Diffusion Handles
Enabling 3D Edits for Diffusion Models by Lifting Activations to 3D

Karran Pandey[1] Paul Guerrero[2] Matheus Gadelha[2] Yannick Hold-Geoffroy[2]
Karan Singh[1] Niloy J. Mitra[2,3]
[1] University of Toronto [2] Adobe Research [3] UCL

diffusionhandles.github.io

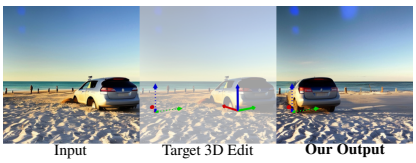
Figure 1. *Diffusion Handles* enable 3D-aware object edits (e.g., 3D translations and rotations), on images generated by diffusion models. Edited images exhibit plausible changes in perspective, occlusion, lighting, shadow, and other 3D effects, without explicitly solving the inverse graphics problem. Our method does not require training or fine-tuning and can be applied to real images through image inversion.

Abstract

Diffusion Handles is a novel approach to enable 3D object edits on diffusion images, requiring only existing pretrained diffusion models depth estimation, without any fine-tuning or 3D object retrieval. The edited results remain plausible, photo-real, and preserve object identity. Diffusion Handles address a critically missing facet of generative image-based creative design. Our key insight is to lift diffusion activations for a selected object to 3D using a proxy depth, 3D-transform the depth and associated activations, and project them back to image space. The diffusion process guided by the manipulated activations produces plausible edited images showing complex 3D occlusion and lighting effects. We evaluate Diffusion Handles: quantitatively, on a large synthetic data benchmark; and qualitatively by a user study, showing our output to be more plausible, and better than prior art at both, 3D editing and identity control.

1. Introduction

Text-to-image diffusion models [45, 46, 48] are the state-of-the-art in image generation. They produce photo-real outputs, effortlessly generate complex, high-resolution images, and support various forms of conditional generation [58]. Pretrained diffusion models can be repurposed to support many image processing tasks [47], such as, image in- or out-painting, superresolution, and denoising. However, there is limited support for object-centric editing in such images, where the 3D composition of scene objects can be changed, while preserving their identity. Existing approaches treat such edits in image space, including: cutting and pasting objects to desired locations using object masks and regenerating the background [2]; using gradient domain edits with some identity control [13]; or leveraging novel view synthesis with fine-tuned diffusion models [30] that are costly to train and can reduce model generality. These approaches are particularly restrictive and do not directly support 3D object edits involving translations, rotations, and changes in scene perspective. Moving the car to a new location on the beach in Figure 1 for example, is non-trivial if the car's identity has to be retained. The 3D edit should successfully capture complex light and shading



effects, as well as a change in perspective (see gizmos in inset), which is hard to achieve by enforcing object pixel intensity invariance used in image-space identity control.

We propose *Diffusion Handles* to support object-level edits aware of the underlying (hidden) 3D object structure. We demonstrate how to generate plausible 3D edits without solving the inverse graphics problem [22, 60] (i.e., without needing to recover the full scene geometry, materials, and illumination). In other words, we enable 3D-aware edits directly on 2D images generated by a diffusion model or actual photos that can be inverted [31] into a diffusion model. At the heart of our method is a novel approach to *lift* diffusion activations to 3D, encoded as coarse proxy depth that can be estimated by a method (e.g., [6]). The lifted activations can then be moved or transformed in 3D scene space and projected back to the image plane using the estimated depth maps. We use these projected activation maps to guide the diffusion process to produce the final edited image (see Figure 1). Our approach is simple and effective, leverages pretrained diffusion models *without fine-tuning or additional training data*, and produces plausible results even when the estimated depths have moderate warps.

Our method enables a range of 3D modifications to objects, such as translations that affect perspective, scaling, and some rotation. We accomplish this by converting the depth map into a point cloud and then applying the desired 3D transformation. To obtain this point cloud, we need the depth map of the object, either from a known template or by estimating it from the image. When working with a known template, precise alignment is necessary. When using an estimated depth, disocclusions can result in unknown depth regions, leading to uncertainty and loss of identity when the transformation is too extreme, especially with large rotations.

We evaluate our method on various editing scenarios, including real and generated images. Since our approach allows for a new type of 3D-aware edits with diffusion-based generated images, we compare it to other editing methods capable of applying similar edits. Specifically, we compare against a 2D editing method using a similar activation-based guidance [13] and a 3D-aware editing method based on novel view synthesis that fine-tunes a diffusion model using 3D information [30]. In contrast, our method allows 3D-aware editing without the need for fine-tuning (see Table 1). We demonstrate the generalizability of our method on a large number of qualitative examples. Additionally, we conduct a user study to compare our results against existing baselines and ablated versions of our approach. In summary, to the best of our knowledge, we present the first editing framework that supports fine-grained 3D control over the object layout in diffusion images, without requiring any additional training.

Table 1. **Comparison to related methods.** Our approach is unique in allowing 3D edits, without any additional training, or 3D data. Our Zero123 baseline allows for 2.5D edits (denoted by *).

	3DIT [30]	DSG [13]	Zero123 [27]	Obj.Stitch [51]	Ours
Training-free?	×	✓	×	×	✓
No 3D data?	×	✓	×	✓	✓
3D edits?	✓	×	✓*	×	✓

2. Related Work

Text-Guided Image Generation. Seminal approaches for creating images from text prompts relied on a combination of image retrieval and composition using user-created layouts [10]. Later, several representation learning techniques targeted creating joint representation for images and text [32, 43, 52]. In the last few years, such multimodal approaches were scaled to hundreds of millions of text-image pairs by using modern deep learning architectures and contrastive learning [44]. Using those representations, initial attempts at image generation did so through a combination of gradient-based optimization and image priors – some hand-crafted [14], others data-driven [11, 40]. However, they suffered from slow runtime and had trouble generating visually appealing imagery. These issues were addressed by several techniques that trained models for outputting images from text prompts. Such approaches relied on autoregressive [8, 12, 56] and diffusion models [15, 36, 45]. Follow-up works also investigated how to provide finer-grained control over the generative process (beyond text prompts) like using regional prompting [4, 57] and additional user-provided image information like depth maps and edges [34, 58]. Despite the photo-real quality of the generated images, those approaches do not allow users to manipulate existing image elements and, more importantly, provide any 3D-aware controls. While our approach relies on existing diffusion models [46], we extend their capabilities (without the need for any additional training) to allow users to manipulate objects in real or generated images in a 3D-aware manner.

