3D-Fixup: Advancing Photo Editing with 3D Priors

YEN-CHI CHENG*, University of Illinois Urbana-Champaign, USA
KRISHNA KUMAR SINGH, Adobe Research, USA
JAE SHIN YOON, Adobe Research, USA
ALEXANDER SCHWING, University of Illinois Urbana-Champaign, USA
LIANG-YAN GUI, University of Illinois Urbana-Champaign, USA
MATHEUS GADELHA, Adobe Research, USA
PAUL GUERRERO, Adobe Research, UK
NANXUAN ZHAO, Adobe Research, USA

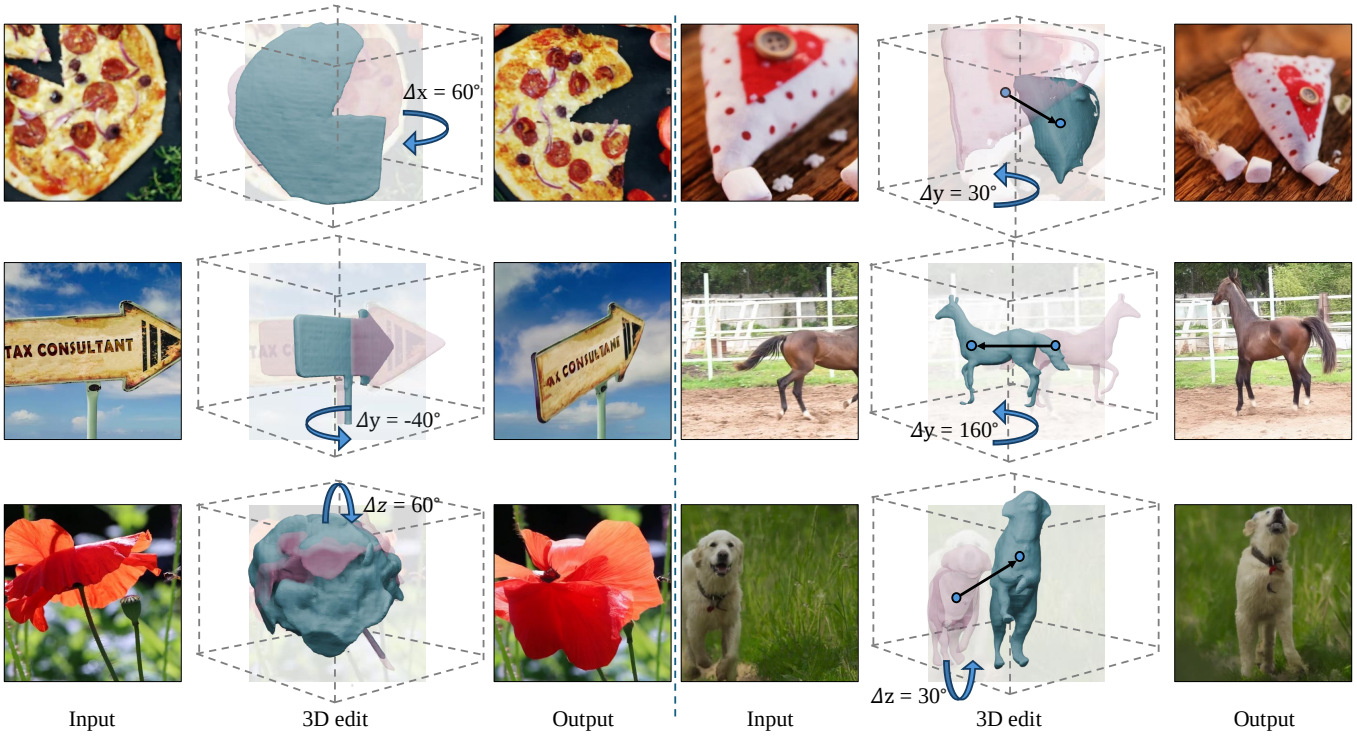

Fig. 1. **3D-aware photo editing. Given a source image with user-specified 3D transformations, our model generates a new image that follows the user's edit while preserving the input identity. The 3D edit is visualized via the transformation between the original mesh (pink) and the edited mesh (cyan).**

Despite significant advances in modeling image priors via diffusion models, 3D-aware image editing remains challenging, in part because the object is only specified via a single image. To tackle this challenge, we propose 3D-Fixup, a new framework for editing 2D images guided by learned 3D priors. The framework supports difficult editing situations such as object translation and 3D rotation. To achieve this, we leverage a training-based approach that harnesses the generative power of diffusion models. As video data naturally encodes real-world physical dynamics, we turn to video data for generating training data pairs, i.e., a source and a target frame. Rather than relying solely on a single trained model to infer transformations between source

*Work was done while Yen-Chi was an intern at Adobe Research.

Authors' Contact Information: Yen-Chi Cheng, ycheng3@illinois.edu, University of Illinois Urbana-Champaign, Urbana, Illinois, USA; Krishna Kumar Singh, krishsin@adobe.com, Adobe Research, San Jose, California, USA; Jae Shin Yoon, jaeyoon@adobe.com, Adobe Research, San Jose, California, USA; Alexander Schwing, aschwing@illinois.edu, University of Illinois Urbana-Champaign, Urbana, Illinois, USA; Liang-Yan Gui, lgui@illinois.edu, University of Illinois Urbana-Champaign, Urbana, Illinois, USA; Matheus Gadelha, gadelha@adobe.com, Adobe Research, San Jose, California, USA; Paul Guerrero, guerrero@adobe.com, Adobe Research, London, UK; Nanxuan Zhao, nanxuanz@adobe.com, Adobe Research, San Jose, California, USA.

ACM XXXX-XXXX/2025/5-ART
https://doi.org/10.1145/nnnnnnn.nnnnnnn

and target frames, we incorporate 3D guidance from an Image-to-3D model, which bridges this challenging task by explicitly projecting 2D information into 3D space. We design a data generation pipeline to ensure high-quality 3D guidance throughout training. Results show that by integrating these 3D priors, 3D-Fixup effectively supports complex, identity coherent 3D-aware edits, achieving high-quality results and advancing the application of diffusion models in realistic image manipulation. The code is provided at https://3dfixup.github.io/.

CCS Concepts: • **Computing methodologies** → **Computer vision**.

Additional Key Words and Phrases: Image editing, 3D, Diffusion Model

ACM Reference Format:
Yen-Chi Cheng, Krishna Kumar Singh, Jae Shin Yoon, Alexander Schwing, Liang-Yan Gui, Matheus Gadelha, Paul Guerrero, and Nanxuan Zhao. 2025. 3D-Fixup: Advancing Photo Editing with 3D Priors. 1, 1 (May 2025), 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Generative image editing is a growing research field that promises intuitive edits of objects in images, even if information about these objects or the scene that contains them is incomplete. For example, moving an object from one side of the image to the other, or rotating it as shown in Figure 1, may require knowledge about lighting, shadows, and occluded parts of the scene that are not available in the image. For these edits, generative models can hallucinate the missing information to obtain a plausible result. Current generative editing methods focus on appearance edits or 2D edits of image patches. However the objects we typically manipulate in images are projections of 3D objects. Some natural edits for 3D objects, like out-of-plane rotations or 3D translations, are thus not possible in most current approaches. However, 3D editing is crucial in applications such as e-commerce, where a 3D object may need to be shown from multiple angles, or digital media production, where it gives artists the ability to re-configure or relight a 3D scene shown in an image, without having to explicitly reconstruct the entire 3D scene, simplifying the creative process. By developing such a 3D editing method on natural images, we aim to bridge the gap between 2D and 3D workflows, making realistic edits more accessible for real-world applications.

