MotionDiff: Training-free Zero-shot Interactive Motion Editing via Flow-assisted Multi-view Diffusion

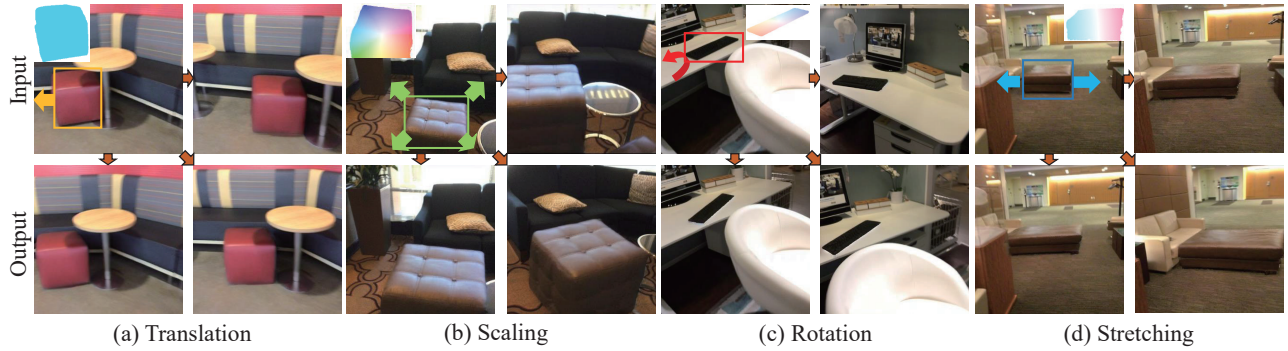
Yikun Ma
Sun Yat-sen University
mayk25@mail2.sysu.edu.cn

Yiqing Li
Sun Yat-sen University
liyq265@mail2.sysu.edu.cn

Jiawei Wu
Sun Yat-sen University
wujw97@mail2.sysu.edu.cn

Xing Luo
Peng Cheng Laboratory
luox@pcl.ac.cn

Zhi Jin *
Sun Yat-sen University
jinzh26@mail.sysu.edu.cn



(a) Translation (b) Scaling (c) Rotation (d) Stretching

Figure 1. We propose **MotionDiff**, a training-free zero-shot flow-assisted diffusion method, which performs complex multi-view motion editing for static scenes. Users can interactively select the object to be edited and add the motion priors of interest. Then, MotionDiff achieves corresponding multi-view motions, *e.g.*, translation, scaling, rotation, and stretching with satisfactory consistency.

Abstract

[1] *Generative models have made remarkable advancements and are capable of producing high-quality content. However, performing controllable editing with generative models remains challenging, due to their inherent uncertainty in outputs. This challenge is particularly pronounced in motion editing, which involves the processing of spatial information. While some physics-based generative methods have attempted to implement motion editing, they typically operate on single-view images with simple motions, such as translation and dragging. These methods struggle to handle complex motions, such as, rotation and stretching, and ensure multi-view consistency, often necessitating resource-intensive retraining. To address these challenges, we propose MotionDiff, a training-free zero-shot diffusion method that leverages optical flow for complex motion editing among multi-view images. Specifically, given a static scene, users can interactively select objects of interest to add motion priors. The proposed Point Kinematic Model (PKM) then estimates corresponding multi-view optical flows during the Multi-view Flow Estimation Stage (MFES). Subsequently, these optical flows are utilized to generate multi-view motion results through decoupled motion representation in the Multi-view Motion Diffusion Stage (MMDS). Extensive experiments demonstrate that MotionDiff outperforms other physics-based generative motion editing methods in achieving high-quality multi-view consistent motion results. Notably, MotionDiff does not require retraining, enabling users to conveniently adapt it for various downstream tasks. Code is available at https://github.com/Mr-Ma-yikun/MotionDiff.*

1. Introduction

Generative models, including Generative Adversarial Networks (GANs) [13] and Diffusion models [30, 50], possess the capability to produce high-quality content. Nevertheless, generating controllable results and executing precise pixel-level editing with these models remain challenge.

Recently, several image editing studies [6, 9, 25] have

[1] * Corresponding author

employed text prompts to execute static editing tasks, such as style transfer and object replacement using generative models. However, accurately processing motion information based on text prompts poses difficulties, leading to suboptimal performance. To address this issue, some methods [19, 20, 26] introduce user-defined physical priors, such as dragging and motion points, to guide generative models for motion editing. Nonetheless, these approaches are generally limited to simple translational dragging and are less effective in handling complex motions, such as scaling and rotation. In contrast, Motion Guidance [12] proposes utilizing optical flow to guide diffusion for complex motion editing. However, inadequate supervision of motion objects results in altered texture details. Additionally, the absence of multi-view motion constraints leads to inconsistent editing performance across different views. More recently, certain methods [41, 45, 49] generate physical priors from single images. However, these methods often require retraining or rely on specific field representations (*e.g.*, NeRF [24], 3DGS [16]), which complicates the editing process and incurs additional resource consumption.

In summary, current physics-based generative motion editing techniques confront three principal challenges: **1)** Drag-based methods excel in managing simple translational motions, but encounter difficulties with more complex rotation and stretching motions. **2)** Most methods emphasize single-view image priors, which obstructs the achievement of multi-view consistent motion editing. **3)** Most methods necessitate retraining or specific field representations, thereby escalating computational resource requirements and data collection expenses.

To address these challenges, we propose a training-free zero-shot multi-view motion editing method called **MotionDiff**. Specifically, it comprises two inference stages: **1) Multi-view Flows Estimation Stage (MFES).** Given a static scene, MFES enables users to conveniently and interactively select the object and motion mode of interest for editing. We then propose the Points Kinematic Model (PKM) to estimate multi-view optical flows based on the motion of 3D point clouds within the static scene. **2) Multi-view Motion Diffusion Stage (MMDS).** This stage takes multi-view images and the estimated optical flows as inputs, guiding a diffusion model to perform motion editing. Our core insight is to decouple the motion representation into a combination of static background, motion objects, and occluded regions. Consequently, we design corresponding guidance strategies without incurring additional computational costs. Extensive experiments demonstrate that MotionDiff can effectively implement complex motion editing, while maintaining multi-view consistency (as shown in Figure 1), surpassing other physics-based generative motion editing methods. Notably, our method provides a user-friendly motion editing solution without retraining.

