Zero-Shot Depth Aware Image Editing with Diffusion Models

Rishubh Parihar* Sachidanand VS* R. Venkatesh Babu
IISc Bangalore

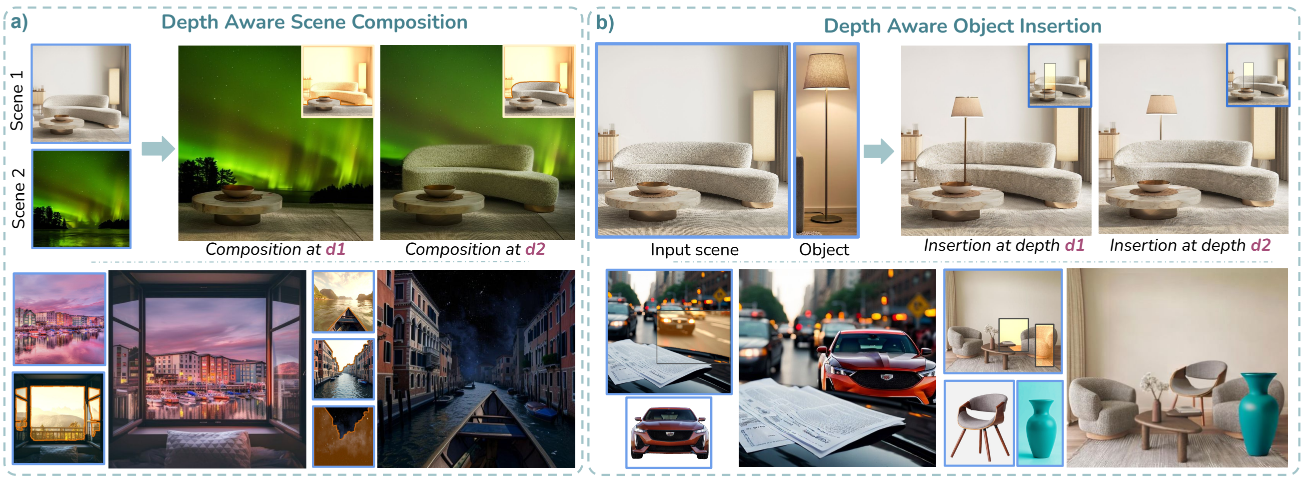

Figure 1. Our method performs precise *depth-aware* image editing at *user-specified depth* **d**. a) Given two input scenes and specified **d**, our method seamlessly composite the foreground (*depth* $<$ **d**) of one scene with the background of another. b) Given a background image, an object image and a 2D bounding box, our method can realistically place the object at depth **d**, with appropriate scene occlusions.

Abstract

*Diffusion models have transformed image editing but struggle with precise depth-aware control, such as placing objects at a specified depth. Layered representations offer fine-grained control by decomposing an image into separate editable layers. However, existing methods simplistically represent a scene via a set of background and transparent foreground layers while ignoring the scene geometry - limiting their effectiveness for depth-aware editing. We propose **D**epth-**G**uided **L**ayer **D**ecomposition - a layering method that decomposes an image into foreground and background layers based on a **user-specified depth value**, enabling precise depth-aware edits. We further propose **F**eature **G**uided **L**ayer **C**ompositing - a zero-shot approach for realistic layer compositing by leveraging generative priors from pretrained diffusion models. Specifically, we guide the internal U-Net features to progressively fuse individual layers into a composite latent at each denoising step. This preserves the structure of individual layers while generating realistic outputs with appropriate color and lighting adjustments without a need for post-hoc harmonization models. We demonstrate our method on two key depth-aware editing tasks: **1)** scene compositing by blending the foreground of one scene with the background of another at a specified depth, and; **2)** object insertion at a user-defined depth. Our zero-shot approach achieves precise depth ordering and high-quality edits, surpassing specialized scene compositing and object placement baselines, as validated across benchmarks and user studies.*

1. Introduction

Recent advancements in diffusion models [1, 2] have significantly improved image editing [3–10]. Though these approaches work well for coarse image modifications, such as altering object appearance, adding attributes or changing image style via text prompts, they lack the precise control over image content that artists and designers require. *Layered image representation* offers finer control by decomposing an image into editable layers, a widely used technique in visual content editing workflows. While recent works have explored layered generation with diffusion models [11–13], their use in layered image editing is largely unexplored.

Current layering approaches [11, 14, 15] decompose images into a background layer and multiple transparent

*equal contribution.

foreground layers each corresponding to a distinct object (Fig. 2). This enables precise editing of existing objects, such as removal, resizing, and translation within the image plane via editing the individual layers. However, this object-centric layering overlooks the spatial geometry of the scene, including the depth of individual objects and their arrangement in 3D space. As a result, they are incapable of performing *depth-aware* editing, such as composing foreground from a scene with background from another at a *specified depth* (Fig. 1a)). Moreover, when composing layers from two different scenes, these methods require additional image harmonization models [16, 17] to adjust lighting and color for photorealistic outputs.

In this work, we propose a novel zero-shot *depth-aware* editing framework that introduces **De**pth-**G**uided **La**yer **D**ecomposition (`DeGLaD`). Given an input image, its depth map (from an off-the-shelf predictor [18]), and a user-specified depth $\mathbf{d}$, `DeGLaD` decomposes the image into *foreground* (depth $< \mathbf{d}$) and *background* (depth $> \mathbf{d}$) layers (Fig. 2) based on the scene depth (Fig. 2). This decomposition enables precise depth-aware editing via independent editing of each layer. For example, to composite two scenes at specified depth $\mathbf{d}$, the background layer from one scene can be replaced with another. Similarly, a novel object can be inserted at depth $\mathbf{d}$ by inpainting the object in the background layer using off-the-shelf inpainting model [19] and composite with the unedited foreground layer, ensuring placement at intended depth. As providing a scalar depth value can be challenging for the user, we offer an intuitive top-view interface that allows users to specify $\mathbf{d}$ with a single click (Suppl.Sec.B).

Directly compositing edited layers in image space leads to unrealistic results, lacking proper lighting and color consistency. To address this, we integrate `DeGLaD` in the latent space of pretrained diffusion models, leveraging their rich generative priors for photorealistic compositing. First, we invert the input images in the diffusion latent space and then apply `DeGLaD` to obtain *latent depth layers*. For the seamless compositing of these layers, we propose **Feat**ure-**G**uided **La**yer **C**ompositing (`FeatGLaC`) - a training-free method that gradually composites the edited layers by guiding the diffusion features towards the target composition at each denoising step, similar to classifier guidance [20]. This progressive compositing approach preserves the structure of individual layers while ensuring realistic compositing with natural lighting and color consistency.

