additional evaluations on three benchmarks, *i.e.*, OmniEraser-Bench [12], OmniPaint-Bench [14], MULAN [11].

OmniEraser-Bench and OmniPaint-Bench Datasets. OmniEraser-Bench [12] and OmniPaint-Bench [14] are two recent benchmarks for object–effect removal, which align with our task setting. For a fair comparison, all methods use their default input sizes (OmniEraser [12] at 1024, all others at 512), then resize the outputs to the same size (short side 512) for evaluation. Table C shows our approach outperforms all baselines across metrics on both benchmarks.

Table D. **Quantitative comparison on MULAN Dataset.** The best and second performances are marked in red and orange, respectively.

Methods	PSNR↑	LPIPS↓	DINO↓	CLIP↓
SDXL-INP [8]	19.91	0.2494	0.1324	0.1312
PowerPaint [15]	21.18	0.2449	0.1087	0.0962
BrushNet [5]	18.22	0.3181	0.2062	0.1893
DesignEdit [3]	23.26	0.2375	0.1114	0.0725
CLIPAway [2]	20.08	0.2666	0.1180	0.1152
FreeCompose [1]	21.30	0.2337	0.0828	0.0703
Attentive Eraser [10]	23.96	0.1960	0.0551	0.0397
RORem [6]	23.53	0.2369	0.0571	0.0438
OmniEraser [12]	21.56	0.2642	0.0728	0.0682
OmniPaint [14]	22.29	0.1915	0.0650	0.0550
ObjectClear (Ours)	24.89	0.1586	0.0468	0.0373

MULAN Dataset. Since the ground truth of some samples in MULAN [11] retains shadows or reflections, it is not suitable for evaluating object–effect removal. Therefore, we randomly sample 500 "effect-free" image pairs from MULAN for our evaluation. As shown in Table D, our method achieves the best performance across all metrics, even surpassing RORem [6], which is trained on MULAN, demonstrating the strong object removal capability of our approach. This also confirm that our model performs robustly on "effect-free" object removal datasets, without introducing negative effects when the objects do not show any shadows or reflections.

C.3. Fair Comparison with Object-Effect Mask

Our Attention-Guided Fusion (AGF) module leverages object-effect masks predicted by the proposed Adaptive Target-Aware Attention to blend the original input background back into the generated result. Importantly, these masks are predicted by our model rather than taken from annotations, ensuring that we do not use any privileged information unavailable to other approaches. Furthermore, existing baseline methods do not have the capability to predict object–effect masks and therefore cannot perform background blending in the same way. This makes AGF an integral part of our model design rather than an external post-processing step, and the comparisons in the main paper are therefore fair.

To further demonstrate that the performance gain is not solely due to the background blending, we perform an additional experiment where all baseline methods are given the ground-truth object-effect mask for blending. As shown in Table E, our method continues to outperform all baselines under this setting, indicating that our superior results primarily come from more effective object–effect generation rather than the availability of blending masks.

Table E. **Quantitative results on RORD-Val with object-effect masks for blending.** All baseline methods are equipped with background blending using ground-truth object-effect masks. The best and second performances are marked in red and orange, respectively. Our method using our predicted object-effect masks achieves the best performance across all metrics.

Method	PSNR↑	LPIPS↓	DINO↓	CLIP↓
SDXL-INP [8] (w/b)	21.67	0.1592	0.0688	0.0932
PowerPaint [15] (w/b)	22.51	0.1472	0.0560	0.0606
BrushNet [5] (w/b)	18.48	0.2421	0.1572	0.1465
DesignEdit [3] (w/b)	23.73	0.1580	0.0721	0.0755
CLIPAway [2] (w/b)	21.96	0.1735	0.0666	0.0734
FreeCompose [1] (w/b)	23.08	0.1603	0.0829	0.0834
Attentive Eraser [10] (w/b)	22.90	0.1700	0.0880	0.0854
RORem [6] (w/b)	25.24	0.1398	0.0387	0.0498
ObjectClear (Ours)	26.24	0.1157	0.0191	0.0299

C.4. Generalization to Multi-Object Removal

The OBER dataset also covers multi-object removal cases, and our model generalizes well to such scenarios (Fig. F). This reflects the strong generalization capability of a network trained on our dataset.