The main contributions can be summarized as follows:

1) We propose MotionDiff, a training-free zero-shot flow-assisted diffusion framework that enables diverse complex multi-view motion editing operations, such as translation, scaling, rotation and stretching.
2) A user-friendly interactive framework is developed within MFES, complemented by the proposed PKM, to derive multi-view optical flows from static scenes.
3) A decoupled inference-only motion representation is designed in MMDS, facilitating convenience for other relevant tasks, like Augmented Reality (AR), Virtual Reality (VR) and Human-Computer Interaction (HCI).

2. Related Works

2.1. Image Editing with Generative Models

Generative models possess the ability to produce realistic and high-quality content, including images [30, 50], videos [17, 36], and 3D models [22, 28]. Prior studies [1, 27, 52] have employed GANs [13] for image editing tasks. Notably, DragGAN [27] stands out by introducing key-handle mechanisms that enable precise control over object movement within images. Nevertheless, due to the limited generative capabilities of GANs, GANs-based methods often encounter challenges in terms of generalization, as well as the authenticity and resolution of the generated content.

The remarkable success of diffusion models [15, 30, 37] has drawn significant attention to image editing [3, 6, 10, 14, 23, 48]. These approaches generally employ target descriptions or feature attention mechanisms for specific editing tasks. However, achieving precise pixel-level editing with physical attributes remains a formidable challenge, primarily due to the inherent uncertainty in outputs and the inadequate representation of spatial motion information.

2.2. Physics-based Generative Motion Editing

Recently, there have been significant surge of interests in physics-based and interactive generative editing [2, 20, 26, 33, 35, 51]. These approaches typically generate physical motion priors, such as vibrations, drags and geometric transformations. Subsequently, they employ retrained diffusion models to process these priors and produce motion results. However, these methods are primarily effective for handling unordered or simple dragging motions, but struggle with complex motions due to limited motion representation capabilities. Thus, Motion Guidance [12] introduces optical flow as a motion prior to enable more sophisticated motion editing. Nonetheless, its guidance strategy does not adequately account for texture details of the motion objects, leading to inconsistent appearance and multi-view performance. In contrast, other methods [21, 41, 45, 46, 49] choose to generate priors (such as depth maps, bounding boxes, *etc.*) from single images, then utilize these priors to

explicitly or implicitly enhance editing more complex 3D assets and multi-view images. However, these methods require retraining from paired datasets [41, 45], or rely on specific NeRF [49] or 3DGS [21, 46] representation fields, which increase computational resource requirements and complicates the practical editing process.

3. Proposed Method

3.1. Preliminaries

Denoising Diffusion. Denoising Diffusion Probabilistic Models (DDPM) [15] introduces the diffusion model for image generation. Using the reparameterization trick, the added noise of input $\boldsymbol{x_0}$ can directly be expressed as:

$$\boldsymbol{x_t} = \sqrt{\overline{\alpha}_t}\boldsymbol{x_0} + \sqrt{1-\overline{\alpha}_t}\boldsymbol{\epsilon}, \quad where\ \boldsymbol{\epsilon} \sim \mathcal{N}(0, I), \quad (1)$$

where α_t represents the learned schedule, $\boldsymbol{\epsilon}$ denotes the noise variable. The computational bottleneck of DDPM is the number of denoising timesteps T, hence, a non-Markovian variant Denoising Diffusion Implicit Models (DDIM) [37] is introduced to reduce the number of T.

$$\begin{aligned}\overline{\boldsymbol{x}}_{\boldsymbol{pre}} = \sqrt{\overline{\alpha}_{pre}}\ \frac{\boldsymbol{x_t} - \sqrt{1-\overline{\alpha}_t}\boldsymbol{\epsilon_\theta}(\boldsymbol{x_t})}{\sqrt{\overline{\alpha}_t}} \\ + \sqrt{1-\overline{\alpha}_{pre}-\sigma_t^2}\,\epsilon_\theta(\boldsymbol{x_t}) + \sigma_t^2\boldsymbol{\epsilon}\,,\end{aligned} \quad (2)$$

where $\epsilon_\theta(\boldsymbol{x_t})$ denotes the denoising network, usually U-Net [31], and σ is the variance hyperparameter.

Guidance Diffusion. Obtaining a specific generative result utilizing diffusion, retraining is usually required. However, its retraining is challenging due to the large number of parameters and the immense data. Fortunately, classifier guidance [8] has explored a guidance inference paradigm without retraining, which is defined as follows:

$$\widetilde{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{x_t}; t; \boldsymbol{y}) = \boldsymbol{\epsilon_\theta}(\boldsymbol{x_t}; t; \boldsymbol{y}) + \sigma_t \nabla_{x_t}\mathcal{L}(\boldsymbol{x_t}), \quad (3)$$

where $\boldsymbol{y}$ denotes the optional conditioning signal, and $\mathcal{L}(\boldsymbol{x_t})$ represents the optimization function.

3.2. Method Overview

In general, MotionDiff achieves motion editing through two inference stages. Firstly, given a static scene, users can interactively select the object to be edited and apply motion priors, and then the proposed PKM estimates multi-view optical flows within MFES. Subsequently, MMDS utilizes these optical flows to guide the Stable Diffusion (SD) [30] for motion editing and decouples the motion representation to obtain multi-view consistent editing results. The details of MFES and MMDS are described in Section 3.3 and 3.4.

3.3. Multi-view Flows Estimation Stage

Optical flow possesses the ability to represent the pixel-level motion. However, directly obtaining the optical flow from a static scene with no motion properties is highly challenging. Therefore, we propose MFES, allowing users to interactively obtain multi-view optical flows, as shown in Figure 2. Specifically, given a static scene, including multi-view images and reconstructed 3D point clouds (hereinafter referred to as "3D points"), we firstly utilize Mask Clustering [47] to segment the 3D points and select the $\boldsymbol{P_o}$ of interest through user-interactive queries.

