

Generative Image Layer Decomposition with Visual Effects

Jinrui Yang^{1,2,*} Qing Liu² Yijun Li² Soo Ye Kim² Daniil Pakhomov²
 Mengwei Ren² Jianming Zhang² Zhe Lin² Cihang Xie¹ Yuyin Zhou¹
¹UC Santa Cruz ²Adobe Research

<https://rayjryang.github.io/LayerDecomp>

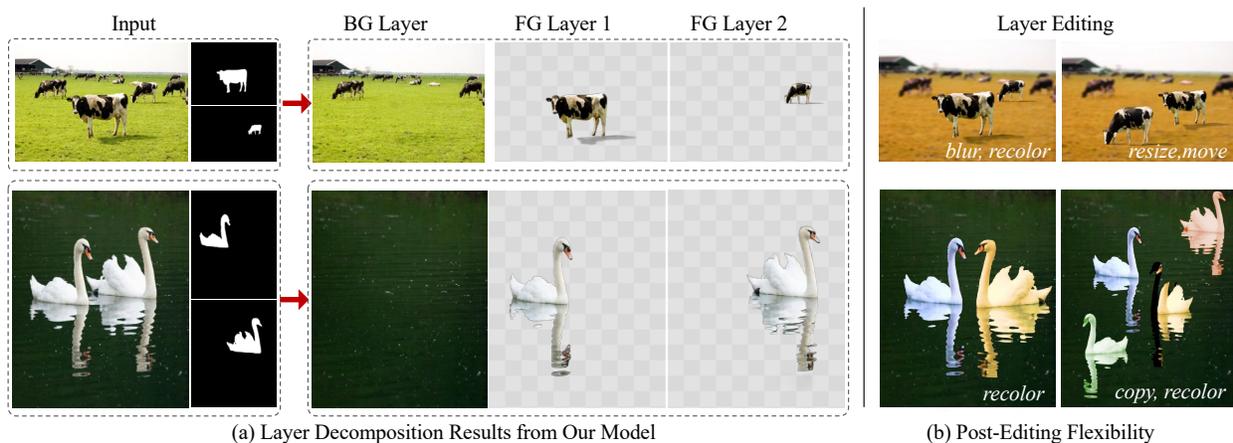


Figure 1. (a) Given an input image and a binary object mask, our model is able to decompose the image into a clean background layer and a transparent foreground layer with preserved visual effects such as shadows and reflections. (b) Subsequently, our decomposition empowers complex and controllable layer-wise editing such as spatial, color and/or style editing.

Abstract

Recent advancements in large generative models, particularly diffusion-based methods, have significantly enhanced the capabilities of image editing. However, achieving precise control over image composition tasks remains a challenge. Layered representations, which allow for independent editing of image components, are essential for user-driven content creation, yet existing approaches often struggle to decompose an image into plausible layers with accurately retained transparent visual effects such as shadows and reflections. We propose **LayerDecomp**, a generative framework for image layer decomposition which outputs photorealistic clean backgrounds and high-quality transparent foregrounds with faithfully preserved visual effects. To enable effective training, we first introduce a dataset preparation pipeline that automatically scales up simulated multi-layer data with synthesized visual effects. To further enhance real-world applicability, we supplement this simulated dataset with camera-captured images containing natural visual effects. Additionally, we propose a consistency loss which enforces the model to learn accurate

representations for the transparent foreground layer when ground-truth annotations are not available. Our method achieves superior quality in layer decomposition, outperforming existing approaches in object removal and spatial editing tasks across several benchmarks and multiple user studies, unlocking various creative possibilities for layer-wise image editing.

1. Introduction

The rapid advancement of large-scale text-to-image diffusion models [3, 7, 32] has greatly improved image editing capabilities, with recent studies [5, 14, 35, 51] demonstrating promising results by training on large-scale datasets of captioned images. However, achieving precise control for user-driven image composition tasks remains challenging. Layered representations, which decompose image components into independently editable layers, are essential for precise user-driven content creation. Most visual content editing software and workflows are layer-based, relying heavily on transparent or layered elements to compose and

*This work was done when Jinrui Yang was a research intern at Adobe Research.

create content. Despite this, few existing approaches have explored layer-based representation for image editing in depth. Recently, LayerDiffusion [48] is proposed to generate transparent layer representations from text inputs; however, this approach is not well-suited for image-to-image editing tasks. Meanwhile, MULAN [38] presents a multi-layer annotated dataset for controllable generation, but it often fails to preserve essential visual effects in the right layers, limiting its adaptability for seamless downstream edits.

This work aims to fill these gaps by decomposing an input image into two layers: a highly plausible clean background layer, and a high-quality transparent foreground layer that retains natural visual effects associated with the target. These decomposed layers will support layer-constrained content modification and allow seamless blending for harmonious re-composition. Additionally, the faithfully preserved natural visual effects in the foreground layer will benefit related research areas, such as shadow detection and shadow generation. However, the lack of publicly available multi-layer datasets with realistic visual effects, such as shadows and reflections, poses a significant challenge for training high-quality layer-wise decomposition models. How to accurately decompose images into layers and learn correct representation for visual effects in the foreground layer without ground-truth data is the key problem to solve.

To address these challenges, we first design a dataset preparation pipeline to collect data from two sources: (1) simulated image triplets consisting of composite images blended from random background and unrelated transparent foreground with generated shadow, allowing us to create a large-scale training set with ground-truth for both branches; and (2) camera-captured images for a scene with and without the target foreground object, ensuring the model can effectively adapt to real-world scenarios. Building on this dataset, we introduce **Layer Decomposition with Visual Effects (LAYERDECOMP)**, a generative training framework that enables large-scale pretrained diffusion transformers to effectively decompose images into editable layers with correct representation of visual effects. The key to our model training lies in a consistency loss, which ensures faithful retention of natural visual effects within the foreground layer while maintaining background coherence. Specifically, for real-world data where visual effect annotations are not available, it is not feasible to directly compute diffusion loss on the foreground layer. Instead, LAYERDECOMP enforces consistency between the original input image and the re-composite result, blended from the two predicted layers, to ensure the model learns correct representation for the transparent foreground layer.

As shown in Figure 1, LAYERDECOMP can effectively decompose an input image into a clean background and a transparent foreground with preserved visual effects. This capability supports downstream editing applications with-

out requiring additional model training or inference. Furthermore, our method exhibits superior layer decomposition quality, outperforming existing approaches in both object removal tasks and object spatial editing tasks across various benchmarks and in multiple user studies.

