

# Diffusion-Based Attention Warping for Consistent 3D Scene Editing

Eyal Gomel Lior Wolf  
Tel-Aviv University

## Abstract

*We present a novel method for 3D scene editing using diffusion models, designed to ensure view consistency and realism across perspectives. Our approach leverages attention features extracted from a single reference image to define the intended edits. These features are warped across multiple views by aligning them with scene geometry derived from Gaussian splatting depth estimates. Injecting these warped features into other viewpoints enables coherent propagation of edits, achieving high fidelity and spatial alignment in 3D space. Extensive evaluations demonstrate the effectiveness of our method in generating versatile edits of 3D scenes, significantly advancing the capabilities of scene manipulation compared to the existing methods. Project page: <https://attention-warp.github.io>*

## 1. Introduction

Recent advances in diffusion models have revolutionized the landscape of 2D image editing, demonstrating unprecedented capabilities in image editing, style transfer and image inpainting [10, 11, 19, 34, 46, 49, 68]. While these achievements have solidified the position of diffusion models as the de facto standard for image editing tasks, extending such capabilities to 3D scene editing presents unique challenges. Several recent contributions have attempted to bridge this gap by applying 2D diffusion models to multiple views of a 3D scene, typically utilizing the same views employed in the scene’s reconstruction [17, 52, 59]. While editing based on a single view proves inadequate for comprehensive 3D manipulation, the simultaneous editing of multiple views introduces significant challenges in maintaining edit consistency across perspectives.

Prior approaches have addressed the consistency challenge through various mechanisms, predominantly focusing on information propagation between views [7, 12, 28, 58]. However, such approaches frequently result in a loss of edit fidelity and conceptual clarity, as the attempt to reconcile potentially conflicting information from multiple views leads to blurred or compromised results.

In contrast, we propose a novel paradigm that leverages

only a single-view edit as the primary manipulation source, then systematically projects the edit attention feature maps onto other views using the underlying 3D scene structure. This innovative use of attention warping ensures that edits are propagated consistently across different perspectives without processing multiple frames simultaneously, significantly reducing computational complexity.

A key innovation in our method is the incorporation of a geometry-guided warping mechanism that utilizes the depth and structural information of the scene to accurately map edits across views, maintaining spatial coherence and alignment with the 3D scene’s structure. Additionally, we propose masking and blending techniques that exploit Gaussian splatting properties, such as Gaussian normal vectors, to prevent warping to occluded or misaligned regions. These techniques ensure smooth transitions and consistency across views, refining the edit quality and preserving realistic integration throughout the 3D model. Our contributions enable high-quality, multi-view-consistent 3D edits that are computationally efficient and robust.

The effectiveness of our approach is demonstrated through extensive experimental validation across a diverse range of editing scenarios and scene types. Our method consistently outperforms existing approaches in terms of edit quality, spatial consistency, and semantic fidelity, as verified through both quantitative metrics and user studies.

## 2. Related Work

Our method builds upon advancements in image-text-based editing using diffusion models [5, 46, 49, 66], Gaussian Splatting [20, 26] representations, and 3D scene editing techniques [7, 8, 12, 17, 28, 52, 56, 58]. We review these key areas and highlight how our approach differs from existing methods.

**Text Based Image Editing w/ Diffusion Models** Diffusion models have become essential for text-guided image editing by leveraging a noise-adding and denoising process to modify images with precision. This process involves iteratively refining a noisy image  $\mathbf{x}_T$  across timesteps  $t$  using learned noise predictions  $\epsilon_\theta(\mathbf{x}_t, \mathbf{t})$ .

Central to diffusion models, especially in text-guided applications, are self-attention and cross-attention mecha-

nisms within its U-Net structure [47]. Self-attention captures dependencies across different regions within the image, enhancing coherence and structure during generation. Cross-attention incorporates external guidance, such as text prompts or conditioning images, by mapping relationships between image features and the conditioning input.

InstructPix2Pix [5] is an image-to-image translation method built upon Stable Diffusion [46]. It fine-tunes Stable Diffusion using synthetic instruction data, enabling the model to perform targeted image editing based on both textual prompts and reference images for structural alignment. This approach is part of a broader class of image-to-image translation methods [1, 18, 25, 36, 38, 43, 51].

ControlNet [66] enhances the general diffusion framework by introducing trainable auxiliary control structures that condition the generation process on additional inputs, such as depth maps, edges, and segmentation masks. Unlike traditional cross-attention conditioning, ControlNet incorporates a parallel trainable path that merges with the original diffusion model.

Our method extends these approaches by editing the content guided by attention mechanisms and also warping the self and cross-attention feature maps across views. This attention warping propagates edits consistently in 3D scenes, ensuring alignment across different viewpoints. This unique handling of attention feature maps sets our method apart by addressing multi-view consistency challenges more effectively than traditional 2D-based approaches.

**3D Scene Representation** Neural Radiance Fields (NeRF) [37] and derivative works [2, 3, 32, 39, 57] have opened up new possibilities in computer vision and computer graphics, including 3D scene reconstruction, editing, segmentation, etc. These models represent scenes as continuous 3D functions, synthesizing novel views by learning volumetric density and color at each 3D point.

3D Gaussian Splatting [26] (3DGS) represents scenes using discrete 3D Gaussian elements, enabling efficient rendering while capturing complex details. Some methods build upon this representation to enhance depth and normal consistency, such as 2DGs [20] which improves 3DGS by collapsing the 3D volume into 2D oriented planar Gaussian disks, ensuring view-consistent geometry and intrinsic surface modeling. 2DGs employs a perspective-accurate splatting process using ray-splat intersections. Depth distortion and normal consistency regularization further enhance the reconstruction quality, supporting detailed geometry.

Our approach leverages the 2DGs representation to facilitate scene edits, incorporating attention-based diffusion models to modify the splats’ attributes while preserving scene integrity.

**3D Scene Editing with Diffusion Models** 3D scene editing and stylization are pivotal in computer vision, enabling diverse applications in neural radiance fields. Ap-

proaches like [24, 30, 69] offer controllable scene modifications that adjust geometry and appearance. Methods such as [21, 40, 65] achieve 3D-consistent stylizations by incorporating mutual 2D-3D learning, while techniques like [16, 61] focus specifically on color manipulation.

Leveraging image-text models [13, 45] for guiding 3D generation has been explored in works such as [23, 54, 55]. DreamFusion [44] introduced *Score Distillation Sampling* (SDS), a method that utilizes gradients from diffusion models to guide 3D model updates. This approach has been adopted in several followup studies [9, 31, 42, 44, 48, 70] to apply diffusion priors for refining 3D representations and enabling complex edits.