Table 1. Nomenclature

$\boldsymbol{K}$	Intrinsic camera matrix	$\boldsymbol{T}$	Camera translation matrix
$\boldsymbol{R}$	Camera rotation matrix	$\boldsymbol{Rot}$	Spatial rotation matrix
$\boldsymbol{P_o}$	Original 3D points	$\boldsymbol{P_m}$	3D points after motion
$\boldsymbol{P_{so}}$	3D **s**parse **o**riginal points	$\boldsymbol{P_{sm}}$	3D **s**parse **m**otion points
$\boldsymbol{P_{ox,y}}$	x, y components of P_o	$\boldsymbol{f_s}$	Single-view optical flow
$\boldsymbol{f_{sx,y}}$	x, y components of $\boldsymbol{f_s}$	$\boldsymbol{f_m}$	Multi-view optical flows
foc	Camera focal length	pp	Camere principal point
$c_{x,y}$	2D pixel coordinates	d	2D depth

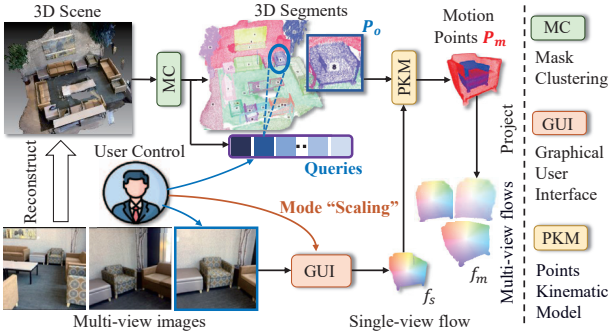


Figure 2. The framework of MFES. Given a static scene, users can interactively select a specific-view image from provided multi-view images and the 3D object of interest through queries (*e.g.*, ID 8 represents a "sofa"). Then the multi-view optical flows are accurately obtained by PKM with user-defined motion mode.

Our core insight is to estimate the 3D points $\boldsymbol{P_m}$ after motion. In this way, we can obtain the corresponding multi-view flows $\boldsymbol{f_m}$ by projecting $\boldsymbol{P_m}$ and $\boldsymbol{P_o}$ to the 2D space:

$$\boldsymbol{f_m} = foc \cdot [\boldsymbol{R}|\boldsymbol{T}]^{-1} \odot (\tfrac{\boldsymbol{P_m}}{d} - \tfrac{\boldsymbol{P_o}}{d}). \quad (4)$$

$\boldsymbol{P_o}$ can be easily obtained utilizing the existing data, while it is challenging to obtain $\boldsymbol{P_m}$. Therefore, we propose the **Points Kinematic Model (PKM)** [2] to estimate $\boldsymbol{P_m}$ for different motion modes, including translation, scaling, rotation, and stretching, as shown in Figure 3. Specifically, we project the selected image to obtain the 3D sparse original points $\boldsymbol{P_{so}}$. Meanwhile, we design a GUI to interactively generate a single-view optical flow $\boldsymbol{f_s}$, and the 3D sparse motion points $\boldsymbol{P_{sm}}$ are represented by $\boldsymbol{f_s}$:

$$\boldsymbol{P_{sm}} = d\boldsymbol{K}[\boldsymbol{R}|\boldsymbol{T}][\frac{c_x + \boldsymbol{f_{sx}} - pp}{foc}, \frac{c_y + \boldsymbol{f_{sy}} - pp}{foc}, 1]. \quad (5)$$

Based on the above definitions, we introduce the details of PKM for different motion modes (refer to Figure 3):

[2] For more formulas details, please refer to the supplementary materials.

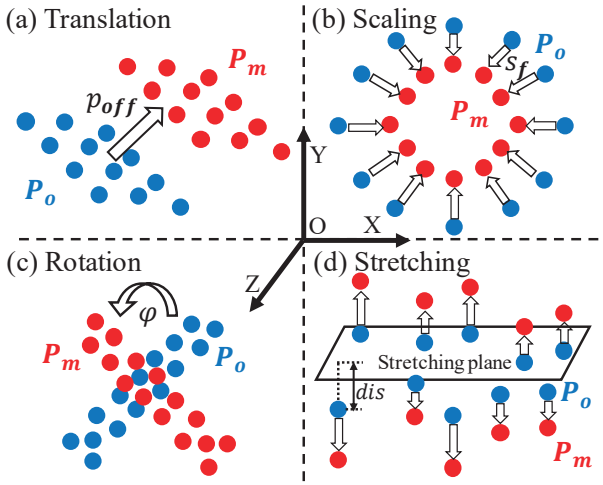


Figure 3. The diagrams of different motion modes in PKM.

1) Translation. The translational motion of 3D points is spatially represented as a 3D offset $\boldsymbol{p_{off}}$ between $\boldsymbol{P_m}$ and $\boldsymbol{P_o}$. We contemplate utilizing $\boldsymbol{P_{sm}}$ and $\boldsymbol{P_{so}}$ to estimate this $\boldsymbol{p_{off}}$ based on Eq. (6):

$$\boldsymbol{P_m} = \boldsymbol{P_o} + \boldsymbol{p_{off}}, \quad \boldsymbol{p_{off}} = \frac{\sum_{i=0}^{N(\boldsymbol{P_{so}})} (\boldsymbol{P_{sm}} - \boldsymbol{P_{so}})}{N(\boldsymbol{P_{so}})}, \tag{6}$$

where $N(\boldsymbol{P_{so}})$ denotes the number of $\boldsymbol{P_{so}}$.

2) Scaling. We assume that there exists a scaling factor s_f between $\boldsymbol{P_o}$ and $\boldsymbol{P_m}$. Therefore, we calculate the s_f based on the single-view optical flow $\boldsymbol{f_s}$.

