Free-Form Scene Editor: Enabling Multi-Round Object Manipulation Like in a 3D Engine

Xincheng Shuai[1], **Zhenyuan Qin**[1], **Henghui Ding**[1*], **Dacheng Tao**[2]

[1]Institute of Big Data, College of Computer Science and Artificial Intelligence, Fudan University, China
[2]College of Computing and Data Science, Nanyang Technological University, Singapore
henghui.ding@gmail.com, dacheng.tao@gmail.com

Abstract

Recent advances in text-to-image (T2I) diffusion models have significantly improved semantic image editing, yet most methods fall short in performing 3D-aware object manipulation. In this work, we present *FFSE*, a 3D-aware autoregressive framework designed to enable intuitive, physically-consistent object editing directly on real-world images. Unlike previous approaches that either operate in image space or require slow and error-prone 3D reconstruction, *FFSE* models editing as a sequence of learned 3D transformations, allowing users to perform arbitrary manipulations, such as translation, scaling, and rotation, while preserving realistic background effects (e.g., shadows, reflections) and maintaining global scene consistency across multiple editing rounds. To support learning of multi-round 3D-aware object manipulation, we introduce *3DObjectEditor*, a hybrid dataset constructed from simulated editing sequences across diverse objects and scenes, enabling effective training under multi-round and dynamic conditions. Extensive experiments show that the proposed *FFSE* significantly outperforms existing methods in both single-round and multi-round 3D-aware editing scenarios.

Code — https://github.com/FudanCVL/FFSE
Extended version — https://arxiv.org/abs/2511.13713

Introduction

Image editing enables users to modify the visual content without requiring expertise in professional software. Recently, advanced methods (Shi et al. 2024; Mu et al. 2024) based on text-to-image (T2I) diffusion models (Rombach et al. 2022; Podell et al. 2023) enable powerful object-centric manipulation, including appearance (Hertz et al. 2023; Tumanyan et al. 2023; Brooks, Holynski, and Efros 2023) or shape modification (Shi et al. 2024; Ling et al. 2024). Although these methods are useful for semantic editing, 3D-aware object manipulation (Qin, Shuai, and Ding 2025) offers more flexibility in many scenarios, as shown in Fig. 1. By equipping models with 3D-aware editing ability, users can manipulate objects as if operating within a 3D engine (Unreal Engine 5 2022), allowing more intuitive and physically-consistent image modifications.

*Corresponding author.

Recently, a growing body of works (Michel et al. 2023; Yenphraphai et al. 2024; Wang et al. 2024; Pandey et al. 2024; Zhang et al. 2024; Jabri et al. 2024; Wu et al. 2024) explore 3D-aware image editing, but still struggle to produce satisfactory results. *Image-space* methods (Liu et al. 2023; Wu et al. 2024; Michel et al. 2023) typically learned 3D priors from established datasets. However, most of methods fail to accomplish comprehensive 3D operations and demonstrate poor generalization on real-world images. *3D-space* methods (Pandey et al. 2024; Yenphraphai et al. 2024; Zhao et al. 2025; Wang et al. 2024) reconstruct scene structure from a single image to support arbitrary 3D manipulations, but are time-consuming and sensitive to noisy geometry estimations (Rombach et al. 2022; Poole et al. 2023; Mildenhall et al. 2022; Liu et al. 2023). Moreover, most existing methods struggle to generate realistic background effects (*e.g.*, shadows, reflections) and often fail to maintain scene consistency across multi-round edits.

Several key challenges remain in achieving effective 3D-aware object manipulation: **1) Object effects.** Existing methods (Michel et al. 2023; Liu et al. 2023) support only limited and simple 3D operations, or often yield low-quality results in complex scenarios (Zhang et al. 2024; Yenphraphai et al. 2024; Pandey et al. 2024). **2) Background effects.** Current approaches struggle to infer realistic environmental interactions caused by object manipulations, such as shadows, reflections, and occlusions. **3) Multi-round editing.** Existing studies lack awareness of scene structure changes, resulting in inconsistencies after multiple edits. **4) User interface.** Previous works often rely on cumbersome inputs (Wu et al. 2024; Michel et al. 2023), or require time-consuming reconstruction process (Yenphraphai et al. 2024; Zhang et al. 2024).

To address the above challenges, we propose *Free-Form Scene Editor* (*FFSE*), an autoregressive generation model that is capable of handling diverse 3D operations in real-world images, akin to modern 3D engines such as *Unreal* (Unreal Engine 5 2022) and *Blender* (B. O. Community 2018). A straightforward way is to leverage existing video datasets (Bain et al. 2021; Zhou et al. 2018) and estimate corresponding transformations using off-the-shelf tools (Teed and Deng 2020; Kirillov et al. 2023; Alzayer et al. 2024; Wu et al. 2024). However, such data often contains undesired background dynamics and noisy annotations, making it unsuitable for learning precise object-centric manipulations. To