We evaluate our method on two novel *depth-aware* editing tasks: ***a)*** photorealistic compositing of scenes at a specified depth with appropriate relighting, and ***b)*** inserting a novel object at a precise depth. To benchmark these new tasks, we introduce a dataset featuring diverse objects and background scenes. Our *zero-shot* method outperforms specialized baselines trained for object insertion and

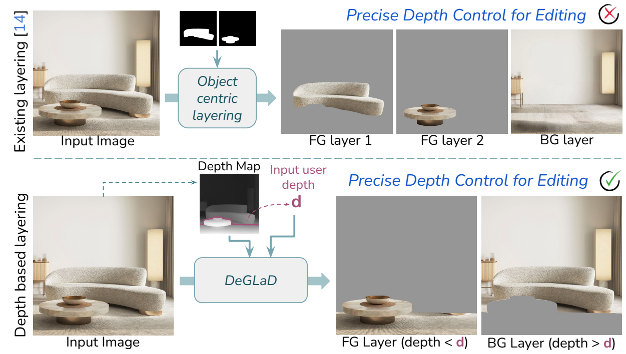

Figure 2. Existing layering method [14] decompose an image into background and foreground object layers but disregard spatial scene geometry, limiting their applicability for depth-aware editing. In contrast, our Depth-Guided Layer Decomposition (`DeGLaD`) decomposes a scene based on the scene depth and a user-specified depth $\mathbf{d}$, enabling precise depth-aware editing such as inserting objects at a precise depth.

scene compositing, demonstrating superior depth awareness and photorealistic compositing supported quantitatively and with user studies. In summary, our key contributions are:

- Depth-Guided Layer Decomposition - a novel layering method to decompose a given image into editable layers using its depth map and a user-specified depth value.
- Feature-Guided Layer Compositing — a zero-shot method that leverages pretrained diffusion models to progressively blend multiple layers with feature guidance, ensuring photorealistic lighting and color harmonization.
- Downstream applications of depth-based layering in novel *depth-aware* editing tasks - object insertion and scene compositing at a user-specified depth.
- Depth Edit Benchmark - A benchmark dataset consisting of diverse images with depth-aware editing annotations for evaluation of the two depth-aware editing tasks.

2. Related Works

Layered Image Generation. Recent methods perform large-scale diffusion model training on transparent layer dataset to perform layered generation [11, 12, 21], facilitating transparent content for editing workflows. Other methods [13, 14, 22, 23] decompose images into layered foreground and background representations or generate only a transparent foreground [24] for compositing-based edits. However, they lack depth-aware editing and are limited to individual object layers. PAIR-Diffusion [25] learns object-centric features to compose multiple images using scene segmentation maps, while [26] leverages a layered latent representation for object movement within a scene. Additionally, works on harmonization [17] and relighting [16, 27, 28] use layered representations for scene compositing but rely on large-scale paired datasets.

Image Editing with Diffusion Models. Text-to-Image models [1, 2, 29] are extensively used for image editing and controlled image synthesis [3, 4, 30, 31]. A set of existing methods manipulate the cross-attention maps [3, 6, 32] during inference to control the image layouts. These include swapping the attention maps [3, 32], or taking attention across the batch of images [33, 34]. Another set of works explores the text conditioning space of the T2I model to achieve more control [6, 35–37]. Others aim to find semantic direction in latent space or the text space [38–40] for editing. However, these approaches focus primarily on appearance-based edits and lack precise 3D control.

3D editing with Generative Models. While diffusion models excel at generating realistic images, they struggle with consistent 3D effects [41, 42]. To introduce 3D control, some methods condition diffusion models on scene normals or depth maps [1, 43], while others train on large-scale 3D-annotated datasets, using 3D bounding boxes [44] or geometric properties [45]. More recent approaches leverage generative priors from pretrained diffusion models for 3D-aware editing [31, 46–48]. Some lift 2D diffusion features to 3D space via depth maps for direct 3D editing [31, 46], while others edit the inferred mesh [49] or point cloud [47, 50] from a single image and refine the rendered image with pretrained diffusion models as post-processing. Another set of methods [51, 52], personalize the diffusion models on multi-view images to achieve 3D editing and view control for personalized object.

Object Insertion. Existing methods formulate object insertion from a single image as an object inpainting task. A widely used approach is to condition the diffusion models on object features extracted from an additional image encoder, enabling object insertion within a specified 2D box [19, 53–57]. Further, some approaches enhance realism of the inserted object by implicitly modeling lighting and shading via curating high-quality datasets [58]. However, these methods lack control over object placement at a specific depth and always generate complete objects without considering occlusions from the scene. In contrast, 3D-based methods enable realistic object insertion [59] and 3D-aware edits [60] but require multiple images to construct accurate 3D representations such as NeRFs. Another approach for object insertion estimates floor planes and scene lighting to place synthetic 3D assets [61], but obtaining realistic 3D assets from a single image remains challenging. Unlike these methods, our approach enables realistic, depth-aware object insertion using only a single object and background image while considering occlusions.

3. Method

3.1. Preliminaries

Diffusion models generate images by iteratively denoising a random noise sample. In the forward diffusion process, image x_0 is corrupted by sequentially adding standard Gaussian noise ϵ to obtain x_t. A denoiser network ϵ_θ is trained to estimate the added noise, conditioned on the timestep and optional conditioning such as text. For generating images, the reverse diffusion process denoises the random noise x_T, with multiple passes through denoising network ϵ_θ. To accelerate the diffusion models, Latent Diffusion Models [1] take a two-stage approach where the input image is first encoded into a lower dimensional latent space of a pretrained variational autoencoder, and the diffusion process is applied in the compressed latent space, reducing the computational demands.

Guidance. Diffusion models generate images by iteratively denoising a random noise sample. Classifier guidance [20] provides a mechanism to steer this iterative sampling process using a predefined energy function $\mathcal{G}$. This enables the ability to condition the generation during inference time without a need for model retraining. For example, to generate class-conditioned images, the energy function as the cross-entropy loss $\mathcal{L}$ between the pretrained classifier's prediction $f(x_t)$ and the given class y as $\mathcal{G} = \mathcal{L}(f(x_t, y))$. During generation, the predicted noise ϵ_θ is adjusted to minimize the classifier loss $\mathcal{L}$ by taking the loss gradient with respect to x_t, with λ as the classifier guidance weight as:

$$\tilde{\epsilon_\theta}(x_t, t) = \epsilon_\theta(x_t, t) + \lambda \nabla_{x_t} \mathcal{L}(f(x_t), y) \tag{1}$$

Several recent guidance approaches achieve inference time conditioning on sketch [62], layout [30, 63], features [31], optical flow [64] and human skeleton [65].

3.2. Depth-Guided Layer Decomposition (`DeGLaD`)

Scene depth serves as an effective representation to model the underlying scene geometry and enables enhanced control over 3D scene structure [31, 43]. Motivated by this, we propose *Depth-Guided Layer Decomposition* - a depth-based layering approach for precise depth-aware editing. The requirements for `DeGLaD` are an input image $\mathbf{x}$, its corresponding depth map $\mathbf{D}$ (can be obtained from off-the-shelf depth predictor [18]). Additionally, the user has to specify a scalar *depth value* $\mathbf{d}$, where the edit needs to be performed. Given these inputs, we decompose the image into *foreground* and *background* layers with corresponding binary masks $\mathbf{M}_{\text{fg}}$ and $\mathbf{M}_{\text{bg}}$, computed as follows:

$$\mathbf{M}_{\text{fg}}(\mathbf{i}, \mathbf{j}) = \mathbb{I}\big(\mathbf{D}(\mathbf{i}, \mathbf{j}) < \mathbf{d}\big), \quad \mathbf{M}_{\text{bg}}(\mathbf{i}, \mathbf{j}) = \mathbb{I}\big(\mathbf{D}(\mathbf{i}, \mathbf{j}) \geq \mathbf{d}\big), \tag{2}$$

where $\mathbf{i}$ and $\mathbf{j}$ denotes the pixel coordinates, and $\mathbb{I}(\cdot)$ is the indicator function. The obtained layers can be edited independently and recomposed for precise depth-aware editing. While we illustrate decomposition at a single depth, our method naturally extends to multiple depths by providing a set of depth values $\{\mathbf{d_1}, ..\mathbf{d_k}\}$, enabling multi-depth editing, such as composing multiple scenes or iteratively inserting objects (Fig. 1).