In summary, our contributions are as follows:

- We introduce a scalable pipeline to generate large-scale simulated multi-layer data with paired ground-truth visual effects for training layer decomposition models.
- We present LAYERDECOMP, a generative training framework that leverages both simulated and real-world data to enable robust layer-wise decomposition with accurate visual effect representation through a consistency loss, facilitating high-quality, training-free downstream edits.
- LAYERDECOMP surpasses existing state-of-the-arts in maintaining visual integrity during layer decomposition, excelling in object removal and object spatial editing, and enabling more creative layer-wise editing.

2. Related Works

2.1. Image Editing

Image editing methods can be broadly categorized into two groups: multi-task editing and local editing. Multi-task editing allows for a wide array of image modifications based on high-level inputs, such as user instructions. For example, Emu-Edit [35] offers a flexible interface for diverse image edits. Other methods, including InstructPix2Pix [4], Prompt2Prompt [10], InstructAny2Pix [17], MagicBrush [47], Imagic [16], PhotoSwap [9], UltraEdit [51], HQ-Edit [13], MGIE [8], and OmniGen [43], focus on fine-grained, user-specific edits. These approaches achieve high precision by aligning closely with user instructions, often through fine-tuned models. However, due to the absence of region references, these methods struggle with accurately locating objects and preserving the integrity of unrelated regions.

Local image editing focuses on tasks like inpainting and object insertion, typically guided by masks or reference images. Early GAN-based methods, such as CMGAN [53], LAMA [37], CoModGAN [52], ProFill [45], CRFill [46], and DeepFillv2 [44], use latent space manipulation for inpainting missing regions. Leveraging the success of text-to-image diffusion models, Repaint [22], SDEdit [24], ControlNet [49], BrushNet [15], and Blended Diffusion [2] combine text guidance with masks or references for customized image inpainting. Further, ObjectStitch [36], HD-Painter [23], and PowerPaint [54] extend this to versatile object editing with text prompts. Beyond local object editing, recent research also focuses on spatial editing techniques that allow more interactive control over object positioning and transformations. Methods like MagicFixup [1], DiffusionHandle [27], DragGAN [26], DiffEditor [25], De-

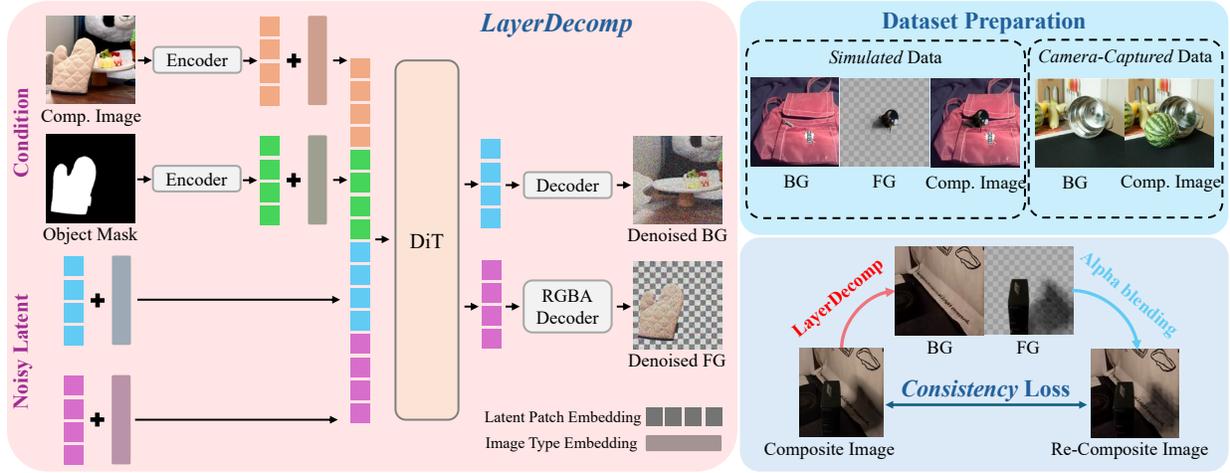


Figure 2. **The framework of LAYERDECOMP.** The model takes four inputs: two conditional inputs, including a composite image and an object mask, and two noisy latent representations of the background and foreground layers. During training, we use simulated image triplets alongside camera-captured background-composite image pairs. We also introduce a pixel-space consistency loss to ensure that natural visual effects such as shadows and reflections are faithfully preserved in the transparent foreground layer.

signEdit [14], and DragAnything [42] highlight a growing emphasis on user-driven, spatially aware editing. However, preserving object appearance and background integrity during spatial edits (e.g., moving and resizing) remains challenging, particularly when complex visual effects like shadows and reflections are involved.

2.2. Image Layer representation

Obtaining high-quality image layer representations is crucial for implementing accurate and diverse editing objectives. This process typically involves image decomposition, layer extraction, and image matting. PACO [21] provides a fine-grained dataset with mask annotations for parts and attributes of common objects. However, these object representations indicate only the regions of parts and objects, lacking transparent layers for flexible editing. MAGICK [6] offers a large-scale matting dataset generated by diffusion models, while MULAN [38] creates RGBA layers from COCO [18] and Laion Aesthetics 6.5 [34] using off-the-shelf detection, segmentation, and inpainting models; however, it does not capture visual effects, limiting its applicability for direct editing. LayerDiffusion [48] and Alfie [31] generate transparent image layers from text prompts, facilitating layer blending. However, text-driven generation limits object identity control in image-specific editing tasks. Once the image decomposition is performed, users typically need to execute image composition to achieve a cohesive final image. Existing methods like ObjectDrop [40] require model fine-tuning to restore visual effects, which can disrupt the original image’s appearance. In contrast, LAYERDECOMP models these effects during training, inher-

ently preserving them to produce a seamless, harmonious composite image without extra fine-tuning.