Image Editing with Generative Models. Generative models have been powering several image editing tasks, like inpainting [28], object insertion and harmonization [25] and stylization [54]. For these traditional tasks, data-driven models offer a way to achieve superior control with less user intervention. They also enabled new tasks like synthesizing images from semantic segmentation maps [34, 39, 58] and text-annotated layouts [3, 4, 9, 15, 26]. More recently, open-ended text-guided image editing has been explored by combining large language models with text-to-image generators [5, 7]. Despite the impressive results, the previous methods do not allow users to preserve the appearance of objects while manipulating the image elements. This issue can be partially addressed by allowing users to edit images by dragging relevant keypoints [33, 38]. Such controls are adequate for performing object deformation but might be

cumbersome for other tasks like changing object positioning in the scene. Closely related to our work, Epstein *et al*. [13] address the problem of editing existing image elements by manipulating the intermediate representation of text-to-image models. They demonstrate how to alter individual object size and 2D position without resorting to any additional training. Unfortunately, none of the aforementioned techniques target 3D object manipulation in images. For that reason, they are incapable of addressing occlusions between edited entities in an image, their 3D position and out-of-plane rotation, for example. On the other hand, our approach is specifically designed to handle these scenarios.

3D-aware Image Editing. Several works have investigated editing 2D images in a 3D-aware manner by changing the viewpoint in which a picture was taken. This was initially attempted with hand-crafted priors [19], but was lately enhanced by data-driven models performing a combination of inpainting and monocular depth estimation [21, 37, 49]. While those techniques allow users to perform fine-grained camera motions, they do not investigate how to manipulate particular objects in the scene. Early attempts were not fully automatic but required significant user input [60] and existing 3D models [23]. More recent work relies in massive 3D object datasets [30] or in training regimes relying in image-based models which hurt their interaction time [42] or generality [35]. Zero123 [27], a notable recent method, does allow pseudo-3D rotations but requires access to 3D models to create a dataset in order to fine-tune a ControlNet [58] backend. In this work, we propose a technique that does not rely on any additional training, does not require large 3D datasets, has an inference time similar to generating an image in text-to-image models, and allows editing in-the-wild images without category-specific restrictions while maintaining realistic image quality.

3. Overview

Our goal is to imbue text-to-image generation pipelines with 3D-aware object edit handles, without requiring any fine-tuning of the generative model. In particular, given an image of a 3D scene and a corresponding text prompt, our method allows the user to perform a 3D transform, such as translation, rotation and scale, on any object described in the text prompt. Figure 1 shows several examples.

The given image is first inverted with a diffusion model, and an image of the *edited* scene is generated by the same diffusion model, guided by additional loss terms that we design to fulfill three tasks: (i) to generate a 3D-transformed version of the edited object, (ii) to otherwise preserve the appearance of all objects in the 3D scene, and (iii) to still allow leveraging the prior of the diffusion model so that the edited object realistically interacts with its environment through lighting, shadows, etc.

We achieve these tasks by lifting activations of the diffusion model to the 3D surface of scene objects, where we can apply 3D transformations. The 3D-transformed activations can then serve as guidance when generating the edited image. We find that the strong prior of the diffusion model makes our method robust to inaccuracies and artifacts of the 3D surfaces we use in our edits, and that the approximate depth obtained from existing depth estimators is sufficient to allow for a wide range of 3D edits.

4. Diffusion Models

Training. During training, a fixed process adds a random amount of noise to an image x to get a noisy image $\tilde{x}(t)$:

$$\tilde{x}(t) = \sqrt{\alpha(t)}\, x + \sqrt{1-\alpha(t)}\, \epsilon, \tag{1}$$

where $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is Gaussian noise, and $t \in [0, T]$ parameterizes a noise schedule α that determines the amount of noise in $\tilde{x}(t)$, with $\alpha(0) = 1$ (no noise) and $\alpha(T) = 0$ (pure noise). A denoiser ϵ_θ with parameters θ is trained to predict the noise ϵ using the following loss:

$$\mathcal{L}_{\text{diff}} = w(t) \| \epsilon_\theta(\tilde{x}(t); t, y, d) - \epsilon \|_2^2 \tag{2}$$

where d is a depth map, y is an encoding of a text prompt, and $w(t)$ is a weighting scheme for different parameters t. The parameter t is sampled uniformly from $[0, T]$ in each training iteration. Once ϵ_θ is trained, $-\epsilon_\theta(\tilde{x}(t); t, y, d)$ defines a vector field in image space that points towards the natural (non-noisy) image manifold.

Inference. At inference time, an image is generated by starting from pure noise $\tilde{x}(T)$, and following the vector field $-\epsilon_\theta(\tilde{x}(t); t, y, d)$ towards the natural image manifold. Multiple different samplers have been proposed [17, 50] to find a trajectory $x_T, x_{T-1}, \ldots, x_0$ with a fixed number of T discrete steps that starts at $x_T := \tilde{x}(T)$ and ends in an image x_0 close to the natural image manifold. Our method is compatible with any standard sampler; we describe the sampler we use in our experiments in the supplemental.