Additionally, the technique of *Iterative Dataset Update* has been proposed, with Instruct-NeRF2NeRF [17] introducing methods that edit individual views and update the dataset to refine 3D scene representations while maintaining coherence. Works leveraging this strategy, such as GaussianEditor [8, 56], IGS2GS [52] and other related methods [53, 62, 64], have shown promising results but still face limitations due to the inherent challenges of using 2D diffusion models for multiview edits and maintaining consistent geometry across views.

Achieving multiview consistency remains a significant challenge in 3D scene editing. Some methods utilize pre-trained 2D models to ensure temporal coherence. Notable works, including ViCA-NeRF [12], DGE [7], GaussCtrl [59], VCEdit [58], and LatentEditor [28], have proposed various approaches to address this challenge. ViCA-NeRF leverages NeRF depth information to establish pixel correspondences across views, enhancing multiview alignment. DGE, inspired by video generation and editing methods, employs 2D image generators for image sequences, editing multiple views simultaneously using spatio-temporal attention and enforcing epipolar constraints to maintain consistency. GaussCtrl utilizes ControlNet conditioned on depth to guide generation, aligning latents across multiple key views and ensuring coherence through the depth-conditioned model. VCEdit consolidates the cross-attention map space between views by leveraging pretrained Gaussians and aligns latents at each diffusion step through a fine-tuned copy of a Gaussian splatting model. LatentEditor focuses on local editing by optimizing NeRF in the diffusion latent space and applying a latent space mask for localized guidance. These methods represent different strategies for enhancing multiview consistency by leveraging latent representations, attention maps, and depth-based conditioning.

Despite these advancements, the current approaches have limitations: 1. Processing multiple views simultaneously restricts the application of specific edit styles to individual views. 2. Multi-view editing in diffusion models can be computationally intensive and memory-demanding. 3. Existing methods often struggle to use a single, non-

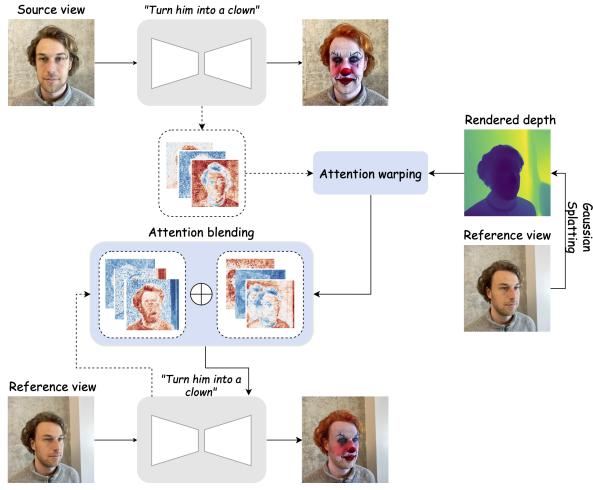


Figure 1. Overview of our method. A single source image is edited using a 2D diffusion model that is conditioned on some prompt. The attention feature maps employed during this process are saved. Given a new reference view, the maps are warped to this view based on the 3D depth map of the reference view. A diffusion model is then applied to the reference view using a blending of the attention feature maps obtained during the diffusion process itself and those that arise from the source view.

diffusion-based edited image to apply consistent edits to a 3D model.

Our approach addresses these limitations by processing one image at a time through a warping mechanism that consistently propagates edits across multiple views. This design provides us the flexibility to select any edited image as the starting point for the warping process, enabling tailored edits that can be applied efficiently. By utilizing attention-based warping, we reduce computational load and memory usage while ensuring that edits are accurately reflected across the 3D model, thereby maintaining coherence and consistency in the final result.

### 3. Method

Our approach enhances 2D diffusion models with attention warping to enable consistent 3D scene editing, by ensuring that edits applied to a single view are coherently propagated across all views of the 3D scene. The method, which is illustrated in Fig. 1, comprises several key components: diffusion-based editing, attention feature map warping, occlusion handling, optional masking, and iterative optimization. Below, we detail each component and the interactions between them.

#### Problem Definition: Diffusion-Based 3D Scene Editing

Given a 3D scene represented by a pretrained 2D Gaussian splatting model  $\mathbf{S} = \{G_i\}_{i=1}^N$ , where each Gaussian  $G_i$  is parameterized by its position, orientation, scale, opacity,

and color, we perform scene editing guided by textual instructions, an unedited image, and either a reference image or a depth map. Specifically, we utilize two types of diffusion models: **InstructPix2Pix** [5]: Takes a textual instruction  $t$  and a guided reference image  $I_G$  to generate edited images. **ControlNet** [66]: Takes a textual instruction  $t$  and depth map  $D$  to guide the editing process.

Depending on the chosen model, the input to the diffusion process is either  $(I, t, I_G)$  or  $(I, t, D)$ . The diffusion model generates edited images  $I'$  based on these inputs, which are then used to fine-tune the Gaussian splatting representation  $S$ .

#### Source View Editing and Attention Feature Map Computation

Select a *source view*  $I_{src}$  from the training set to apply the initial edit. The diffusion model processes  $I_{src}$  along with the corresponding instruction to produce an edited image  $I'_{src}$ . During this editing process, attention feature maps  $\mathbf{F}_{src} = \{\mathbf{F}_{src}^l\}_{l=1}^L$  are computed at each layer  $l$  of the diffusion model, capturing the regions of the image that are influenced by the edit. Each attention feature map  $\mathbf{F}_{src}^l$  comprises both self-attention and cross-attention components:

$$\mathbf{F}_{src}^l = \{\mathbf{F}_{self}^l, \mathbf{F}_{cross}^l\}, \quad \text{where:} \quad (1)$$

$\mathbf{F}_{self}^l = \text{SelfAttention}(\mathbf{Q}, \mathbf{K}, \mathbf{V})$  captures the internal relationships within the source view  $I_{src}$ .  $\mathbf{F}_{cross}^l = \text{CrossAttention}(\mathbf{Q}_{cross}, \mathbf{K}_{cross}, \mathbf{V}_{cross})$  integrates contextual information from external sources or conditioning inputs, which in our case is the textual prompt  $t$ .

**Selected Attention Feature Maps** To focus on detailed and spatially fine-grained edits, we utilize attention feature maps exclusively from the up-sampling blocks at high resolutions (32 and 64). This selection ensures that the most relevant and high-resolution attention information is propagated during the warping process.

Editing both the self- and the cross-attention allows  $\mathbf{F}_{src}^l$  to encapsulate both the internal dynamics of the source view and the influence of external context, facilitating more coherent and context-aware edits.