For **shrinkage**, the s_f can be expressed as:

$$s_f = \frac{\sum_{i=0}^{N} ||\boldsymbol{f_s}||}{N \cdot Max(||\boldsymbol{f_s}||)}, \quad ||\boldsymbol{f_s}|| = \sqrt{\boldsymbol{f_{sx}}^2 + \boldsymbol{f_{sy}}^2}, \tag{7}$$

where $Max(\cdot)$ denotes the maximum value, while N represents the number of non-zero optical flow values.

For **enlargement**, we obtain the s_f by calculating the ratio of the occluded region $\boldsymbol{o_r}$ to the $\boldsymbol{f_s}$. Therefore, we need to create the $\boldsymbol{o_r}$, and the s_f can be calculated:

$$s_f = \frac{\boldsymbol{o_r}}{r(\boldsymbol{f_s})}, \quad \boldsymbol{o_r} = L(c_x, c_y, \boldsymbol{f_s}) \cup r(\boldsymbol{f_s}), \tag{8}$$

where $L(\cdot)$ represents the linear sampling function, $r(\cdot)$ represents the region of non-zero optical flow.

Thus, the $\boldsymbol{P_m}$ of above scaling motion can be defined as:

$$\boldsymbol{P_m} = s_f \cdot \boldsymbol{P_o}. \tag{9}$$

3) Rotation. For rotation, all 3D points have the same rotation angle φ. Fortunately, we can conveniently obtain φ based on the designed GUI. Therefore, the spatial rotation matrix $\boldsymbol{Rot}$ can be defined as:

$$\boldsymbol{Rot} = \begin{bmatrix} cos\varphi & -sin\varphi & 0 \\ sin\varphi & cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{10}$$

$$\boldsymbol{P_m} = \boldsymbol{Rot} \odot (\boldsymbol{P_o} - \boldsymbol{p_c}) + \boldsymbol{p_c}, \tag{11}$$

where $\boldsymbol{p_c}$ represents the centroid of $\boldsymbol{P_o}$.

4) Stretching. For stretching, we assume $\boldsymbol{P_o}$ are segmented by a 2D stretching plane and are stretched to varying degrees along this plane. Since the $\boldsymbol{f_s}$ is obtained in the XY plane, the stretching plane is perpendicular to XY plane and parallel to Z-axis:

$$Ax + By + D = 0. \tag{12}$$

Therefore, in order to solve this plane in 3D space, we need to find two 3D points $\boldsymbol{P_1} = (x_1, y_1, z_1), \boldsymbol{P_2} = (x_2, y_2, z_2)$ located on this plane. Since Eq.(12) defines the relationship between the stretching plane and image, we can conveniently obtain two 2D points on the stretching line from GUI. Then, we project them to get two 3D points, thus calculating the unique 2D stretching plane:

$$\begin{cases} A = y_2 - y_1, \quad B = x_1 - x_2, \\ D = (x_2 - x_1)y_1 - (y_2 - y_1)x_1 \end{cases}, \tag{13}$$

therefore, the distance of $\boldsymbol{P_o}$ to this plane $\boldsymbol{dis}$ and the stretching factor $\boldsymbol{t_f}$ are expressed as:

$$\boldsymbol{dis} = \frac{A\boldsymbol{P_{ox}} + B\boldsymbol{P_{oy}} + D}{\sqrt{A^2 + B^2}}, \quad \boldsymbol{t_f} = \frac{\boldsymbol{dis}}{|Max(\boldsymbol{dis})|}. \tag{14}$$

Therefore, the $\boldsymbol{P_m}$ of stretching motion are defined as:

$$\boldsymbol{P_m} = \boldsymbol{P_o} + \boldsymbol{t_f} \odot Max(\boldsymbol{P_{sm}} - \boldsymbol{P_{so}}). \tag{15}$$

In summary, by designing the PKM, we can estimate the 3D motion points $\boldsymbol{P_m}$, and subsequently obtain multi-view optical flows $\boldsymbol{f_m}$ according to $\boldsymbol{P_m}$ and $\boldsymbol{P_o}$ in Eq. (4).

3.4. Multi-view Motion Diffusion Stage

Controlling diffusion models for motion editing often requires retraining, which is demanding on computational resources cost and data. Therefore, in MMDS (Figure 4), we propose a training-free zero-shot diffusion model, leveraging the multi-view optical flows obtained from MFES to guide Stable Diffusion (SD) [30] in completing motion editing task. **Our core insight is to decouple the motion process into a combination of static background, motion objects, and occluded region, as shown in Figure 5.**

Specifically, in all diffusion steps, we utilize DDIM Inversion [37] to obtain the non-moving region latents to prevent the **static background** structure from being tampered with. Meanwhile, we design the Flow Guidance Strategy (FGS) in all steps to guide SD for motion editing. From the T step, we introduce the processed Latent Space Fusion (LSF) strategy to maintain the texture details of the generated **motion objects**. To guarantee multi-view consistency of the generated **occluded region**, we introduce the Background Grid Constrain (BGC). Finally, we utilize VAE [18]