Guidance. The vector field $\epsilon_\theta(\tilde{x}(t); t, y, d)$ can be guided to minimize a custom energy $\mathcal{G}(\tilde{x}(t); t, y, d)$ by biasing each step with the gradient $\nabla_{\tilde{x}(t)}\, \mathcal{G}$ of the energy. Apart from this form of guidance, most samplers also use classifier-free guidance [16] to more closely follow the text prompt y, by moving the vector $\epsilon_\theta(\tilde{x}(t); t, y, d)$ away from the vector $\epsilon_\theta(\tilde{x}(t); t, \emptyset, d)$ obtained with the encoding $\emptyset$ of an empty text prompt (also called *null-text*). Similar to previous work [13], we combine the two forms of guidance:

$$\begin{aligned} \epsilon_\theta^{\mathcal{G}}(\tilde{x}(t); t, y, d) \;=\; & (1+\mu)\, \epsilon_\theta(\tilde{x}(t); t, y, d) \\ & - \mu\, \epsilon_\theta(\tilde{x}(t); t, \emptyset, d) \\ & + \lambda\, \nabla_{\tilde{x}(t)}\, \mathcal{G}(\tilde{x}(t); t, y, d). \end{aligned} \tag{3}$$

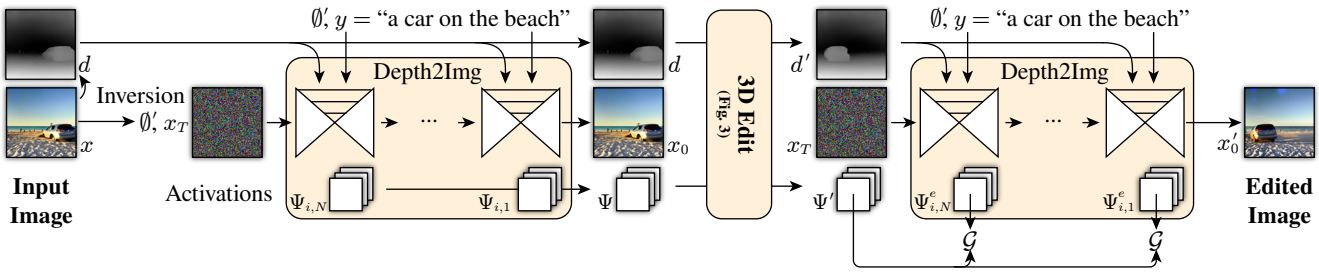


Figure 2. **Overview.** Starting from an input image x, we first estimate depth d and invert the image into a depth-to-image diffusion model, giving us activations Ψ that reconstruct the input image. A 3D transform supplied by a user can then be applied to Ψ by lifting them to the 3D surfaces given by the depth map (Figure 3 shows details). Using the 3D transformed Ψ' as guidance in a diffusion model allows us to generate an edited image that adheres to the edit, preserves the identity of the input image, and is plausible.

The main focus of our work is designing a guidance energy $\mathcal{G}$ that biases the diffusion steps to produce a 3D-edited version of an input image. Section 5 up to 5.1 describes how we obtain the features needed for this energy, and Section 5.2 defines the energy.

Latent Diffusion. In our experiments, we use a latent diffusion model where images x_t contain features from a pre-trained latent space, rather than RGB values; but our method is also compatible with non-latent diffusion models, so we keep our description general.

5. 3D Edits for Diffusion Models

To perform 3D image edits that preserve both realism and identity of the original image, we 3D-transform the feature spaces of intermediate layers in a pre-trained diffusion model, the *activations* Ψ. These activations describe the appearance and identity of objects in a generated image. We use activations in the decoder of the denoiser ϵ_θ, but only use layers with sufficient resolution to avoid inaccurate guidance; we use layers 2 and 3 (the last two layers) in the decoder of the StableDiffusion v2 [46] depth-conditioned denoiser. We denote activations of layer i in denoising step t as $\Psi_{i,t}$.

To apply a 3D edit to a given image x with text prompt y, we proceed in three steps: (i) we invert the image to reconstruct it with a diffusion model and save activations Ψ of the generation process; (ii) we apply the 3D edit to the activations; (iii) we re-generate the image using the edited activations as guidance. In the remainder of this section we are going to describe these three steps in more detail.

Inverting the Input Image. Given an input image x, corresponding text prompt encoding y, we obtain the activations by inverting the image with our diffusion model using Null-Text Inversion [31]. As our diffusion model is also conditioned by a depth map, the depth map can either be given as input (for example, from a synthetic 3D scene), or we estimate it using an existing monocular depth estimator [6]. Note that the depth-conditioned diffusion model is tolerant to inaccuracies and noise in the depth map, as it was trained on estimated depth maps. The inversion gives us an initial noise x_T and an updated null-text encoding $\emptyset'$ that we can use to reconstruct the input image x in an inference pass $x_T, \ldots, x_0$ of the diffusion model, such that $x_0 \approx x$. During inference pass, we record activations $\Psi_{i,t}$ of all relevant layers i and time steps t.

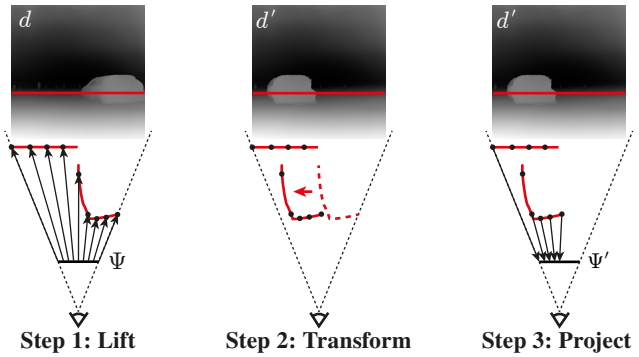


Figure 3. **3D edit of depth and activations.** (i) Activations Ψ are lifted to 3D surfaces; (ii) the depth d is 3D-transformed, together with the lifted Ψ; (iii) the transformed Ψ are projected back to the image plane. Section 5.1 gives a detailed description of these steps.

5.1. Performing the 3D-Edit

To perform a 3D edit in our 2D image x, we define a simple warping mechanism $\mathcal{W}$. Given a flow field $F : [0,1]^2 \mapsto \mathbb{R}^2$, and a signal $X : [0,1]^2 \mapsto C$, $\mathcal{W}$ is defined as

$$\mathcal{W}[X, F](u) = X(u - F(u)), \tag{4}$$

where u is a coordinate in $[0,1]^2$. This operator will be used throughout our method to warp signals defined on a 2D domain, such as attention maps and activations.