**Attention Feature Map Warping to Target Views** To ensure consistency across different views, the maps from the source view are warped to target views. This warping leverages depth information and camera transformations. For a target view  $I_{tgt}$ , the warped feature maps  $\mathbf{F}_{warp}^{l,tgt} = \{\mathbf{F}_{warp}^{l,tgt}\}_{l=1}^L$  are obtained as follows:

$$\mathbf{F}_{warp}^{l,tgt} = \mathcal{W}(\mathbf{F}_{src}^l, \mathbf{D}_{tgt}, \mathbf{T}_{src}, \mathbf{T}_{tgt}), \quad (2)$$

where  $\mathcal{W}$  denotes the warping function that utilizes the depth map  $\mathbf{D}_{tgt}$ , along with the camera transformation matrices  $\mathbf{T}_{src}$ ,  $\mathbf{T}_{tgt}$ , to map the attention from the source view to the target view.

The depth map is obtained from the Gaussian splitting

model S. To ensure consistency between views, we compute the normal vectors for each Gaussian in both source and target views, denoted as  $\mathbf{n}_i^{\text{src}}$  and  $\mathbf{n}_i^{\text{tgt}}$ , respectively. This helps manage the differing appearances of objects from various angles. These Gaussians are used to compute the depth map  $\mathbf{D}_{\text{tgt}}$ , which serves as an input to Equation 2, utilized to determine the warp visibility mask. Gaussians with normal angle differences exceeding a threshold  $\theta_{\max} = 60^\circ$ , i.e., the cases in which  $\mathbf{n}_i^{\text{src}} \cdot \mathbf{n}_i^{\text{tgt}} \geq \cos(\theta_{\max})$ , are excluded from the depth rendering process.

Computing the depth based only on Gaussians with similar orientations helps to ensure that only reliable Gaussians contribute to the attention warping, reducing artifacts from mismatched geometries and enhancing the accuracy of the visibility mask.

The camera transformations  $\mathbf{T}$  are obtained by combining various components, including the intrinsic camera parameters  $\mathbf{K}$ : the focal lengths  $f_x$  and  $f_y$ , and the principal point offsets  $c_x$  and  $c_y$ , as well as the 3D rotation matrix  $\mathbf{R}$ . The warping process aligns a target image to a source view by projecting 3D coordinates from the target to the source image plane and is given here completeness. First, pixel coordinates  $\mathbf{p}_{\text{tgt}}$  from the target image are unprojected into 3D space using the depth map  $\mathbf{D}_{\text{tgt}}$  and intrinsic matrix  $\mathbf{K}_{\text{tgt}}$ :  $\mathbf{P}_{\text{tgt}}^c = (\mathbf{K}_{\text{tgt}}^{-1} \mathbf{p}_{\text{tgt}}^T \mathbf{D}_{\text{tgt}})^T$ .

The 3D point  $\mathbf{P}_{\text{tgt}}^c$  is converted to world coordinates using the extrinsic matrix  $\mathbf{R}_{\text{tgt}}$ :  $\mathbf{P} = \mathbf{R}_{\text{tgt}}^{-1}(\mathbf{P}_{\text{tgt}}^c)_h$ , where  $h$  denotes homogeneous coordinates. The world coordinates  $\mathbf{P}$  are transformed to the source camera's coordinates using  $\mathbf{R}_{\text{src}} \cdot \mathbf{P}_{\text{src}}^c = (\mathbf{R}_{\text{src}} \mathbf{P}_h^T)^T$ .

The 2D pixel coordinates in the source view are:

$$u = \frac{f_x \mathbf{P}_{\text{src},x}^c}{\mathbf{P}_{\text{src},z}^c} + c_x, \quad v = \frac{f_y \mathbf{P}_{\text{src},y}^c}{\mathbf{P}_{\text{src},z}^c} + c_y$$

To identify out-of-bounds regions, a mask  $\mathbf{M}$  is defined:

$$\mathbf{M}(u, v) = \begin{cases} 1, & \text{if } 0 \leq u < W \text{ and } 0 \leq v < H \\ 0, & \text{otherwise} \end{cases}, \quad (3)$$

where  $W$  and  $H$  are the target image width and height.

The warping process applies uniformly to both self-attention and cross-attention feature maps within  $\mathbf{F}_{\text{src}}^l$ , ensuring that all relevant attention information is accurately propagated across views.

The warped attention feature maps  $\mathbf{F}_{\text{warp}}^{l,\text{tgt}}$  serve as an additional input of the diffusion model to guide the target view editing:

$$\mathbf{I}'_{\text{tgt}} = \text{DM}(\mathbf{I}_{\text{tgt}}, \mathbf{x}_{\text{tgt}}, \mathbf{t}, \mathbf{F}_{\text{warp}}^{\text{tgt}}), \quad (4)$$

where DM is the diffusion model,  $\mathbf{I}_{\text{tgt}}$  is the target view image,  $\mathbf{x}_{\text{tgt}}$  can be either the guided image  $I_G^{\text{tgt}}$  or the depth map  $\mathbf{D}_{\text{tgt}}$ ,  $\mathbf{t}$  is the textual instruction, and  $\mathbf{F}_{\text{warp}}^{\text{tgt}}$  represents

the set of warped attention feature maps guiding the diffusion process. The usage of the last input parameter within the diffusion model is detailed in Sec. 3.

### Modifying the DM Attention and Handling Occlusions

During the diffusion process for the target image, we blend the warped attention with the attention computed directly from the target view's edit, denoted as  $\mathbf{F}_{\text{new}}^{\text{tgt}}$ , as follows:

$$\mathbf{F}_{\text{masked}}^{l,\text{tgt}} = \mathbf{F}_{\text{warp}}^{l,\text{tgt}} \circ \mathbf{M} + \mathbf{F}_{\text{new}}^{l,\text{tgt}} \circ (1 - \mathbf{M}) \quad (5)$$

$$\mathbf{F}_{\text{final}}^{l,\text{tgt}} = \alpha \circ \mathbf{F}_{\text{masked}}^{l,\text{tgt}} + (1 - \alpha) \circ \mathbf{F}_{\text{new}}^{l,\text{tgt}}, \quad (6)$$

where  $\alpha \in [0, 1]$  is a blending coefficient controlling the influence of the warped attention  $\mathbf{F}_{\text{warp}}^{l,\text{tgt}}$ , and  $\mathbf{M}$  is the binary mask defined in Eq. 3, which ensures that warped attention is only applied to visible regions, while non-visible regions rely solely on the new attention from the target view edit. This blending leads to attention that is correctly applied based on the visibility of regions, maintaining the integrity and realism of the 3D scene during edits. **Decaying the Blending Coefficient:** We gradually decay the blend coefficient  $\alpha$  during the denoising process to balance the influence of the warped and new attention feature maps and reduce the risk of introducing out-of-distribution features as the process progresses.  $\alpha_t = \alpha_0 \cdot (\frac{T-t}{T})$ , where  $\alpha_0 = 0.9$  is the initial blend coefficient,  $t$  is the current denoising timestep, and  $T$  is the total number of timesteps. The warped attention feature maps help define the overall structure early in the process, while later iterations focus on refining details, balancing between warped and new maps.