3. Approach

3.1. Overview of LAYERDECOMP Framework

As shown in Fig 2, the LAYERDECOMP framework builds upon Diffusion Transformers (DiTs) [29] to denoise multi-layer image outputs in the latent space encoded by the VAE encoders $g_\phi^{\text{RGB}}(\cdot)$ and $g_\psi^{\text{RGBA}}(\cdot)$. Specifically, the DiT model $f_\theta(\cdot)$ takes two types of conditional input $\mathbf{c} = (\mathbf{y}_{\text{comp}}, \mathbf{y}_{\text{obj}})$, which are the latents of the original composite image, and the decomposition object mask, respectively, *i.e.*, $\mathbf{c} = (\mathbf{y}_{\text{comp}}, \mathbf{y}_{\text{obj}}) = (g_\phi^{\text{RGB}}(\mathbf{I}_{\text{comp}}^{\text{RGB}}), g_\phi^{\text{RGB}}(\mathbf{M}_{\text{obj}}^{\text{RGB}}))$. By taking the conditional image embeddings, the model targets to denoise the noisy latent $\mathbf{x}_t = (\mathbf{x}_t^{\text{bg}}, \mathbf{x}_t^{\text{fg}})$, to recover the latents of the clean background image and the transparent foreground layer $\mathbf{x}_0 = (\mathbf{x}_0^{\text{bg}}, \mathbf{x}_0^{\text{fg}}) = (g_\phi^{\text{RGB}}(\mathbf{I}_{\text{bg}}^{\text{RGB}}), g_\psi^{\text{RGBA}}(\mathbf{I}_{\text{fg}}^{\text{RGBA}}))$. The training loss follows the standard denoising diffusion loss [12, 19]:

$$\mathcal{L}_{\text{dm}} = \mathbb{E}_{t \sim \mathcal{U}(\{1, \dots, T\}), \epsilon, \mathbf{x}_t} [\|\epsilon_\theta(\mathbf{x}_t; \mathbf{c}, t) - \epsilon\|_2^2], \quad (1)$$

s.t. $\mathbf{x}_0 \sim q_{\text{data}}(\mathbf{x}_0)$, $\epsilon \sim \mathcal{N}(\mathbf{0}, I)$, $\mathbf{x}_t = \sqrt{\alpha_t} \mathbf{x}_0 + \sqrt{1 - \alpha_t} \epsilon$.

Specifically, the noisy input \mathbf{x}_t and image conditions \mathbf{c} are initially divided into non-overlapping patches and converted into patch embeddings. The patch embeddings of each type of images, such as background, foreground or any conditions, are added with a corresponding *type embedding* and then concatenated into a sequence. Subsequently, the model follows the standard DiT architecture, where the patch embeddings are processed through multiple

transformer blocks. With such design, the image conditions provide comprehensive contextual information through the self-attention in the transformer blocks to enhance the denoising, and the loss is only computed on the positions corresponding to the noisy latents.

Note that the latent of the foreground image needs to be encoded with RGBA channels, we leverage an RGBA-VAE fine-tuned from the original VAE by following LayerDiffusion [48] which makes minimal disturbance to the original latent space. Moreover, as the foreground image is not always available, *e.g.*, in the case of real-world camera-captured data, the noisy input and output corresponding to the foreground are masked out from \mathcal{L}_{dm} computation if $\mathbf{I}_{\text{fg}}^{\text{RGBA}}$ is absent in the training stage.

3.2. Consistency Loss for Visual Effects Learning

To handle cases where real-world data lacks ground-truth annotations, we introduce a consistency loss that enables the learning of natural visual effects in the transparent foreground layer without explicit annotation. Intuitively, as shown in Fig 2, the consistency loss is applied in the decoded pixel space to encourage the predicted foreground can faithfully reconstruct the composite input after blending with the background layers.

More specifically, given a composite image $\mathbf{I}_{\text{comp}}^{\text{RGB}}$, at any denoising timestep t , we reparameterize our model prediction back to the estimation of the clean latent \mathbf{x}_0 as:

$$\hat{\mathbf{x}}_0(\mathbf{x}_t; \mathbf{c}, t) = \frac{1}{\sqrt{\alpha_t}} (\mathbf{x}_t - \sqrt{1 - \alpha_t} \cdot \epsilon_\theta(\mathbf{x}_t; \mathbf{c}, t)). \quad (2)$$

Given Eq. 2, we compute the estimated $\hat{\mathbf{x}}_0(\mathbf{x}_t; \mathbf{c}, t) = (\hat{\mathbf{x}}_0^{\text{bg}}, \hat{\mathbf{x}}_0^{\text{fg}})$ of background and foreground at time step t and decode them into pixel space to get $\hat{\mathbf{I}}_{\text{bg}}^{\text{RGB}} = h_{\phi'}^{\text{RGB}}(\hat{\mathbf{x}}_0^{\text{bg}})$ and $\hat{\mathbf{I}}_{\text{fg}}^{\text{RGBA}} = h_{\psi'}^{\text{RGBA}}(\hat{\mathbf{x}}_0^{\text{fg}})$, via the decoder of RGB-VAE and RGBA-VAE, respectively. The results are combined through alpha blending to produce the estimated composite image $\hat{\mathbf{I}}_{\text{comp}}^{\text{RGB}} = \mathcal{A}(\hat{\mathbf{I}}_{\text{bg}}^{\text{RGB}}, \hat{\mathbf{I}}_{\text{fg}}^{\text{RGBA}})$. The consistency loss is thus:

$$\mathcal{L}_{\text{consist}} = \mathbb{E}_t \sum_{i=1}^H \sum_{j=1}^W \left| \mathbf{I}_{\text{comp}}^{\text{RGB}}(i, j) - \hat{\mathbf{I}}_{\text{comp}}^{\text{RGB}}(i, j) \right|, \quad (3)$$

where H and W indicates the height and the width of the composite image, respectively.

The consistency loss enables LAYERDECOMP to learn faithful representations of transparent visual effects in the foreground layer, which is essential for accurately decomposing natural shadows and reflections in real-world data, especially in the absence of ground-truth annotations.

3.3. Dataset Preparation

To effectively train LAYERDECOMP, we curated a hybrid dataset that combines simulated and real-world data. Ideally, training LAYERDECOMP requires image triplets: an

input image $\mathbf{I}_{\text{comp}}^{\text{RGB}}$, a transparent foreground layer containing the target object and its visual effects $\mathbf{I}_{\text{fg}}^{\text{RGBA}}$, and a background image without the foreground object $\mathbf{I}_{\text{bg}}^{\text{RGB}}$. While collecting natural triplet images with specialized devices or through manual annotation might be feasible, it is costly and impractical for large-scale data needs. Conversely, synthesizing such triplet data directly with generative models presents significant challenges. Observations from existing approaches, such as HQ-Edit [13] and LayerDiffusion [48], indicate that generative models often inadvertently modify areas outside the target foreground, making it difficult to produce truly aligned image layers with consistent content. Additionally, accurately representing transparent visual effects, such as shadows and reflections in the foreground layer with an alpha channel for transparency, remains unexplored in existing works. To address these limitations, we developed a simulated data pipeline to create triplet images and supplemented it with a smaller portion of real-world $\mathbf{I}_{\text{comp}}^{\text{RGB}}$ and $\mathbf{I}_{\text{bg}}^{\text{RGB}}$ pairs to enhance robustness.