3D-aware Flow Field. A key differentiating factor of our approach is that we compute the flow field F in a 3D-aware manner. This is done in three steps (see Figure 3).

Step 1 - Lift: the lifting function $L_d : [0,1]^2 \mapsto \mathbb{R}^3$ assigns a 3D coordinate to every point in the 2D domain of the image X (assuming coordinates normalized to $[0,1]^2$),

based on the depth map d. We assume the same $55°$ field of view (fov) our depth estimator [6] was trained on.

Step 2 - Transform: the user defines a function $T : \mathbb{R}^3 \mapsto \mathbb{R}^3$ that modifies the position of the points in 3D space. Groups of 3D points that correspond to an object can be identified with the help of any off-the shelf image segmentation model. In our experiments, we opt for a SAM [24]-based approach, see the supplemental for details. We call the mask of the selected segment the *object mask* M_o. Lifting M_o to 3D identifies the 3D points corresponding to the object of interest, which can then be manipulated by the user with a rigid 3D transformation. All other 3D points remain unchanged.

Step 3 - Project: the projection function $P : \mathbb{R}^3 \mapsto \mathbb{R}^2$ projects 3D coordinates back to the 2D image plane assuming the same camera parameters as the lifting function.

By composing all three steps, we define an operation that transforms 2D coordinates in the original image to the corresponding position in the edited image. Our 3D-aware flow F is based on the inverse of this operation:

$$F(u) = u - (P \circ T \circ L_d)^{-1}(u). \tag{5}$$

Note that the inverse may not be defined for some 2D coordinates, since the operation $P \circ T \circ L_d$ is not always bijective; it may create overlapping regions (like occlusions) and holes (like disocclusions). We handle overlapping regions by picking the coordinates closest to the camera. To handle holes, we create a *valid mask* M_v of regions that are not holes $M_v \coloneqq \mathbb{1}_{\text{range}(P \circ T \circ L_d)}$, and only guide regions inside this mask when generating the edited image.

Edited Maps. We warp the activations $\Psi_{i,t}$ with this 3D-aware flow field to get edited activations $\Psi'_{i,t}$:

$$\Psi'_{i,t} \coloneqq \mathcal{W}[\rho(\Psi_{i,t}), F] \tag{6}$$

where ρ denotes bilinear interpolation.

Edited Depth. To obtain an edited depth map d', we separately construct the edited depth for the transformed object d'_o and for the remaining (static) scene d'_b, before recompositing them. Treating them separately allows us to inpaint any holes in the static part of the scene that might be created by the 3D edit by leveraging the prior of a large 2D diffusion model.

Specifically, the depth for the static part of the scene d'_b is obtained by removing the transformed object from the image x using an existing object removal method [53], with the object mask M_o as input, resulting in an image x_b without the transformed object. d'_b is then estimated from x_b using a monocular depth estimator [6]. We obtain depth of the transformed object d'_o from the distance between the camera and the transformed 3D points $T \circ L_d$:

$$d'_o(u) \coloneqq \|\mathcal{W}[T \circ L_d, F](u)\|_2, \tag{7}$$

where we assume that the camera is at the origin.

The transformed object depth d'_o and the depth of the remaining scene d'_b are then composited seamlessly using Poisson Image Editing [41] to obtain the edited depth d'.

5.2. Generating the Edited Image

We generate the edited image x'_0 using a diffusion process that is conditioned on the text prompt y and the edited depth d', and uses the initial noise x_T and the null-text $\emptyset$ obtained from the inversion described at the start of this section. Without any additional guidance, the resulting image would closely approximate the input image x. Thus, we guide the diffusion process to follow the edited activations Ψ'.Guiding the diffusion process to follow the edited activations encourages the resulting image to preserve the identity of objects from the original scene and to follow the edited object layout. We add two energy terms.

The *object guidance* energy $\mathcal{G}_o$ focuses on the edited object only. It is the L2 distance between the activations of the diffusion process Ψ^e and the edited activations Ψ':

$$\mathcal{G}_o \coloneqq \sum_{i,t} w^o_{i,t} \sum_u \left(M'_o(\Psi^e_{i,t} - \Psi'_{i,t})\right)^2(u), \tag{8}$$

where Ψ^e are the activations of the denoiser in the diffusion process, and $M'_o \coloneqq \mathcal{W}[M_o, F] \cdot M_v$ is the valid part of the warped object mask, i.e. the mask of the foreground object in the edited image. $w^o_{i,t}$ is a per-step and per-layer weight we set according to a schedule. See below for details.

The *background guidance* energy $\mathcal{G}_b$ is defined similarly to the object guidance energy, but focuses on the static part of the scene only:

$$\mathcal{G}_b \coloneqq \sum_{i,t} w^b_{i,t} \frac{\left(\sum_u M'_b \Psi^e_{i,t}(u) - \sum_u M'_b \Psi'_{i,t}(u)\right)^2}{\sum_u M'_b(u)}, \tag{9}$$

where $M'_b \coloneqq 1 - M'_o$ and w^b_t is set according to a similar schedule as $w^o_{i,t}$, see below for details. We compare the average of the activations over the image, as we expect some parts of the static scene to change, for example due lighting or shadows, disocclusions, etc.

We set weights $w^o_{i,t}$ and $w^b_{i,t}$ according to a *guidance schedule*. (i) We guide only up until time step $38/50$, and then zero the guidance. (ii) We cycle between guiding different layers in each time step, guiding layer 3 in the first step, layer 2 in the second step, both layers in the third step, and repeat this cycle in subsequent steps. (iii) Finally, we adjust the relative weighting of $w^o_{i,t}$ and $w^b_{i,t}$ according to the desired level of foreground and background preservation. Section 6 discusses and motivates these design choices and gives examples of different settings in an ablation.