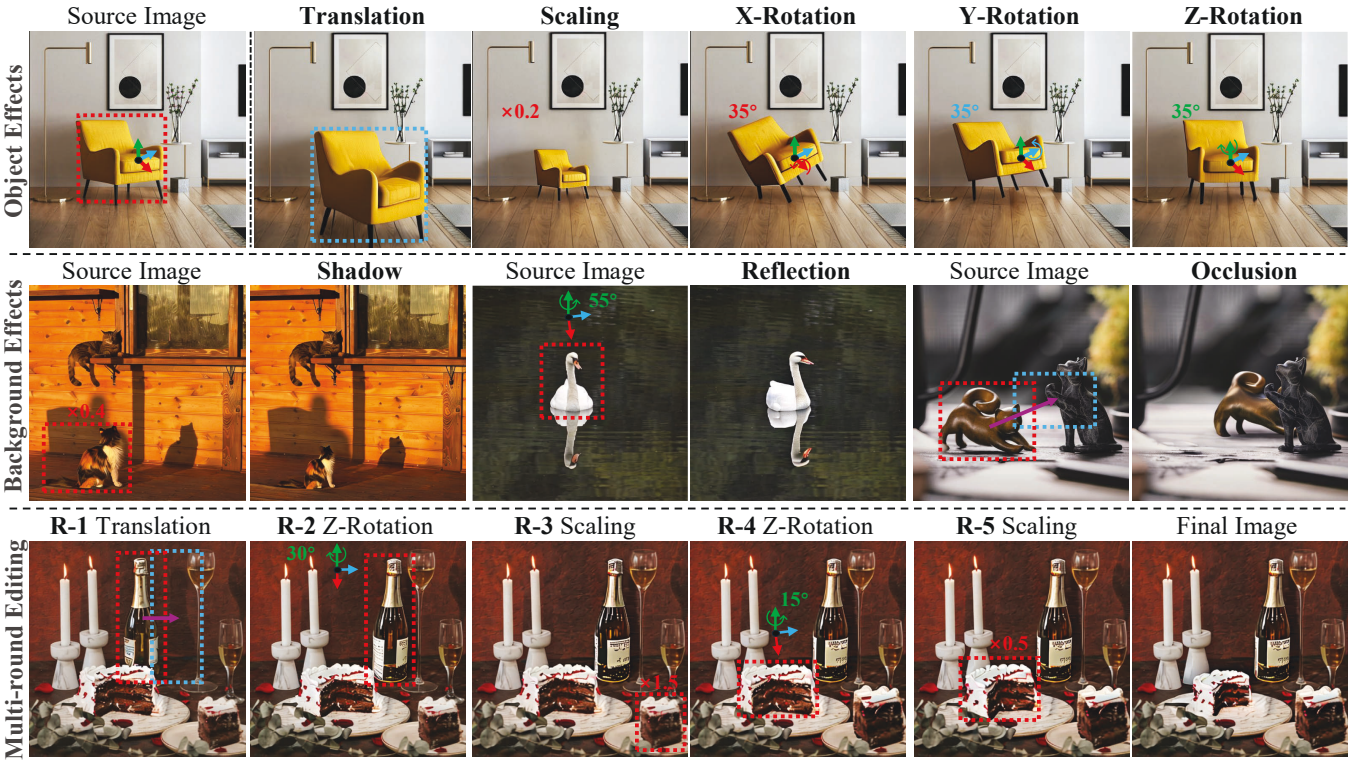

Figure 1: 3D-aware object manipulation results of our *Free-Form Scene Editor* (*FFSE*). 1) Object effects. *FFSE* can process a variety of 3D operations, including challenging transformations such as rotations. 2) Background effects. *FFSE* generates realistic environmental interaction resulting from object manipulations, such as shadows and occlusions. 3) Multi-round editing. *FFSE* maintains consistency of scene elements across multiple editing iterations. Moreover, the proposed *FFSE* provides a user-friendly interface without time-consuming 3D reconstruction.

address this issue, we construct a hybrid dataset, *3DObjectEditor*, which employs image sequences generated with physical simulation to emulate realistic manipulation dynamics. Unlike existing datasets (Michel et al. 2023), *3DObjectEditor* is specifically designed to support multi-round editing and covers a broader range of object and scene categories, along with higher domain diversity. Trained on *3DObjectEditor*, our *FFSE* model can perform various 3D operations iteratively, while generating plausible background effects and maintaining scene consistency across edits. In addition, our multi-stage training strategy enables the model's robust performance on real-world images.

In summary, our main contributions are as follows: **1)** We introduce *3DObjectEditor*, a hybrid dataset that simulates image sequences resulting from diverse 3D operations across a wide range of objects and scenes. **2)** We propose *FFSE*, an autoregressive framework for 3D-aware image editing that produces realistic object and background effects while maintaining scene consistency. **3)** Extensive experiments demonstrate the superior performance of the proposed *FFSE* over existing methods.

Related Works

Image Editing. Image editing (Meng et al. 2022; Hertz et al. 2023; Shuai et al. 2024; Yang et al. 2024) has been significantly enhanced by powerful text-to-image (T2I) diffusion models. Existing models enable fine-grained image editing, like appearance modification, object removal, and image inpainting.

Object-centric Spatial Manipulation. Current methods for object-centric manipulation rely on spatial cues like point pairs (Ling et al. 2024; Shi et al. 2024) or 2D masks (Mou et al. 2024; Epstein et al. 2023; Li et al. 2025; Shuai et al. 2025), but struggle with precise object-level transformations and 3D-aware operations. While some approaches reconstruct 3D scenes for better control (Zhao et al. 2025; Yenphraphai et al. 2024), they suffer from time-consuming optimization and noisy geometry estimation.

Methodology

Overview

To simulate the behavior of interactive 3D engines (B. O. Community 2018; Unreal Engine 5 2022) and enable manipulation of real-world images, our goal is to train a neural network for iterative 3D-aware object manipulation. We first define a formal setting involving a scene state space S, an operation space O, and a state transition function $p_{tf}(s'|s, o)$, where $s, s' \in S$ represent the current and next scene states, and $o \in O$ denotes an editing operation. In addition, we

define an observation space X, where each element is a projected view of the state space S via a mapping function f_m.

Specifically, the state space S represents the underlying configuration of scene elements, such as object and background components, while the observation space X corresponds to images captured from camera views through the rendering function f_m. The operation space O includes a set of 3D transformations $\{o^T, o^S, o^X, o^Y, o^Z\}$, corresponding to translation, scaling, and rotations around x/y/z axes aligned with the forward, left, and upward directions in the object space, respectively. The transition function $p_{tf}(s'|s, o)$ can be implemented by a physical simulator that updates the scene state based on the current state and applied operation.

Given the editing history $h_r = \{(x_i, o_i)\}_{i=0}^{r-1}$ up to r-th round, where $x_i \in X$ and x_0 is the source image, our goal is to model the observation distribution $p(x_r|h_r)$ using a diffusion-based generative model p_θ. To train this model, we collect a dataset $\{(h_{r_j}, x_{r_j})\}_{j=1}^{N_D}$ via a rule-based data generation pipeline below.

Data Generation

There is no well-established public dataset tailored for learning 3D-aware object manipulation in multi-round editing. A suitable dataset must meet several key criteria (see extended version), while the existing data sources violate some of them. Therefore, we introduce *3DObjectEditor*, a hybrid dataset that combines real D_{real} and synthetic D_{syn} samples.