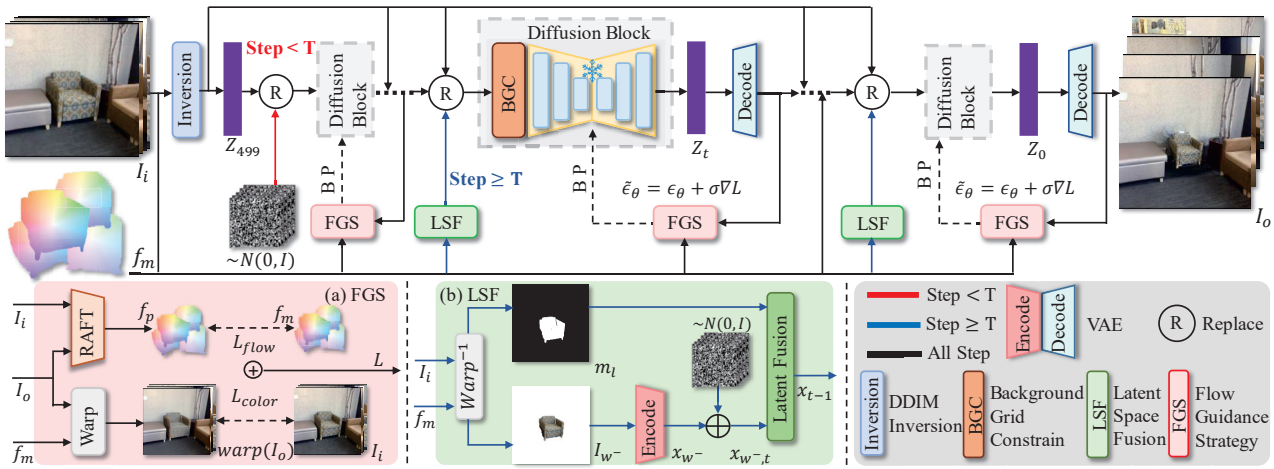


Figure 4. The pipeline of MMDS. Given multi-view images and their optical flows, we perform DDIM inversion in all diffusion steps to preserve the static background from being tampered with. In the first T steps, only FGS is used as the guidance to ensure a reasonable motion trend. From $T+1$ step, both FGS and LSF are utilized to obtain corresponding motion with fidelity of texture details. Meanwhile, BGC is embedded into the diffusion model in all steps to maintain multi-view consistency.

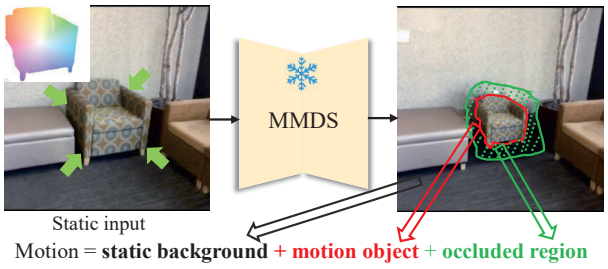


Figure 5. The motion decoupled representation.

to decode the predicted noise latents and obtain multi-view motion results. In the following, we describe the details of decoupled motion representation strategies in MMDS:

Flow Guidance Strategy. In FGS, we aim to utilize multi-view optical flows to guide SD in zero-shot motion editing. Firstly, we employ Recurrent All-Pairs Field Transforms (RAFT) [40] to predict the optical flows $\boldsymbol{f_p}$ from the output images $\boldsymbol{I_o}$ and input images $\boldsymbol{I_i}$, and then calculate the flow loss $\mathcal{L}_{flow}$. Simultaneously, we apply the warp transformation to $\boldsymbol{I_o}$ and compute the color loss $\mathcal{L}_{color}$:

$$\mathcal{L}_{flow} = ||\boldsymbol{f_m} - \boldsymbol{f_p}||_1, \ \boldsymbol{f_p} = RAFT(\boldsymbol{I_i}, \boldsymbol{I_o}), \quad (16)$$

$$\mathcal{L}_{color} = ||\boldsymbol{I_i} - warp(\boldsymbol{I_o})||_1. \quad (17)$$

Subsequently, these two losses are used to update gradients in Eq. (3) and guide the diffusion for motion editing.

Latent Space Fusion. However, directly applying FGS leads to texture details changes of **motion objects**, as shown in Figure 8(d). Thus, we propose LSF to achieve high-quality texture fidelity without increasing the computational cost. Specifically, we first perform inverse warp transformation on $\boldsymbol{I_i}$ to obtain the target object $\boldsymbol{I_{w^-}}$ and its latent space mask $\boldsymbol{m_l}$, then encode the $\boldsymbol{I_{w^-}}$ using a VAE encoder:

$$\boldsymbol{I_{w^-}} = warp^{-1}(\boldsymbol{I_i}, \boldsymbol{f_m}), \ \boldsymbol{x_{w^-}} = E(\boldsymbol{I_{w^-}}), \quad (18)$$

where E denotes the VAE encoder.

Then, in accordance with the forward process of DDIM, we add noise to the encoded image $\boldsymbol{x_{w^-}}$ based on Eq. (1):

$$\boldsymbol{x_{w^-,t}} = \sqrt{a_t}\boldsymbol{x_{w^-}} + \sqrt{1-a_t}\boldsymbol{\epsilon}, \boldsymbol{\epsilon} \sim \mathcal{N}(0, I), \quad (19)$$

where $\sqrt{a_t}$ denotes the hyperparameter related to the process of adding noise. Finally, we perform latent fusion of the forward noise $\overline{\boldsymbol{x_{t-1}}}$ predicted by the Eq. (2):

$$\boldsymbol{x_{t-1}} = \boldsymbol{m_l} \odot \boldsymbol{x_{w^-,t}} + (1 - \boldsymbol{m_l}) \odot \overline{\boldsymbol{x_{t-1}}}. \quad (20)$$

Through LSF, we can maintain the texture details of the motion objects, shown in Figure 8(f).

Background Grid Constrain. Benefiting from FGS and LSF strategies, we can achieve motion editing with consistent texture details of motion objects. However, the lack of effective multi-view consistency constraints results in uncontrollable content generation within the **occluded region**, as shown in Figure 9(b). Previous works [4, 44] found that using grid constraints during the generation process can benefit the multi-view consistency. Therefore, we aim to introduce a relevant constraint that preserves the consistency of occluded region across different views. Hence, we propose the Background Grid Constrain (BGC), as shown in Figure 6, which is defined as follows:

$$\widetilde{\epsilon_\theta}(\boldsymbol{x_t};\ t;\ \boldsymbol{y}) = G^{-1}(\epsilon_\theta(G(\boldsymbol{x_t});\ t;\ \boldsymbol{y})), \quad (21)$$

where G denotes the grid transformation.

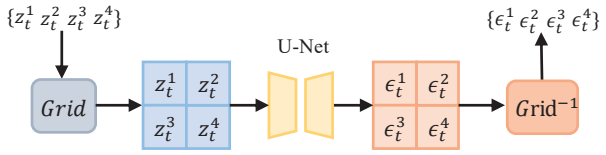


Figure 6. The framework of BGC, which can enforce multi-view consistency without introducing additional computational costs.