We define the final guidance energy $\mathcal{G}$ as, $\mathcal{G} \coloneqq \mathcal{G}_o + \mathcal{G}_b$. We use the gradient of this guidance energy to bias each step of the diffusion process, as described in Eq.(3) as, $\epsilon^{\mathcal{G}}_\theta(x_t; t, y, \emptyset', d)$ resulting in the edited image x'_0.

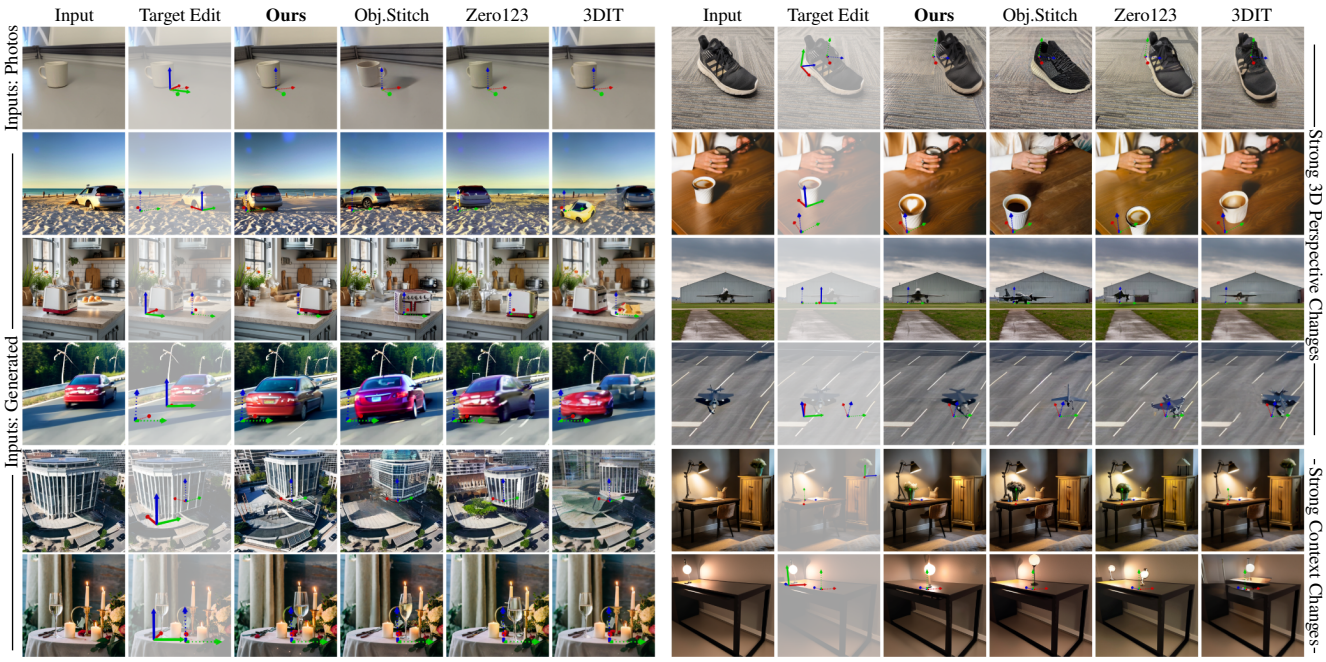


Figure 4. **Qualitative comparison.** We compare our method to three baselines. A target edit (from solid to dotted axes) is applied to an object from the input image . We show that our method better achieves our three goals: identity preservation, edit adherence, and plausibility.

6. Results

Datasets. We created two datasets to evaluate our method. The `PhotoGen` dataset consists of 31 edits in 26 images images (five of the images have two different edits) that we either generated, photographed, or licensed and downloaded. It contains 5 photographs and 21 generated images. In the `Benchmark` dataset we aim at minimizing our (the author's) bias in the choice of images and edits. It consists of 50 edits in 50 images (one edit per image) that we generated with a depth-to-image diffusion model using depth from synthetic 3D scenes. The 3D scenes are generated automatically by randomly choosing a 3D asset from 10 categories in the ModelNet40 [55] dataset, and placing the asset at a random location on a ground plane. Edits are randomly chosen from 3D translations and 3D rotations. Both the parameters of the initial placement and of the edit are constrained to ensures objects remain withing the view frustum and exhibit a limited amount of disocclusion after the edit.

Baselines. We compare to several state-of-the-art methods that share our goal of image editing with generative models. *Object3DIT* [30] fintunes Zero123 [27] with synthetic data to enable either 3D rotations, scaling, or translation on a ground plane. *ObjectStitch* [51] allows transplanting objects from one image to a given 2D position in another image; we use it to transplant objects to a different location in the same image and remove the original, unedited object using the same object removal method [53] we use in our depth edit. We create another baseline that uses *Zero123* to get a novel view of the foreground object, removes the original foreground object [53], moves the novel view to a new image location, and inpaints a 15-pixel-wide region around the novel view using Firefly [1] to improve image coherence. We also experimented with Diffusion Self-Guidance [13], but found that the public code performs far worse in terms of identity preservation and edit adherence than the published version (which uses the proprietary Imagen [48]), as confirmed by the authors. For fairness, we instead show an ablation that comes close to this method.

Qualitative Comparison. See Figure 4. ObjectStitch generates scenes with good plausibility, but as it does not provide 3D controls, we observe low edit adherence (i.e., the output does not match the target edit); we also observe relatively low identity preservation. Zero123 and 3DIT have better identity preservation, but Zero123 struggles to generate good novel views for objects that are for from the dataset it was fine-tuned on (like the somewhat blurry red car), and the fixed inpainting region limits the plausibility of secondary effects like shadows and lighting in the edited scene (see for example the lack of shadows for the coffee cup). 3DIT is biased even more strongly than Zero123 by the synthetic scenes it was fintuned on, in which objects have a limited range of sizes and types, and are viewed from a limited range of angles. 3DIT lacks generalization, for example, it fails to perform an edit (e.g. the wine glasses), or places a novel object into the scene (e.g. the car on the beach). Our method shows better identity preservation and edit adherence due to

Figure 5. **Stop-motion edits.** Intermediate edits along two edit trajectories demonstrate consistency and identity preservation.

our detailed 3D-aware guidance, and better plausibility since we do not perform any fine-tuning that would bias the prior of the pre-trained diffusion model towards more constrained scene types. The supplementary provides comparisons on the full `PhotoGen` dataset. Figure 5 shows multiple intermediate edits along two edit trajectories, demonstrating the consistency and identity preservation of our method.