Realistic Domain D_{real}. We construct image sequences in D_{real} using following steps: **1) Asset collection.** We leverage MULAN (Tudosiu et al. 2024), which contains RGBA images decomposed from scenes in MS-COCO (Lin et al. 2014) and LAION (Schuhmann et al. 2021) datasets. We only use the MS-COCO part. **2) Scene construction.** Based on labeled foreground & background tags, we randomly select a background and several object images as scene elements. For each object, its center position (x_p, y_p) in the background image and depth value d are initialized randomly. **3) Sequence construction.** For each editing step, an object is randomly chosen for manipulation, with the operation sampled from the restricted space $\{o^T, o^S\}$ in D_{real}. To simulate different distances from camera view, we determine the object's size (the shortest side of image) through $(d - d_{\text{max}})\frac{s_{\text{min}} - s_{\text{max}}}{d_{\text{max}} - d_{\text{min}}} + s_{\text{min}}$, where $[s_{\text{min}}, s_{\text{max}}]$ and $[d_{\text{min}}, d_{\text{max}}]$ denote size and depth bounds, respectively. The size bounds s_{min} and s_{max} are computed by scaling their predefined values $\hat{s}_{\text{min}}$ and $\hat{s}_{\text{max}}$ with the object's current scaling factor f_s, *i.e.*, $s = \hat{s} \times f_s$. During translation, we randomly update the object's position (x_p, y_p) and depth d, while in scaling, f_s is adjusted. The final image is rendered using the painter's algorithm: the background is placed first, followed by foreground objects rendered in depth order from farthest to nearest.

Synthetic Domain D_{syn}. Relying solely on D_{real} is insufficient, as it lacks support for precise object rotation and realistic background effects. Therefore, we employ Blender (B. O. Community 2018) for high-fidelity simulation: **1) Asset collection.** High-resolution panoramic images and 3D scenes from PolyHaven (PolyHaven 2022) and Sketchfab (Sketchfab 2022) are used as backgrounds. For objects, over 6,000 assets are filtered from Objaverse-LVIS (Deitke et al. 2023b) and Objaverse-XL (Deitke et al. 2023a), including animated models to increase pose diversity. **2) Scene construction.** A background and a set of object assets are randomly selected to form a scene. Directional and point lights are added to avoid underexposed renderings. Objects are appropriately scaled and randomly placed on the ground plane. For animated assets, a random keyframe is selected to provide diverse object poses. **3) Sequence construction.** At each step, any 3D operation from $\{o^T, o^S, o^X, o^Y, o^Z\}$ is applied to a randomly chosen object. Finally, Blender's Cycles ray tracer simulates realistic physics, producing high-quality environmental interaction such as shadows.

Free-Form Scene Editor

Existing methods (Michel et al. 2023; Yenphraphai et al. 2024; Wang et al. 2024; Pandey et al. 2024; Zhang et al. 2024; Wu et al. 2024) demonstrate limited capability in 3D-aware object manipulation, particularly in handling realistic *object effects*, *background effects*, and maintaining consistency in *multi-round editing*. To address these challenges, we propose *Free-Form Scene Editor* (*FFSE*), an approach that approximates $p(x_r|h_r)$ using a neural network parameterized by θ. The model estimates the new observation x_r in an autoregressive manner, conditioned on the editing history h_r.

The overall framework of *FFSE* is shown in Fig. 2. For training efficiency, we build upon a pretrained video generation backbone (Blattmann et al. 2023). To model the sequential editing process, we maintain a frame buffer and an operation buffer to store historical observations and user-specified operations. To encode h_r, *operation encoder* processes the past operations $\{o_i\}_{i=0}^{r-1}$ from the operation buffer. The outputs are injected into the main branch via *operation self-attention* to guide the editing behavior. In parallel, *frame encoder* encodes previous observations $\{x_i\}_{i=0}^{r-1}$ to capture scene dynamics and structural context. It also receives a target location mask to guide object placement in the current step. To improve the consistency of the edited object, we introduce *context self-attention* to enhance the standard self-attention layers. Furthermore, to prevent learning the domain-specific content, we introduce *Domain LoRA* for each domain in *3DObjectEditor*, and employ a multi-stage training strategy to ensure robust performance on real-world images.

Encoding of the History. We apply different components to process operations and observations, respectively. Specifically, the operation o_i in round i can be decomposed into *source region* and *operation type&value*. In our implementation, the centroid l_i^p and bounding box l_i^b are used as *source region* to locate the object before manipulation with different granularities. On the other hand, *operation type&value* indicates the relative transformation of the object from the last frame. Concretely, o_i^T is presented by the normalized pixel offset, while o_i^S and o_i^X, o_i^Y, o_i^Z are encoded by the scaling factor and rotation angle around the corresponding axis. Formally, *operation encoder* encodes *source region* and *operation type & value* by fourier embedding and MLP, which is denoted by $f(\cdot) = \text{MLP}(\text{Fourier}(\cdot))$. The encoded features of *source region* c_i^{src} and *operation type & value* c_i^{opt}

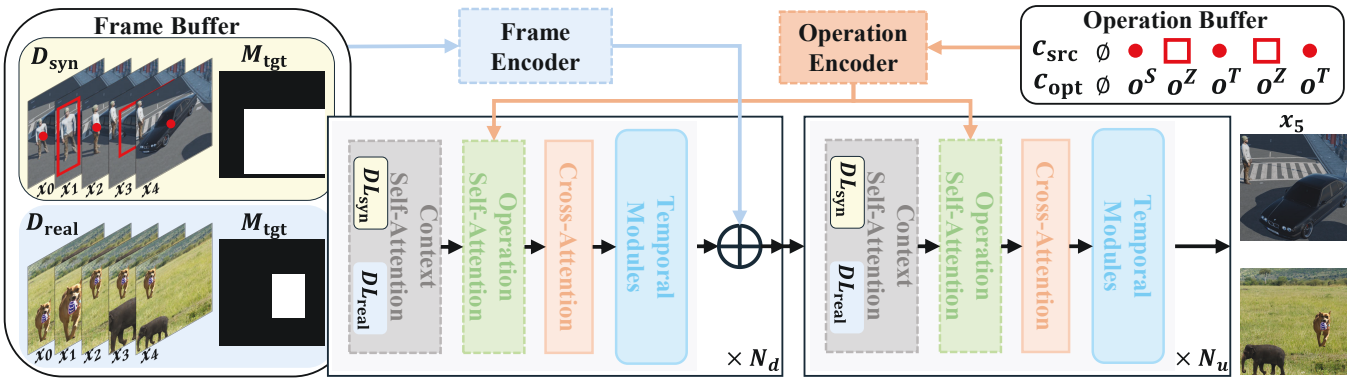

Figure 2: Overall framework of *Free-Form Scene Editor* (*FFSE*) with dashed boxes indicating introduced learnable modules, where the middle blocks and convolutional layers from the base model are omitted for simplicity. N_d and N_u denote the number of down and up blocks, respectively. Two 6-length editing sequences are shown as an example, where only DL_{syn} is active since the current training batch is sampled from D_{syn}. Historical observations and operations are processed by *frame encoder* and *operation encoder*, respectively, to capture scene structure changes. The output of *frame encoder* is added to down block features, while the output from *operation encoder* is injected into the main branch via *operation self-attention*. Additionally, standard self-attention modules are enhanced by *context self-attention* to improve the appearance consistency of the edited object.