**Optional Masking with Language-SAM** To enhance edit precision, and following previous contributions [7, 8, 59], we optionally apply a language-guided segmentation mask using Language SAM (combining SAM with Grounding DINO) [29, 33, 35]. This mask  $\mathbf{M}_s$  restricts the diffusion editing process to specific regions of the source view:

$$\mathbf{I}'_{\text{src}} = \text{DM}(\mathbf{I}_{\text{src}}, \mathbf{t}) \circ \mathbf{M}_s \quad (7)$$

The mask ensures that only the targeted regions are modified, preserving the integrity of the surrounding scene. This approach is particularly effective for edits focusing on specific objects within the image, allowing for precise modifications while leaving the rest of the scene unchanged.

**Iterative Optimization** To achieve high consistency and convergence, the editing process is performed iteratively over a small number of iterations (3 iterations in all experiments). The editing pipeline is summarized in Alg. 1; Each iteration involves the following steps:

1. **Subset Editing:** Select a subset of views from the dataset and apply the diffusion-based editing process, generating warped attention feature maps for these views.
2. **GS Optimization:** Fine-tune the Gaussian splatting representation  $\mathbf{S}$  based on the edited images to align with

---

**Algorithm 1** Diffusion-Based Attention Warping for 3D Scene Editing

---

**Require:** Pretrained Gaussian splatting model  $\mathbf{S}$ , textual instruction  $\mathbf{t}$ , image  $\mathbf{I}$  and reference image or depth map  $\mathbf{x}_{tgt}$ , number of stages  $S$

- 1: Generate edited image  $\mathbf{I}'_{src} = DM(\mathbf{I}, \mathbf{x}_{tgt}, \mathbf{t})$
- 2: **for** stage  $s = 1$  to  $S$  **do**
- 3:   Select subset of views  $\{\mathbf{I}_{tgt}\}$
- 4:   **for** each target view  $\mathbf{I}_{tgt}$  **do**
- 5:     Warp  $\mathbf{F}_{warp}^{src}$  attention feature maps:  $\mathbf{F}_{warp}^{tgt}$
- 6:     Generate target view DM  $(\mathbf{I}_{tgt}, \mathbf{t}, \mathbf{x}_{tgt}, \mathbf{F}_{warp}^{tgt})$
- 7:     Fine-tune Gaussian splatting model  $\mathbf{S}$  with  $\mathbf{I}_{tgt}$
- 8:   **end for**
- 9: **end for**

---

the modifications.

This iterative approach allows the model to progressively refine the scene, ensuring that edits remain consistent across all views and that the Gaussian splatting representation accurately reflects the desired changes.

**Subset Editing:** We follow common practice by working on a subset of the data, using 40 samples to ensure a balance between computational efficiency and comprehensive evaluation. The subset of views is selected randomly from images that have not been edited up to that stage, ensuring diverse perspectives are incorporated without reusing previously edited views. This approach maintains variety and helps capture more comprehensive updates across the scene.

**GS Optimization:** Our optimization follows 2DGS [20] and includes L1 and LPIPS [67] RGB losses to measure the discrepancy between the rendered and edited images  $\mathbf{I}'$ . Additionally, we incorporate normal consistency and depth distortion losses for surface alignment and controlled weight distribution along the rays. **L1 Loss:** Calculates pixel-wise absolute differences between edited and target images to maintain fidelity. **LPIPS Loss:** Evaluates perceptual similarity using deep feature representations to ensure realistic results. **Normal Consistency Loss:** Aligns splat normals with depth map gradients:  $\mathcal{L}_n = \sum_i \omega_i (1 - n_i^T N)$ , where  $i$  indexes intersected splats,  $\omega_i$  is the blending weight,  $n_i$  is the splat normal facing the camera, and  $N$  is the normal from the depth map gradient. **Depth Distortion Loss:** Minimizes distance between ray-splat intersections for concentrated weight distribution  $\mathcal{L}_d = \sum_{i,j} \omega_i \omega_j |z_i - z_j|$ , where  $\omega_i$  is the blending weight, and  $z_i$  is the depth value at the intersection point  $i$ .

## 4. Experiments

To thoroughly evaluate our method, we conducted tests on six diverse scenes using 17 unique prompts to assess performance across a range of scenarios. These included the same

scenes used for the evaluation of DGE [7], enabling direct comparison. The selected prompts covered object-centric and non-object-centric scenes, indoor and outdoor environments, and human face edits. While detailed evaluation data is not consistently provided in prior works, we emphasize transparency to encourage reproducibility and comparability in future research.

We evaluated our method on several benchmark datasets commonly used in 3D scene editing and rendering tasks: IN2N [17], Mip-NeRF360 [4], and BlendedMVS [63]. All experiments were conducted on a 512×512 images. These datasets challenge our method with varied lighting, complex geometries, and textures, demonstrating its adaptability across scenarios.

**Baselines** We compared our method with several recent state-of-the-art baselines, including IGS2GS [52] (a Gaussian splatting version of IN2N [17]), GaussCtrl [59], and DGE [7]. IGS2GS and DGE are based on the Instruct-Pix2Pix [5] diffusion model for image editing, while Gauss-Ctrl leverages ControlNet [66]. These baselines were chosen for their demonstrated ability to produce high-quality edits and their relevance to 3D editing, allowing us to comprehensively evaluate how our method performs against the current state-of-the-art. Additional promising methods, such as VCEdit [58] and LatentEditor [28], were not included in our experiments due to the lack of publicly available implementations.

**Evaluation Metrics** To provide a comprehensive evaluation of our method, we employed a range of metrics covering both standard and perceptual measures. While prior works often focus on a limited set of metrics, we aimed for a broader assessment to give a complete overview of our method’s performance. Next, we outline the metrics used.

**Edit PSNR:** This metric calculates the Peak Signal-to-Noise Ratio (PSNR) between the edited images generated by the diffusion model and the rendered images, quantifying the fidelity of the edits. **CLIP Similarity:** A standard metric for perceptually comparing images and text. We encode the training set using the CLIP [45] model and separately encode the target prompt into CLIP space. The cosine similarity between these encodings measures how closely the edited images align with the intended target prompt.

**CLIP Directional Similarity:** This metric [14] assesses the consistency of changes between images and text. We compute the cosine similarity between: 1. The difference in CLIP space between the original training set and the rendered training set. 2. The difference between the source and target prompts. This metric captures how well the direction of change in the image corresponds to the intended change described by the prompts.