Simulated Data: To create image triplets, we first collected a large-scale object assets consisting of unoccluded foreground objects with synthesized shadows. We used entity segmentation [30] to select “thing” objects from natural images, and applied depth estimation to infer occlusion relations to exclude incomplete objects. We then applied a shadow synthesis method [39] to generate a shadow intensity map for each object on a white background. By integrating the intensity map into the alpha channel, we obtained comprehensive object assets in RGBA format. During training, we adjusted the scale and position of each foreground asset to align with the properties of a randomly selected background image $\mathbf{I}_{\text{bg}}^{\text{RGB}}$, resulting in a finalized foreground layer $\mathbf{I}_{\text{fg}}^{\text{RGBA}}$. By blending the two layers together, we obtained a composite image $\mathbf{I}_{\text{comp}}^{\text{RGB}}$, completing the triplet data needed for model training. Although the composite results may lack fully realistic geometry and harmonized content, this approach enables large-scale training and allows the model to learn the appropriate representations for the two output layers in the decomposition task.

Camera-Captured Data: We also include a small set of real-world camera-captured image pairs, denoted as $\mathbf{I}_{\text{com}}^{\text{RGB}}$ and $\mathbf{I}_{\text{bg}}^{\text{RGB}}$, similar to the *counterfactual* dataset proposed by ObjectDrop [40]. The real-world data enhances the model’s ability to generalize to natural images containing authentic shadows and a broader range of visual effects, such as reflections, which are crucial for accurate foreground-background decomposition in real-world tasks.

4. Experiments

Implementation. LAYERDECOMP is finetuned from a 5B-parameter DiT model pre-trained for text-to-image genera-



Figure 3. **Object removal - comparison with mask-based methods.** Our model, using tight input masks, generates more visually plausible results with fewer artifacts compared to ControlNet Inpainting [49], SD-XL Inpainting [32], and PowerPaint [54], which all require loose mask input. Besides, our model delivers coherent foreground layers and supports more advanced downstream editing tasks.

tion, with the text encoder removed for the layer decomposition task. RGBA-VAE is finetuned from the DiT VAE using L1, GAN, and perceptual losses [50]. Following LayerDiffusion [48], we extend the input/output from 3 to 4 channels and introduce an extra latent code to adapt the original VAE’s latent space. It takes images with 512×512 resolution and encode them into 64×64 latent feature maps. Image type embedding is learnable through linear layers. For the simulated dataset, we use 100k foreground object images with precomputed shadows on plain backgrounds, selecting unoccluded main objects using segmentation and depth heuristics. For backgrounds, we source 5M stock images and blend the foregrounds based on object size and placement. Our camera-captured dataset consists of 6,000 image pairs. During training, we use the Adam optimizer and set the learning rate at $1e-5$. Training is conducted with a total batch size of 128 on 16 A100 GPUs for 80,000 iterations. During inference, all results are generated using DDIM sampling with 50 steps.

4.1. Ablations

To quantitatively assess the advantages of incorporating visual effects in the foreground layer and the effectiveness of our proposed consistency loss, we construct a held-out evaluation dataset of 635 images for ablation studies. This dataset includes camera-captured composite images with the corresponding backgrounds. Visual examples will be provided in the supplementary materials. The quality of

Table 1. **Ablation study of LAYERDECOMP on held-out test set.** “BG” denotes the decomposed background, and “Comp” represents the re-composited image created by our two-layer results.

Model	PSNR \uparrow		LPIPS \downarrow		FID \downarrow		CLIP-FID \downarrow	
	BG	Comp	BG	Comp	BG	Comp	BG	Comp
V_0 :RGB-only	28.21	-	0.0732	-	21.00	-	4.551	-
V_1 : V_0 +RGBA FG (obj.)	28.28	27.53	0.0708	0.0649	18.48	18.83	2.487	2.329
V_2 : V_0 +RGBA FG (obj.+v.e.)	28.56	28.66	0.0691	0.0612	17.99	16.87	2.539	2.172
Ours: $V_2 + \mathcal{L}_{\text{consist}}$	29.27	30.53	0.0618	0.0494	16.04	12.75	1.813	1.564

the decomposed background layers can be directly evaluated using standard metrics, such as PSNR, LPIPS [50], FID [11, 28], and CLIP-FID [28]. To further evaluate the quality of the decomposed foreground layers, we apply alpha blending to re-composite the background and foreground together, comparing the result with the original composite image. As shown in Tab. 1, compared to a naïve DiT baseline which outputs only an RGB background, adding an RGBA foreground layer not only enables decomposition but also improves background quality. Incorporating visual effects in the foreground layer and introducing a consistency loss further enhance model performance. This demonstrates the decomposition task may implicitly improve the model’s understanding of the input scene, leading to superior results in both layers’ predictions.

Table 2. **Comparison of LAYERDECOMP with mask-based object removal methods.** Loose (L) and Tight (T) mask-based results are shown where applicable. PSNR^m and SSIM^m are computed on the shadow mask region to assess the model’s shadow removal ability.