in the i-th round are formalized as:

$$\begin{aligned} c_i^{\text{src}} &= [f(l_i^p), f(l_i^b)], \\ c_i^{\text{opt}} &= [f(o_i^T), f(o_i^S), f(o_i^X), f(o_i^Y), f(o_i^Z)], \end{aligned} \tag{1}$$

where $[\cdot]$ is channel-wise concatenation. Besides, we set the corresponding elements to learnable "null" embeddings for conditions not provided. As shown in Fig. 2, we concatenate the "null" conditions $\emptyset$ with $\{c_i^{\text{src}}\}_{i=0}^{r-1}$ and $\{c_i^{\text{opt}}\}_{i=0}^{r-1}$ along the sequence dimension, denoted as c_{src} and c_{opt}, respectively, where $\emptyset$ indicates that the initial observation x_0 is not derived from any operation. Finally, inspired from previous works (Li et al. 2023), we integrate assembled features $[c_{\text{src}}, c_{\text{opt}}]$ into the network by injecting *operation self-attention* between the *context self-* and cross-attention layers from spatial modules:

$$\hat{v} = \bar{v} + \beta \cdot \tanh(\gamma) \cdot \text{TS}\big(\text{SelfAttn}([\bar{v}, \text{repeat}([c_{\text{src}}, c_{\text{opt}}])])\big), \tag{2}$$

where $\bar{v}$ is the features from *context self-attention*. Concretely, $[c_{\text{src}}, c_{\text{opt}}]$ is repeated in spatial dimension to align with $\bar{v}$. TS truncates the features to select visual tokens. γ is a learnable scalar and β modulates the control effect in inference time.

On the other hand, we represent *target region* by the binary mask M_{tgt}, which is derived from the bounding box of the target location in the current round. This design enhances the location accuracy of the edited object. Then, it is concatenated with previous observations $\{x_j\}_{j=0}^{r-1}$, and the final input is fed to the *frame encoder* as indicated in Fig. 2, which is a lightweight branched network. Finally, the output is added to the spatial features from the down blocks. We randomly omit M_{tgt} during training by applying an all-zero mask, which allows the model to implicitly learn the *target region* from the current operation and observations, avoiding cumbersome input from users in inference time.

Context Self-attention. To maintain the appearance consistency of edited objects after operations, we introduce *context self-attention* (*CSA*) to enhance the ordinary self-attention modules by referring the edited object in the r-th round to the same object from the last observation, expressed as:

$$\bar{v}_r = v_r + \lambda M_{\text{tgt}} \text{softmax}(A_{r,r-1} + \frac{Q_r^{'}(K_{r-1}^{'})^T}{\sqrt{d}}) V_{r-1}^{'}, \tag{3}$$

where subscripts r and $r-1$ represent sliced features corresponding to the current and the last rounds, respectively, and $'$ indicates that the variables are calculated by newly injected layers. For example, v_r is the r-th round feature from ordinary self-attention modules. The learnable λ adjusts the effects from the last observation, and M_{tgt} prevents the influence in irrelevant pixels. Furthermore, $A_{r,r-1}$ is a $hw \times hw$ matrix and the element in $[i, j]$ is set to 0 only if $\text{Vec}(M_r)[i]$ and $\text{Vec}(M_{r-1})[j]$ are all located in the object region, where $\text{Vec}(\cdot)$ represents that the matrix is flattened to vector, M_r and M_{r-1} are masks derived by object bounding boxes in the current and the last steps. Other elements in $A_{r,r-1}$ are all set to an infinitesimal value, ensuring that only the features inside the object region are involved in mutual computation.

Multi-stage Training Strategy. We propose a multi-stage training strategy to learn from *3DObjectEditor*. First, we train the newly injected modules with the whole dataset. To prevent overfitting to domain-specific content from different sources, we adopt *Domain LoRA* (Hu et al. 2022) DL_{real} and DL_{syn} for D_{real} and D_{syn}, respectively. During training, the corresponding LoRA modules are loaded into the network according to the domain identifier of the sample. This setup allows the model to capture object effects caused by specific operations shared between D_{real} and D_{syn}, while enhancing generalization for diverse in-the-wild images. As shown in Fig. 2, these LoRA modules are injected only into the *context self-attention* layers. The training objective for this stage is:

$$\arg\min_{\theta, DL_{\text{real,syn}}} E_{(h_r, x_r) \sim D_{\text{real,syn}}} \left[\left\| \varepsilon_{\theta, DL_{\text{real,syn}}} \left(x_{0:r}^t, t, h_r \right) - \epsilon \right\|^2 \right] \tag{4}$$