User Study. We quantitatively compare our method to all baselines in terms of the three desirable goals: identity preservation, edit adherence, and plausibility, with a user study on a subset of 11 edits in 11 images from our `PhotoGen` dataset, including one photograph. We separately evaluate each desirable goal by showing users pairs of images and asking them to select the image that better fulfills the goal. We form 66 random image pairs, where each pair compares a random result from our method to the corresponding result from a random baseline. We split these 66 pair into 3 groups of 22 pairs each, one group for each goal. A total of 22 users participated in the study, with mixed expertise in image editing. Each user compared all 22 pairs for each goal (484 data points per goal, and an average of 161 data points for each of the three method pairings). For the plausibility goal, we additionally compare to the original input image as an upper bound for the achievable plausibility (average of 121 data points per method pair). Both the order of pairs for each goal, and of methods in each pair was randomized. See supplemental for details.

Results are shown in Figure 6. We can see that users clearly preferred our method over the baselines in all three goals. In some cases, users even found our results to be more plausible than the original input images, although, as we would expect, the original images were still more plausible on average. The results support our observations from the qualitative results: ObjectStitch has good plausibiliy, but relatively low identity preservation and edit adherence. 3DIT and Zero123 have better identity preservation and edit adherence, but lower plausibility. 3DIT has especially low plausibility due to its biased diffusion prior.

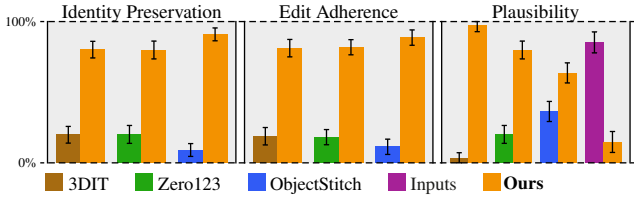


Figure 6. **User study.** We compare how well each method achieves our three main goals. Each pair of bars show the percentage of users that preferred our method (orange) or a baseline (other color) with 95% confidence intervals. The *inputs* bar represents an upper bound to the plausibility of an edited image.

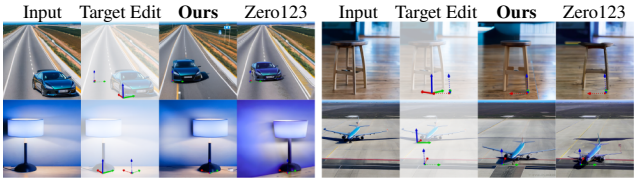


Figure 7. **Synthetic benchmark.** Comparison to Zero123 [27] on a few examples of our `Benchmark` dataset.

Synthetic Benchmark. We automatically generate images and edits to reduce selection bias in the results. We use synthetic depth, which comes as a by-product of the automatic image generation. This ensures that scene depth is reasonable, which factors out the influence of errors in the depth estimate from our experiments. The full benchmark on all 50 scenes is given in the supplementary, Figure 7 shows a qualitative comparison on four scenes. We see good identity preservation, edit adherence and plausibility in these scenes, suggesting that our method works robustly on random scenes given a reasonable depth. We additionally provide a quantitative comparison to Zero123, ObjectStitch and 3DIT on the full benchmark dataset in Table 2 that evaluates edit adherence and identity preservation. *Edit adherence* is evaluated using the Intersection over Union (IoU) between the a SAM-based [24] segmentation of the foreground object in the edited image and a ground truth segmentation mask obtained from the synthetic depth. *Identity Preservation* is evaluated using a cycle consistency metric that transform the edited image back to the original object configuration and measures the difference to the original input image using both the L1 distance and LPIPS [59]. As seen in Table 2, our method consistently outperforms the three baselines on both edit adherence and identity preservation.

Table 2. **Quantitative comparison on the `Benchmark` dataset.** We compare *identity preservation*, based on the cycle consistency of performing the edit, followed by its inverse; and *edit adherence*, as measured by the IoU between image region covered by the edited foreground object and the corresponding ground truth image region.

	Identity Preservation		Edit Adherence
	$E_{\text{id}}^{\text{L1}}(\times 10)\downarrow$	$E_{\text{id}}^{\text{LPIPS}}\downarrow$	$S_{\text{edit}}\uparrow$
Obj.Stitch [51]	0.89	<u>0.25</u>	0.37
Zero123 [27]	1.05	0.31	<u>0.52</u>
3DIT [30]	<u>0.74</u>	0.27	0.15
Ours	**0.71**	**0.19**	**0.85**

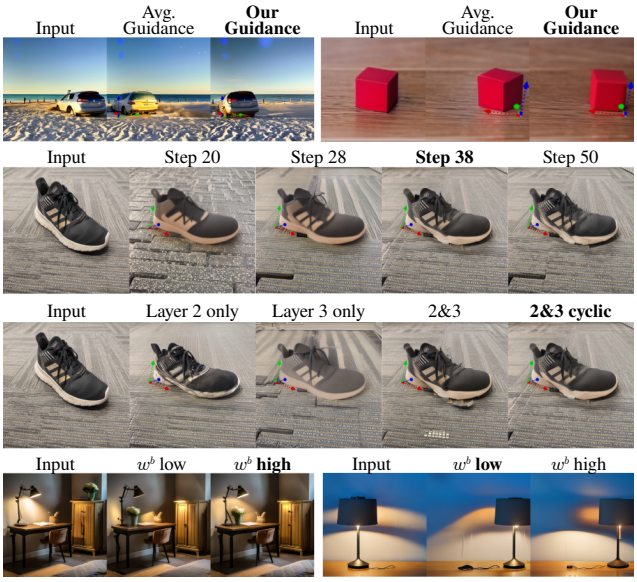


Figure 8. **Ablation Study.** We show the effect of several design choices of our method. See the *Ablation Study* paragraph for details.