We also included two custom metrics to evaluate how well the rendered training set matches the edited image: **DINO Single Image Similarity:** This metric calculates the

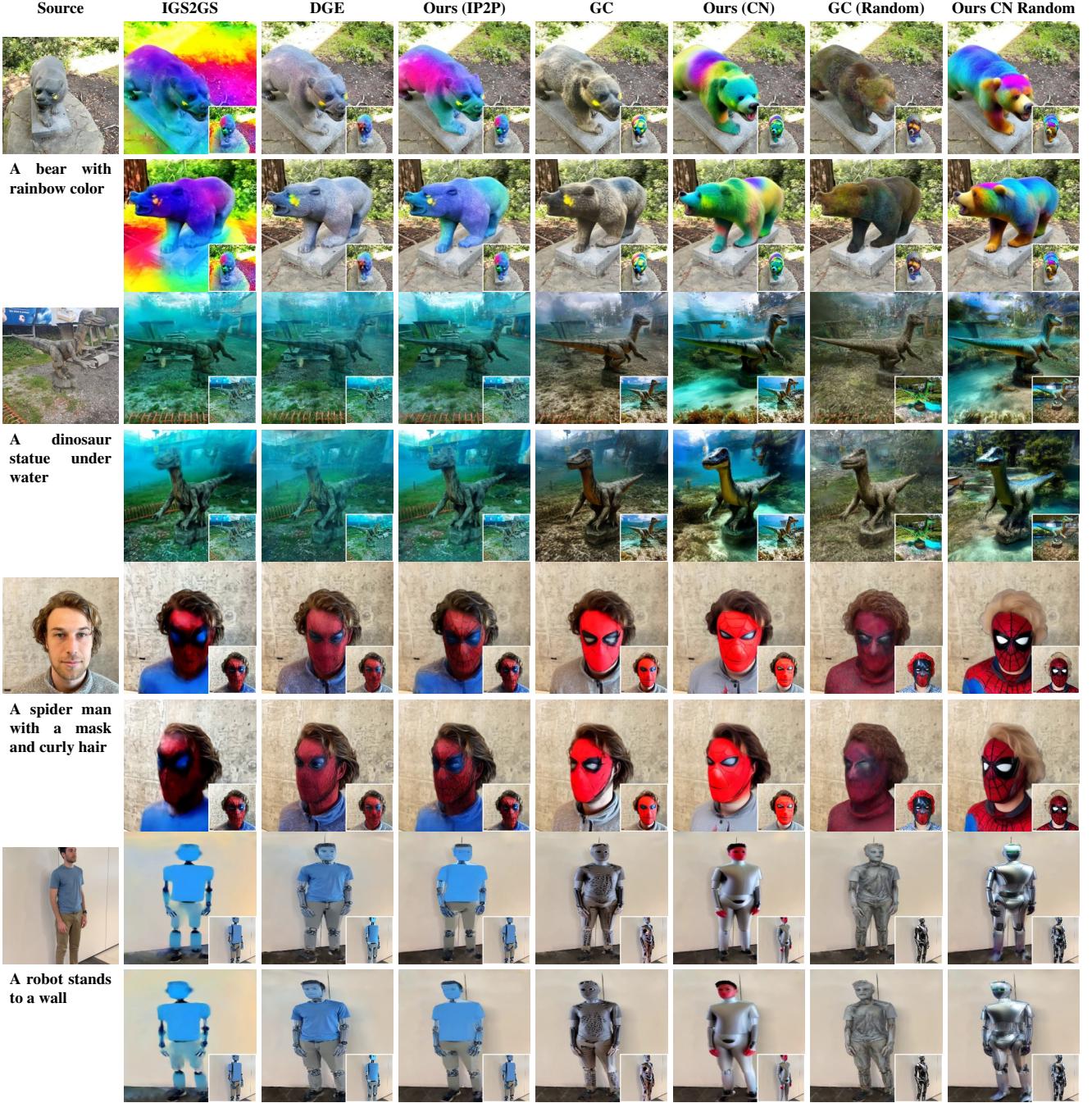


Figure 2. A comparison of scene editing methods across various scenes is presented. Each sample shows two views, with the modified source image shown as an inset. Additional examples are provided in Figs. VIII, IX, X, XI and XII.

mean similarity in DINOv2 [6, 41] space between the edited source image and the training set. DINO embeddings are known for capturing detailed visual semantics, making this metric effective for assessing visual alignment with the target appearance. **CLIP Single Image Similarity:** Similar to DINO Single Image Similarity, but with CLIP embeddings.

**Results** We evaluated our method against state-of-the-art techniques in two main categories: models based on InstructPix2Pix (IP2P), including IGS2GS and DGE, and models based on ControlNet guided by depth (GuassCtrl). GaussCtrl was tested using both latent inversion and random latent initialization methods. In the table, the la-

Method	Edit PSNR	Clip Similarity	Clip Dir. Sim.	DINO Single Img Sim.	Clip Single Img Sim.
IGS2GS (default params)	19.745	0.253	0.150	0.518	0.771
IGS2GS (improved)	21.429	0.267	0.149	0.627	0.841
DGE	25.667	0.257	0.160	0.692	0.862
<b>Ours IP2P</b>	22.780	0.264	0.144	0.727	<b>0.868</b>
GaussCtrl	<b>26.575</b>	0.252	0.138	0.670	0.843
<b>Ours ControlNet</b>	22.522	0.268	0.172	<b>0.756</b>	0.859
GaussCtrl-Random	20.438	0.236	0.103	0.517	0.767
<b>Ours ControlNet Random</b>	20.212	<b>0.276</b>	<b>0.189</b>	0.743	0.852

Table 1. Comparison of Methods: The methods above the separator use InstructPix2Pix for prompt-based editing, while those below rely on ControlNet-based editing.

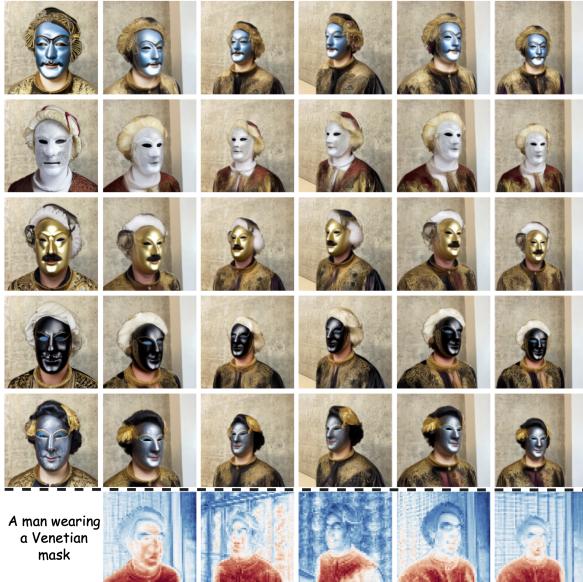


Figure 3. Obtaining variability. Our method (ControlNet variant) with different random seeds to produce diverse stylistic variations. Each row illustrates how varying the random seed impacts the visual output, resulting in unique edits while preserving the overall content structure. Additionally, the figure includes the warped feature map of the source view (left column) to provide insight into how the attention is distributed across the edited images.

bel ‘‘Random’’ indicates the use of ControlNet with a randomly initialized latent, as opposed to the latent inversion approach [49] employed in GaussCtrl. IGS2GS is applied both with its default parameters and with a better set of parameters we found, in which the main difference is using fewer iterations (3000 instead of 7000).