3.3. Feature-Guided Layer Composition (`FeatGLaC`)

Directly compositing the obtained layers in the image space leads to unnatural results, lacking color harmonization and proper lighting (Fig. 3). To address this, we integrate `DeGLaD` in the latent space of pretrained diffusion models and progressively compose

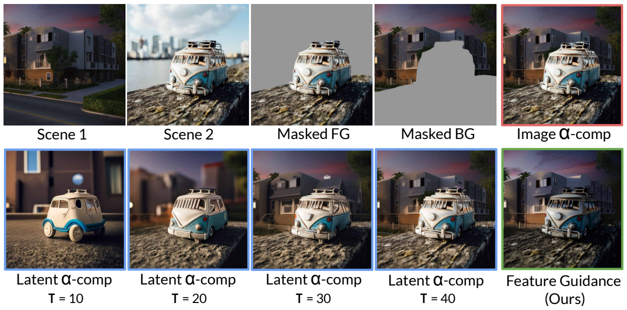

Figure 3. **Ablation for compositing layers: a)** Naively α compositing the layers in the image space (Image α-comp) results in unnatural 'cut-paste' artifacts without adjusting color and scene lighting. ***b)*** Performing α compositing in the diffusion latent space at $t = \tau$ (Eq.3) followed by denoising (Latent α-comp τ) has an inherent tradeoff. Compositing at an early stage of diffusion ($\tau = 40$) results in identity loss due to excessive denoising, and compositing at a later stage ($\tau = 10$) results in unnatural blending similar to Image α-comp. Our diffusion feature guidance-based layer compositing generates photorealistic composition while preserving the structure and identity of individual images.

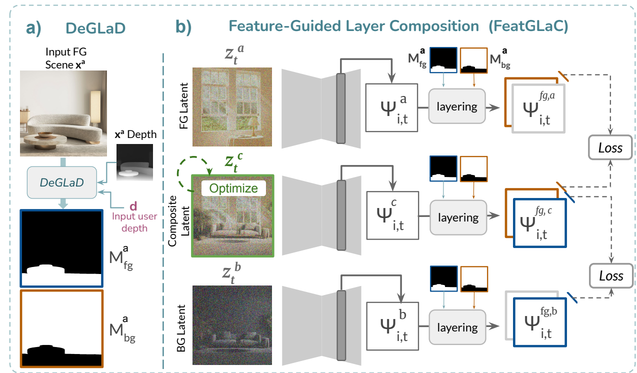

Figure 4. **Overall framework for scene compositing: a)** Given an input foreground image $\mathbf{x^a}$ and a user specified depth $\mathbf{d}$, we decompose the image with Depth-Guided Layer Decomposition (`DeGLaD`) and obtain foreground $\mathbf{M^a_{fg}}$ and background $\mathbf{M^a_{bg}}$ masks. **b)** We compose the foreground latent $\mathbf{z^a_t}$ with background latent $\mathbf{z^b_t}$ using the obtained mask with diffusion feature guidance. We guide the features of the composite latent $\Psi^c_{i,t}$ using the activations for foreground $\Psi^a_{i,t}$ and background features $\Psi^b_{i,t}$ and update the composite latent $\mathbf{z^c_t}$ for K iterations at each denoising step.

the layers during denoising to generate realistic outputs. We explain this approach using an example where the foreground layer from scene $\mathbf{x_a}$ (extracted by binary mask $\mathbf{M^a_{fg}}$) is composed with the background from scene $\mathbf{x_b}$. For composing foreground from $\mathbf{x_a}$ with background of $\mathbf{x_b}$, we define the background mask for $\mathbf{x_b}$ to be the same as that of $\mathbf{x_a}$, i.e., $\mathbf{M^b_{bg}} = \mathbf{M^a_{bg}}$. We start by inverting $\mathbf{x_a}$ and $\mathbf{x_b}$ with null-text inversion [66] to obtain the corresponding latents $\mathbf{z^a_{0:T}}$ and $\mathbf{z^b_{0:T}}$.

Baseline. One straightforward approach is to first α-composite the latents $\mathbf{z^a_\tau}$ and $\mathbf{z^b_\tau}$ at intermediate timestep τ to obtain a composite intermediate latent $\mathbf{z^c_\tau}$:

$$\mathbf{z^c_\tau} = \mathbf{M^a_{fg}} * \mathbf{z^a_\tau} + \mathbf{M^b_{bg}} * \mathbf{z^b_\tau} \tag{3}$$

where $\mathbf{M^a_{fg}}$ is downsampled to match the dimension of latent $\mathbf{z_t}$. The composed latent $\mathbf{z^c_\tau}$ is then denoised with the diffusion model for the remaining $\mathbf{T} - \tau$ timesteps for realistic blending of the two layers [5]. Though this framework seems promising, it has an inherent tradeoff between realistic blending with complex scene effects and preserving layer identity, as shown in Fig. 3. A large τ (close to clean image) does not provide enough freedom to recover the complex scene effects with denoising, and a small τ (close to noisy image) generates plausible composition but changes the scene contents significantly.

Composition with guidance. Rather than directly α-compositing the inverted latents, we introduce a more gradual fusion strategy that incorporates feature guidance at each denoising step. We call this approach ***Fea**ture-**G**uided **La**yer **C**omposition*, `FeatGLaC` in short. We start by initializing the composite latent $\mathbf{z^c_T}$ as the background latent $\mathbf{z^b_T}$ and iteratively denoise it with feature guidance, similar to classifier guidance [20]. Following prior works [31, 62], which demonstrate that the internal features of the denoising U-Net are highly expressive and enable fine-grained control over generation, we guide these features towards the target composition to achieve seamless blending. We denote the U-Net features as $\Psi_{i,t}$, where $\mathbf{i}$ is the diffusion model layer index and $\mathbf{t}$ is the diffusion timestep. At each timestep, we extract the features $\Psi^a_{i,t}$, $\Psi^b_{i,t}$ and $\Psi^c_{i,t}$ from $\mathbf{z^a_t}$, $\mathbf{z^b_t}$ and composite latent $\mathbf{z^c_t}$ respectively. Next, we define the diffusion guidance energy $\mathcal{G}$ for progressive composition of these layers.

Intuition: *We force the foreground layer (fg) of $\Psi^c_{i,t}$ to be close to foreground layer of $\Psi^a_{i,t}$ and the background layer (bg) of $\Psi^c_{i,t}$ to be close to the background layer of $\Psi^b_{i,t}$ as shown in Fig. 4. This is implemented by defining $\mathcal{G} =$*

$$\sum_i ||\mathbf{M^a_{fg}} * (\mathbf{\Psi^a_{i,t}} - \mathbf{\Psi^c_{i,t}})||^2 + ||\mathbf{M^b_{bg}} * (\mathbf{\Psi^b_{i,t}} - \mathbf{\Psi^c_{i,t}})||^2 \tag{4}$$

We compute the gradients of guidance energy $\mathcal{G}$ with respect to composite latent $\mathbf{z^c_t}$ and backpropagate them to update the next sample prediction as $\tilde{\mathbf{z}}^c_t = \mathbf{z^c_t} - \nabla_{\mathbf{z^c_t}}\mathcal{G}$ for $\mathbf{K}$ iterations at each denoising timestep. This gradual layer composition approach strikes a good tradeoff in preserving layer identity and generating photorealistic compositions. Fig. 1 & 3.

***Notably**, the proposed `FeatGLaC` framework is model-agnostic and can be integrated into any pretrained diffusion model. We demonstrate its flexibility by applying it to a depth-conditioned diffusion model for scene composition and an inpainting diffusion model for object insertion, leveraging their specialized editing capabilities for depth-aware tasks.*

3.4. Application in Depth-Aware Editing

We implement layer decomposition with `DeGLaD` and composition with `FeatGLaC` to perform depth-aware scene composition in Sec. 3.4.1 and object insertion in Sec. 3.4.2. A detailed algorithm of our method for both these tasks is provided in Supp.Sec.A. requires users to specify a depth value $\mathbf{d}$ for the scene where the edit needs to be performed, which may not be user-friendly. To address this, we alternatively provide an intuitive interface, as discussed in Supp.Sec.B that visualizes segmented scene from a top-down view, allowing users to easily specify $\mathbf{d}$ with a single user click.