Model	RORD [33]				MULAN [38]				DESObAv2 [20]	
	PSNR \uparrow	LPIPS \downarrow	FID \downarrow	CLIP-FID \downarrow	PSNR \uparrow	LPIPS \downarrow	FID \downarrow	CLIP-FID \downarrow	PSNR ^m \uparrow	SSIM ^m \uparrow
CNI [49]	20.45 ^L 22.01 ^T	0.235 ^L 0.182 ^T	50.40 ^L 53.71 ^T	8.853 ^L 9.262 ^T	17.79	0.321	65.03	9.396	36.94 ^L	0.491 ^L
SDI [32]	19.88 ^L 20.81 ^T	0.205 ^L 0.166 ^T	53.73 ^L 56.28 ^T	11.38 ^L 11.10 ^T	16.04	0.303	65.74	11.54	34.21 ^L	0.527 ^L
PP [54]	20.88 ^L 21.26 ^T	0.231 ^L 0.201 ^T	39.48 ^L 56.56 ^T	8.596 ^L 11.32 ^T	17.17	0.314	55.80	9.988	29.33 ^L	0.369 ^L
Ours	24.56^L24.79^T	0.133^L0.132^T	21.77^L21.73^T	5.735^L5.778^T	19.13	0.244	39.26	6.332	38.57^T	0.640^T

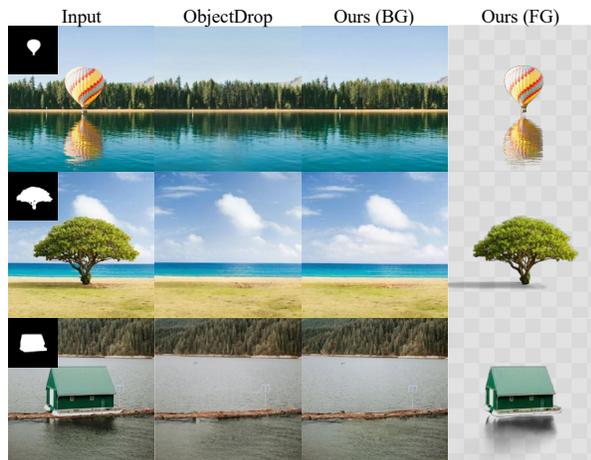


Figure 4. **Object removal - comparison with ObjectDrop [21].** Based on their released examples, our model demonstrates comparable quality in photorealistic object removal in the background layer, while decomposing the foreground with intact visual effects.

4.2. Comparison on Object Removal

To assess the quality of the background layers predicted by LAYERDECOMP, we evaluate the model on the object removal task, comparing it to several state-of-the-art approaches, including mask-based methods (ControlNet Inpainting [49], SD-XL Inpainting [32], PowerPaint [54], ObjectDrop [40]) and instruction-driven models (Emu-Edit [35], MGIE [8], and OmniGen [43]).

Quantitative evaluation among mask-based inpainting methods is conducted on three public benchmarks: **RORD** [33], a real-world object removal dataset collected from video data with human-labeled loose and tight object masks; **MULAN** [38], a synthesized multi-layer dataset that provides instance-wise RGBA decompositions for COCO and LAION images; and **DESObAv2** [20], a real-world image dataset with shadow mask annotations and synthesized image pairs where instance shadows are removed. Standard metrics, including PSNR, LPIPS, FID, and CLIP-FID, are used on RORD and MULAN, while regional similarity such as masked PSNR and masked SSIM are used for shadow re-

Table 3. **User study for instruction-driven object removal.**

Methods	Removal	Result	Background	Overall
	Effectiveness	Plausibility	Integrity	
Emu-Edit [35]	5.00%	4.38%	3.33%	4.79%
Ours	57.08%	77.92%	76.25%	83.54%
OmniGen [43]	4.07%	3.15%	2.96%	3.89%
Ours	67.04%	80.56%	84.63%	87.78%

moval in DESObAv2. More details of dataset preparation will be included in the supplementary materials.

Most existing mask-based object removal approaches cannot automatically detect and remove visual effects associated with the target object, necessitating a loose mask input. Indeed, as shown in Tab. 2, using a loose mask introduces more inpainting area and thus hurts PSNR and LPIPS, but significantly improves result fidelity (i.e., FID and CLIP-FID) for all compared methods. In contrast, LAYERDECOMP demonstrates robustness to mask tightness and outperforms all methods by a large margin. For shadow removal on DESObAv2, LAYERDECOMP surpasses other methods even without utilizing the shadow annotation included in the loose mask. As shown in Fig. 3, LAYERDECOMP provides substantial improvements over existing methods, generating more photorealistic background layers with fewer artifacts and minimal visual effect residues. Comparing with ObjectDrop, a leading model in object removal that can automatically eliminate visual effect with a tight input mask, we applied LAYERDECOMP to the example images released by their work. As shown in Fig. 4, LAYERDECOMP achieves similar high-quality backgrounds with effective object removal. Additionally, LAYERDECOMP decomposes the foreground with realistic visual effects, enabling further editing capabilities not supported by ObjectDrop. More visual comparisons will be included in the supplementary materials.

To compare with instruction-driven methods, we conduct a user study on 60 randomly selected images from the **Emu-Edit Remove Set** [35]. We ask 17 independent researchers to compare results from LAYERDECOMP and an existing method, focusing on three quality aspects: *removal effectiveness*, *result plausibility*, and *background integrity*.

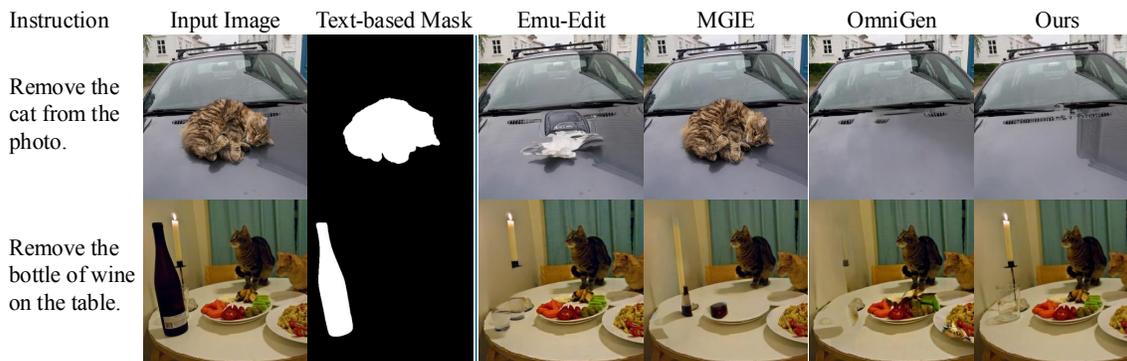


Figure 5. **Object removal - comparison with instruction-driven methods.** Combining with a text-based grounding method, our model can effectively remove target objects and preserve background integrity, while existing instruction-based editing methods, such as Emu-Edit [35], MGIE [8], and OmniGen [43], may struggle to fully remove the target or maintain background consistency.