The current challenge for 3D editing of objects in images lies in maintaining consistent object appearance across different angles and lighting conditions, which is essential for creating seamless edits. Existing approaches have attempted to address these issues using either optimization or feed-forward deep net techniques. Optimization-based approaches, such as Image Sculpting [Yenphraphai et al. 2024], begin by constructing a rough 3D shape of the object, followed by directly editing the 3D shape, and finally performing refinement to obtain the final edits. While this method achieves high-quality results, it is computationally intensive and slow, limiting its practical applications. In contrast, feed-forward approaches like 3DIT [Michel et al. 2024] and Magic Fixup [Alzayer et al. 2024] leverage conditional diffusion models to guide the editing process, making them relatively fast and efficient. However, these methods are primarily limited by their dependence on 2D data and synthetic training sets, which either lack depth and spatial understanding or real-world data understanding. Besides, the reliance on text prompts restricts the precision and granularity of the user control, often leading to outputs that may diverge from the user's intention.

To address these challenges, we propose a feed-forward method that utilizes real-world video data enriched with 3D priors, allowing for realistic 3D-aware editing of objects in natural images. We design a novel data generation pipeline to overcome the challenge of collecting large-scale 3D-aware image editing datasets in real-world scenarios. Our pipeline generates training data by leveraging 3D transformations estimated between frames in a video, and combines those with the priors obtained from an image-to-3D model. This intermediate 3D guidance serves as a crucial bridge, enabling the model to learn 3D-aware editing without requiring explicit 3D annotations for every frame. By utilizing both the dynamic information from videos and the structural insights provided by 3D priors, our approach captures real-world physical dynamics while facilitating fine-grained control over edits. This innovative design allows the model to generalize effectively to natural scenes, bridging the gap between synthetic and real-world applications. In Figure 1, we show some 3D-aware edits performed by our approach which allows fine-grained 3D user control while preserving object identity.

Our contributions are threefold: (1) we develop a novel data pipeline for generating 3D-aware training data from real-world videos, bridging 2D inputs with 3D editing capabilities; (2) we design an efficient feed-forward model that performs precise 3D editing on natural images using this 3D guidance; and (3) we conduct extensive evaluations, demonstrating that our method achieves realistic 3D edits and outperforms state-of-the-art approaches.

2 Related Work

2.1 3D-Aware Generative Image Editing

3D-aware image editing methods provide 3D control for image objects while maintaining consistency w.r.t. object identity, pose, and lighting during editing. Object3DIT [Michel et al. 2024] is one of the earliest 3D-aware editing methods, directly operating on 3D transformation parameters as a condition in a feed-forward deep-net architecture. The method is however limited by its small synthetic training set, reducing its generality and the precision of its edits. Diffusion Handles [Pandey et al. 2024] and GeoDiffuser [Sajnani et al. 2024] are recent approaches that are general and precise, but use inference-time optimization to align the output to a particular edit, limiting robustness and inference speed. The space of supported 3D transformations is somewhat limited since those techniques make no attempt in explicitly reasoning about unseen parts of objects. In contrast, Image Sculpting [Yenphraphai et al. 2024] leverages off-the-shelf single-view reconstruction models to enable impressive 3D-aware editing results, but also requires a computationally demanding inference-time optimization. Unlike prior approaches, our work focuses on fine-tuning large image diffusion models specifically for this task. This allows our method to exhibit remarkable robustness to challenging editing operations that could not be performed with existing baselines (see Figure 8). Additionally, no inference-time optimization is needed, allowing for fast evaluation.

2.2 Generative Image Editing

Drag-based methods have emerged as a prominent paradigm for interactive image editing, offering users precise control over object movement and transformation. Early methods like DragGAN [Pan et al. 2023] utilized GANs for point-to-point dragging but faced challenges with generalization and editing quality. More recent methods, such as DragDiffusion [Mou et al. 2023], InstantDrag [Shin et al. 2024], and EasyDrag [Hou et al. 2024], extend this concept to diffusion models, leveraging fine-tuned or reference-guided approaches to enhance photorealism. DragonDiffusion [Mou et al. 2023] stands out by avoiding fine-tuning and employing energy functions with visual cross-attention, enabling diverse editing tasks within and across images. DiffEditor [Mou et al. 2024] and DiffUHaul [Avrahami et al. 2024] further refine drag-style editing, addressing challenges like entanglement and enhancing consistency in dragging results. For articulated object interactions, DragAPart [Li et al. 2025] focuses on part-level motion understanding, allowing edits like opening drawers or repositioning parts. In contrast, generative editing methods that do not rely on drag-based interactions offer alternative workflows for tasks like object insertion, removal, and repositioning. ObjectDrop [Winter et al. 2024] models the effects of objects on scenes using counterfactual supervision, enabling realistic object manipulation. Meanwhile, SEELE [Wang et al. 2024a] formulates subject repositioning as a prompt-guided inpainting task, preserving image fidelity while offering precise spatial control. Magic Fixup [Alzayer et al. 2024] employs diffusion models to transform coarsely edited images into photorealistic outputs, leveraging video data to learn how objects adapt under various conditions. Similarly, ObjectStitch [Song et al. 2023] and IMPRINT [Song et al. 2024] focus on object compositing while preserving identity and harmonizing with the background, making them valuable for realistic image manipulation. However, unlike our approach, none of the methods in this paragraph benefit from a 3D-aware prior or provide controls to support 3D-aware edits like out-of-plane rotations or 3D translations.

2.3 Image-to-video with motion control

Image-to-video methods with motion control [Bahmani et al. 2025; Guo et al. 2025; Shi et al. 2024; Wang et al. 2024b] are related to generative image editing to some extent, as any generated frame could be taken as an edited image. However, edits are limited to motions that are plausible in a video and, to our best knowledge, none of the methods provide 3D control.

3 Approach

Our goal is 3D editing (e.g., out-of-plane rotation and translation) of a chosen object within an image. Existing 3D editing approaches that use inference-time optimization [Pandey et al. 2024; Yenphraphai et al. 2024] suffer from excessive inference times, making them impractical in real-world applications. In contrast, feedforward approaches [Michel et al. 2024] suffer from a lack of high-quality training data, limiting generality and control. We propose a feedforward 3D editing method that offers precise control and good generality. The editing workflow is illustrated in Figure 2. As shown, the 3D edits we consider include out-of-plane rotations and translations

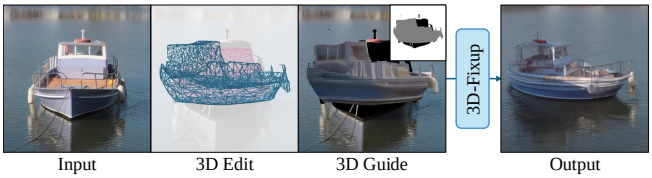

Fig. 2. *Inference pipeline.* We assume editing instructions (possibly converted from text prompts) are in the form of 3D operations like rotation and translation. Given a mask indicating the object to be edited, we first perform image-to-3D [Xu et al. 2024] to reconstruct the mesh. We then apply the user's desired 3D edit to obtain the 3D guidance. Here the 3D edit is visualized as the transformation between original mesh (pink wire-frame) and the edited mesh (cyan wire-frame). Finally, the model outputs the 3D aware editing result.

that change the perspective of the object. Formally, given a source image I_{src}, the user selects the object to be modified. The selection is represented via the mask M_{src}, which we use to obtain a rough 3D reconstruction. The user then performs a 3D edit of the rough 3D reconstruction. Upon rendering the modified 3D reconstruction we obtain the 3D guidance $I_{\text{guide}} \in \mathbb{R}^{H\times W\times 3}$, which is used to generate the desired editing result.