Ablation Study. We ablate four design choices, see Figure 8. (i) First, we compare our form of local guidance for the foreground vs. using the average over the foreground region in Eq. 8, similar to DSG [13]. Our local guidance significantly improves identity preservation and edit adherence. (ii) Second, we guide up to a different maximum number of steps. Intuitively, the last time steps allow the model to reconcile the edited object with the scene, by creating details such as contact shadows and lighting. We found that giving the model space to do this reconciliation without guidance increases plausibility. Guiding too few steps, on the other hand, reduces identity preservation. (iii) Third, we show the effect of using different choices of layers in the guidance schedule for each time step. Guiding the second layer of the denoiser decoder only tends to preserve texture style, but loses some identity preservation and edit adherence, while guiding the third layer only tends to have the opposite effect. Ideally we want to preserve all three properties, but we found that guiding both layers introduces artifacts, possibly because the guidance of different layers can be contradictory. Our cyclic schedule reduces artifacts by guiding both layers. (iv) Finally, we balance foreground and background weights w^o and w^b. In both scenes, the lighting of the foreground object and background are at odds (e.g., on the left, the vase is originally unlit and the background at the target position has strong lighting). Setting w^b low relative to w^o preserves foreground lighting, but changes background lighting, and vice-versa for high w^b.

7. Conclusion

We have presented Diffusion Handles to enable 3D-aware object level edits on 2D images, which may be generated or real photographs. We do not require additional training or 3D supervision data, and avoid explicitly solving the inverse graphics problem. We demonstrated that by lifting intermediate diffusion activations to 3D using estimated depth, and transforming the activations with user-specified 3D edits, one can produce realistic images with a good balance between plausibility and identity control while respecting the target edits. In our extensive tests, we demonstrated the superiority of our proposed approach against other contemporary baselines, both quantitatively and qualitatively.

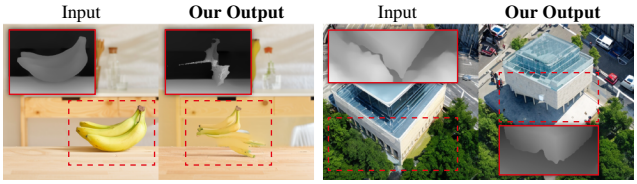


Figure 9. **Limitations.** Our method relies on a reasonable depth map. Large edits that reveal strong distortions of a depth estimate or missing parts of the depth result in low-quality output.

Limitations and Future Work. Although our method is robust to the quality of the estimated depths, which are often warped strongly in view direction, large edits that make this warping apparent, and edits that reveal parts of the objects hidden in the original view may give undesirable results (see Figure 9). In the future, we would like to use shape priors to infill the estimated depth maps in occluded regions. One exciting option would be to use recent image-to-Nerf models [20, 29] to perform such a regularization. Another limitation of our method is that identity preservation, while better than existing methods, is still not perfect. In the future, we expect generative image models to also produce additional channels (e.g., albedo, normal, specular, illumination) that would allow more physically grounded control over object identity that is hard to achieve directly using only RGB information. Additionally, we plan to experiment with removing the text prompt, as it may not be necessary for inversion. Finally, we would like to extend our method to produce video output by animating 3D edits similar to Figure 5, but with more frames. The challenging part will be ensuring temporal smoothness while preserving object identity without additional training. We expect to use pre-trained video diffusion models [18].

References

[1] Adobe. Firefly. 6

[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. 1

[3] Omri Avrahami, Thomas Hayes, Oran Gafni, Sonal Gupta, Yaniv Taigman, Devi Parikh, Dani Lischinski, Ohad Fried,

and Xi Yin. Spatext: Spatio-textual representation for controllable image generation. In *CVPR*, 2023. 2

[4] Omer Bar-Tal, Lior Yariv, Yaron Lipman, and Tali Dekel. Multidiffusion: Fusing diffusion paths for controlled image generation. *Int. Conf. Machine Learning*, 2023. 2

[5] James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, Wesam Manassra, Prafulla Dhariwal, Casey Chu, Yunxin Jiao, and Aditya Ramesh. Improving image generation with better captions. 2

[6] 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. 2, 4, 5

[7] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions. In *CVPR*, 2023. 2

[8] Huiwen Chang, Han Zhang, Jarred Barber, AJ Maschinot, Jose Lezama, Lu Jiang, Ming-Hsuan Yang, Kevin Murphy, William T Freeman, Michael Rubinstein, et al. Muse: Text-to-image generation via masked generative transformers. *Int. Conf. Machine Learning*, 2023. 2

[9] Minghao Chen, Iro Laina, and Andrea Vedaldi. Training-free layout control with cross-attention guidance. *arXiv preprint arXiv:2304.03373*, 2023. 2

[10] Tao Chen, Ming-Ming Cheng, Ping Tan, Ariel Shamir, and Shimin Hu. Sketch2photo: internet image montage. *ACM Transactions on Computer Graphics*, 2009. 2

[11] Katherine Crowson, Stella Rose Biderman, Daniel Kornis, Dashiell Stander, Eric Hallahan, Louis Castricato, and Edward Raff. Vqgan-clip: Open domain image generation and editing with natural language guidance. In *ECCV*, 2022. 2

[12] Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, and Jie Tang. Cogview: Mastering text-to-image generation via transformers. In *NeurIPS*, 2021. 2

[13] Dave Epstein, Allan Jabri, Ben Poole, Alexei A Efros, and Aleksander Holynski. Diffusion self-guidance for controllable image generation. *arXiv preprint arXiv:2306.00986*, 2023. 1, 2, 3, 6, 8