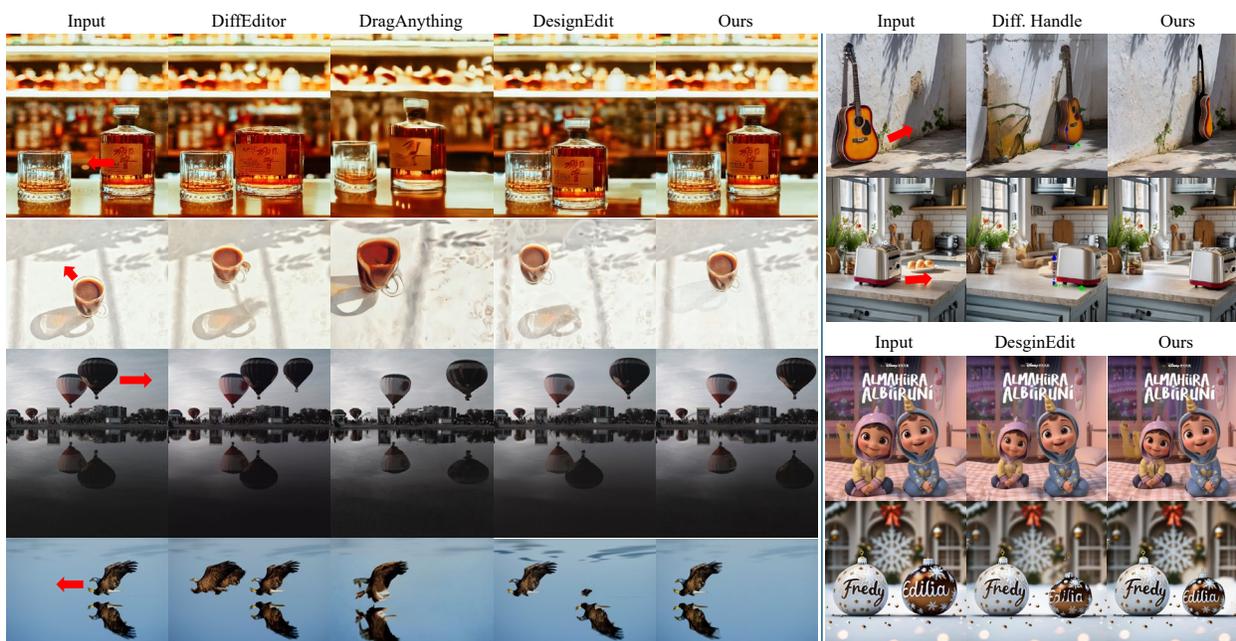


Figure 6. **Object spatial editing.** Our model enables precise object moving and resizing with seamless handling of visual effects, resulting in highly effective and realistic edits that preserve content identity. When applied to examples released by specific works, such as DiffusionHandle [27] and DesignEdit [14], our model also achieves satisfying results.

In cases where both methods achieve satisfactory results, users could mark it as a “tie”. For a fair comparison, we use a grounding model to generate text-based masks to input to LAYERDECOMP. In total, 3060 data points are collected in this study. As shown in Tab. 3, LAYERDECOMP is clearly preferred in at least 83% of the testing cases for overall quality. Example object removal results are visualized in Fig. 5. While Emu-Edit and MGIE struggle to fully remove the target object, OmniGen is more effective but does not reliably preserve the background integrity. Our model, in contrast, successfully removes the target object

while preserving most background fine details.

4.3. Comparison on Object Spatial Editing

We further compare LAYERDECOMP for object spatial editing tasks, including object moving and resizing, against several state-of-the-art methods: DiffEditor [25], DragAnything [41], DesignEdit [14], and Diffusion Handles [27]. For user study, we select 15 editing samples using web images for each task, and ask 23 independent researcher to compare results from LAYERDECOMP and an existing method, focusing on three quality aspects: *edit effective-*

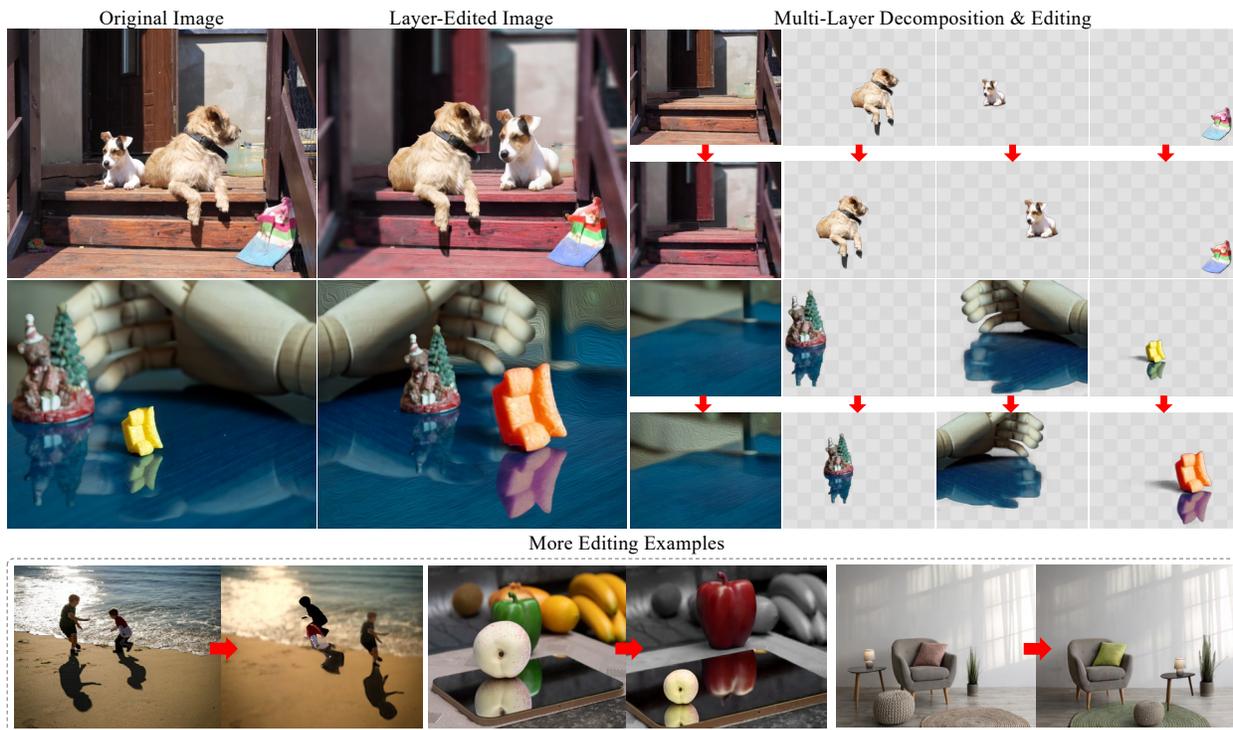


Figure 7. **Multi-layer Decomposition and Creative layer-editing.** By sequentially applying our model, we can decompose multiple foreground layers with distinct visual effects, which can then be used for further creative editing tasks.