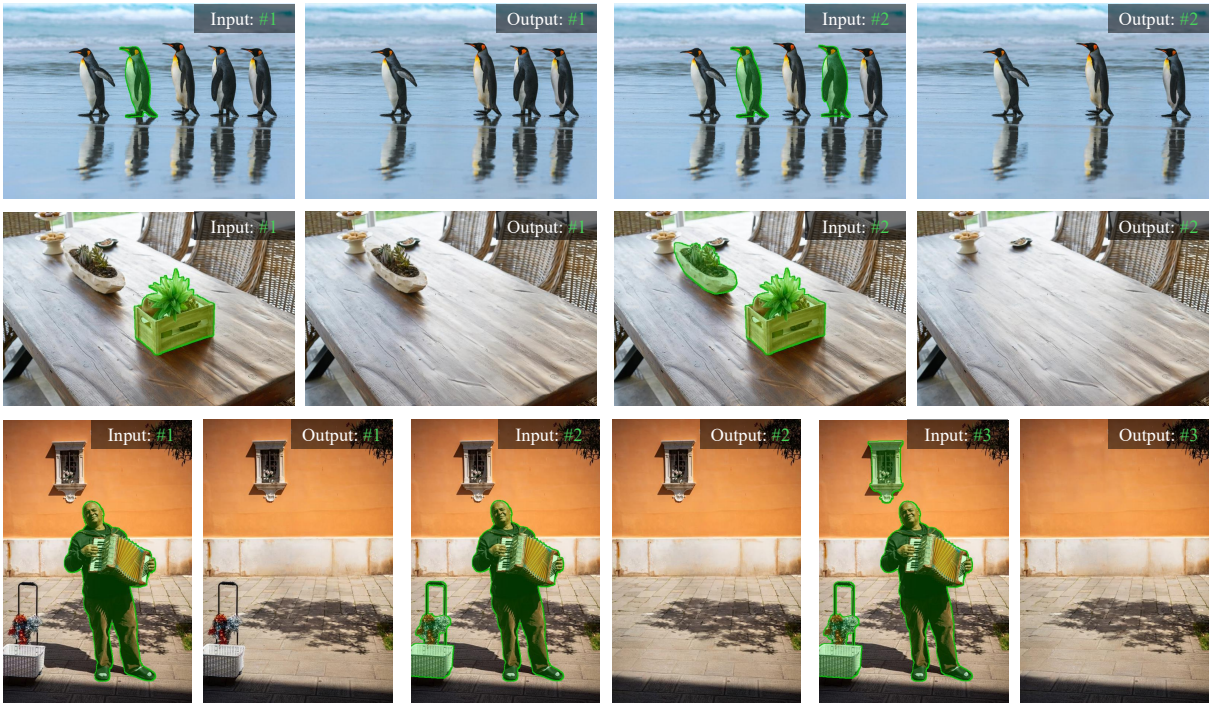


Figure F. **Multi-object removal.** Our method removes one or multiple objects simultaneously using masks to indicate the targets.

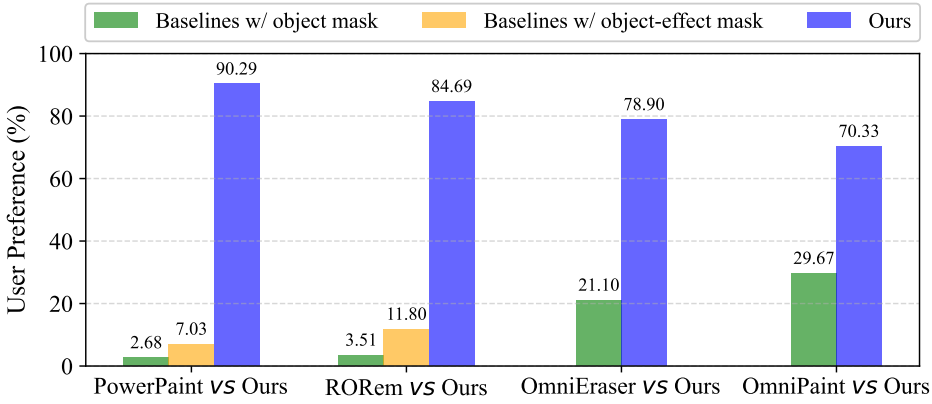


Figure G. **User Study.** Our ObjectClear is preferred by human voters over four representative state-of-the-art methods (PowerPaint [15], RORem [6], OmniEraser [12], and OmniPaint [14].