As shown in Table 1, our method outperforms all compared models in single-image similarity metrics, demonstrating superior alignment with the target images in both DINO and CLIP spaces. For CLIP similarity and directional similarity, our method shows better performance than

GaussCtrl. In comparison with IP2P-based models, while there are instances where our method shows minor differences, it often performs on par or better. Recognizing that these metrics may not fully capture the subjective quality of edits, we also conducted qualitative analyses and user studies for a more comprehensive evaluation. Although some methods achieve higher scores in the Edit PSNR metric, it is important to note that this metric is biased, since it favors methods that edit fewer reference images as part of their 3D model finetuning. When fewer images are edited, the 3D resulting model is better fitted to these views, especially since the information is propagated in such a way that the image editing is based on the progress of the 3D model. This, of course, comes at the price of relying on reference views that are inconsistent with the prompt.

To complement our quantitative evaluation, Fig. 2 compares our method against the baselines across various scenes and prompts, showcasing how our approach maintains superior edit quality and consistency. Note that significant differences in the edited reference images between methods arise from the way consistency is achieved between the reference views, even for the same diffusion model seed.

Lastly, Fig. 3 demonstrates the versatility of our single-image editing capability, where we can apply edits using the same prompt but achieve different styles. The GaussCtrl method shows no versatility due to the reliance on DDIM inversion [68]. The other baselines obtain different results for different seeds, but due to their propagation mechanism, display less variability, see Fig. IV.

**User Study** We conducted a user study involving 28 participants, each answering 10 questions across 6 different scenes. In each question, users were shown a source image, a prompt, and two novel views generated by each method. Participants were asked to choose the image they believed best represented the edit, focusing on both edit quality and coherence with the given prompt. We did not limit participants’ evaluation criteria, allowing them to assess based on their judgment. To ensure a fair comparison, the ques-

Method	Selection Frequency
DGE	0.142
IGS2GS	0.265
Ours IP2P	<b>0.594</b>
GaussCtrl	0.148
GaussCtrl Random	0.058
Ours ControlNet	0.154
Our ControlNet Random	<b>0.639</b>

Table 2. User study results for the two groups of methods.

tions were divided into two sets for IP2P-based models and for ControlNet-based models, preventing cross-model biases during evaluation. The results of the user study are summarized in Table 2. Evidently, there is a clear reference to our method in both groups, where in the second there is an advantage to the method that is randomly initialized.

Name	PSNR	C. Sim.	Dir. Sim.	D. Im.	Sim. C.	Im. Sim.
1 stage	18.015	0.271	0.175	0.726	0.826	
2 stage	17.196	0.278	0.205	0.768	0.871	
Only SA	18.992	0.278	0.209	0.771	<b>0.879</b>	
W/O decay	20.123	0.277	0.210	0.767	0.871	
W/O M	<b>20.661</b>	0.280	0.203	0.765	0.873	
Full method	18.949	<b>0.281</b>	<b>0.211</b>	<b>0.773</b>	0.877	

Table 3. Ablation study results. C=Clip D=Dino.

**Ablation Study** We conducted thorough ablations to evaluate the impact of different components of our method, including: **Warping mask**: removing the warping mask as described in Equation 2. **Blending decay mechanism**: a constant blending coefficient throughout the denoising process, rather than decaying it over time. **Self-attention**: injecting only self-attention feature maps into the diffusion model. **Iterative optimization**: We ran our method using both single-stage and two-stage executions instead of the three iterations we use in our experiments. The ablations were performed on four different scenes from different datasets. The results of these ablations are summarized in Table 3. As can be seen, each component contributes to the overall success. The number of iterations helps in an incremental way, mostly to the Clip Directional Similarity score. Removing the cross attention from the blending deteriorates the metrics that measure text alignment, but improves the clip image similarity. The decay and the mask seem to assist in multiple metrics. Various ablations improve the PSNR. However, as mentioned above, this metric can be high despite the overall results being weaker. A visual comparison can be found in Fig. VII.

## 5. Discussion and Limitations

The success of our attention-warping approach in maintaining consistency across multiple views suggests broader applications beyond static 3D scenes. A natural extension would be to apply similar principles to video editing, where optical flow could replace depth-based warping for propagating edits across frames. Unlike recent video editing methods like [15, 27, 60] that rely on temporal diffusion models or frame-by-frame processing, our attention-warping technique could potentially offer more precise control over edit propagation while maintaining temporal coherence, without the necessity of multiple frames processing. This approach would be particularly advantageous compared to methods that depend on direct feature matching or temporal consistency losses, as our warped based attention approach could better preserve fine-grained edit details while ensuring smooth transitions between frames. The success of our approach in handling occlusions and view-dependent artifacts through the visibility mask and the gaussians normal mask also provides insights into how attention mechanisms can be made more geometry-aware, which could be valuable for improving other 3D-aware generation and editing methods.

However, our method does face several important limitations. **Geometry Dependence:** The quality of our edits heavily relies on the accuracy of the underlying geometric reconstruction. In cases where the Gaussian splatting model fails to capture accurate depth information or produces noisy geometry, the warping process can lead to artifacts or inconsistent edits across views. **Limited Edit Scope:** While our method handles a wide range of edits, it can struggle with certain types of modifications that require significant geometric changes or involve heavy occlusions. For example, adding large objects that should appear consistently across multiple views remains challenging, as the method primarily focuses on appearance changes rather than structural modifications. **Diffusion Model Constraints:** Our method, which is based on diffusion models, can be limited by their inherent capabilities and limitations.

## 6. Conclusions

We have presented a novel approach for consistent 3D scene editing that leverages attention features from diffusion models through geometric-aware warping. By capturing edit intentions from a single reference view and systematically propagating them across multiple viewpoints, our method achieves high-quality, consistent edits while avoiding the computational overhead of processing multiple views simultaneously. Extensive experiments demonstrate that our approach outperforms existing methods in maintaining edit fidelity across viewpoints, as validated through both quantitative metrics and user studies.