To provide the necessary supervision for training the feedforward model, i.e., to obtain a source image I_{src}, a guidance image I_{guide}, and a ground truth target image I_{tgt}, we construct a new dataset derived from videos. For this, we develop the data processing pipeline that we describe in Section 3.1. As videos naturally capture 3D motion as well as variations in lighting and background conditions, they offer a rich source of real-world data. By integrating this dataset into the training process, our method learns to handle complex 3D transformations while ensuring photorealism and maintaining the fidelity of the edited subject.

Using this dataset, we fine-tune a pretrained diffusion model conditioned on 1) the edited guidance image I_{guide}, and 2) the source image I_{src}. Importantly, unlike prior 2D editing methods, the guidance image is obtained by 3D-transforming a full 3D reconstruction of the chosen object in the image. We describe architecture and training setup of this model in Section 3.2.

3.1 Constructing the Dataset from Videos

Given a video, we create data pairs by sampling two frames, a source image I_{src} and a target image I_{tgt}. We use both images to compute a guidance image I_{guide}. Figure 3 provides an overview of our data processing pipeline. We discuss details next.

Flow for sampling the source and target image. We compute optical flow across all frames in a given video. If the accumulated flow across the entire video is too small, we discard the video. If the accumulated flow exceeds a threshold, we sample two frames from the video clip which we refer to as I_{src} and I_{tgt}.

Obtaining mask for the main object. We use Grounded-SAM [Ren et al. 2024] to obtain object masks and filter cases with privacy, aesthetic, or insignificant 3D transformation issues. Notice that Grounded-SAM struggles to detect occluded or unusually shaped instances, which are rare. To automatically identify the main object,

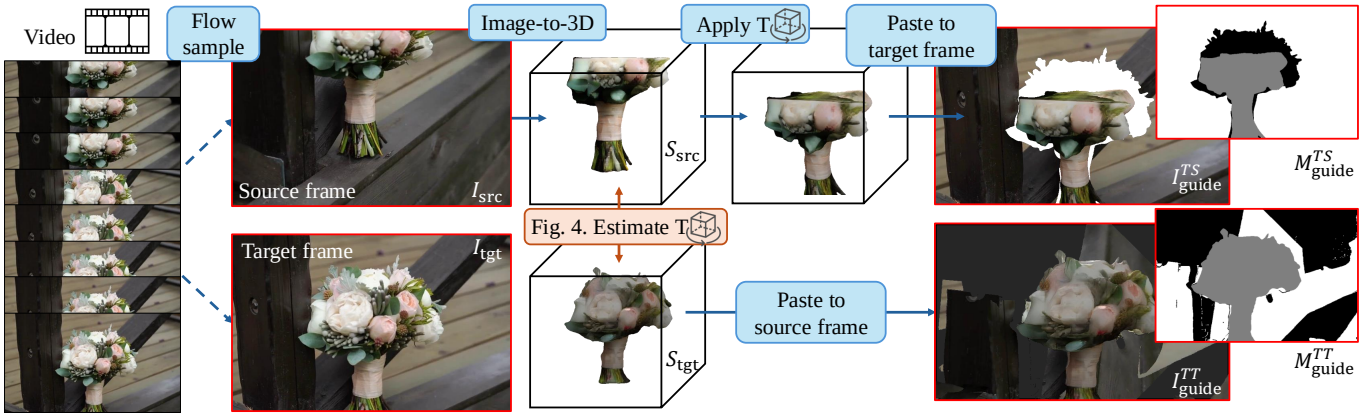

Fig. 3. *Data pipeline: Overview.* Given a video, we sample two frames, the source frame I_{src} and the target frame I_{tgt}, using optical flow as a cue: we discard videos where the flow indicates little motion through the entire clip. Using Image-to-3D methods, we reconstruct a mesh for the desired object for both frames. We then estimate the 3D transformation $\mathbf{T}$ (see Figure 4) between the source frame mesh and the target frame mesh. Availability of the transformation $\mathbf{T}$ enables two ways to create the training data: (1) in "Transform Source", we paste the rendering of the transformed source mesh onto the target frame; (2) in "Transform Target", we paste the rendering of the target mesh onto the source frame. Data examples are shown in Figure 5.

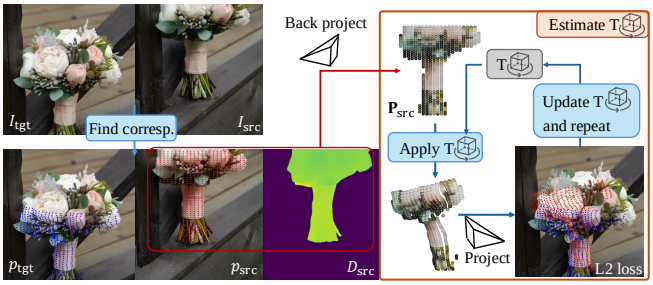

Fig. 4. *Data pipeline: Estimation of the 3D transformation* $\mathbf{T}$. We estimate the 3D transformation $\mathbf{T}$ by leveraging correspondences between the source frame and the target frame. Given frames between the source frame and the target frame, we first perform tracking to obtain corresponding points. We then initialize the parameters for the 3D transformation $\mathbf{T}$ and use an optimization to improve $\mathbf{T}$: (1) We unproject the 2D correspondences on the source frame to 3D pointclouds and apply the current $\mathbf{T}$ to transform points to the target image; (2) we project points back to 2D and compare via an L2 loss with the 2D correspondences of the target frame.

we define an "instance score" that prioritizes centrally located objects that occupy significant portions of the frame. The score is the weighted sum of inverted border score S_b and area score S_a across the video. They are calculated by

$$S_b = 1 - \frac{p^{\text{inst}}_{\text{border}}}{p_{\text{border}}}, \quad S_a = \frac{p_{\text{inst}}}{p_{\text{image}}}, \tag{1}$$

where p_{border} is the number of border pixels in the whole image, $p^{\text{inst}}_{\text{border}}$ is the number of instance pixels which touch the border, p_{inst} is the total number of pixels of the instance, and p_{image} is the number of pixels of the whole image. The instance score is $S_{\text{inst}} = 0.6 * S_b + 0.4 * S_a$. We select the highest-scoring instance.

Image to 3D reconstruction. Given the source image I_{src} and the target image I_{tgt}, we perform image to 3D reconstruction using InstantMesh [Xu et al. 2024]. Specifically, given the foreground masks $M_{\text{src}} \in \mathbb{R}^{H \times W}$ and $M_{\text{tgt}} \in \mathbb{R}^{H \times W}$, we first crop the foreground object and pad the image to a square while ensuring that the object is centered. InstantMesh then employs Zero123++ [Shi et al. 2023] to generate multiview images. Subsequently, InstantMesh operates on the multiview images to compute the reconstructed meshes S_{src} and S_{tgt} for the source and target images.