C.5. User Study

To enable a more comprehensive evaluation, we conducted a user study on object removal results for in-the-wild images. We compared ObjectClear with four representative state-of-the-art methods: PowerPaint [15], RORem [6], OmniEraser [12], and OmniPaint [14]. For a fair comparison, we evaluated PowerPaint and RORem under two settings: (1) conditioned on the object mask and (2) conditioned on the object-effect mask, OmniEraser and OmniPaint are conditioned only on the object mask. All ObjectClear results were generated using only the object mask, and we compared our outputs against both settings of PowerPaint and RORem, as well as the object-mask setting of OmniEraser and OmniPaint.

We invite a total of 30 participants for this user study. Each volunteer was presented 80 randomly selected image quadruples, consisting of: *an input image*, *two results* from a baseline method under different mask conditions, and *our result* (for OmniEraser and OmniPaint, only the object-mask result was provided, consistent with our setting, thus forming a triple set). Participants were asked to select the best removal result based on two criteria: the realism of the object region and the preservation of background details. As summarized in Fig G, ObjectClear outperforms the baselines under both mask settings. Notably, although some baseline methods benefited from access to object-effect masks, our ObjectClear won more user preference with the object mask only.

C.6. Results with User Strokes

In practical applications, users often interact with visual systems through imprecise or casually drawn inputs, such as rough scribbles or incomplete masks. These inputs may vary significantly in shape, location, and accuracy. Therefore, it is essential for a robust object removal network to effectively process such arbitrary mask inputs without relying on carefully crafted annotations. Benefiting from our mask augmentation strategy and Adaptive Target-Aware Attention mechanism, our network demonstrates strong robustness to diverse mask inputs. In this subsection, we simulate user strokes and feed them into the network along with the images. The resulting outputs and attention maps show that our network can accurately identify and attend to the object and its associated effects, even with imprecise masks, as illustrated in Fig. H.

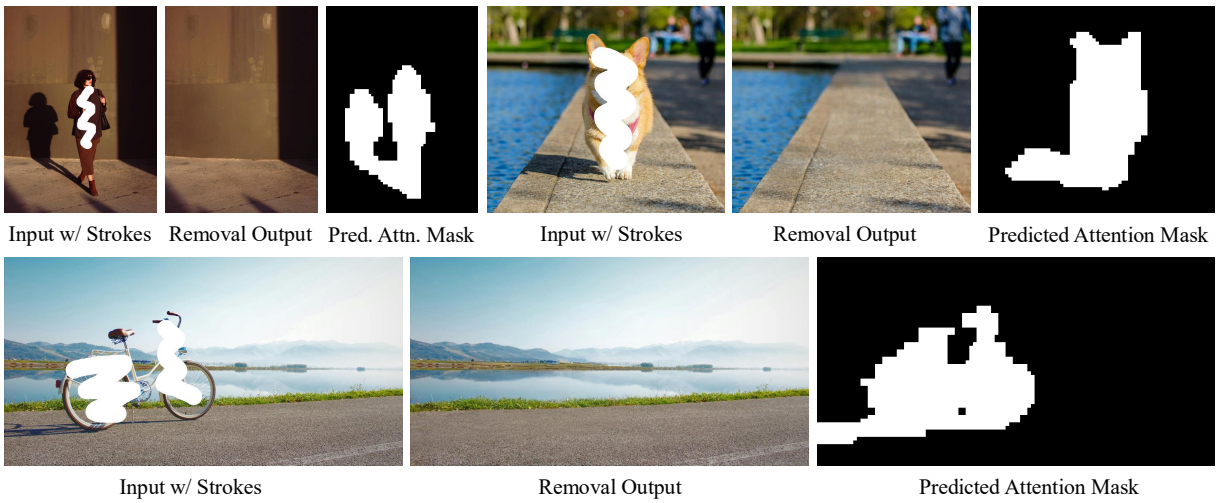


Figure H. **Results with User Strokes.** We simulate user strokes and feed them into the network along with the images. The resulting outputs and attention maps show that our network can accurately identify and attend to the object and its effects, even with imprecise masks.