3.4.1. Depth Aware Scene Composition

The goal of this task is to seamlessly compose the foreground of one scene, $\mathbf{x_a}$, with the background of another, $\mathbf{x_b}$, at the user-specified depth $\mathbf{d}$. We first decompose the images into depth-based layers using `DeGLaD` and compose the layers with `FeatGLaC` as discussed in Sec. 3.2 & 3.3 and Fig. 2. To ensure structure preservation of individual layers during composition, we incorporate a pretrained depth-conditioned diffusion model [1] for guidance. Specifically, we condition the model using an α-composited depth map, obtained by blending $\mathbf{D_a}$ (depth of $\mathbf{x_a}$) and $\mathbf{D_b}$ (depth of $\mathbf{x_b}$) with their respective foreground and background masks ($\mathbf{M^a_{fg}}$, $\mathbf{M^a_{bg}}$). This composite depth input allows the model to maintain the structure of individual layers during the composition.

3.4.2. Depth Aware Object Insertion

We introduce a novel task of realistically inserting a given object $\mathbf{x_0}$ in an input scene $\mathbf{x}$ at a user-specified depth $\mathbf{d}$ and inside a 2D bounding box $\mathbf{b}$. Existing approaches [19, 53, 54] can insert a given object in the specified 2D bounding box but do not provide explicit depth-aware control during object insertion. To this end, we *lift* an object insertion model $\mathcal{H}$ [19] and make it depth-aware. We first perform null-text inversion [66] on $\mathbf{x}$ to obtain latent $\mathbf{z_{0:T}}$ and store corresponding diffusion U-Net features $\Psi_{i,t}$ for guidance. Next, we obtain the foreground ($\mathbf{M_{fg}}$) and background ($\mathbf{M_{bg}} = 1 - \mathbf{M_{fg}}$) layers for the input scene $\mathbf{x}$ using `DeGLaD` at specified depth $\mathbf{d}$. The object insertion model $\mathcal{H}$ takes input scene $\mathbf{x}$, object image $\mathbf{x_o}$ and 2D bounding box $\mathbf{b}$ as input and iteratively denoises a edit latent $\mathbf{z^e_T}$ initialized from $\mathcal{N}(0, I)$ to inpaint the object in the bounding box. To perform depth-aware object insertion, we extract the features $\Psi^e_{i,t}$ of *edit latent* $\mathbf{z^e_t}$ from $\mathcal{H}$ at each timestep $\mathbf{t}$ and apply `FeatGLaC` to compose with only unedited foreground allowing the background layer to change.

Intuition. *We force the foreground (depth $< \mathbf{d}$) features $\Psi^e_{i,t}$ of the edit latent $\mathbf{z^e_t}$ to be close to the foreground features $\Psi^{i,t}$ of the inverted image latent $\mathbf{z_t}$ at each denoising step. This is implemented by defining the guidance energy $\mathcal{G}$* as:

$$\mathcal{G} = \sum_i ||\mathbf{M_{fg}} * (\mathbf{\Psi_{i,t}} - \mathbf{\Psi^e_{i,t}})||^2 \tag{5}$$

We use the above guidance loss to refine the intermediate edit latent $\mathbf{z^e_t}$ for $\mathbf{K}$ iterations at each denoising step. Empirically, replacing the foreground layer of $\mathbf{z^e_t}$ with the foreground layer of $\mathbf{z_t}$ at an intermediate timestep τ, followed by the guidance update for the remaining timesteps, yields more accurate object insertion results. We hypothesize that this improves $\mathcal{H}$'s ability to interpret object depth, especially scene occlusions, leading to more seamless compositions (Fig. 6).

4. Experiments

We perform extensive experiments to evaluate our method for depth-aware editing. In this section, we first discuss the implementation and dataset details, followed by experiments on scene composition, object insertion, and ablation studies. Additional experiment and dataset details are provided in the supp. document We strongly encourage reviewing the attached project page (*index.html*) in supplementary.zip for high-resolution visual results.

Figure 5. **Depth-aware object insertion.** Our method enables realistically placing multiple objects at precise user specified depths. Further our method can place the same object at multiple locations and depths in a given scene.

4.1. Implementation Details

For scene composition, we use the depth-conditioned Stable Diffusion v2-depth [1], and for object insertion, we use the Anydoor inpainting model [19]. The guidance is applied to features from the last and penultimate layers of the diffusion U-Net, which enhances edit plausibility, which is also shown in our ablations. For scene composition, guidance is applied from timesteps 0 to 38, updating the latent $\mathbf{z^c_t}$ for $K = 5$ iterations per step. For object insertion, guidance is given from timesteps 30 to 50, with the latent $\mathbf{z^e_t}$ updated for $K = 3$ iterations per step. Additionally, we use Depth Anything [18] to obtain the depth for object insertion and Zoedepth [67] for scene composition, as the depth-conditioned model SD v2-depth is pretrained with metric depth map. We extracted captions of input scenes using captioning model [68] to perform null-text inversion. Also, for scene composition, we compose the captions of two input scenes to condition the diffusion model during guided generation.

4.2. Dataset

Since we are the first to introduce the two depth-aware editing tasks, no public dataset exists for their evaluation. To address this, we curated the ***Depth Edit Benchmark*** - a dataset comprising two subsets tailored for the extensive evaluation of depth-aware scene composition and object insertion.

Object insertion. We gathered a collection of 490 scene-object image pairs from the web. Each pair includes annotations for the corresponding scene's depth map, depth value $\mathbf{d}$, where the object can be plausibly placed, and a corresponding 2D bounding box for object insertion. This dataset includes diverse objects from indoor and outdoor environments, with possible occlusion for inserted objects to effectively assess depth-aware object insertion.

Scene composition. We curated a dataset with $2,844$ image pairs with diverse foreground and background scenes, sourced from the SSHarmonization dataset [69] and the web. The dataset covers

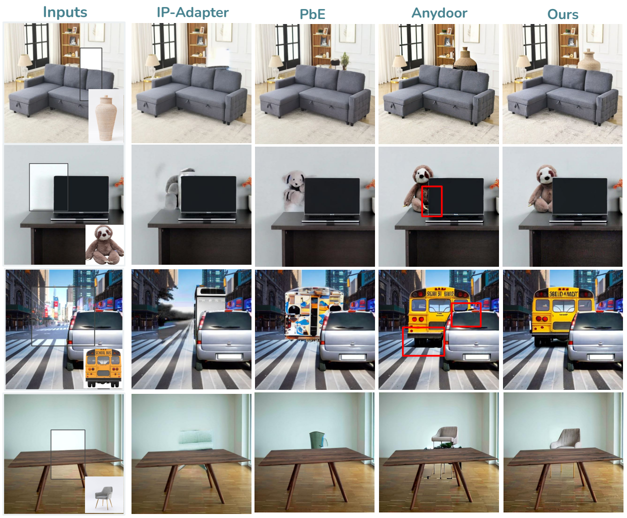

Figure 6. **Comparison for depth-aware object insertion:** IP-Adapter and PbE struggles to insert objects with consistent identity within an amodal bounding box. Anydoor achieves plausible placement but generates artifacts along the mask border (marked in red). Our method enables realistic object insertion while preserving both object identity and scene consistency.

a broad range of indoor and outdoor environments with varying lighting, composition, and appearance. Each image is annotated with depth maps (extracted from [67]) and the depth value $\mathbf{d}$ for each foreground scene for plausible scene composition. Additionally, we generate text prompts using an off-the-shelf image captioning model [68].