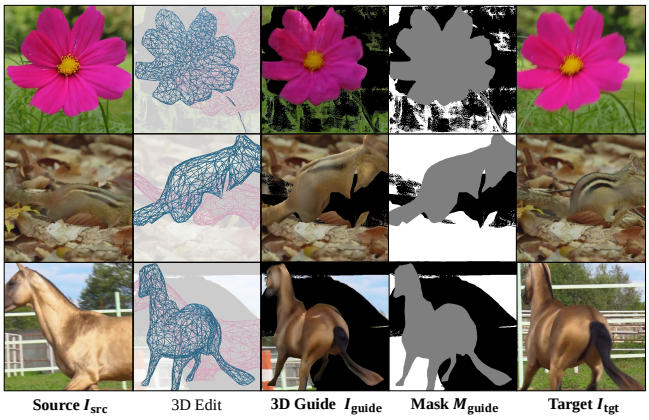

Fig. 5. *Examples of the training data.* Given a video, we use the steps described in Figure 3 to obtain the training data, i.e., the source image I_{src} and the target image I_{tgt}. The guidance image is obtained via the developed data pipeline. The mask has three values: 0 indicates the hole created by the coarse edit and the model needs to inpaint by looking at the details of the reference; 0.5 refers to the rendering of the object; and 1.0 denotes the original background.

Estimating 3D transformation with tracking. To obtain a coarse edit in an automated manner, we use the object meshes for the source and target image to estimate a 3D transformation. This process is illustrated in Figure 4. Concretely, to estimate the 3D transformation $\mathbf{T}$ between the object in the source image I_{src} and the target image I_{tgt}, we first compute N correspondences p_{src} and p_{tgt} along with their visibility maps v_{src} and v_{tgt}, using SpaTracker [Xiao

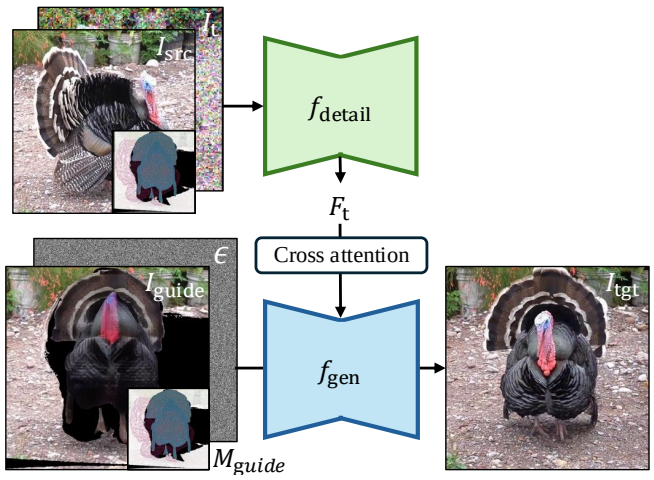

Fig. 6. *Overview of the training pipeline.* We develop a conditional diffusion model for 3D-aware image editing. It consists of two networks: f_{gen} and f_{detail}. During training, given the inputs—target frame I_{tgt}, 3D guidance I_{guide}, mask M_{guide}, and detail feature F_t—f_{gen} learns the reverse diffusion process to predict the noise ϵ and reconstruct I_{tgt}. To better preserve identity and fine-grained details from the source image I_{src}, f_{detail} takes as input the source image I_{src}, its noisy counterpart I_t, and the mask M_{guide}, and extracts detail features F_t. We apply cross-attention between F_t and the intermediate features of f_{gen} to incorporate content and details from I_{src} during the reverse diffusion process.

et al. 2024]. The input to the tracking model is the video segment containing the two frames I_{src} and I_{tgt}, along with the source mask M_{src}. Depth maps D_{src} and D_{tgt} are rendered from the meshes S_{src} and S_{tgt}, respectively.

Using the depth maps and correspondences, we backproject the 2D points into 3D space:

$$\mathbf{P}_{\text{src}} = \Pi^{-1}(p_{\text{src}}, D_{\text{src}}, \mathbf{K}), \tag{2}$$

$$\mathbf{P}_{\text{tgt}} = \Pi^{-1}(p_{\text{tgt}}, D_{\text{tgt}}, \mathbf{K}), \tag{3}$$

where Π^{-1} denotes the unprojection operation and $\mathbf{K} \in \mathbb{R}^{3\times3}$ is the intrinsic camera matrix. The 3D points $\mathbf{P} \in \mathbb{R}^3$ are calculated as:

$$\mathbf{P} = D(p) \cdot \mathbf{K}^{-1} \begin{bmatrix} p_x \\ p_y \\ 1 \end{bmatrix}, \tag{4}$$

where $D(p)$ is the depth at pixel $p = (p_x, p_y)$.

The translation component of $\mathbf{T}$ is initialized by calculating the centroid offset between the two point clouds:

$$\mathbf{t} = \mathbf{c}_{\text{tgt}} - \mathbf{c}_{\text{src}}, \quad \mathbf{c} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{P}_i. \tag{5}$$

To initialize the rotation, a coarse grid search is performed jointly over the X, Y, and Z axes, using a step size of 10° within the range $[0^\circ, 360^\circ]$. The transformation $\mathbf{T}$ is optimized by minimizing the re-projection loss:

$$\mathcal{L}_{\text{reproj}} = \sum_{i=1}^{N} \|p_{\text{tgt},i} - \Pi(\mathbf{T}\mathbf{P}_{\text{src},i}, \mathbf{K})\|_2^2, \tag{6}$$

where Π is the projection operation:

$$\Pi(\mathbf{P}, \mathbf{K}) = \begin{bmatrix}\mathbf{K} & \mathbf{0}\end{bmatrix} \mathbf{P}. \tag{7}$$

The optimization iteratively adjusts $\mathbf{T}$ until convergence. Eq. 6 assumes $\mathbf{T}$ to be rigid transformations. However, for non-rigid transformations, Eq. 6 finds a close rigid approximation. During training, the model learns from non-rigid transformation as ground truth while using a rigid approximation in guidance I_{guide}. This discrepancy is often desirable, as it encourages the model to adapt non-rigidly, ensuring the edited object fits naturally into its new context. For example, when rotating the dog in Fig. 1 or the horse in Fig. 9, subtle posture adjustments, such as foot placement, help the resulting scene remain plausible.

Creating the training data. With the optimized transformation $\mathbf{T}$ computed, we have two settings to obtain the guidance image and the editing mask. We refer to the first setting as "Transform Source" (TS): the estimated 3D transformation $\mathbf{T}$ is applied to the source mesh S_{src}. The transformed mesh is rendered and pasted onto the target image I_{tgt} based on the target mask M_{tgt} to form the guidance image I^{TS}_{guide}. The editing mask M^{TS}_{guide} has 1.0 for the static background, 0.5 for the rendered regions, and 0.0 for the holes created by cropped the object in the I_{tgt}. We refer to the second setting as "Transform Target" (TT): we transform and render the target mesh S_{tgt} onto the source frame I_{src} to obtain the guidance image I^{TT}_{guide}. The background is warped based on the flow computed between the source frame and the target frame following Alzayer *et al.* [Alzayer et al. 2024]. M^{TT}_{guide} is similar to M^{TS}_{guide} for the background (1.0) and rendered regions (0.5), while the holes (0.0) indicated the warped regions. Thus, the final training tuple for "Transform Source" and "Transform Target" are $(I_{\text{source}}, I^{TS}_{\text{guide}}, M^{TS}_{\text{guide}}, I_{\text{target}})$, and $(I_{\text{source}}, I^{TT}_{\text{guide}}, M^{TT}_{\text{guide}}, I_{\text{target}})$ respectively. We sample data pair randomly from TT or TS when training the model. Both the editing masks M^{TS}_{guide} and M^{TT}_{guide} guide the model to inpaint missing regions (0.0), enhance 3D-transformed areas (0.5), and preserve intact content (1.0). Examples of the collected data are illustrated in Figure 5.