C.7. Object Insertion and Movement.

As shown in Fig. I, even when only the target objects are specified for insertion and movement, ObjectClear is capable of generating plausible and natural shadows and reflections accordingly.

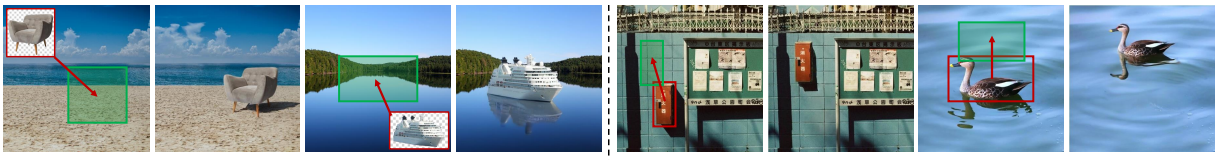


Figure I. **Object Insertion and Movement.** In addition to accurately inserting or repositioning objects, our ObjectClear also generates plausible and natural shadows and reflections accordingly.

C.8. More Comparisons on In-the-wild Data

In this subsection, we conduct comprehensive comparisons of ObjectClear with state-of-the-art methods across two categories: *image inpainting methods* and *object removal methods*, on the in-the-wild data.

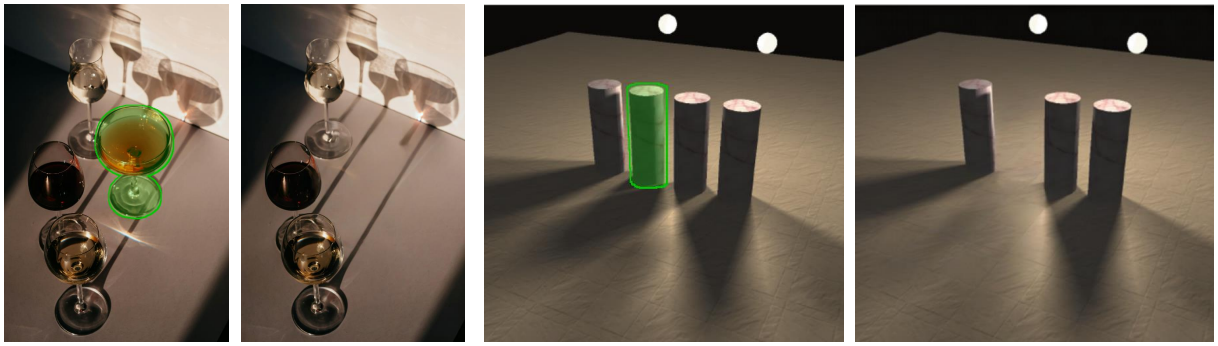
Comparison with Image Inpainting Methods. We compare our method with state-of-the-art image inpainting methods, including PowerPaint [15], Attentive Eraser [10], DesignEdit [3], and RORem [6]. Since these inpainting methods often struggle to remove object-associated effects when only provided with an object mask, we supply them with our annotated Object-Effect Masks for fair comparison, while our ObjectClear uses only Object Masks as input. As shown in Fig. M, although these inpainting methods can remove shadows with the guidance of additional effect regions, they frequently introduce undesirable changes to the original background. In contrast, ObjectClear effectively eliminates the target object and its associated effects using only the object mask, while preserving background content with high fidelity.

Comparison with Object Removal Methods. We further compare ObjectClear with object removal methods (OmniEraser [12] and OmniPaint [14]) under a unified input setting, all methods use Object Masks as input to ensure fairness. As illustrated in Fig. L, our ObjectClear demonstrates superior stability compared to the two baselines: it not only successfully eliminates the target object and its associated effects but also avoids unintended changes to the original background, preserving background fidelity with high precision. To further highlight the background preservation advantage, we visualize the pixel-wise difference maps of the compared methods in Fig. M. ObjectClear maintains minimal pixel differences in non-target background areas, which clearly validates its ability to preserve the original background while accurately removing the object and its effects.