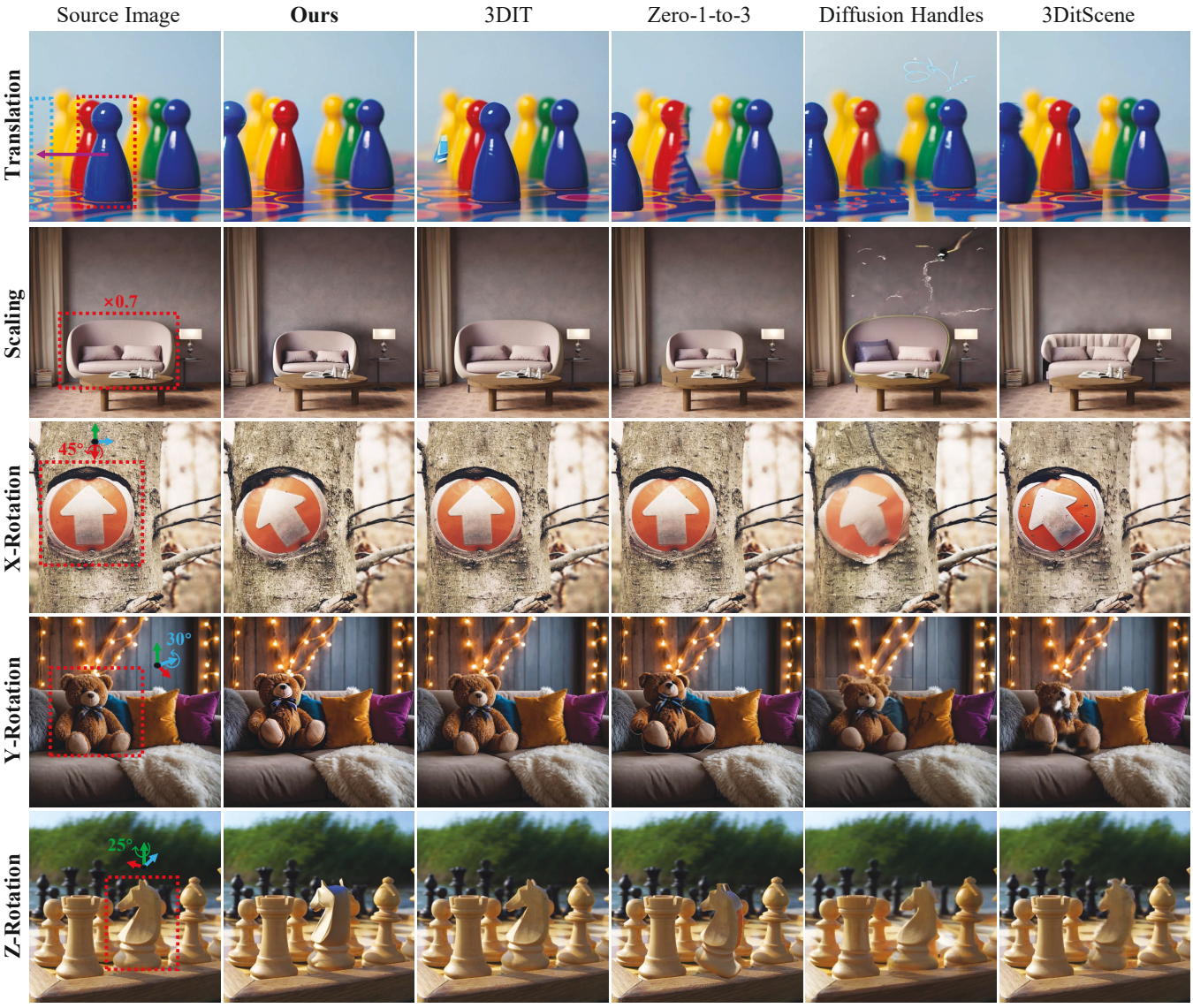

Figure 3: Evaluation of object effects under different 3D operations in single-round editing.

where $x^t_{0:r}$ is the noisy image sequence in time t, obtained by injecting Gaussian noise ϵ into $x_{0:r}$ via diffusion forward process. r is sampled from a uniform distribution bounded by $[r_{\text{min}}, r_{\text{max}}]$.

Due to the insufficient simulation in D_{real}, the model trained with Eq. (4) may generate unrealistic results. To enhance the quality of generated background effects such as shadows, we finetune the model solely with D_{syn}. In this stage, we load DL_{syn} into network while only optimizing θ:

$$\arg\min_{\theta} E_{(h_r, x_r)\sim D_{\text{syn}}} \left[\left\| \varepsilon_{\theta, DL_{\text{syn}}} \left(x^t_{0:r}, t, h_r \right) - \epsilon \right\|^2 \right]. \quad (5)$$

Inference. Our framework supports iterative user interaction, enabling the multi-round object manipulation from an initial source image x_0. It is worth noting that *Domain LoRA* modules are omitted during inference to preserve the quality of the base generative model.

Experiment

Implementation Details. We build *FFSE* upon the pretrained image-to-video model SVD (Blattmann et al. 2023), training it with the Adam optimizer (initial learning rate: $1e^{-4}$) on 4 NVIDIA A800 80G GPUs. All experiments use 512×512 resolution and a batch size of 8, with $r_{\text{min}} = 1$ and $r_{\text{max}} = 12$. In Stage 1, we jointly train $\{DL_{\text{real}}, DL_{\text{syn}}\}$ and newly introduced parameters θ using D_{real} and D_{syn} for 80K iterations. In Stage 2, we load DL_{syn} to the model and further optimize θ on D_{syn} for 10K iterations.

Evaluation Details. For assessing the performance of models, we concentrate on the following aspects. **1) Maintenance of source content.** We use PSNR and SSIM for assessing the overall visual similarity. **2) Identity preservation of the edited object.** For the operated object, we use CLIP-Score (Radford et al. 2021) and DINO-Score (Caron et al.

Task	Method	Content Consistency		Identity Preservation	
		PSNR ↑	SSIM$_{\times 10^2}$ ↑	DINO ↑	CLIP ↑
Single-round Editing	3DIT (Michel et al. 2023)	20.12	68.76	61.38	80.96
	Zero-1-to-3 (Liu et al. 2023)	<u>23.84</u>	<u>71.97</u>	65.42	83.27
	Diffusion Handles (Pandey et al. 2024)	18.83	58.33	71.33	88.53
	3DitScene (Zhang et al. 2024)	17.67	53.39	<u>73.69</u>	<u>89.11</u>
	***FFSE* (ours)**	**26.31**	**79.54**	**82.39**	**91.67**
Multi-round Editing	3DIT	18.31	57.62	60.19	78.27
	Zero-1-to-3	<u>19.81</u>	<u>64.77</u>	<u>61.67</u>	<u>82.38</u>
	Diffusion Handles	13.79	50.47	59.06	78.24
	3DitScene	10.75	43.24	42.17	76.35
	***FFSE* (ours)**	**24.96**	**74.99**	**79.51**	**90.42**

Table 1: Quantitative evaluation in single-round and multi-round editing.

2021) to compute the semantic similarity between the objects before and after the manipulation. For multi-round editing, we average the results of adjacent image pairs from generated frames. Furthermore, since the operation effects on the object and background are difficult to evaluate, we conduct a user study to represent the human preference. For the validation dataset, we source high-quality images from public websites like Unsplash (Unsplash 2020), Pixabay (pixabay 2020), and Pexels (Pexels 2020) to construct diverse and complex scenes from the real world. Specifically, we collect 50 images in total and manually select the objects to be edited. Next, we use an off-the-shelf tool (Liu et al. 2024) to estimate the bounding box and centroid as *source region*. Then, several *target region* and *operation type&value* pairs are assigned for objects to construct samples for different operations. We also construct operation sequences to assess multi-round editing performance. Finally, for single-round editing, we obtain 45 test cases for translation, 48 for scaling, 30 for rotation around the x/y axis, and 40 for rotation around the z axis. We also get 30 examples for the multi-round editing experiment, where the sequence length is fixed to 6.