Specifically, a latent tensor $\mathcal{Z}$ with the shape of $(4, 4, 64, 64)$ undergoes a transformation through G, its size changes to $(1, 4, 128, 128)$. Then, U-Net [31] is utilized to predict the noise latent $\boldsymbol{\epsilon}$, and we perform inverse grid transformation G^{-1} to obtain the latent with shape $(4, 4, 64, 64)$.

Through this grid-based constrain, BGC can leverage the multi-view global information to complement each other, thus improving multi-view consistency without increasing the computation cost. In summary, through the designed MFES and MMDS, our MotionDiff can achieve complex multi-view consistent motion editing for a static scene.

4. Experiments

4.1. Implementations Details

We implement our method on four 48G NVIDIA RTX A6000 GPUs, and only a single GPU is required to complete an inference process. Our experiments are conducted on the widely-recognized ScanNet200 [7, 32], which comprises of 312 scenes with multi-view images and 3D points. For segmenting 3D points and estimating optical flows, we utilize the frozen Mask Clustering [47] and RAFT [40]. The SD-1.5 [30] model is served as our denoising framework. All input and output images are processed at 512 $\times$ 512 resolution. DDIM sampling is adopted with 500 steps and the classifier-free guidance scale is set to 7.5. To ensure a fair experimental setup, we follow Motion Guidance [12] protocol, incorporating five distinct motion modes. Subsequently, we perform both quantitative and qualitative comparisons against benchmark methods, including DiffEditor [26], Motion Guidance [12], and MagicFixup [2].

4.2. Quantitative Comparisons

Previous employed image editing metrics, such as FID [34] and CLIP Similarity [11, 29], are inadequate for motion editing task due to the significant spatial geometric transformations and the emergence of diverse novel regions. Additionally, the absence of paired input-motion datasets renders traditional subjective metrics like PSNR and SSIM [43] unsuitable. Consequently, we introduce three multi-view motion editing evaluation metrics:

Motion Position Accuracy (MPA), calculated by RAFT to represent the motion precision between input and output:

$$MPA = \lambda_{mpa}\ \mathcal{L}_{flow}, \tag{22}$$

-	Translation	Scaling	Rotation	Stretching
DE	14.7/10.5/33.51	15.7/19.4/33.34	18.1/14.3/27.45	7.4/16.3/36.31
MG	59.5/25.1/29.31	53.0/20.1/28.33	39.3/14.6/22.84	35.7/30.8/35.53
MF	26.5/9.2/33.24	10.0/15.6/34.29	17.0/14.1/29.78	6.2/7.0/37.78
Ours	**10.8/4.6/34.38**	**6.7/6.2/34.76**	**10.6/11.1/30.02**	**3.1/3.4/38.62**

Table 2. Quantitative comparisons with other physics-based generative motion editing methods. Each column represents one specific motion on **MPA** $\downarrow$ **/ ATP** $\downarrow$ **/ MVC** $\uparrow$ metrics. The **DE, MG, MF** represent DiffEditor, Motion Guidance, and MagicFixup, respectively. MotionDiff achieves the SOTA quantitative performance.

Appearance Texture Fidelity (ATF), denoting the extent of texture appearance change in the motion objects:

$$ATF = \lambda_{atf}\ ||\boldsymbol{m} \odot \boldsymbol{I}_{\boldsymbol{w}^-},\ \boldsymbol{m} \odot \boldsymbol{I}_{\boldsymbol{o}}||_1, \tag{23}$$

where $\boldsymbol{m}$ represents the non-zero regions masks of $\boldsymbol{f_m}$. We empirically set λ_{mpa} to 10, λ_{atf} to 100.

Multi-View Consistency (MVC), we adopt Overlapping PSNR proposed by MVDiffusion [38]:

$$MVC = P(\boldsymbol{I_1}, pt(\boldsymbol{I_2}, \boldsymbol{K} \odot \boldsymbol{R_1^{-1}} \odot \boldsymbol{R_2} \odot \boldsymbol{K^{-1}})), \tag{24}$$

where P represents the standard PSNR, $\boldsymbol{I_{1,2}}$ denote two similar images, pt signifies the perspective transformation.

In Table 2, we present quantitative comparisons of each motion modes. Other methods exhibit unsatisfactory results in terms of MPA and ATF metrics, primarily due to inadequate motion accuracy and compromised appearance fidelity. Additionally, these methods struggle to maintain multi-view consistency, resulting in lower MVC values. In contrast, MotionDiff demonstrates SOTA performance across all metrics, attributed to its capability in achieving precise texture-consistent multi-view motion editing.

4.3. Qualitative Comparisons

To comprehensively evaluate the performance of MotionDiff, we provide detailed qualitative comparisons against other benchmark methods.

We input multi-view images, and fairly apply corresponding motions to all methods. As illustrated in Figure 7, we present visual comparisons for translation, scaling, rotation, and stretching, respectively. Each row showcases the editing outcomes of different methods, while each column represents the different views from a single scene. To better display the motion results, we zoom in the **box regions**. It demonstrates that DiffEditor [26] can manage simple motions, such as translation and scaling to some extent, but is limited in handling more complex motions (*e.g.*, the appearance of the blackboard in rotation is unsatisfactory). Additionally, Motion Guidance [12] struggles to maintain the appearance of motion objects, due to lacking an effective texture supervision strategy, for instance, the texture details of the red pillow are lost during translation. Moreover, MagicFixup [2] achieves satisfactory results in single-view motion editing, but its multi-view motion editing performance

Figure 7. Visual comparisons between MotionDiff and other physics-based motion editing methods on translation, scaling, rotation, and stretching tasks. Our method achieves better motion editing performance while maintaining multi-view consistency.

is limited due to the inadequate consistency constraints. In contrast, MotionDiff excels in executing diverse motions, while effectively preserving multi-view consistency.