[14] Kevin Frans, Lisa Soros, and Olaf Witkowski. Clipdraw: Exploring text-to-drawing synthesis through language-image encoders. In *NeurIPS*, 2022. 2

[15] Oran Gafni, Adam Polyak, Oron Ashual, Shelly Sheynin, Devi Parikh, and Yaniv Taigman. Make-a-scene: Scene-based text-to-image generation with human priors. 2022. 2

[16] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 3

[17] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 3

[18] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *arXiv:2204.03458*, 2022. 8

[19] Derek Hoiem, Alexei A. Efros, and Martial Hebert. Automatic photo pop-up. In *ACM Transactions on Computer Graphics*, 2005. 3

[20] Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. Lrm: Large reconstruction model for single image to 3d, 2023. 8

[21] Varun Jampani, Huiwen Chang, Kyle Sargent, Abhishek Kar, Richard Tucker, Michael Krainin, Dominik Kaeser, William T Freeman, David Salesin, Brian Curless, and Ce Liu. Slide: Single image 3d photography with soft layering and depth-aware inpainting. In *CVPR*, 2021. 3

[22] Natasha Kholgade, Tomas Simon, Alexei Efros, and Yaser Sheikh. 3d object manipulation in a single photograph using stock 3d models. *ACM Transactions on graphics (TOG)*, 33 (4):1–12, 2014. 2

[23] Natasha Kholgade, Tomas Simon, Alexei A. Efros, and Yaser Sheikh. 3d object manipulation in a single photograph using stock 3d models. *ACM Transactions on Computer Graphics*, 2014. 3

[24] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023. 5, 7

[25] Jiajie Li, Jian Wang, Chen Wang, and Jinjun Xiong. Image harmonization with diffusion model, 2023. 2

[26] Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. Gligen: Open-set grounded text-to-image generation. In *CVPR*, pages 22511–22521, 2023. 2

[27] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *ICCV*, 2023. 2, 3, 6, 7

[28] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models, 2022. 2

[29] Lu Mi, Abhijit Kundu, David Ross, Frank Dellaert, Noah Snavely, and Alireza Fathi. im2nerf: Image to neural radiance field in the wild, 2022. 8

[30] Oscar Michel, Anand Bhattad, Eli VanderBilt, Ranjay Krishna, Aniruddha Kembhavi, and Tanmay Gupta. Object 3dit: Language-guided 3d-aware image editing. *arXiv preprint arXiv:2307.11073*, 2023. 1, 2, 3, 6, 7

[31] 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. 2, 4

[32] Y. Mori, H. Takahashi, and R. Oka. Image-to-word transformation based on dividing and vector quantizing images with words. In *MISRM'99 First International Workshop on Multimedia Intelligent Storage and Retrieval Management*, 1999. 2

[33] Chong Mou, Xintao Wang, Jiechong Song, Ying Shan, and Jian Zhang. Dragondiffusion: Enabling drag-style manipulation on diffusion models. *arXiv preprint arXiv:2307.02421*, 2023. 2

[34] Chong Mou, Xintao Wang, Liangbin Xie, Yanze Wu, Jian Zhang, Zhongang Qi, Ying Shan, and Xiaohu Qie. T2i-adapter: Learning adapters to dig out more controllable

ability for text-to-image diffusion models. *arXiv preprint arXiv:2302.08453*, 2023. 2

[35] Thu Nguyen-Phuoc, Christian Richardt, Long Mai, Yong-Liang Yang, and Niloy Mitra. Blockgan: Learning 3d object-aware scene representations from unlabelled images. In *NeurIPS*, 2020. 3

[36] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021. 2

[37] Simon Niklaus, Long Mai, Jimei Yang, and Feng Liu. 3d ken burns effect from a single image. *ACM Transactions on Computer Graphics*, 2019. 3

[38] 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

[39] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *CVPR*, 2019. 2

[40] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *ICCV*, 2021. 2

[41] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, pages 577–582. 2023. 5

[42] Ryan Po and Gordon Wetzstein. Compositional 3d scene generation using locally conditioned diffusion. *arXiv preprint arXiv:2303.12218*, 2023. 3

[43] Ariadna Quattoni, Michael Collins, and Trevor Darrell. Learning visual representations using images with captions. In *CVPR*, 2007. 2

[44] 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. In *Int. Conf. Machine Learning*, 2021. 2

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

[46] 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, 4

[47] Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–10, 2022. 1

[48] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022. 1, 6

[49] Meng-Li Shih, Shih-Yang Su, Johannes Kopf, and Jia-Bin Huang. 3d photography using context-aware layered depth inpainting. In *CVPR*, 2020. 3

[50] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 3

[51] 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. 2, 6, 7

[52] Nitish Srivastava and Russ R Salakhutdinov. Multimodal learning with deep boltzmann machines. In *NeurIPS*, 2012. 2

[53] 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. *arXiv preprint arXiv:2109.07161*, 2021. 5, 6

[54] Zhizhong Wang, Lei Zhao, and Wei Xing. Stylediffusion: Controllable disentangled style transfer via diffusion models, 2023. 2

[55] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015. 6

[56] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, Ben Hutchinson, Wei Han, Zarana Parekh, Xin Li, Han Zhang, Jason Baldridge, and Yonghui Wu. Scaling autoregressive models for content-rich text-to-image generation. *Transactions on Machine Learning Research*, 2022. 2

[57] Yu Zeng, Zhe Lin, Jianming Zhang, Qing Liu, John Collomosse, Jason Kuen, and M. Patel, Vishal. Scenecomposer: Any-level semantic image synthesis. 2023. 2

[58] 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. 1, 2, 3

[59] 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. 7

[60] Youyi Zheng, Xiang Chen, Ming-Ming Cheng, Kun Zhou, Shi-Min Hu, and Niloy J. Mitra. Interactive images: Cuboid proxies for smart image manipulation. *ACM Transactions on Computer Graphics*, 2012. 2, 3