Compared Methods. For compared methods, we focus on two algorithm families and choose the methods that are open source and can handle most of the 3D operations described in our paper. For image space methods, we choose 3DIT (Michel et al. 2023) and Zero-1-to-3 (Liu et al. 2023) as compared methods. Specifically, we crop the source region from the image and apply Zero-1-to-3 to get the transformed object, which is then overlaid on the target region of the inpainted background image. Compared 3D methods include Diffusion Handles (Pandey et al. 2024) and 3DitScene (Zhang et al. 2024), which manipulate estimated point clouds or reconstructed 3DGS (Kerbl et al. 2023). Since 3DitScene and Diffusion Handles do not implement scaling operations, we move the object from/closer to the camera to emulate scaling down/up. Besides, for operations that are not supported by the compared method, we directly return the source image.

Comparisons with State-of-the-Art Methods

Comparison on Single-round Editing. In this experiment, we primarily evaluate the performance in single-round editing, concentrating on the effects of object and background caused by specific manipulations. Fig. 3 demonstrates the performance of object effects from different 3D operations. For image space methods, 3DIT only supports limited operation types and suffers from poor generalization ability. As a result, it fails to accomplish most of the edits. On the other hand, although it suffers from a cumbersome workflow, Zero-1-to-3 can handle most of the 3D operations. However, it fails to achieve the goal in some complicated scenarios (rows 3,5 in Fig. 3). Furthermore, the noisy estimation of the inpainting model leads to unwanted artifacts in the occluded area, as shown in row 1 of Fig. 3. In contrast, 3D space methods outperform in operations requiring geometric knowledge, such as object rotation. Nevertheless, they are limited by the time-consuming reconstruction process and low-quality results caused by noisy geometry estimation. In comparison, our method accomplishes all operations with high fidelity and quality. For example, *FFSE* can recover or manipulate the occluded object (the red checker piece in row 1 and the sofa in row 2). For rotation around principal axes, our method generates accurate object transformations, while achieving realistic physical effects, such as the reflection of light on the chess piece in the last row of Fig. 3. The quantitative results in Tab. 1 and the user study in the extended version also demonstrate that *FFSE* outperforms in consistency, quality, and operation fidelity.

We assess the plausibility of background effects in Fig. 4. As shown in rows 1,2 of Fig. 4, the compared methods fail to create realistic shadows and reflections, caused by the limited knowledge of the interaction between the object and environment. In the 1st row of Fig. 4, most of the methods fail to remove the dog's shadow in the source area and generate the correct effect in the target location. Although 3DIT is trained on data created by physical simulation, it is limited by the poor generalization ability on real-world images. The last row of Fig. 4 evaluates the case of occlusion. Among the compared methods, 3DitScene achieves better performance using reconstructed scene structure, but with artifacts in the source region. Other methods struggle to place the object behind the teapot, exhibiting incorrect occlusion. Our method generates reasonable environmental interactions in all cases. The user study in the extended version also demonstrates the physical plausibility of our method.

Comparison on Multi-round Editing. In this experiment, we evaluate the multi-round editing performance of *FFSE*

Setting	Content Consistency		Identity Preservation	
	PSNR ↑	SSIM$_{\times 10^2}$ ↑	DINO ↑	CLIP ↑
w/ D_{real}	25.86	79.31	<u>81.92</u>	<u>91.11</u>
w/ D_{syn}	24.37	74.51	73.31	86.43
w/o stage 2	<u>25.92</u>	<u>79.33</u>	78.77	89.82
w/o DL (a)	25.37	76.54	79.53	89.75
w/o DL (b)	24.53	73.25	74.92	88.13
w/o *CSA*	24.81	75.17	75.65	88.71
***FFSE*(Ours)**	**26.31**	**79.54**	**82.39**	**91.67**

Table 2: Ablation studies.

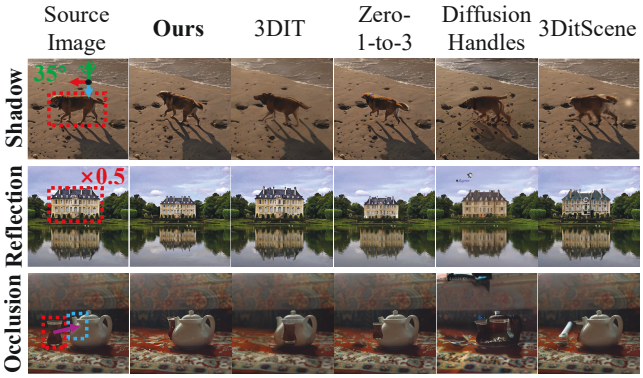

Figure 4: Evaluation of background effects in single-round editing. The figures demonstrate that *FFSE* generates more physically-plausible environmental interactions.

and representative methods from image space and 3D space algorithm families. For the qualitative experiment, we assess methods' awareness of scene structure changes through changing the occlusion relationships among objects. As shown in columns 2-4 of Fig. 5, we translate the teapot in front of the cup and then move it away. Both Zero-1-to-3 (Liu et al. 2023) and Diffusion Handles (Pandey et al. 2024) fail to recover the cup. Besides, as the number of editing rounds increases, the quality of their results deteriorates progressively due to accumulated errors. In contrast, our method accurately restores the occluded object, exhibiting high consistency after multiple manipulations. Tab. 1 also reveals that *FFSE* achieves better performance in multi-round editing.

Ablation Studies

In this experiment, we perform ablation studies to illustrate the impacts of our multi-stage training strategy and *context self-attention* (*CSA*). Fig. 6 compares different settings in the same editing scenario. Due to the unrealistic physical simulation, the model trained with D_{real} manipulates the object in a copy-and-paste manner with incorrect shadow in the background. The model trained with D_{syn} generates low-quality images with oversaturated color, which means it overfits to the rendering style. The performance of the model trained with a single stage falls between the above two models, resulting in unrealistic shadows. To verify the effectiveness of *Domain LoRA* DL_{syn} and DL_{real}, we additionally train a model based on a single set of LoRA modules. When

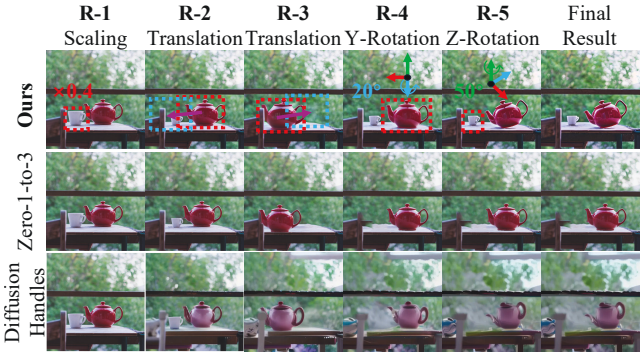

Figure 5: Evaluation in multi-round editing. *FFSE* accomplishes the operation in each editing round, and maintains high consistency of scene elements. As indicated in columns 2-4 of our method, the cup is first occluded by the teapot, and then becomes visible.