4.4. Ablation Studies

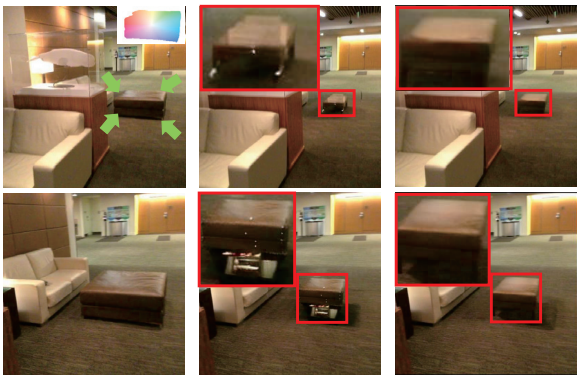
To validate the efficacy of the proposed strategies, we design corresponding ablation studies. The visualization and metric outcomes are presented in Figure 8, 9 and Table 3.

W/o inversion. In Figure 8(b), the absence of DDIM inversion substitution makes it difficult to maintain a **static background**, generating uncontrollable content.

W/o FGS. As depicted in Figure 8(c), the completion of physical motion becomes challenging without FGS, as there

(a) Input (b) w/o inversion (c) w/o FGS (d) w/o LSF (e) LSF in image space (f) Ours

Figure 8. Visual comparisons of our ablation studies of MotionDiff.



(a) Input (b) w/o BGC (c) Ours

Figure 9. Visual ablation study of BGC. It demonstrates that BGC is advantageous in ensuring consistency across multi-view images.

is no motion prior provided to the diffusion model.

W/o LSF. As illustrated in Figure 8(d), the absence of the LSF strategy leads to inadequate supervision over **motion objects**, thereby failing to guarantee the texture details.

LSF in image space. To evaluate the effectiveness of the latent fusion strategy, we transform it into the image space. However, as shown in Figure 8(e), due to the substantial distribution difference between these two spaces, the quality of the generated content hardly be guaranteed.

W/o BGC. As indicated in Figure 9(b), the absence of multi-view constraints results in inconsistent generation of the **occluded region**. In summary, extensive ablation studies validate the efficacy of the proposed strategies. Through their integration, MotionDiff accomplishes sophisticated multi-view consistent motion editing without retraining.

-	w/o inversion	w/o FGS	w/o LSF	LSF in image	w/o BGC	Ours
MPA ↓	9.5	9.4	7.7	7.7	7.2	**6.8**
ATF ↓	5.6	10.2	**5.1**	7.5	5.3	**5.1**
MVC ↑	28.96	32.14	31.79	30.23	30.22	**34.92**

Table 3. Quantitative results of the ablation studies.

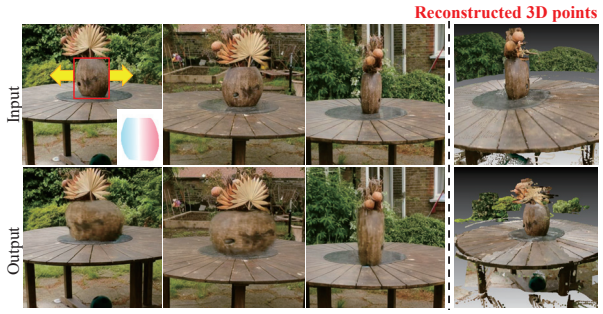


Figure 10. Given sparse-view images, MotionDiff obtains corresponding editing results with reconstructed 3D points by DUSt3R.

4.5. Extension Experiments

Given that ScanNet dataset provides 3D points for each scene, we utilize these directly for simplicity. **However, it is important to note that our method can also operate using multi-view images as the only input**, with 3D points being reconstructed through established techniques, such as DUSt3R [42] and MV-DUSt3R+ [39]. To validate this, we select arbitrary **outdoor sparse-view** images from Mip-NeRF 360 [5], and reconstruct corresponding 3D points using DUSt3R. Figure 10 demonstrates the motion results, further validating the editing practicality of our method.

5. Conclusion and Limitations

A training-free zero-shot diffusion framework MotionDiff is proposed, facilitating diverse complex multi-view motion editing tasks. An convenient interactive framework is designed within MFES to estimate optical flow, while the motion process is decoupled in MMDS, allowing users to freely adopt it for other down-stream tasks. Nonetheless, the proposed PKM somewhat limits the motion modes. establishing corresponding models for other motions in PKM is required, while we believe it is straightforward to implement. In future work, we will focus on integrating more general motion representation for generative editing.

6. Acknowledgment

This work was supported by Guangdong S&T Program No. 2024B1111060003, the National Natural Science Foundation of China under Grant U24A20251, 62071500.

References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4432–4441, 2019. 2

[2] Hadi Alzayer, Zhihao Xia, Xuaner Zhang, Eli Shechtman, Jia-Bin Huang, and Michael Gharbi. Magic fixup: Streamlining photo editing by watching dynamic videos. *ACM Trans. Graph.*, 2025. 2, 6

[3] Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 18208–18218, 2022. 2

[4] Amir Bar, Yossi Gandelsman, Trevor Darrell, Amir Globerson, and Alexei Efros. Visual prompting via image inpainting. *Advances in Neural Information Processing Systems*, 35:25005–25017, 2022. 5

[5] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5470–5479, 2022. 8

[6] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18392–18402, 2023. 1, 2

[7] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5828–5839, 2017. 6

[8] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021. 3

[9] Dave Epstein, Allan Jabri, Ben Poole, Alexei Efros, and Aleksander Holynski. Diffusion self-guidance for controllable image generation. *Advances in Neural Information Processing Systems*, 36:16222–16239, 2023. 1

[10] Kunyu Feng, Yue Ma, Bingyuan Wang, Chenyang Qi, Haozhe Chen, Qifeng Chen, and Zeyu Wang. Dit4edit: Diffusion transformer for image editing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2969–2977, 2025. 2

[11] Rinon Gal, Or Patashnik, Haggai Maron, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. Stylegan-nada: Clip-guided domain adaptation of image generators. *ACM Transactions on Graphics (TOG)*, 41(4):1–13, 2022. 6

[12] Daniel Geng and Andrew Owens. Motion guidance: Diffusion-based image editing with differentiable motion estimators. *International Conference on Learning Representations*, 2024. 2, 6

[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in Neural Information Processing Systems*, 27, 2014. 1, 2

[14] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-or. Prompt-to-prompt image editing with cross-attention control. In *The Eleventh International Conference on Learning Representations*. 2