Our dataset is sourced from 8 million licensed videos. We discard videos exceeding 500 frames due to high compute costs, reducing the dataset to 2 million. After filtering cases depicting humans (privacy and aesthetics issues), illustrations and drone videos (insignificant 3D transformations), and videos without detected entities, we retain 375k clips. Further motion-based flow filtering reduces this to 50k videos. Note that the proposed data pipeline is general and applicable to any video dataset. Processing each video takes ~95.7s and requires 14.41GB of memory on a single A100 GPU. The pipeline includes flow filtering, mask extraction, image-to-3D, tracking, and 3D estimation, with videos averaging 50–500 frames.

3.2 3D Editing with a Diffusion Model

Our diffusion model aims to generate realistic images that complete the 3D edit specified by the guidance image I_{guide}. Hence, the structure of the image should be preserved as outlined in the guidance. For regions indicated by the mask M_{guide}, the model performs the following operations: inpainting missing regions ($M_{\text{guide}} = 0.0$),

modifying ambiguous regions ($M_{\text{guide}} = 0.5$), and preserving content and identity in confident regions ($M_{\text{guide}} = 1.0$).

An overview of our architecture is provided in Figure 6. We base our architecture on MagicFixup [Alzayer et al. 2024] and adopt two networks in our pipeline: a generator for generating the output image f_{gen} and an extractor for extracting details f_{detail} from the source image I_{src}. We use those networks in a diffusion process which operates as follows: the diffusion forward process progressively adds Gaussian noise to an image, yielding a sequence of intermediate states:

$$x_t \sim \mathcal{N}(\sqrt{\alpha_t}x_{t-1}, (1-\alpha_t)\mathbf{I}), \tag{8}$$

which gradually resemble a standard Gaussian as diffusion time t increases towards a final timestep T. Here α_t defines the noise schedule at diffusion time t. The model learns the reverse process: a standard Gaussian input $x_T \sim \mathcal{N}(0, \mathbf{I})$ is gradually denoised toward intermediate states x_t before eventually arriving at the final estimated image x_0:

$$x_{t-1} = f_{\text{gen}}(x_t, I_{\text{guide}}, M_{\text{guide}}, F_t, t; \theta). \tag{9}$$

At each denoising step t, the model is conditioned on the guidance I_{guide}, mask M_{guide}, and the features F_t extracted by the extractor f_{detail}. We initialize the process from a noisy version of the guidance image, i.e., we use

$$x_T = \sqrt{\bar{\alpha}_T}I_{\text{guide}} + \sqrt{1-\bar{\alpha}_T}\epsilon, \tag{10}$$

where $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ and $\bar{\alpha}_T = \prod_{s=1}^{T} \alpha_s$. This initialization ensures alignment with the guidance while bridging the gap between training and inference domains. The extractor f_{detail} operates on the source image I_{src} for referencing and with the goal to preserve fine-grained details and object identity. To ensure compatibility with the diffusion process, we add noise to the source image:

$$I_t = \sqrt{\bar{\alpha}_t}I_{\text{src}} + \sqrt{1-\bar{\alpha}_t}\epsilon, \quad \epsilon \sim \mathcal{N}(0, \mathbf{I}), \quad \bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s. \tag{11}$$

From this noisy source image, the extractor computes the feature

$$F_t = [f_t^1, \ldots, f_t^n] = f_{\text{detail}}([I_t, I_{\text{src}}, M_{\text{guide}}]; t), \tag{12}$$

for each self-attention block. Here n is the number of attention blocks, and $[\cdot]$ denotes concatenation along the channel dimension. These features are injected into the model through cross-attention layers, enabling details preservation from the source image to outputs during synthesis. For each layer i, at step t, the features F_t extracted by the extractor f_{detail} serve as keys K and values V, while the features of the generator f_{gen} $[g_t^1, \ldots, g_t^n]$ act as queries Q. Formally, we have

$$A_t^i = \text{softmax}(\frac{Q_t^i K_t^{i\top}}{\sqrt{d}}), \quad \text{and} \quad G_t^i = A_t^i V_t^i, \tag{13}$$

where Q_t^i, K_t^i, and V_t^i are query, key, and value projections of the respective features. This mechanism ensures that fine details from I_{src} are faithfully transferred to the synthesized output.

During inference, user instructions (e.g., text prompts, drags on 3D objects) are converted into 3D transformations $\mathbf{T}$ (out-of-plane rotations and translations). Using InstantMesh [Xu et al. 2024], we perform image-to-3D reconstruction to generate a 3D mesh of the subject. Applying $\mathbf{T}$ to the mesh, we obtain the guidance for editing. The model uses this guidance I_{guide} along with the mask M_{guide} to produce the final output. Figure 2 illustrates this framework.

3.3 Implementation details

For fair comparisons, following the state-of-the-art Magic-Fixup [Alzayer et al. 2024], we train our model starting from pretrained weights of Stable Diffusion 1.4. Training samples are drawn from data settings—TT, TS, MF—with probabilities (0.35, 0.35, 0.3), where MF is sampling from Magic Fixup's data. To encourage identity preservation, we drop the conditioning on I_{src} with a 0.2 probability, forcing the model to rely on I_{src}'s context. We train the model with a batch size of 8, using AdamW [Loshchilov 2017] and a learning rate of 1e−5 on 8 NVIDIA A100 GPUs for about two days. We use a linear diffusion noise schedule, with $\alpha_1 = 0.9999$, $\alpha_T = 0.98$, and $T = 1000$. We use DDIM for sampling with 100 steps during inference time. The images were all cropped to 512×512 for training.

4 Experiments

We evaluate the proposed method both qualitatively and quantitatively on a set of edits. For this, we curated a set of user edits to show the use cases of the model in practical applications. We also created a test dataset which contains large 3D transformations to validate the proposed method.

Dataset. We use stock video as our dataset for training and testing. The training data consists of around 50k data points which contains common objects with motions in the scene. We randomly sample diverse scenes and objects while maintaining a reasonable scale when constructing the test set.

Baseline. To validate the effectiveness of the proposed method, we compare to seven baselines: *Magic Fixup* [Alzayer et al. 2024], *Object 3DIT* [Michel et al. 2024], *Zero-1-to-3* [Liu et al. 2023], *InstantDrag* [Shin et al. 2024], *MOFA-Video* [Niu et al. 2024], *Blended LDM* [Avrahami et al. 2023], and finally *Instruct-pix2pix* [Brooks et al. 2023]. Please refer to the supplementary materials for details regarding the baselines.

Metrics. For quantitative evaluation, we use LPIPS [Zhang et al. 2018] and FID [Heusel et al. 2017] metrics. LPIPS measures the perceptual similarity to assess fidelity to the ground truth using a neural network such as AlexNet [Krizhevsky et al. 2012] or VGGNet [Simonyan and Zisserman 2014]. FID (Fréchet Inception Distance) evaluates the realism of generated images by comparing their distribution to that of real data.