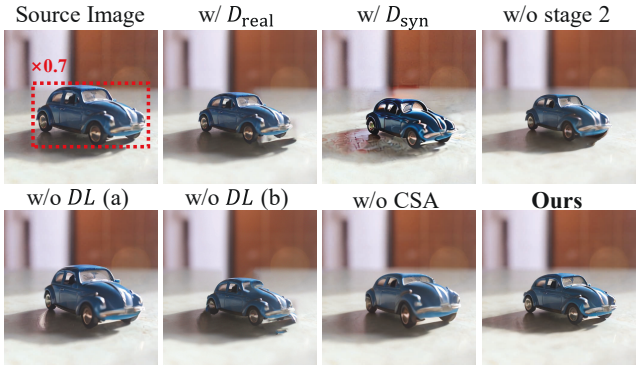

Figure 6: Ablation studies.

the LoRA set is removed during inference ("w/o DL (a)" in Fig. 6), the model fails to perform the correct operation since the LoRA modules are coupled with operation modules. When the LoRA set is fully loaded ("w/o DL (b)"), the image quality is compromised by the artifacts. On the other hand, the model without *CSA* exhibits lower consistency of object appearance. The object-level metrics in Tab. 2 also demonstrate the improvement in consistency from *CSA*.

Conclusion

We introduced *FFSE*, a 3D-aware autoregressive image editing framework that enables users to perform iterative object manipulations directly on real-world images. Trained on our dataset *3DObjectEditor* that combines realistic and synthetic sequences across diverse objects and scenes, *FFSE* learns to perceive structural changes and maintain consistency across multiple editing rounds. By integrating condition-specific components and a multi-stage training scheme, our model can handle various 3D operations and generalize well to in-the-wild scenarios. Extensive experiments show that *FFSE* outperforms previous methods in both single-round and multi-round editing, delivering high-quality results.

Acknowledgments

This project was supported by the National Natural Science Foundation of China (NSFC) under Grant No. 62472104. Dr Tao's research is partially supported by NTU RSR and Start Up Grants.

References

Alzayer, H.; Xia, Z.; Zhang, X.; Shechtman, E.; Huang, J.; and Gharbi, M. 2024. Magic-Fixup: Streamlining Photo Editing by Watching Dynamic Videos. *arXiv*.

B. O. Community. 2018. *Blender - a 3d modelling and rendering package*.

Bain, M.; Nagrani, A.; Varol, G.; and Zisserman, A. 2021. Frozen in Time: A Joint Video and Image Encoder for End-to-End Retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 1708–1718.

Blattmann, A.; Dockhorn, T.; Kulal, S.; Mendelevitch, D.; Kilian, M.; Lorenz, D.; Levi, Y.; English, Z.; Voleti, V.; Letts, A.; et al. 2023. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv*.

Brooks, T.; Holynski, A.; and Efros, A. A. 2023. InstructPix2Pix: Learning to Follow Image Editing Instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 18392–18402.

Caron, M.; Touvron, H.; Misra, I.; Jégou, H.; Mairal, J.; Bojanowski, P.; and Joulin, A. 2021. Emerging Properties in Self-Supervised Vision Transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 9630–9640.

Deitke, M.; Liu, R.; Wallingford, M.; Ngo, H.; Michel, O.; Kusupati, A.; Fan, A.; Laforte, C.; Voleti, V.; Gadre, S. Y.; VanderBilt, E.; Kembhavi, A.; Vondrick, C.; Gkioxari, G.; Ehsani, K.; Schmidt, L.; and Farhadi, A. 2023a. Objaverse-XL: A Universe of 10M+ 3D Objects. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Deitke, M.; Schwenk, D.; Salvador, J.; Weihs, L.; Michel, O.; VanderBilt, E.; Schmidt, L.; Ehsani, K.; Kembhavi, A.; and Farhadi, A. 2023b. Objaverse: A Universe of Annotated 3D Objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 13142–13153.

Epstein, D.; Jabri, A.; Poole, B.; Efros, A. A.; and Holynski, A. 2023. Diffusion Self-Guidance for Controllable Image Generation. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Hertz, A.; Mokady, R.; Tenenbaum, J.; Aberman, K.; Pritch, Y.; and Cohen-Or, D. 2023. Prompt-to-Prompt Image Editing with Cross-Attention Control. In *International Conference on Learning Representations (ICLR)*.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2022. LoRa: Low-rank adaptation of large language models. *International Conference on Learning Representations (ICLR)*.

Jabri, A.; van Steenkiste, S.; Hoogeboom, E.; Sajjadi, M. S. M.; and Kipf, T. 2024. DORSal: Diffusion for Object-centric Representations of Scenes et al. In *International Conference on Learning Representations (ICLR)*.

Kerbl, B.; Kopanas, G.; Leimkühler, T.; and Drettakis, G. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics*, 139:1–139:14.

Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A. C.; Lo, W.-Y.; et al. 2023. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 4015–4026.

Li, Y.; Liu, H.; Wu, Q.; Mu, F.; Yang, J.; Gao, J.; Li, C.; and Lee, Y. J. 2023. GLIGEN: Open-Set Grounded Text-to-Image Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 22511–22521.

Li, Z.; Luo, H.; Shuai, X.; and Ding, H. 2025. AnyI2V: Animating Any Conditional Image with Motion Control Generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Lin, T.; Maire, M.; Belongie, S. J.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft COCO: Common Objects in Context. In *European Conference on Computer Vision (ECCV)*, 740–755.

Ling, P.; Chen, L.; Zhang, P.; Chen, H.; Jin, Y.; and Zheng, J. 2024. Freedrag: Feature dragging for reliable point-based image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 6860–6870.

Liu, R.; Wu, R.; Hoorick, B. V.; Tokmakov, P.; Zakharov, S.; and Vondrick, C. 2023. Zero-1-to-3: Zero-shot One Image to 3D Object. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 9264–9275.