## References

- [1] Omer Bar-Tal, Dolev Ofri-Amar, Rafail Fridman, Yoni Kashtan, and Tali Dekel. Text2live: Text-driven layered image and video editing. In *European conference on computer vision*, pages 707–723. Springer, 2022.
- [2] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5855–5864, 2021.
- [3] 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.
- [4] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. *CVPR*, 2022.
- [5] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions. In *CVPR*, 2023.
- [6] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.
- [7] Minghao Chen, Iro Laina, and Andrea Vedaldi. Dge: Direct gaussian 3d editing by consistent multi-view editing. *arXiv preprint arXiv:2404.18929*, 2024.
- [8] Yiwen Chen, Zilong Chen, Chi Zhang, Feng Wang, Xiaofeng Yang, Yikai Wang, Zhongang Cai, Lei Yang, Huaping Liu, and Guosheng Lin. Gaussianeditor: Swift and controllable 3d editing with gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21476–21485, 2024.
- [9] Xinhua Cheng, Tianyu Yang, Jianan Wang, Yu Li, Lei Zhang, Jian Zhang, and Li Yuan. Progressive3d: Progressively local editing for text-to-3d content creation with complex semantic prompts. *arXiv preprint arXiv:2310.11784*, 2023.
- [10] Yingying Deng, Fan Tang, Weiming Dong, Chongyang Ma, Xingjia Pan, Lei Wang, and Changsheng Xu. Stytr2: Image style transfer with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11326–11336, 2022.
- [11] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis. *ArXiv*, abs/2105.05233, 2021.
- [12] Jiahua Dong and Yu-Xiong Wang. Vica-nerf: View-consistency-aware 3d editing of neural radiance fields. *Advances in Neural Information Processing Systems*, 36, 2024.
- [13] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.
- [14] 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.
- [15] Michal Geyer, Omer Bar-Tal, Shai Bagon, and Tali Dekel. Tokenflow: Consistent diffusion features for consistent video editing. *arXiv preprint arXiv:2307.10373*, 2023.
- [16] Bingchen Gong, Yuehao Wang, Xiaoguang Han, and Qi Dou. Recolornerf: Layer decomposed radiance fields for efficient color editing of 3d scenes. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 8004–8015, 2023.
- [17] Ayaan Haque, Matthew Tancik, Alexei Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-nerf2nerf: Editing 3d scenes with instructions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.
- [18] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022.
- [19] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *arXiv preprint arxiv:2006.11239*, 2020.
- [20] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *SIGGRAPH 2024 Conference Papers*. Association for Computing Machinery, 2024.
- [21] Yi-Hua Huang, Yue He, Yu-Jie Yuan, Yu-Kun Lai, and Lin Gao. Stylizednerf: consistent 3d scene stylization as stylized nerf via 2d-3d mutual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18342–18352, 2022.
- [22] Inbar Huberman-Spiegelglas, Vladimir Kulikov, and Tomer Michaeli. An edit friendly ddpm noise space: Inversion and manipulations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12469–12478, 2024.
- [23] Ajay Jain, Ben Mildenhall, Jonathan T Barron, Pieter Abbeel, and Ben Poole. Zero-shot text-guided object generation with dream fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 867–876, 2022.
- [24] Kacper Kania, Kwang Moo Yi, Marek Kowalski, Tomasz Trzcinski, and Andrea Tagliasacchi. Conerf: Controllable neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18623–18632, 2022.
- [25] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6007–6017, 2023.
- [26] 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.
- [27] Levon Khachatryan, Andranik Moysian, 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.
- [28] Umar Khalid, Hasan Iqbal, Nazmul Karim, Jing Hua, and Chen Chen. Latenteditor: Text driven local editing of 3d scenes. *arXiv preprint arXiv:2312.09313*, 2023.
- [29] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023.
- [30] Verica Lazova, Vladimir Guzov, Kyle Olszewski, Sergey Tulyakov, and Gerard Pons-Moll. Control-nerf: Editable feature volumes for scene rendering and manipulation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4340–4350, 2023.
- [31] Yuhan Li, Yishun Dou, Yue Shi, Yu Lei, Xuanhong Chen, Yi Zhang, Peng Zhou, and Bingbing Ni. Focaldreamer: Text-driven 3d editing via focal-fusion assembly. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3279–3287, 2024.
- [32] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *Advances in Neural Information Processing Systems*, 33:15651–15663, 2020.
- [33] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023.
- [34] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11461–11471, 2022.
- [35] Luca Medeiros. Language segment-anything. [github.com/luca-medeiros/lang-segment-anything](https://github.com/luca-medeiros/lang-segment-anything), 2024.
- [36] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jianjun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021.
- [37] 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. *Communications of the ACM*, 65(1):99–106, 2021.
- [38] 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.
- [39] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4):1–15, 2022.
- [40] Thu Nguyen-Phuoc, Feng Liu, and Lei Xiao. Snerf: stylized neural implicit representations for 3d scenes. *arXiv preprint arXiv:2207.02363*, 2022.
- [41] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINov2: Learning robust visual features without supervision. In *arXiv 2304.07193*, 2024.
- [42] Jangho Park, Gihyun Kwon, and Jong Chul Ye. Ed-nerf: Efficient text-guided editing of 3d scene using latent space nerf. *arXiv preprint arXiv:2310.02712*, 2023.
- [43] Gaurav Parmar, Krishna Kumar Singh, Richard Zhang, Yijun Li, Jingwan Lu, and Jun-Yan Zhu. Zero-shot image-to-image translation. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–11, 2023.
- [44] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.
- [45] 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.
- [46] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.
- [47] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 234–241. Springer, 2015.
- [48] Etaï Sella, Gal Fiebelman, Peter Hedman, and Hadar Averbuch-Elor. Vox-e: Text-guided voxel editing of 3d objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 430–440, 2023.
- [49] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.
- [50] Matthew Tancik, Ethan Weber, Eron Ng, Ruilong Li, Brent Yi, Justin Kerr, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, David McAllister, and Angjoo Kanazawa. Nerfstudio: A modular framework for neural radiance field development. In *ACM SIGGRAPH 2023 Conference Proceedings*, 2023.
- [51] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. 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*, pages 1921–1930, 2023.
- [52] Cyrus Vachha and Ayaan Haque. Instruct-gs2gs: Editing 3d gaussian splats with instructions. <https://instruct-gs2gs.github.io/>, 2024.
- [53] Binglun Wang, Niladri Shekhar Dutt, and Niloy J Mitra. Proteusnerf: Fast lightweight nerf editing using 3d-aware image