Table 4. **User study for object spatial editing.**

	Methods	Edit Effectiveness	Result Plausibility	Content Integrity	Overall
Moving	DesignEdit [14]	1.67%	1.11%	1.11%	2.22%
	Ours	71.67%	90.56%	77.22%	94.44%
	DragAnything [41]	1.82%	1.21%	2.42%	1.21%
	Ours	67.88%	93.33%	95.15%	96.36%
Resizing	DesignEdit [14]	2.22%	3.89%	3.33%	3.89%
	Ours	69.44%	82.78%	71.67%	87.22%
	DiffEditor [25]	1.11%	1.11%	1.21%	1.11%
	Ours	95.15%	96.36%	92.21%	96.36%

ness, result plausibility, and content integrity. In this study, 2070 data points are collected. As shown in Tab. 4, LAYERDECOMP is at least 87% more preferred in the spatial editing tasks, featuring superior plausibility in editing results. As shown in Fig. 6, LAYERDECOMP enables seamless spatial editing for objects in various scenes. By preserving intact visual effects in the transparent foreground layer, shadows and reflections move naturally with the editing targets, allowing harmonious re-composition to be effortlessly achieved through alpha blending. When compared to DiffusionHandle [27] and DesignEdit [14] on their released examples, LAYERDECOMP demonstrates comparable results in most scenarios, including graphic design examples without requiring additional fine-tuning.

4.4. Multi-Layer Decomposition and Editing

LAYERDECOMP can be applied sequentially to an original input image with different instance masks, decomposing multiple layers along with their visual effects, as examples shown in Fig. 7. This process enables creative and complex layer-based editing for each individual layer, including spatial manipulation, recoloring, and filtering. Once editing is complete, the re-composition of all layers maintains a natural and realistic appearance, as demonstrated in Fig. 7.

5. Discussions

In conclusion, our model achieves superior image layer decomposition, outperforming existing methods in object removal and spatial editing across benchmarks and user studies. By decomposing images into a realistic background and transparent foreground with preserved visual effects, it enables diverse creative editing possibilities. Our consistency loss facilitates accurate foreground representation learning even without hard-to-obtain ground truth. While robust, the model may require refinement for challenging cases like incomplete masks, heavy occlusions, or intricate colorful shadows. Future directions include enhancing performance under such conditions and expanding support for a broader range of visual effects to enable more generalizable and versatile image decomposition.

References

- [1] Hadi Alzayer, Zhihao Xia, Xuaner Zhang, Eli Shechtman, Jia-Bin Huang, and Michael Gharbi. Magic fixup: Streamlining photo editing by watching dynamic videos. *arXiv preprint arXiv:2403.13044*, 2024. 2
- [2] Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18208–18218, 2022. 2
- [3] James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions. *Computer Science*. <https://cdn.openai.com/papers/dall-e-3.pdf>, 2(3):8, 2023. 1
- [4] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. *arXiv preprint arXiv:2211.09800*, 2022. 2
- [5] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18392–18402, 2023. 1
- [6] Ryan D Burgert, Brian L Price, Jason Kuen, Yijun Li, and Michael S Ryoo. Magick: A large-scale captioned dataset from matting generated images using chroma keying. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22595–22604, 2024. 3
- [7] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*, 2024. 1
- [8] Tsu-Jui Fu, Wenze Hu, Xianzhi Du, William Yang Wang, Yinfei Yang, and Zhe Gan. Guiding instruction-based image editing via multimodal large language models. In *International Conference on Learning Representations (ICLR)*, 2024. 2, 6, 7
- [9] Jing Gu, Yilin Wang, Nanxuan Zhao, Tsu-Jui Fu, Wei Xiong, Qing Liu, Zhifei Zhang, He Zhang, Jianming Zhang, Hyun-Joon Jung, et al. Photoswap: Personalized subject swapping in images. In *Advances in Neural Information Processing Systems*, 2024. 2
- [10] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022. 2
- [11] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 5
- [12] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 3
- [13] Mude Hui, Siwei Yang, Bingchen Zhao, Yichun Shi, Heng Wang, Peng Wang, Yuyin Zhou, and Cihang Xie. Hq-edit: A high-quality dataset for instruction-based image editing. *arXiv preprint arXiv:2404.09990*, 2024. 2, 4
- [14] Yueru Jia, Yuhui Yuan, Aosong Cheng, Chuke Wang, Ji Li, Huizhu Jia, and Shanghang Zhang. Designdedit: Multi-layered latent decomposition and fusion for unified & accurate image editing. *arXiv preprint arXiv:2403.14487*, 2024. 1, 3, 7, 8
- [15] 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. *arXiv preprint arXiv:2403.06976*, 2024. 2
- [16] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6007–6017, 2023. 2
- [17] Shufan Li, Harkanwar Singh, and Aditya Grover. Instructany2pix: Flexible visual editing via multimodal instruction following. *arXiv preprint arXiv:2312.06738*, 2023. 2
- [18] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 3
- [19] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022. 3
- [20] Qingyang Liu, Jianting Wang, and Li Niu. Desobav2: Towards large-scale real-world dataset for shadow generation. *arXiv preprint arXiv:2308.09972*, 2023. 6
- [21] Zhengzhe Liu, Qing Liu, Chirui Chang, Jianming Zhang, Daniil Pakhomov, Haitian Zheng, Zhe Lin, Daniel Cohen-Or, and Chi-Wing Fu. Object-level scene deocclusion. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024. 3, 6
- [22] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11461–11471, 2022. 2
- [23] Hayk Manukyan, Andranik Sargsyan, Barsegh Atanyan, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. Hd-painter: high-resolution and prompt-faithful text-guided image inpainting with diffusion models. *arXiv preprint arXiv:2312.14091*, 2023. 2
- [24] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021. 2
- [25] Chong Mou, Xintao Wang, Jiechong Song, Ying Shan, and Jian Zhang. Diffeditor: Boosting accuracy and flexibility on diffusion-based image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8488–8497, 2024. 2, 7, 8
- [26] Xingang Pan, Ayush Tewari, Thomas Leimkühler, Lingjie Liu, Abhimitra Meka, and Christian Theobalt. Drag your