Liu, S.; Zeng, Z.; Ren, T.; Li, F.; Zhang, H.; Yang, J.; Jiang, Q.; Li, C.; Yang, J.; Su, H.; et al. 2024. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European Conference on Computer Vision (ECCV)*, 38–55.

Meng, C.; He, Y.; Song, Y.; Song, J.; Wu, J.; Zhu, J.; and Ermon, S. 2022. SDEdit: Guided Image Synthesis and Editing with Stochastic Differential Equations. In *International Conference on Learning Representations (ICLR)*.

Michel, O.; Bhattad, A.; VanderBilt, E.; Krishna, R.; Kembhavi, A.; and Gupta, T. 2023. OBJECT 3DIT: Language-guided 3D-aware Image Editing. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Mildenhall, B.; Srinivasan, P. P.; Tancik, M.; Barron, J. T.; Ramamoorthi, R.; and Ng, R. 2022. NeRF: representing scenes as neural radiance fields for view synthesis. *European Conference on Computer Vision (ECCV)*, 99–106.

Mou, C.; Wang, X.; Song, J.; Shan, Y.; and Zhang, J. 2024. DragonDiffusion: Enabling Drag-style Manipulation on Diffusion Models. In *International Conference on Learning Representations (ICLR)*.

Mu, J.; Gharbi, M.; Zhang, R.; Shechtman, E.; Vasconcelos, N.; Wang, X.; and Park, T. 2024. Editable Image Elements for Controllable Synthesis. In *European Conference on Computer Vision (ECCV)*, 39–56.

Pandey, K.; Guerrero, P.; Gadelha, M.; Hold-Geoffroy, Y.; Singh, K.; and Mitra, N. J. 2024. Diffusion Handles Enabling 3D Edits for Diffusion Models by Lifting Activations to 3D. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 7695–7704.

Pexels. 2020. *https://www.pexels.com*.

pixabay. 2020. *https://pixabay.com*.

Podell, D.; English, Z.; Lacey, K.; Blattmann, A.; Dockhorn, T.; Müller, J.; Penna, J.; and Rombach, R. 2023. SDXL: Improving Latent Diffusion Models for High-Resolution Image Synthesis. *arXiv*.

PolyHaven. 2022. *https://polyhaven.com*.

Poole, B.; Jain, A.; Barron, J. T.; and Mildenhall, B. 2023. DreamFusion: Text-to-3D using 2D Diffusion. In *International Conference on Learning Representations (ICLR)*.

Qin, Z.; Shuai, X.; and Ding, H. 2025. SceneDesigner: Controllable Multi-Object Image Generation with 9-DoF Pose Manipulation. In *NeurIPS*.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; Krueger, G.; and Sutskever, I. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *International Conference on Machine Learning (ICML)*, 8748–8763.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-Resolution Image Synthesis with Latent Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 10674–10685.

Schuhmann, C.; Vencu, R.; Beaumont, R.; Kaczmarczyk, R.; Mullis, C.; Katta, A.; Coombes, T.; Jitsev, J.; and Komatsuzaki, A. 2021. LAION-400M: Open Dataset of CLIP-Filtered 400 Million Image-Text Pairs. *arXiv*.

Shi, Y.; Xue, C.; Liew, J. H.; Pan, J.; Yan, H.; Zhang, W.; Tan, V. Y. F.; and Bai, S. 2024. DragDiffusion: Harnessing Diffusion Models for Interactive Point-Based Image Editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 8839–8849.

Shuai, X.; Ding, H.; Ma, X.; Tu, R.; Jiang, Y.-G.; and Tao, D. 2024. A survey of multimodal-guided image editing with text-to-image diffusion models. *arXiv*.

Shuai, X.; Ding, H.; Qin, Z.; Luo, H.; Ma, X.; and Tao, D. 2025. Free-Form Motion Control: Controlling the 6D Poses of Camera and Objects in Video Generation. In *ICCV*.

Sketchfab. 2022. *https://sketchfab.com*.

Teed, Z.; and Deng, J. 2020. RAFT: Recurrent All-Pairs Field Transforms for Optical Flow. In *European Conference on Computer Vision (ECCV)*, 402–419.

Tudosiu, P.; Yang, Y.; Zhang, S.; Chen, F.; McDonagh, S.; Lampouras, G.; Iacobacci, I.; and Parisot, S. 2024. MULAN: A Multi Layer Annotated Dataset for Controllable Text-to-Image Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 22413–22422.

Tumanyan, N.; Geyer, M.; Bagon, S.; and Dekel, T. 2023. Plug-and-Play Diffusion Features for Text-Driven Image-to-Image Translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 1921–1930.

Unreal Engine 5. 2022. *https://www.unrealengine.com/*. Https://www.unrealengine.com/.

Unsplash. 2020. *https://unsplash.com*.

Wang, R.; Xiang, J.; Yang, J.; and Tong, X. 2024. Diffusion Models are Geometry Critics: Single Image 3D Editing Using Pre-trained Diffusion Priors. In *European Conference on Computer Vision (ECCV)*, 441–458.

Wu, Z.; Rubanova, Y.; Kabra, R.; Hudson, D. A.; Gilitschenski, I.; Aytar, Y.; van Steenkiste, S.; Allen, K. R.; and Kipf, T. 2024. Neural Assets: 3D-Aware Multi-Object Scene Synthesis with Image Diffusion Models. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Yang, S.; Zhang, L.; Ma, L.; Liu, Y.; Fu, J.; and He, Y. 2024. Magicremover: Tuning-free Text-guided Image inpainting with Diffusion Models. In *International Conference on Learning Representations (ICLR)*.

Yenphraphai, J.; Pan, X.; Liu, S.; Panozzo, D.; and Xie, S. 2024. Image Sculpting: Precise Object Editing with 3D Geometry Control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 4241–4251.

Zhang, Q.; Xu, Y.; Wang, C.; Lee, H.; Wetzstein, G.; Zhou, B.; and Yang, C. 2024. 3DitScene: Editing Any Scene via Language-guided Disentangled Gaussian Splatting. *arXiv*.

Zhao, R.; Zhang, Z.; Yang, Z.; and Yang, Y. 2025. 3D Object Manipulation in a Single Image using Generative Models. *arXiv*.

Zhou, T.; Tucker, R.; Flynn, J.; Fyffe, G.; and Snavely, N. 2018. Stereo magnification: learning view synthesis using multiplane images. *ACM Transactions on Graphics*, 65.