4.1 Comparison with Baselines

We compare the proposed method with several state-of-the-art image-editing methods, which operate on different types of conditions, such as the 3D transform, a drag (point correspondence), and a text prompt. The results are shown in Figure 7. Similar to our approach, 3DIT [Michel et al. 2024] and Zero123 [Liu et al. 2023] use the 3D transform as condition. However, 3DIT fails to generate plausible results because of the domain gap between its synthetic training data and real-world images, while Zero123 struggles with identity preservation. For the dragging-based methods InstantDrag [Shin et al. 2024] and MOFA-Video [Niu et al. 2024], we

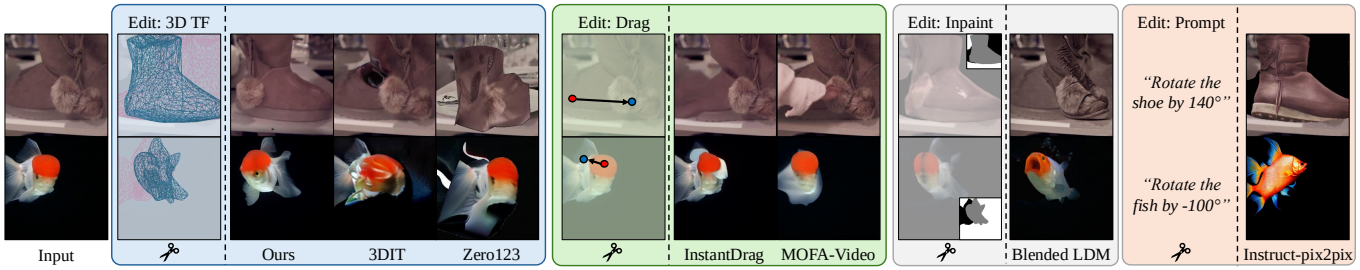

Fig. 7. **Comparison with baselines.** We compare several state of the art baselines with different kinds of conditions, such as 3D transforms, drags, inpainting masks, and text prompts. We can see that none of the baselines accurately follow the target 3D transform while preserving identity. Baselines that directly use 3D transforms suffer from a lack of good training data, and using other types of conditions makes it hard to unambiguously specify the 3D transform.

Table 1. **Quantitative comparison to baselines**. We compare to the baselines using the LPIPS and FID metrics. The result shows that the 3D editing of the proposed method is closer to the ground truth and real distribution.

Model	LPIPS ↓	FID (5k) ↓	FID (30k) ↓
Magic Fixup [Alzayer et al. 2024]	0.5776	174.6926	27.1542
3DIT [Michel et al. 2024]	0.4493	145.3389	23.8392
Zero123 [Liu et al. 2023]	0.6803	202.2304	44.8570
Instruct-pix2pix [Brooks et al. 2023]	0.7532	231.7562	73.3829
InsantDrag [Shin et al. 2024]	0.4810	163.6477	34.6370
Blended LDM [Avrahami et al. 2023]	0.5012	185.0291	41.3255
MOFA-Video [Niu et al. 2024]	0.3283	149.1247	20.9583
Ours	**0.2397**	**132.1145**	**13.0228**

Table 2. Quantitative results for training with different sets of training data.

Data setting	LPIPS ↓	FID ↓
Transform source	0.3874	151.6589
Transform source + MF	0.3321	145.3312
Transform source + Transform Target + MF	**0.2397**	**132.1145**

Table 3. **Ablation study of conditioning.** We evaluate the effect of different conditioning mechanisms with two mask configurations: (0.0,0.5,1.0) v.s. (0.0,1.0), and the impact of dropout of image features.

Configurations	Mask (0.0, 1.0)	Without dropout	**Ours**
FID ↓	17.5615	21.2870	**13.0228**

use the known correspondence between the source and transformed mesh to define an input drag. We find that drags are too ambiguous to clearly define a 3D transform and both methods struggle to interpret larger drags, such as the rotation of the goldfish or the shoes. Blended LDM [Avrahami et al. 2023] takes the guidance image and the mask as inputs and adopts Blended Diffusion [Avrahami et al. 2022] to refine the coarse edit, which does not preserve identity. Finally, Instruct-pix2pix [Brooks et al. 2023] is instructed by a text prompt, but suffers from its ambiguity. In contrast, our proposed method can generate high-quality edits for large 3D transformations while preserving identity.

We also present a comparison with Magic Fixup in Figure 9. The results demonstrate that our proposed method achieves more realistic images, benefiting from 3D-transformation-based guidance. For example, our method effectively handles pose changes, such as adjusting the camera's viewing direction or modifying the poses of subjects, as shown in the horse, jaguar, and cake examples. In contrast, Magic Fixup struggles with such edits.

4.2 3D Editing with Continuous Rotations

We also demonstrate that the proposed method can handle extensive 3D edits on common objects, as illustrated in Figure 8. In each scenario, we progressively increase the rotation from 0 to 180 degrees along the y-axis, applying it to the reconstructed mesh to generate the 3D-transformation-based guidance image. The results show that our method successfully deals with out-of-plane 3D rotation edits, from minor adjustments to substantial transformations, highlighting the model's 3D-awareness during editing.

4.3 Quantitative Comparison of 3D Editing

We compute metrics to evaluate the performance of methods as shown in Tab. 1. LPIPS is calculated for each model by measuring the similarity between its outputs and the ground-truth images. The final LPIPS score is obtained as the mean value across all pairs. FID assesses the realism of the generated results by comparing the distribution of the generated images to that of real video frames. The results demonstrate that the proposed method produces outputs that are highly realistic and align well with the real data as indicated by lower FID and LPIPS values. In terms of the detail preservation, we address the challenging task of hallucinating novel views and missing parts based on I_{guide}. Since it relies only on single-view image to generate unseen regions, preserving identity and fine details is inherently difficult. However, LPIPS in Table 1 shows that 3D-Fixup achieves better detail preservation than prior methods. Finally, we also compare the runtime with Image-sculpting [Yenphraphai et al. 2024], an optimization-based method for image editing. The runtime is $\sim 877s$ per sample, while ours can achieve $\sim 2s$ for 50 DDIM steps.

4.4 Ablation Study of Data Setting and Conditioning

We evaluate the importance of each data setting in Tab. 2. Using the setting with all the data yields the best performance in terms of FID and LPIPS. We also evaluate the effect of different conditioning mechanisms in Tab. 3. First, we consider different mask settings: $(0.0, 0.5, 1.0)$ vs. $(0.0, 1.0)$. Then we study the impact of dropout of image features for cross-attention. The results suggest that our conditioning outperforms other configurations.

5 Conclusion

We introduced 3D-Fixup, a workflow that tackles the problem of 3D-aware image editing. To make 3D-aware image editing efficient, we adopt a feedforward method. To train such a model, data is crucial. Since collecting data with corresponding 3D edits is time-consuming and expensive, we developed an automatic framework to collect suitable data from real-world videos. The resulting method bridges the gap between 2D image editing and 3D transformations, enabling realistic edits that preserve content identity while maintaining fidelity across various perspectives.

We demonstrated the effectiveness of our approach through extensive experiments, showcasing its ability to handle large out-of-plane rotations and translations, as well as challenging scenarios involving significant pose changes. Quantitative evaluations using LPIPS and FID metrics validate the realism and accuracy of our edits, outperforming state-of-the-art baselines such as Magic Fixup.

We found that intricate details, such as sprinkles on donuts or textures on clothing, are sometimes not preserved well, likely due to image encoder limitations. Additionally, 3D-Fixup produces suboptimal results when I_{guide} is of low quality. This occurs when the image-to-3D step performs poorly due to occlusion, incompleteness, or a suboptimally detected mask, which can be mitigated by outpainting masks/objects. Future work may explore extending the framework to handle more complex scenes with multiple objects and refining the underlying 3D priors to enhance generalization across diverse datasets.