Efficiency Comparison with Object Removal Methods. ObjectClear is significantly more efficient than both OmniEraser [12] and OmniPaint [14]. While the latter two are built on Flux ($\sim$ 12B parameters), ObjectClear is based on SDXL ($\sim$ 3.7B), making our model considerably lighter. As a result, ObjectClear runs over $5\times$ faster during inference, *i.e.*, 1.63 s/image (ObjectClear) vs. 9.42 s/image (OmniPaint) at 512×512 resolution on an A100 GPU.

C.9. Limitations

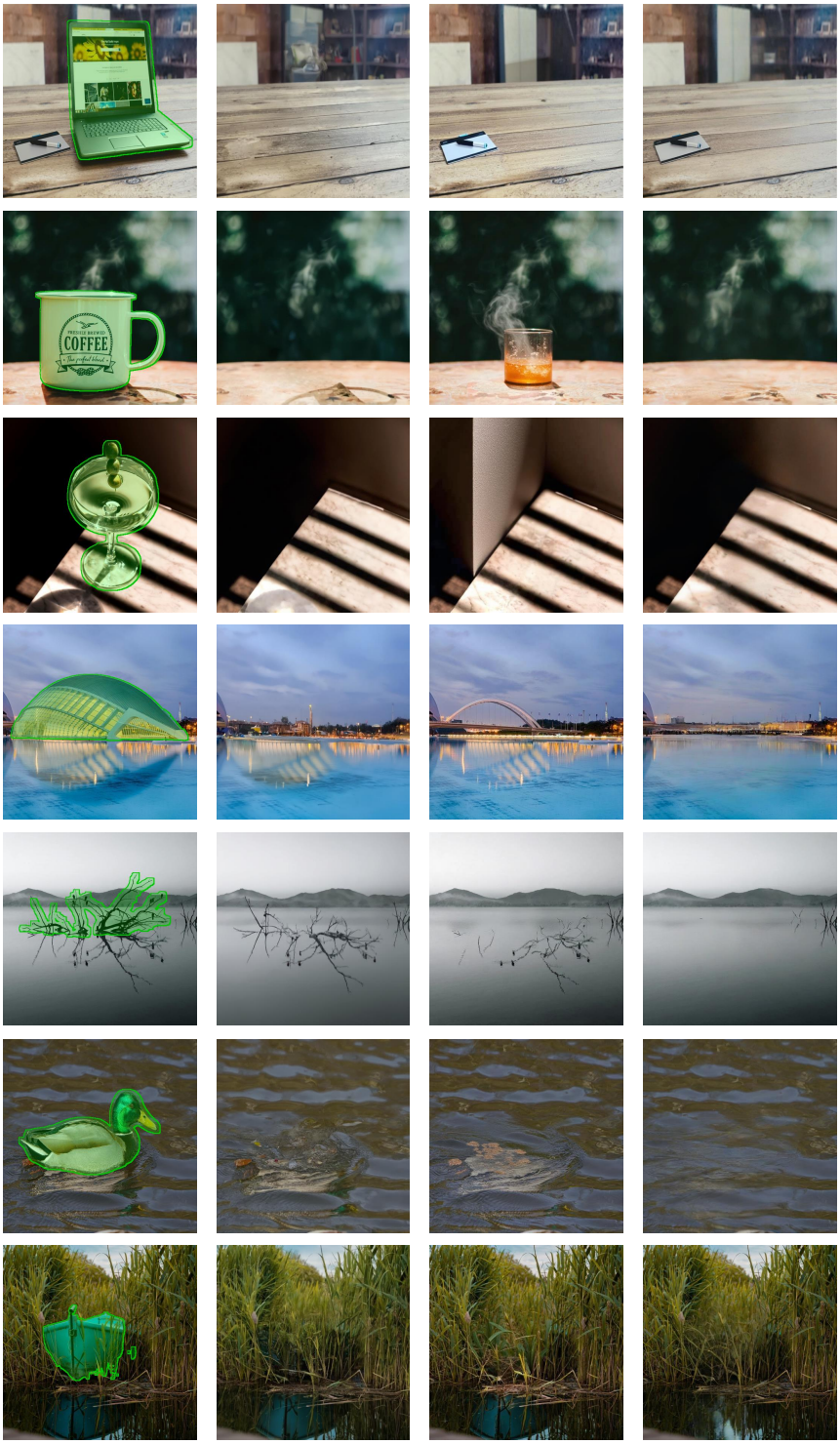
While ObjectClear exhibits strong performance in removing objects and their associated effects, it still faces challenges in highly complex scenarios. Specifically, in cases with overlapping shadows from multiple objects or complex lighting conditions, it can be difficult to disentangle which shadows belong to which objects. As a result, the model may fail to remove the shadows of the target object (Fig. Ja) or remove shadows of other objects (Fig. Jb). Effectively disentangling object-specific shadows in such complex scenes remains an important direction for future work.