- gan: Interactive point-based manipulation on the generative image manifold. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–11, 2023. 2
- [27] Karran Pandey, Paul Guerrero, Matheus Gadelha, Yannick Hold-Geoffroy, Karan Singh, and Niloy J Mitra. Diffusion handles enabling 3d edits for diffusion models by lifting activations to 3d. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7695–7704, 2024. 2, 7, 8
- [28] Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On aliased resizing and surprising subtleties in gan evaluation. In *CVPR*, 2022. 5
- [29] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023. 3
- [30] Lu Qi, Jason Kuen, Yi Wang, Jiuxiang Gu, Hengshuang Zhao, Philip Torr, Zhe Lin, and Jiaya Jia. Open world entity segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(7):8743–8756, 2022. 4
- [31] Fabio Quattrini, Vittorio Pippi, Silvia Cascianelli, and Rita Cucchiara. Alfie: Democratising rgba image generation with no \$\$\$ *arXiv preprint arXiv:2408.14826*, 2024. 3
- [32] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 1, 5, 6
- [33] Min-Cheol Sagong, Yoon-Jae Yeo, Seung-Won Jung, and Sung-Jea Ko. Rord: A real-world object removal dataset. In *BMVC*, page 542, 2022. 6
- [34] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022. 3
- [35] Shelly Sheynin, Adam Polyak, Uriel Singer, Yuval Kirstain, Amit Zohar, Oron Ashual, Devi Parikh, and Yaniv Taigman. Emu edit: Precise image editing via recognition and generation tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8871–8879, 2024. 1, 2, 6, 7
- [36] Yizhi Song, Zhifei Zhang, Zhe Lin, Scott Cohen, Brian Price, Jianming Zhang, Soo Ye Kim, and Daniel Aliaga. Object-stitch: Object compositing with diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18310–18319, 2023. 2
- [37] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2149–2159, 2022. 2
- [38] Petru-Daniel Tudosiu, Yongxin Yang, Shifeng Zhang, Fei Chen, Steven McDonagh, Gerasimos Lampouras, Ignacio Jacobacci, and Sarah Parisot. Mulan: A multi layer annotated dataset for controllable text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22413–22422, 2024. 2, 3, 6
- [39] Tianyu Wang, Jianming Zhang, Haitian Zheng, Zhihong Ding, Scott Cohen, Zhe Lin, Wei Xiong, Chi-Wing Fu, Luis Figuroa, and Soo Ye Kim. Metashadow: Object-centered shadow detection, removal, and synthesis. *arXiv preprint arXiv:2412.02635*, 2024. 4
- [40] Daniel Winter, Matan Cohen, Shlomi Fruchter, Yael Pritch, Alex Rav-Acha, and Yedid Hoshen. Objectdrop: Bootstrapping counterfactuals for photorealistic object removal and insertion. *arXiv preprint arXiv:2403.18818*, 2024. 3, 4, 6
- [41] Weijia Wu, Zhuang Li, Yuchao Gu, Rui Zhao, Yefei He, David Junhao Zhang, Mike Zheng Shou, Yan Li, Tingting Gao, and Di Zhang. Draganything: Motion control for anything using entity representation. In *European Conference on Computer Vision*, pages 331–348. Springer, 2024. 7, 8
- [42] Weijia Wu, Zhuang Li, Yuchao Gu, Rui Zhao, Yefei He, David Junhao Zhang, Mike Zheng Shou, Yan Li, Tingting Gao, and Di Zhang. Draganything: Motion control for anything using entity representation. In *European Conference on Computer Vision*, pages 331–348. Springer, 2025. 3
- [43] Shitao Xiao, Yueze Wang, Junjie Zhou, Huaying Yuan, Xingrun Xing, Ruiran Yan, Shuting Wang, Tiejun Huang, and Zheng Liu. Omnigen: Unified image generation. *arXiv preprint arXiv:2409.11340*, 2024. 2, 6, 7
- [44] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4471–4480, 2019. 2
- [45] Yu Zeng, Zhe Lin, Jimei Yang, Jianming Zhang, Eli Shechtman, and Huchuan Lu. High-resolution image inpainting with iterative confidence feedback and guided upsampling. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIX 16*, pages 1–17. Springer, 2020. 2
- [46] Yu Zeng, Zhe Lin, Huchuan Lu, and Vishal M Patel. Cr-fill: Generative image inpainting with auxiliary contextual reconstruction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 14164–14173, 2021. 2
- [47] Kai Zhang, Lingbo Mo, Wenhui Chen, Huan Sun, and Yu Su. Magicbrush: A manually annotated dataset for instruction-guided image editing. *Advances in Neural Information Processing Systems*, 36, 2024. 2
- [48] Lvmin Zhang and Maneesh Agrawala. Transparent image layer diffusion using latent transparency. *arXiv preprint arXiv:2402.17113*, 2024. 2, 3, 4, 5
- [49] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023. 2, 5, 6
- [50] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the*

- IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. [5](#)
- [51] Haozhe Zhao, Xiaojian Ma, Liang Chen, Shuzheng Si, Ru-
jie Wu, Kaikai An, Peiyu Yu, Minjia Zhang, Qing Li, and
Baobao Chang. Ultraedit: Instruction-based fine-grained im-
age editing at scale. *arXiv preprint arXiv:2407.05282*, 2024.
[1](#), [2](#)
- [52] Shengyu Zhao, Jonathan Cui, Yilun Sheng, Yue Dong, Xiao
Liang, Eric I Chang, and Yan Xu. Large scale image comple-
tion via co-modulated generative adversarial networks. In *In-
ternational Conference on Learning Representations (ICLR)*,
2021. [2](#)
- [53] Haitian Zheng, Zhe Lin, Jingwan Lu, Scott Cohen, Eli
Shechtman, Connelly Barnes, Jianming Zhang, Ning Xu,
Sohrab Amirghodsi, and Jiebo Luo. Image inpainting with
cascaded modulation gan and object-aware training. In *Eu-
ropean Conference on Computer Vision*, pages 277–296.
Springer, 2022. [2](#)
- [54] 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. *arXiv
preprint arXiv:2312.03594*, 2023. [2](#), [5](#), [6](#)