[15] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 2, 3

[16] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42 (4), 2023. 2

[17] Levon Khachatryan, Andranik Movsisyan, Vahram Tadevosyan, Roberto Henschel, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. Text2video-zero: Text-to-image diffusion models are zero-shot video generators. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15954–15964, 2023. 2

[18] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 4

[19] Ruining Li, Chuanxia Zheng, Christian Rupprecht, and Andrea Vedaldi. Dragapart: Learning a part-level motion prior for articulated objects. *ECCV 2024*, 2024. 2

[20] Zhengqi Li, Richard Tucker, Noah Snavely, and Aleksander Holynski. Generative image dynamics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24142–24153, 2024. 2

[21] Guan Luo, Tian-Xing Xu, Ying-Tian Liu, Xiao-Xiong Fan, Fang-Lue Zhang, and Song-Hai Zhang. 3d gaussian editing with a single image. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 6627–6636, 2024. 2, 3

[22] Yikun Ma, Dandan Zhan, and Zhi Jin. Fastscene: Text-driven fast indoor 3d scene generation via panoramic gaussian splatting. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pages 1173–1181, 2024. 2

[23] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*. 2

[24] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, pages 405–421, 2020. 2

[25] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6038–6047, 2023. 1

[26] Chong Mou, Xintao Wang, Jiechong Song, Ying Shan, and Jian Zhang. Diffeditor: Boosting accuracy and flexibility on diffusion-based image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8488–8497, 2024. 2, 6

[27] Xingang Pan, Ayush Tewari, Thomas Leimkühler, Lingjie Liu, Abhimitra Meka, and Christian Theobalt. Drag your gan: Interactive point-based manipulation on the generative image manifold. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–11, 2023. 2

[28] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *The Eleventh International Conference on Learning Representations*. 2

[29] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 6

[30] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 1, 2, 3, 4, 6

[31] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015. 3, 6

[32] David Rozenberszki, Or Litany, and Angela Dai. Language-grounded indoor 3d semantic segmentation in the wild. In *European Conference on Computer Vision*, pages 125–141. Springer, 2022. 6

[33] Rahul Sajnani, Jeroen Vanbaar, Jie Min, Kapil Katyal, and Srinath Sridhar. Geodiffuser: Geometry-based image editing with diffusion models. *arXiv preprint arXiv:2404.14403*, 2024. 2

[34] Maximilian Seitzer. pytorch-fid: FID Score for PyTorch. https://github.com/mseitzer/pytorch-fid, 2020. Version 0.3.0. 6

[35] Yujun Shi, Chuhui Xue, Jun Hao Liew, Jiachun Pan, Hanshu Yan, Wenqing Zhang, Vincent YF Tan, and Song Bai. Dragdiffusion: Harnessing diffusion models for interactive point-based image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8839–8849, 2024. 2

[36] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. In *The Eleventh International Conference on Learning Representations*. 2

[37] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*. 2, 3, 4

[38] Shitao Tang, Fuyang Zhang, Jiacheng Chen, Peng Wang, and Yasutaka Furukawa. Mvdiffusion: Enabling holistic multi-view image generation with correspondence-aware diffusion. In *Advances in Neural Information Processing Systems*, pages 51202–51233, 2023. 6

[39] Zhenggang Tang, Yuchen Fan, Dilin Wang, Hongyu Xu, Rakesh Ranjan, Alexander Schwing, and Zhicheng Yan. Mv-dust3r+: Single-stage scene reconstruction from sparse views in 2 seconds. *CVPR*, 2025. 8

[40] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer, 2020. 5, 6

[41] Ruicheng Wang, Jianfeng Xiang, Jiaolong Yang, and Xin Tong. Diffusion models are geometry critics: Single image 3d editing using pre-trained diffusion priors. In *ECCV*, pages 441–458, 2024. 2, 3

[42] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20697–20709, 2024. 8

[43] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 6

[44] Ethan Weber, Aleksander Holynski, Varun Jampani, Saurabh Saxena, Noah Snavely, Abhishek Kar, and Angjoo Kanazawa. Nerfiller: Completing scenes via generative 3d inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20731–20741, 2024. 5

[45] Ziyi Wu, Yulia Rubanova, Rishabh Kabra, Drew Hudson, Igor Gilitschenski, Yusuf Aytar, Sjoerd van Steenkiste, Kelsey Allen, and Thomas Kipf. Neural assets: 3d-aware multi-object scene synthesis with image diffusion models. *Advances in Neural Information Processing Systems*, 37: 76289–76318, 2025. 2, 3

[46] Tianyi Xie, Zeshun Zong, Yuxing Qiu, Xuan Li, Yutao Feng, Yin Yang, and Chenfanfu Jiang. Physgaussian: Physics-integrated 3d gaussians for generative dynamics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4389–4398, 2024. 2, 3

[47] Mi Yan, Jiazhao Zhang, Yan Zhu, and He Wang. Maskclustering: View consensus based mask graph clustering for open-vocabulary 3d instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28274–28284, 2024. 3, 6

[48] Zexuan Yan, Yue Ma, Chang Zou, Wenteng Chen, Qifeng Chen, and Linfeng Zhang. Eedit: Rethinking the spatial and temporal redundancy for efficient image editing. *arXiv preprint arXiv:2503.10270*, 2025. 2

[49] Jiraphon Yenphraphai, Xichen Pan, Sainan Liu, Daniele Panozzo, and Saining Xie. Image sculpting: Precise object editing with 3d geometry control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4241–4251, 2024. 2, 3

[50] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In

Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 3836–3847, 2023. 1, 2

[51] Tianyuan Zhang, Hong-Xing Yu, Rundi Wu, Brandon Y Feng, Changxi Zheng, Noah Snavely, Jiajun Wu, and William T Freeman. Physdreamer: Physics-based interaction with 3d objects via video generation. In *European Conference on Computer Vision*, pages 388–406, 2025. 2

[52] Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. In-domain gan inversion for real image editing. In *European Conference on Computer vision*, pages 592–608. Springer, 2020. 2