(a) Under-Removal with Effect Interactions (b) Over-Removal with Effect Interactions

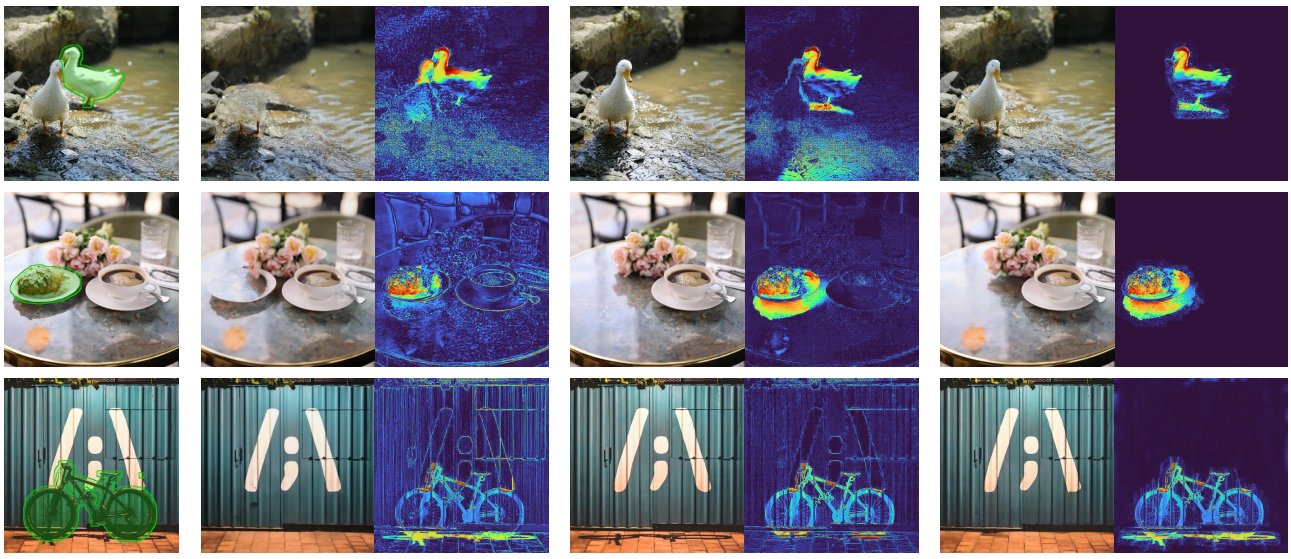
Figure J. **Limitations.** In complex scenes where multiple objects may produce overlapping or intertwined effects (*e.g.*, shadows and reflections), our method can consistently remove the objects but may sometimes fail to precisely eliminate the associated effects. This results in either (a) *under-removal of the target effect* or (b) *over-removal of effects belonging to nearby objects*.

Figure K. **Comparison with Image Inpainting Models.** Since some methods struggle to remove object effects when provided with only the object mask, we supply them with our annotated Object-Effect Mask for a fair comparison, while our ObjectClear uses only Object Mask as input. Although these methods are able to remove shadows with the additional effect region, they often introduce undesirable changes to the original background. In contrast, ObjectClear effectively removes the object and its associated effects using only the object mask, while preserving the background content with high fidelity.



Input with Mask | OmniEraser | OmniPaint | ObjectClear (Ours)

Figure L. **Comparison with Object Removal Models.** All compared methods use Object Masks as input for fair comparison. Our ObjectClear demonstrates superior stability: it not only successfully eliminates the target object and its associated effects, but also avoids unintended changes to the original background, preserving background fidelity with high precision.