4.3. Object Insertion

To our knowledge, we are the first to perform depth-aware object insertion using only a single object and background image; hence, we compare with reference-based inpainting baselines. We use state-of-the-art reference conditioned inpainting methods IP-Adapter [53], Paint by example (PbE) [54], and Anydoor [19] to inpaint the given object in a scene in a specified bounding box. These methods take a bounding box as input and place the object without accounting for occlusions. For a fair comparison, we adapt them for depth-aware placement by using the foreground layer mask to occlude the bounding box with overlapping objects (Fig. 6), resulting in an amodal bounding box mask. This will preserve the foreground regions during inpainting and give us the illusion that the object is placed behind other objects.

Method	DINO-sim ↑	KID ↓	Δ depth ↓	Clip-sim ↑
IP-Adapter [53]	0.244	5.3	9.366	27.81
PbE [54]	0.273	4.9	6.733	60.12
Anydoor [19]	0.507	4.9	3.176	83.23
Ours	**0.545**	**4.8**	**2.989**	**84.86**

Table 1. **Depth-Aware object insertion comparison.** KID and Δ **depth** are reported in x10^2 units.

Metrics. We evaluate object insertion method for object identity, realism of the output, correctness of the inserted object, and depth consistency for the inserted object. We use DINO [70] feature similarity (DINO-sim) between the generated object in the bounding box and the reference object to measure identity preservation. To measure image realism, we compute KID [71] against COCO [72] as our evaluation set is relatively smaller to compute FID. To evaluate whether the object is actually placed, we use CLIP [73] similarity (CLIP-sim) between *'a photo of object-name'* and the cropped image from the generated image. If the object is correctly generated, the CLIP score should be higher. To assess depth consistency, we compute the discrepancy between the predicted object depth and the user-specified input depth. We estimate the depth of the generated image (using [18]), and compute the mean object depth of the object segment (obtained with SAM [74]). We report normalized Δ depth across the dataset, where lower values indicate more consistent depth-aware placement.

Analysis. We present the results for depth-aware object insertion in Fig. 6, and Tab. 1. The reference-conditioned inpainting models, such as the IP-adapter and Paint-by-example (PbE), struggle to generate accurate objects in the amodal mask as they have been trained to primarily inpaint unoccluded objects with 2D bounding boxes. This is quantified with a poor CLIP-sim metric in Tab. 1. Anydoor is able to generate consistent objects; however, it generates significant border artifacts (marked in red), resulting in an unnatural composition. Our approach generates realistic compositions with accurate object insertion (highest Clip-sim) and superior identity preservation (highest Dino-sim) as compared to all the object insertion baselines. Further, the object is naturally placed at an accurate depth, as evident with lower Δ **depth** scores. Note that the identity preservation of the object is limited by the base inpainting model and is not a limitation of our guidance method. However, we can improve the identity by doing a post-processing step; experiments are in the Supp.Sec.F1 document.

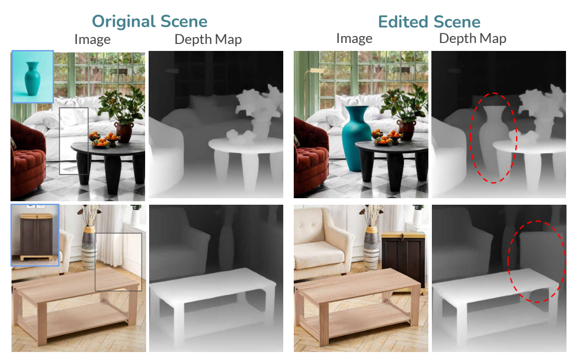

Figure 7. **Depth consistency in object insertion:** The depth map after object insertion appears visually consistent, confirming the object's placement accurately within the scene geometry.

Inserted objects are consistent with input depth. To analyze the depth consistency of the inserted object, we visualize the depth map from [75] after object insertion in Fig. 7. The visualization confirms the object is places at accurate scene depth between the foreground and background scene objects.

4.4. Scene Composition

We compare our method with the following baselines: ***a)*** ***`DeGLaD` Image*** - We perform `DeGLaD` in the image space and compose the edited *image layers* with α compositing. Further, we perform image harmonization using [17] as post-processing for realistic blending. ***b)*** ***`DeGLaD` + SDEdit*** [5] - We perform SDEdit (noise

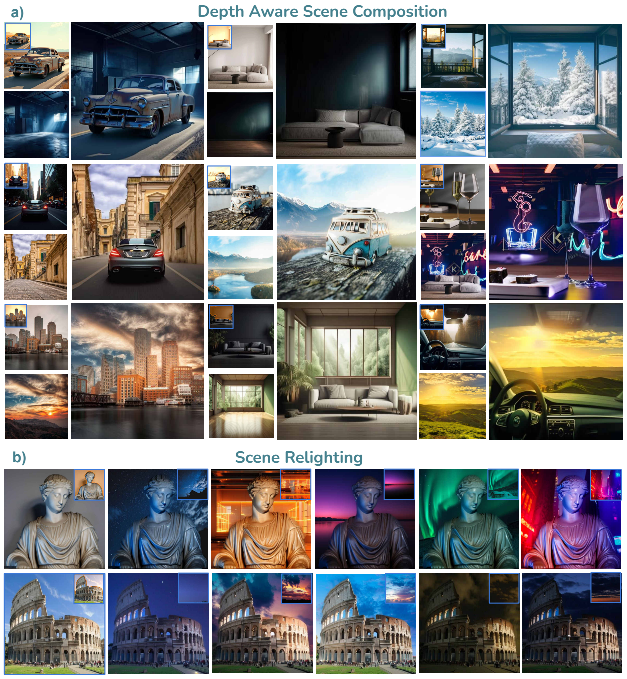

Figure 8. **Depth-aware scene compositing. a)** Our method seamlessly blends input scenes at a specified depth with accurate color and lighting adjustments. **b)** Additionally, our method enables scene relighting by compositing a foreground object with a plain background featuring strong lighting effects.

and denoise with diffusion model) on the output of `DeGLaD` composition in the image space for generating realistic compositions. ***c) PAIR Diffusion [25]*** allows for localized control during generation by copying content from a masked reference image. We use the layer masks ($\mathbf{M_{fg}}$ and $\mathbf{M_{bg}}$) to segment out the foreground and background regions and then pass desired reference scene images to PAIR Diffusion for generating the composed scene.

Metrics. We measure the visual quality of the composed scene and identity (structure and appearance) preservation of the foreground and background. We report FID with the COCO dataset to quantify the realism of the generated image. To evaluate identity preservation, we report the average LPIPS distance between the background and the foreground region of the composite image with the input images. For realistic scene composition, both LPIPS and FID should be low, indicating superior identity preservation and realism.

Method	LPIPS ↓	FID ↓
Pair-Diffusion	0.45	140.54
`DeGLAD` Image	0.036	132.6
`DeGLaD` + SDEdit	0.395	106.24
`DeGLaD` Diff	0.263	123.32

Table 2. Scene compositing comparison

Analysis. We present our results and comparisons in Fig. 9 and Tab. 2. `DeGLaD` Image achieves harmonization of the foreground to improve blending; however, it still struggles with *cut-pasting* appearance (e.g., bed scene) when the layered mask is not perfect, leading to unrealistic generation (inferior FID score). `DeGLaD` + SDEdit and PAIR-diffusion change the scene structure while generating consistent images in some examples. Our method generates photorealistic depth-aware scene composition with accurate scene illumination while preserving the scene structure. Notably, our method is robust to minor errors in the layered mask, as shown in Fig. 9 bed example. Additionally, our method can realistically relight the scene by providing different sky backgrounds.