Acknowledgments

Work supported in part by NSF grants 2008387, 2045586, 2106825, MRI 1725729, NIFA award 2020-67021-32799, and the Amazon-Illinois Center on AI for Interactive Conversational Experiences.

References

Hadi Alzayer, Zhihao Xia, Xuaner Zhang, Eli Shechtman, Jia-Bin Huang, and Michael Gharbi. 2024. Magic Fixup: Streamlining Photo Editing by Watching Dynamic Videos. *arXiv preprint arXiv:2403.13044* (2024).

Omri Avrahami, Ohad Fried, and Dani Lischinski. 2023. Blended latent diffusion. *ACM transactions on graphics (TOG)* 42, 4 (2023), 1–11.

Omri Avrahami, Rinon Gal, Gal Chechik, Ohad Fried, Dani Lischinski, Arash Vahdat, and Weili Nie. 2024. DiffUHaul: A Training-Free Method for Object Dragging in Images. *arXiv preprint arXiv:2406.01594* (2024).

Omri Avrahami, Dani Lischinski, and Ohad Fried. 2022. Blended diffusion for text-driven editing of natural images. In *CVPR*.

Sherwin Bahmani, Xian Liu, Wang Yifan, Ivan Skorokhodov, Victor Rong, Ziwei Liu, Xihui Liu, Jeong Joon Park, Sergey Tulyakov, Gordon Wetzstein, et al. 2025. Tc4d: Trajectory-conditioned text-to-4d generation. In *European Conference on Computer Vision*. Springer, 53–72.

Tim Brooks, Aleksander Holynski, and Alexei A Efros. 2023. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 18392–18402.

Yuwei Guo, Ceyuan Yang, Anyi Rao, Maneesh Agrawala, Dahua Lin, and Bo Dai. 2025. Sparsectrl: Adding sparse controls to text-to-video diffusion models. In *European Conference on Computer Vision*. Springer, 330–348.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems* 30 (2017).

Xingzhong Hou, Boxiao Liu, Yi Zhang, Jihao Liu, Yu Liu, and Haihang You. 2024. EasyDrag: Efficient Point-based Manipulation on Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8404–8413.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* 25 (2012).

Ruining Li, Chuanxia Zheng, Christian Rupprecht, and Andrea Vedaldi. 2025. Dragapart: Learning a part-level motion prior for articulated objects. In *European Conference on Computer Vision*. Springer, 165–183.

Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. 2023. Zero-1-to-3: Zero-shot one image to 3d object. In *Proceedings of the IEEE/CVF international conference on computer vision*. 9298–9309.

I Loshchilov. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101* (2017).

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

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

Chong Mou, Xintao Wang, Jiechong Song, Ying Shan, and Jian Zhang. 2024. Diffeditor: Boosting accuracy and flexibility on diffusion-based image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8488–8497.

Muyao Niu, Xiaodong Cun, Xintao Wang, Yong Zhang, Ying Shan, and Yinqiang Zheng. 2024. Mofa-video: Controllable image animation via generative motion field adaptions in frozen image-to-video diffusion model. In *European Conference on Computer Vision*. Springer, 111–128.

Xingang Pan, Ayush Tewari, Thomas Leimkühler, Lingjie Liu, Abhimitra Meka, and Christian Theobalt. 2023. Drag your gan: Interactive point-based manipulation on the generative image manifold. In *ACM SIGGRAPH 2023 Conference Proceedings*. 1–11.

Karran Pandey, Paul Guerrero, Metheus Gadelha, Yannick Hold-Geoffroy, Karan Singh, and Niloy J. Mitra. 2024. Diffusion Handles: Enabling 3D Edits for Diffusion Models by Lifting Activations to 3D. *CVPR* (2024).

Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, Zhaoyang Zeng, Hao Zhang, Feng Li, Jie Yang, Hongyang Li, Qing Jiang, and Lei Zhang. 2024. Grounded SAM: Assembling Open-World Models for Diverse Visual Tasks. arXiv:2401.14159 [cs.CV]

Rahul Sajnani, Jeroen Vanbaar, Jie Min, Kapil Katyal, and Srinath Sridhar. 2024. GeoDiffuser: Geometry-Based Image Editing with Diffusion Models. arXiv:2404.14403 [cs.CV]

Ruoxi Shi, Hansheng Chen, Zhuoyang Zhang, Minghua Liu, Chao Xu, Xinyue Wei, Linghao Chen, Chong Zeng, and Hao Su. 2023. Zero123++: a Single Image to Consistent Multi-view Diffusion Base Model. arXiv:2310.15110 [cs.CV]

Xiaoyu Shi, Zhaoyang Huang, Fu-Yun Wang, Weikang Bian, Dasong Li, Yi Zhang, Manyuan Zhang, Ka Chun Cheung, Simon See, Hongwei Qin, et al. 2024. Motion-i2v: Consistent and controllable image-to-video generation with explicit motion modeling. In *ACM SIGGRAPH 2024 Conference Papers*. 1–11.

Joonghyuk Shin, Daehyeon Choi, and Jaesik Park. 2024. InstantDrag: Improving Interactivity in Drag-based Image Editing. In *SIGGRAPH Asia 2024 Conference Papers*. 1–10.

Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

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

Yizhi Song, Zhifei Zhang, Zhe Lin, Scott Cohen, Brian Price, Jianming Zhang, Soo Ye Kim, He Zhang, Wei Xiong, and Daniel Aliaga. 2024. Imprint: Generative object compositing by learning identity-preserving representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8048–8058.

Jiawei Wang, Yuchen Zhang, Jiaxin Zou, Yan Zeng, Guoqiang Wei, Liping Yuan, and Hang Li. 2024b. Boximator: Generating rich and controllable motions for video synthesis. *arXiv preprint arXiv:2402.01566* (2024).

Yikai Wang, Chenjie Cao, Ke Fan, Qiaole Dong, Yifan Li, Xiangyang Xue, and Yanwei Fu. 2024a. Repositioning the Subject within Image. *arXiv preprint arXiv:2401.16861* (2024).

Daniel Winter, Matan Cohen, Shlomi Fruchter, Yael Pritch, Alex Rav-Acha, and Yedid Hoshen. 2024. ObjectDrop: Bootstrapping Counterfactuals for Photorealistic Object Removal and Insertion. *arXiv preprint arXiv:2403.18818* (2024).

Yuxi Xiao, Qianqian Wang, Shangzhan Zhang, Nan Xue, Sida Peng, Yujun Shen, and Xiaowei Zhou. 2024. SpatialTracker: Tracking Any 2D Pixels in 3D Space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 20406–20417.

Jiale Xu, Weihao Cheng, Yiming Gao, Xintao Wang, Shenghua Gao, and Ying Shan. 2024. Instantmesh: Efficient 3d mesh generation from a single image with sparse-view large reconstruction models. *arXiv preprint arXiv:2404.07191* (2024).

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

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 586–595.