Input with Mask | OmniEraser & Diff. Map | OmniPaint & Diff. Map | ObjectClear (Ours) & Diff. Map

Figure M. **Comparison of Object Removal Models via Difference Maps.** We present results of representative object removal methods, and their pixel-wise difference maps relative to the input. ObjectClear successfully removes the target object and its associated shadows while preserving the background. This advantage is clearly reflected by the low pixel-wise differences in non-target areas.

References

[1] Zhekai Chen, Wen Wang, Zhen Yang, Zeqing Yuan, Hao Chen, and Chunhua Shen. FreeCompose: Generic zero-shot image composition with diffusion prior. In *ECCV*, 2024. 5, 6

[2] Yiğit Ekin, Ahmet Burak Yildirim, Erdem Eren Çağlar, Aykut Erdem, Erkut Erdem, and Aysegul Dundar. CLIPAway: Harmonizing focused embeddings for removing objects via diffusion models. In *NeurIPS*, 2024. 5, 6

[3] Yueru Jia, Yuhui Yuan, Aosong Cheng, Chuke Wang, Ji Li, Huizhu Jia, and Shanghang Zhang. DesignEdit: Multi-layered latent decomposition and fusion for unified & accurate image editing. In *AAAI*, 2025. 5, 6, 9

[4] Longtao Jiang, Zhendong Wang, Jianmin Bao, Wengang Zhou, Dongdong Chen, Lei Shi, Dong Chen, and Houqiang Li. SmartEraser: Remove anything from images using masked-region guidance. In *CVPR*, 2025. 1

[5] Xuan Ju, Xian Liu, Xintao Wang, Yuxuan Bian, Ying Shan, and Qiang Xu. BrushNet: A plug-and-play image inpainting model with decomposed dual-branch diffusion. In *ECCV*, 2024. 5, 6

[6] Ruibin Li, Tao Yang, Song Guo, and Lei Zhang. RORem: Training a robust object remover with human-in-the-loop. In *CVPR*, 2025. 5, 6, 7, 9

[7] Qingyang Liu, Junqi You, Jianting Wang, Xinhao Tao, Bo Zhang, and Li Niu. Shadow generation for composite image using diffusion model. In *CVPR*, 2024. 3

[8] 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. *ICLR*, 2024. 5, 6

[9] Min-Cheol Sagong, Yoon-Jae Yeo, Seung-Won Jung, and Sung-Jea Ko. RORD: A real-world object removal dataset. In *BMVC*, 2022. 3

[10] Wenhao Sun, Benlei Cui, Xue-Mei Dong, and Jingqun Tang. Attentive Eraser: Unleashing diffusion model's object removal potential via self-attention redirection guidance. In *AAAI*, 2025. 5, 6, 9

[11] Petru-Daniel Tudosiu, Yongxin Yang, Shifeng Zhang, Fei Chen, Steven McDonagh, Gerasimos Lampouras, Ignacio Iacobacci, and Sarah Parisot. MULAN: A multi layer annotated dataset for controllable text-to-image generation. In *CVPR*, 2024. 3, 5, 6

[12] Runpu Wei, Zijin Yin, Shuo Zhang, Lanxiang Zhou, Xueyi Wang, Chao Ban, Tianwei Cao, Hao Sun, Zhongjiang He, Kongming Liang, and Zhanyu Ma. OmniEraser: Remove objects and their effects in images with paired video-frame data. *arXiv preprint arXiv:2501.07397*, 2025. 1, 3, 5, 6, 7, 9

[13] Daniel Winter, Matan Cohen, Shlomi Fruchter, Yael Pritch, Alex Rav-Acha, and Yedid Hoshen. ObjectDrop: Bootstrapping counterfactuals for photorealistic object removal and insertion. In *ECCV*, 2024. 1, 2, 3

[14] Yongsheng Yu, Ziyun Zeng, Haitian Zheng, and Jiebo Luo. OmniPaint: Mastering object-oriented editing via disentangled insertion-removal inpainting. In *ICCV*, 2025. 5, 6, 7, 9

[15] Junhao Zhuang, Yanhong Zeng, Wenran Liu, Chun Yuan, and Kai Chen. A task is worth one word: Learning with task prompts for high-quality versatile image inpainting. In *ECCV*, 2024. 5, 6, 7, 9