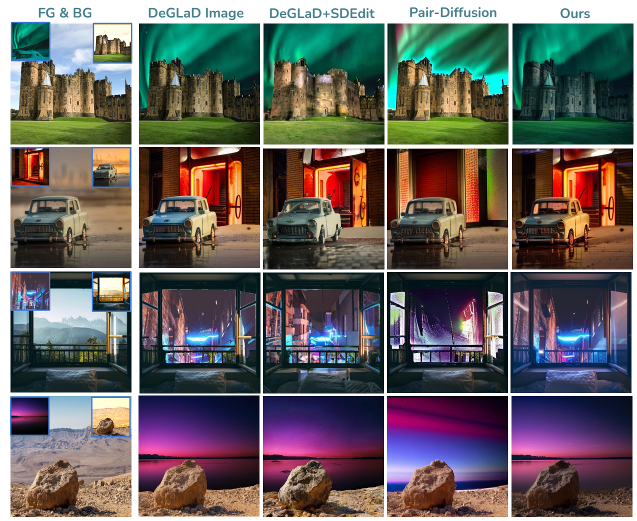

Figure 9. **Comparison for scene compositing.** `DeGLaD` Image result in cut-pasting artifacts leading to unnatural outputs. `DeGLaD`+SDEdit and Pair Diffusion generate unnatural compositions and distort the identity in some cases. Our metho realistically composite the two scenes in a depth-aware manner with consistent intra-scene illumination.

Composed scene follows accurate depth ordering. To analyze the depth consistency, we visualize the histogram of the input and the output scene in Fig. 10, which shows our method preserves the distribution of depth present in the foreground and background scene even during composition.

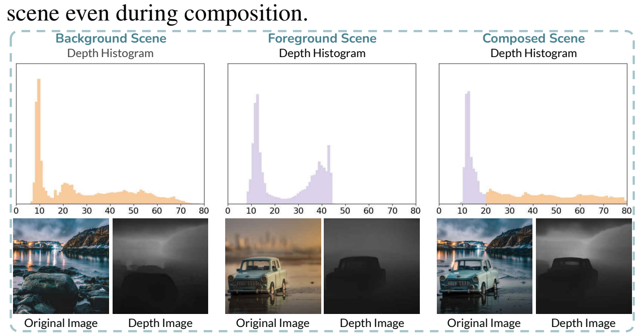

Figure 10. **Depth consistency in scene editing:** The initial depth regions in the composite image align with the foreground depth maps, while the later regions correspond to the background depth maps, indicating the preserved depth distribution.

4.5. User study

Due to the unavailability of well-established benchmarks for the task, we perform a user study to evaluate our approach across multiple aspects. We perform a user study to evaluate our method for depth-aware scene editing. We evaluate object insertion for the realism of the *placement*, *identity preservation*, and *depth consistency*. For the task of scene composition, we evaluate for the *realism* of the composition and *depth consistency*. The study was performed on 15 source images for each task, and 40 volunteers participated with varied expertise in image editing. We created 60 image pairs for object insertion and 40 pairs for scene composition, with each pair consisting of our generated output and a randomly sampled baseline. We divide this dataset into groups of 20 image pairs for separate analysis on each editing goal. Each user compared 20 pairs for each of the goals for the two tasks. The order of image pairs and the methods within each pair were randomized. The results of the user study are present in Fig. 11

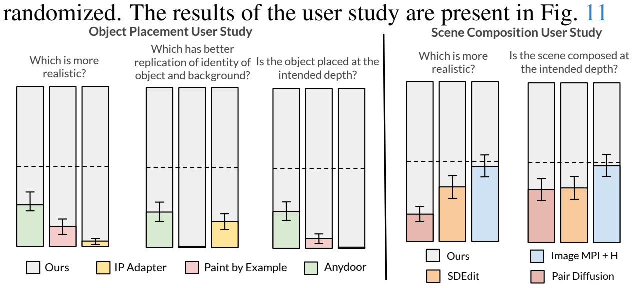

Figure 11. **User study** compiled from 40 responses. Our object insertion method is better than the baselines in *realism*, *identity preservation* and *depth consistency*. Similarly, for scene compositing, our approach surpasses the baselines in both *realism* and *depth consistency*, with image DeGLaD Image yielding comparable results but it suffers from image cut-pasting artifacts (Fig. 9).

Object insertion. Our method significantly outperforms all baselines in terms of realism, identity preservation, and depth consistency. PbE and IP-adapter perform poorly across all three goals, indicating the challenge of depth-aware placement task. Our approach excels in depth consistency metrics, indicating that our method effectively performs depth-aware editing while producing highly realistic images. This can also be seen while visualising the generated image depth map Fig. 11 where the object depth map is consistent with the surrounding.

Scene composition. As indicated in the user study, DeGLaD Image performs comparably to our approach for both goals. However, the harmonization model used in DeGLaD baseline, is specifically trained on a large-scale dataset for the task of blending objects in the background scene whereas our method is zero-shot. Further, applieng DeGLaD in the image space suffers with cut-paste artifacts as shown in Fig. 9. As compared to all the other scene compositing baselines, our approach achieves significantly better performance.

4.6. Ablations

We ablate over the design choices for scene compositing in Fig. 12. We follow the same guidance parameters for the object insertion task as well. Additional quantitative ablations are provided in the Supp.Sec.C - Tab.1 & 2.

Guidance Timesteps. We ablate over the timestep range from $0-50$ for applying the FeatGLaC guidance. Guiding only for small timesteps $(0-20)$ results in significant structure changes for the foreground and background scenes. On the contrary, providing guidance for all the timesteps preserves the structure but leads to unnatural composition (lighting mismatch). We found that guiding until an intermediate range of timesteps (0-38) and allowing the image to denoise freely for the remaining steps strikes a good balance, resulting in realistic compositions.

Guidance Layers. We ablate over the U-Net decoder features used to calculate FeatGLaC guidance loss, and using all the decoder layers for guidance results in significant artifacts. We observe that guidance with the first decoder layers can significantly hurt the generation. Finally, we achieve a combination of layer 3 (weight 8.5) and layer 2 (weight 0.2) works well in most cases. Using only one of these layers resulted in subpar compositions.

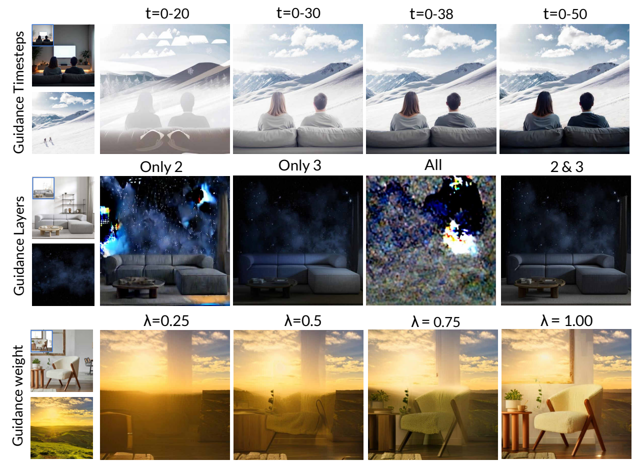

Figure 12. Ablation for scene compositing guidance parameters

Guidance weight. After finalizing the layers to be used for FeatGLaC guidance, we tried different weights for the guidance factor. Specifically, we ablate over a guidance multiplier λ for foreground guidance. Having a smaller λ results in generating only a background region, we achieve a good composition with $\lambda = 1$. Notably, λ is also a control parameter that a user uses to control the effect of the background on the foreground scene.