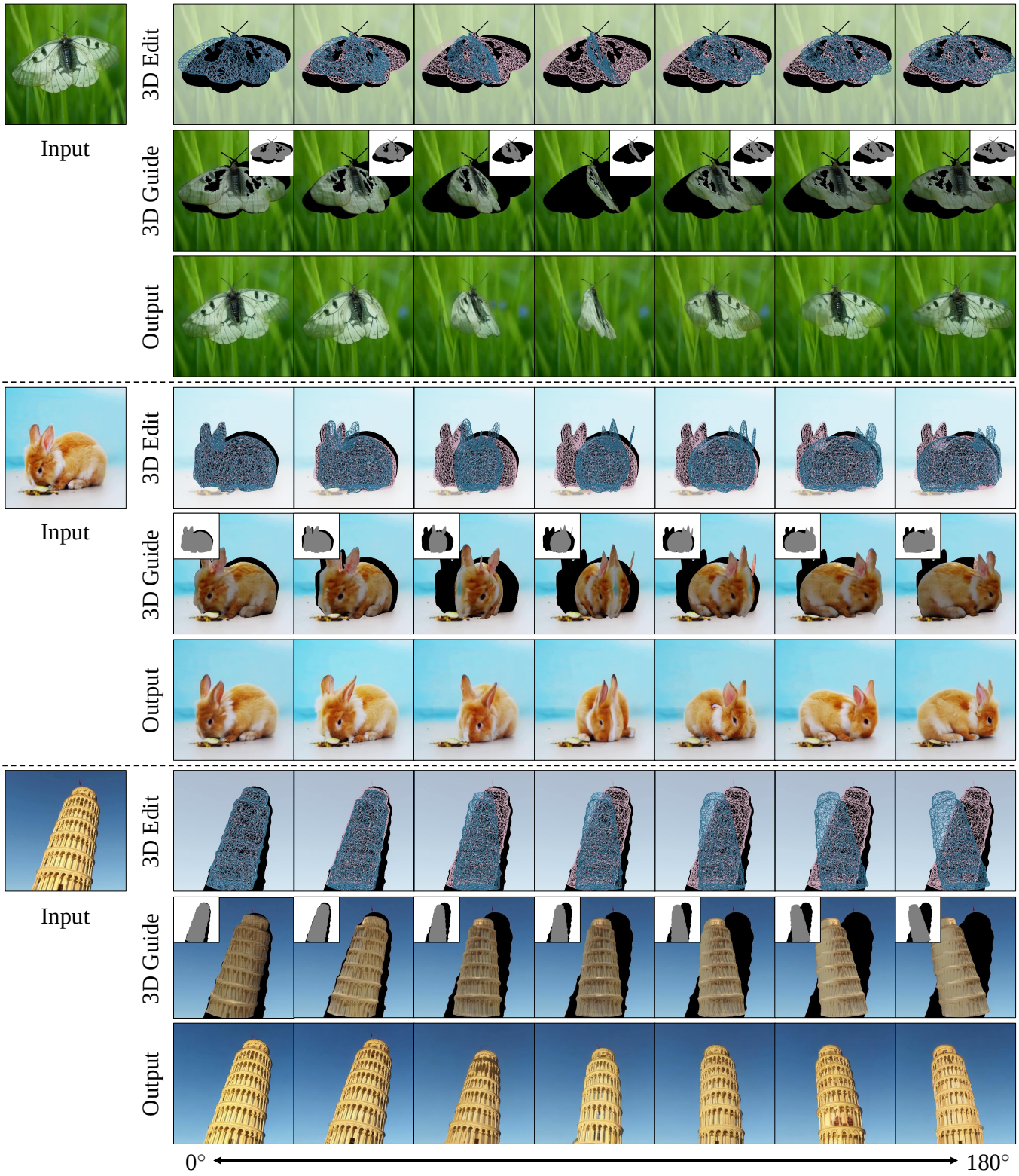

Fig. 8. **3D editing with continuous rotations.** We demonstrate that the proposed method can handle extensive 3D edits on common objects. In each scenario, we progressively increase the rotation from 0 to 180 degrees along the y-axis, applying it to the reconstructed mesh to generate the 3D-transformation-based image guidance. The results show that our method can handle large out-of-plane 3D rotations during editing.

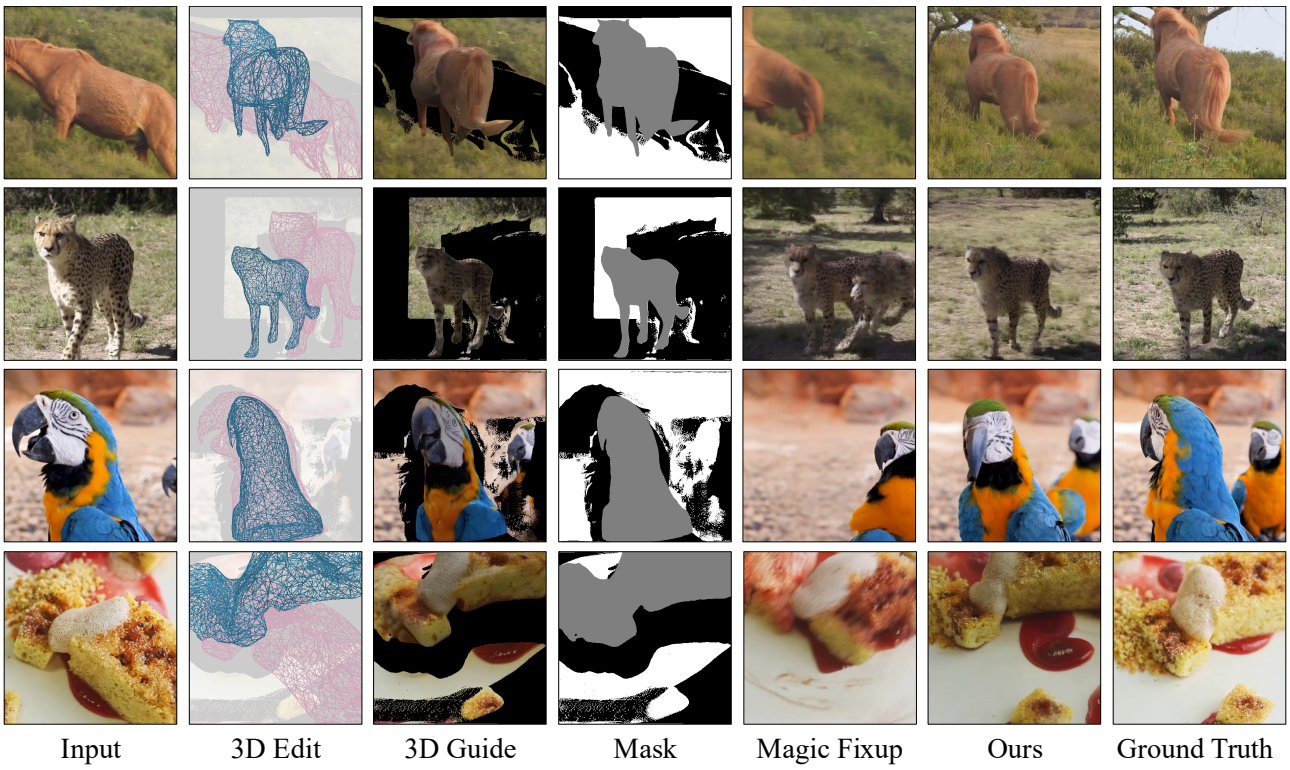

Fig. 9. **Comparison with Magic Fixup.** The results demonstrate that the proposed method achieves more realistic outputs by leveraging the 3D-transformation-based guidance. For instance, our method effectively handles pose changes, such as adjusting the camera's viewing direction for the cakes and jaguar, or modifying the poses of the horse and parrot.