5. Conclusion and Discussion

Limitations. While our approach is highly effective, it has some limitations. Since our method builds on pretrained diffusion models, it inherits their biases. For object insertion, we rely on an inpainting model that may distort object identity in complex cases, such as objects with intricate textures (Fig. 13). Integrating the advancements in recent inpainting models can improve the identity in such cases. Additionally, our guidance mechanism involves optimization at each denoising step, increasing computational cost.

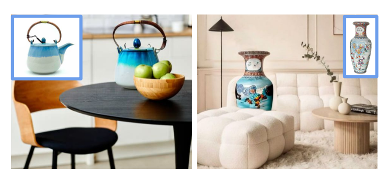

Figure 13. Failure cases.

Conclusion. In this work, we propose a zero-shot framework for depth-aware image editing. We introduce a novel depth-based layering approach that decomposes an image based on a user-specified depth value, enabling precise depth control. Additionally, we present a layer composition method that progressively blends layers using diffusion feature guidance at each denoising step, ensuring realistic layer composition. We demonstrate the effectiveness of our approach on two novel tasks: depth-aware object insertion and scene composition, achieving highly plausible edits with accurate depth control. Our work offers a fresh perspective on image layering and its applications in depth-aware editing.

References

[1] 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, 2, 3, 5

[2] 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, 2

[3] 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. 1, 3

[4] 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. 3

[5] 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. 4, 6

[6] Or Patashnik, Daniel Garibi, Idan Azuri, Hadar Averbuch-Elor, and Daniel Cohen-Or. Localizing object-level shape variations with text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23051–23061, 2023. 3

[7] Manuel Brack, Felix Friedrich, Katharia Kornmeier, Linoy Tsaban, Patrick Schramowski, Kristian Kersting, and Apolinário Passos. Ledits++: Limitless image editing using text-to-image models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8861–8870, 2024.

[8] Zongze Wu, Nicholas Kolkin, Jonathan Brandt, Richard Zhang, and Eli Shechtman. Turboedit: Instant text-based image editing. In *European Conference on Computer Vision*, pages 365–381. Springer, 2024.

[9] Gilad Deutch, Rinon Gal, Daniel Garibi, Or Patashnik, and Daniel Cohen-Or. Turboedit: Text-based image editing using few-step diffusion models. In *SIGGRAPH Asia 2024 Conference Papers*, pages 1–12, 2024.

[10] Yusuf Dalva, Kavana Venkatesh, and Pinar Yanardag. Fluxspace: Disentangled semantic editing in rectified flow transformers. *arXiv preprint arXiv:2412.09611*, 2024. 1

[11] Lvmin Zhang and Maneesh Agrawala. Transparent image layer diffusion using latent transparency. *ACM Transactions on Graphics (TOG)*, 43(4):1–15, 2024. 1, 2

[12] Runhui Huang, Kaixin Cai, Jianhua Han, Xiaodan Liang, Renjing Pei, Guansong Lu, Songcen Xu, Wei Zhang, and Hang Xu. Layerdiff: Exploring text-guided multi-layered composable image synthesis via layer-collaborative diffusion model. In *European Conference on Computer Vision*, pages 144–160. Springer, 2024. 2

[13] Yusuf Dalva, Yijun Li, Qing Liu, Nanxuan Zhao, Jianming Zhang, Zhe Lin, and Pinar Yanardag. Layerfusion: Harmonized multi-layer text-to-image generation with generative priors. *arXiv preprint arXiv:2412.04460*, 2024. 1, 2

[14] Jinrui Yang, Qing Liu, Yijun Li, Soo Ye Kim, Daniil Pakhomov, Mengwei Ren, Jianming Zhang, Zhe Lin, Cihang Xie, and Yuyin Zhou. Generative image layer decomposition with visual effects. *arXiv preprint arXiv:2411.17864*, 2024. 1, 2

[15] Pengzhi Li, Qinxuan Huang, Yikang Ding, and Zhiheng Li. Layerdiffusion: Layered controlled image editing with diffusion models. In *SIGGRAPH Asia 2023 Technical Communications*, pages 1–4. 2023. 1

[16] Mengwei Ren, Wei Xiong, Jae Shin Yoon, Zhixin Shu, Jianming Zhang, HyunJoon Jung, Guido Gerig, and He Zhang. Relightful harmonization: Lighting-aware portrait background replacement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6452–6462, 2024. 2

[17] Zhanghan Ke, Chunyi Sun, Lei Zhu, Ke Xu, and Rynson W.H. Lau. Harmonizer: Learning to perform white-box image and video harmonization. In *European Conference on Computer Vision (ECCV)*, 2022. 2, 6

[18] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10371–10381, 2024. 2, 3, 5, 6

[19] Xi Chen, Lianghua Huang, Yu Liu, Yujun Shen, Deli Zhao, and Hengshuang Zhao. Anydoor: Zero-shot object-level image customization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6593–6602, 2024. 2, 3, 5, 6

[20] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 2, 3, 4

[21] Zipeng Qi, Guoxi Huang, Zebin Huang, Qin Guo, Jinwen Chen, Junyu Han, Jian Wang, Gang Zhang, Lufei Liu, Errui Ding, et al. Layered rendering diffusion model for zero-shot guided image synthesis. *arXiv preprint arXiv:2311.18435*, 2023. 2

[22] Jinrui Yang, Qing Liu, Yijun Li, Soo Ye Kim, Daniil Pakhomov, Mengwei Ren, Jianming Zhang, Zhe Lin, Cihang Xie, and Yuyin Zhou. Generative image layer decomposition with visual effects. *arXiv preprint arXiv:2411.17864*, 2024. 2

[23] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22413–22422, 2024. 2

[24] Omer Bar-Tal, Dolev Ofri-Amar, Rafail Fridman, Yoni Kasten, and Tali Dekel. Text2live: Text-driven layered image and video editing. In *European conference on computer vision*, pages 707–723. Springer, 2022. 2

[25] Vidit Goel, Elia Peruzzo, Yifan Jiang, Dejia Xu, Xingqian Xu, Nicu Sebe, Trevor Darrell, Zhangyang Wang, and Humphrey Shi. Pair diffusion: A comprehensive multimodal

object-level image editor. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8609–8618, 2024. 2, 7

[26] Jiawei Ren, Mengmeng Xu, Jui-Chieh Wu, Ziwei Liu, Tao Xiang, and Antoine Toisoul. Move anything with layered scene diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6380–6389, 2024. 2

[27] Junuk Cha, Mengwei Ren, Krishna Kumar Singh, He Zhang, Yannick Hold-Geoffroy, Seunghyun Yoon, HyunJoon Jung, Jae Shin Yoon, and Seungryul Baek. Text2relight: Creative portrait relighting with text guidance. *arXiv preprint arXiv:2412.13734*, 2024. 2

[28] Sumit Chaturvedi, Mengwei Ren, Yannick Hold-Geoffroy, Jingyuan Liu, Julie Dorsey, and Zhixin Shu. Synthlight: Portrait relighting with diffusion model by learning to re-render synthetic faces. *arXiv preprint arXiv:2501.09756*, 2025. 2

[29] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022. 2

[30] Dave Epstein, Allan Jabri, Ben Poole, Alexei Efros, and Aleksander Holynski. Diffusion self-guidance for controllable image generation. *Advances in Neural Information Processing Systems*, 36:16222–16239, 2023. 3

[31] 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. 3, 4

[32] Mingdeng Cao, Xintao Wang, Zhongang Qi, Ying Shan, Xiaohu Qie, and Yinqiang Zheng. Masactrl: Tuning-free mutual self-attention control for consistent image synthesis and editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22560–22570, 2023. 3

[33] Yupeng Zhou, Daquan Zhou, Ming-Ming Cheng, Jiashi Feng, and Qibin Hou. Storydiffusion: Consistent self-attention for long-range image and video generation. *arXiv preprint arXiv:2405.01434*, 2024. 3

[34] Yoad Tewel, Omri Kaduri, Rinon Gal, Yoni Kasten, Lior Wolf, Gal Chechik, and Yuval Atzmon. Training-free consistent text-to-image generation. *ACM Transactions on Graphics (TOG)*, 43(4):1–18, 2024. 3

[35] Yuxin Zhang, Weiming Dong, Fan Tang, Nisha Huang, Haibin Huang, Chongyang Ma, Tong-Yee Lee, Oliver Deussen, and Changsheng Xu. Prospect: Prompt spectrum for attribute-aware personalization of diffusion models. *ACM Transactions on Graphics (TOG)*, 42(6):1–14, 2023. 3

[36] Andrey Voynov, Qinghao Chu, Daniel Cohen-Or, and Kfir Aberman. p+: Extended textual conditioning in text-to-image generation. *arXiv preprint arXiv:2303.09522*, 2023.

[37] Yuval Alaluf, Elad Richardson, Gal Metzer, and Daniel Cohen-Or. A neural space-time representation for text-to-image personalization. *ACM Transactions on Graphics (TOG)*, 42(6):1–10, 2023. 3

[38] Mingi Kwon, Jaeseok Jeong, and Youngjung Uh. Diffusion models already have a semantic latent space. *arXiv preprint arXiv:2210.10960*, 2022. 3

[39] Manuel Brack, Felix Friedrich, Dominik Hintersdorf, Lukas Struppek, Patrick Schramowski, and Kristian Kersting. Sega: Instructing text-to-image models using semantic guidance. *Advances in Neural Information Processing Systems*, 36:25365–25389, 2023.

[40] Stefan Andreas Baumann, Felix Krause, Michael Neumayr, Nick Stracke, Vincent Tao Hu, and Björn Ommer. Continuous, subject-specific attribute control in t2i models by identifying semantic directions. *arXiv preprint arXiv:2403.17064*, 2024. 3

[41] Ayush Sarkar, Hanlin Mai, Amitabh Mahapatra, Svetlana Lazebnik, David A Forsyth, and Anand Bhattad. Shadows don't lie and lines can't bend! generative models don't know projective geometry... for now. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28140–28149, 2024. 3

[42] Rishi Upadhyay. *Improving Projective Geometry in Diffusion Models*. PhD thesis, UCLA, 2024. 3

[43] 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. 3

[44] Shariq Farooq Bhat, Niloy Mitra, and Peter Wonka. Loosecontrol: Lifting controlnet for generalized depth conditioning. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024. 3

[45] Oscar Michel, Anand Bhattad, Eli VanderBilt, Ranjay Krishna, Aniruddha Kembhavi, and Tanmay Gupta. Object 3dit: Language-guided 3d-aware image editing. *Advances in Neural Information Processing Systems*, 36, 2024. 3

[46] Rahul Sajnani, Jeroen Vanbaar, Jie Min, Kapil Katyal, and Srinath Sridhar. Geodiffuser: Geometry-based image editing with diffusion models. *arXiv preprint arXiv:2404.14403*, 2024. 3

[47] Ruicheng Wang, Jianfeng Xiang, Jiaolong Yang, and Xin Tong. Diffusion models are geometry critics: Single image 3d editing using pre-trained diffusion priors. *arXiv preprint arXiv:2403.11503*, 2024. 3

[48] 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. 3

[49] Jiraphon Yenphraphai, Xichen Pan, Sainan Liu, Daniele Panozzo, and Saining Xie. Image sculpting: Precise object editing with 3d geometry control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4241–4251, 2024. 3

[50] Qihang Zhang, Yinghao Xu, Chaoyang Wang, Hsin-Ying Lee, Gordon Wetzstein, Bolei Zhou, and Ceyuan Yang. 3ditscene: Editing any scene via language-guided disentangled gaussian splatting. *arXiv preprint arXiv:2405.18424*, 2024. 3

[51] Nupur Kumari, Grace Su, Richard Zhang, Taesung Park, Eli Shechtman, and Jun-Yan Zhu. Customizing text-to-image

diffusion with camera viewpoint control. *arXiv preprint arXiv:2404.12333*, 2024. 3

[52] Or Patashnik, Rinon Gal, Daniel Cohen-Or, Jun-Yan Zhu, and Fernando de la Torre. Consolidating attention features for multi-view image editing. In *ACM SIGGRAPH Conference and Exhibition on Computer Graphics and Interactive Techniques in Asia*, 2024. 3

[53] Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023. 3, 5, 6

[54] Binxin Yang, Shuyang Gu, Bo Zhang, Ting Zhang, Xuejin Chen, Xiaoyan Sun, Dong Chen, and Fang Wen. Paint by example: Exemplar-based image editing with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18381–18391, 2023. 5, 6

[55] Yizhi Song, Zhifei Zhang, Zhe Lin, Scott Cohen, Brian Price, Jianming Zhang, Soo Ye Kim, and Daniel Aliaga. Objectstitch: Object compositing with diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18310–18319, 2023.

[56] 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 *European Conference on Computer Vision*, pages 150–168. Springer, 2024.

[57] Nataniel Ruiz, Yuanzhen Li, Neal Wadhwa, Yael Pritch, Michael Rubinstein, David E Jacobs, and Shlomi Fruchter. Magic insert: Style-aware drag-and-drop. *arXiv preprint arXiv:2407.02489*, 2024. 3

[58] 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

[59] Mohamad Shahbazi, Liesbeth Claessens, Michael Niemeyer, Edo Collins, Alessio Tonioni, Luc Van Gool, and Federico Tombari. Inserf: Text-driven generative object insertion in neural 3d scenes. *arXiv preprint arXiv:2401.05335*, 2024. 3

[60] Ayaan Haque, Matthew Tancik, Alexei A Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-nerf2nerf: Editing 3d scenes with instructions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19740–19750, 2023. 3

[61] Yunhao Ge, Hong-Xing Yu, Cheng Zhao, Yuliang Guo, Xinyu Huang, Liu Ren, Laurent Itti, and Jiajun Wu. 3d copy-paste: Physically plausible object insertion for monocular 3d detection. *Advances in Neural Information Processing Systems*, 36, 2024. 3

[62] Andrey Voynov, Kfir Aberman, and Daniel Cohen-Or. Sketch-guided text-to-image diffusion models. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–11, 2023. 3, 4

[63] Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Soumyadip Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 843–852, 2023. 3

[64] Daniel Geng and Andrew Owens. Motion guidance: Diffusion-based image editing with differentiable motion estimators. In *The Twelfth International Conference on Learning Representations*. 3

[65] Grace Luo, Trevor Darrell, Oliver Wang, Dan B Goldman, and Aleksander Holynski. Readout guidance: Learning control from diffusion features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8217–8227, 2024. 3

[66] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6038–6047, 2023. 4, 5

[67] Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth. *arXiv preprint arXiv:2302.12288*, 2023. 5, 6

[68] Abdou. vit-swin-base-224-gpt2-image-captioning. In *https://huggingface.co/Abdou/vit-swin-base-224-gpt2-image-captioning*, 2022. 5, 6

[69] Yifan Jiang, He Zhang, Jianming Zhang, Yilin Wang, Zhe Lin, Kalyan Sunkavalli, Simon Chen, Sohrab Amirghodsi, Sarah Kong, and Zhangyang Wang. Ssh: A self-supervised framework for image harmonization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4832–4841, 2021. 5

[70] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021. 6

[71] Mikołaj Bińkowski, Dougal J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD GANs. In *International Conference on Learning Representations*, 2018. 6

[72] 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. 6

[73] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. 6

[74] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023. 6

[75] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3), 2022. 6