- context. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 7(1):1–17, 2024.
- [54] Can Wang, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao. Clip-nerf: Text-and-image driven manipulation of neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3835–3844, 2022.
- [55] Can Wang, Ruixiang Jiang, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao. Nerf-art: Text-driven neural radiance fields stylization. *IEEE Transactions on Visualization and Computer Graphics*, 2023.
- [56] Junjie Wang, Jiemin Fang, Xiaopeng Zhang, Lingxi Xie, and Qi Tian. Gaussianeditor: Editing 3d gaussians delicately with text instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20902–20911, 2024.
- [57] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*, 2021.
- [58] Yuxuan Wang, Xuanyi Yi, Zike Wu, Na Zhao, Long Chen, and Hanwang Zhang. View-consistent 3d editing with gaussian splatting. In *European Conference on Computer Vision*, pages 404–420. Springer, 2025.
- [59] Jing Wu, Jia-Wang Bian, Xinghui Li, Guangrun Wang, Ian Reid, Philip Torr, and Victor Prisacariu. GaussCtrl: Multi-View Consistent Text-Driven 3D Gaussian Splatting Editing. *ECCV*, 2024.
- [60] Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Stan Weixian Lei, Yuchao Gu, Yufei Shi, Wynne Hsu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7623–7633, 2023.
- [61] Qiling Wu, Jianchao Tan, and Kun Xu. Palettenerf: Palette-based color editing for nerfs. *arXiv preprint arXiv:2212.12871*, 2022.
- [62] Jiale Xu, Xintao Wang, Yan-Pei Cao, Weihao Cheng, Ying Shan, and Shenghua Gao. Instructp2p: Learning to edit 3d point clouds with text instructions. *arXiv preprint arXiv:2306.07154*, 2023.
- [63] Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. *Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [64] Lu Yu, Wei Xiang, and Kang Han. Edit-diffnerf: Editing 3d neural radiance fields using 2d diffusion model. *arXiv preprint arXiv:2306.09551*, 2023.
- [65] Kai Zhang, Nick Kolkin, Sai Bi, Fujun Luan, Zexiang Xu, Eli Shechtman, and Noah Snavely. Arf: Artistic radiance fields. In *European Conference on Computer Vision*, pages 717–733. Springer, 2022.
- [66] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *IEEE International Conference on Computer Vision (ICCV)*, 2023.
- [67] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.
- [68] Yuxin Zhang, Nisha Huang, Fan Tang, Haibin Huang, Chongyang Ma, Weiming Dong, and Changsheng Xu. Inversion-based style transfer with diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10146–10156, 2023.
- [69] Chengwei Zheng, Wenbin Lin, and Feng Xu. Editablenerf: Editing topologically varying neural radiance fields by key points. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8317–8327, 2023.
- [70] Jingyu Zhuang, Chen Wang, Liang Lin, Lingjie Liu, and Guanbin Li. Dreameditor: Text-driven 3d scene editing with neural fields. In *SIGGRAPH Asia 2023 Conference Papers*, pages 1–10, 2023.

## A. Advantages of Single View Editing

In this section, we present a comprehensive discussion on the benefits of single-view editing in 3D scene editing. Although certain methods can achieve impressive edits with multi-view consistency [7, 28, 58, 59], they lack the flexibility to allow users to control the desired edit style. Why is this control important? It enables users to customize scene edits precisely to their preferences. Additionally, it provides the option to edit a single image without involving a diffusion model by using a single inversion process [68] to propagate edits across the scene.

Figure IV illustrates visual examples of our results, showing edits made with the same source image and prompt but varying in style. We also provide comparisons to other methods. It is important to note that methods like GaussCtrl [59] and Ours ControlNet, which rely on image inversion, are excluded from these comparisons as they use pre-defined inverted images rather than random latents.

**Limitations of Other Methods:** While some existing methods, such as GaussCtrl, do not offer the flexibility to choose a random edit style due to their reliance on image inversion, we adapted GaussCtrl to support random-based generation for comparison purposes. However, even with this adaptation, GaussCtrl struggles to align consistently with a single source edit style, as demonstrated in Figure IV. Although methods like IGS2GS [52] and DGE [7] can generate edits using different seeds, they have significant limitations. IGS2GS is capable of producing varying results; however, the generated edits are often of lower quality and lack the ability to allow users to select a specific style explicitly. On the other hand, DGE struggles to produce noticeable variations in edits, even when using different seeds. As shown in Figure IV, these methods may produce edits that are inconsistent with the initial source-edited image or fail to offer meaningful diversity. For instance, as shown in the first example, although the source image has pink skin, both IGS2GS and DGE generate results with white skin. In contrast, our method is not constrained by the chosen edit style. Users can freely select any edited image to use as the basis for their scene edits, providing superior flexibility and user control over the editing process. **User-Generated Edits:** As further illustrated in Figure V, we provide a visualization of the DDPM inversion [22] process. In this setup, the user supplies a non-diffusion-based edited image, which is inverted into the latent space using DDPM inversion. Our method is then applied to this inverted edit. The top row in the figure shows the user-generated edited image followed by three novel views generated using our method, while the bottom row displays the user-generated edited image and three corresponding views generated using the DGE [7] method. This comparison demonstrates that, although DGE is a state-of-the-art view-consistent editing method, it re-

quires more than a single input image to produce such edits. In contrast, our method achieves high-quality, consistent results directly from a single user-provided edited image, highlighting a key advantage of our approach.

## B. Ablation Study Details

In this section, we provide a detailed discussion and visual comparison for the ablation study of our method, as shown in Figure VI. All ablation experiments are conducted using the “Ours ControlNet Random” method. Below, we analyze the impact of different components of our approach:

**Iterative Process:** We examine the effect of using a non-iterative process (*1-stage* in the figure). It is evident that using only a single stage results in the method struggling to generate coarse and fine details.

**Self-Attention Only:** This ablation demonstrates the effect of injecting only the attention feature maps from the self-attention layers. While this produces reasonable results, it struggles to reconstruct finer details, such as forehead wrinkles.

**Without Decay:** Here, we evaluate the impact of removing the decay mechanism for alpha blending between the warped and new attention feature maps. As shown in the figure, the absence of this mechanism results in suboptimal blending and reduced image quality.

**Without Warp Mask:** In this scenario, the warping mask is omitted during the warping process, leading to visible artifacts caused by the introduction of out-of-distribution features during image generation. For instance, noticeable color leakage occurs between visible and occluded regions, such as the eyelid area and the left side of the man’s hair, as well as between the lips and the neck.

Each ablation highlights the importance of the respective components in ensuring high-quality, consistent, and artifact-free results. We conduct our ablations on *face*, *bear* and *dinosaur* scenes from the datasets accordingly [4, 17, 63].

## C. Edit PSNR Visual Comparison

As highlighted in the main paper, the *Edit PSNR* metric is inherently biased and often fails to reflect the true quality of edits accurately. To illustrate this limitation, we present visual examples in Figure VII. These examples demonstrate that edits with result in poor visual quality can have high PSNR values. For instance, while GaussCtrl [59] achieves a significantly higher *Edit PSNR* score compared to our method, its final edit does not adhere to the given prompt: “*a photo of a rainbow-colored bear in the forest*.” This discrepancy underscores the inadequacy of relying solely on PSNR as a metric for evaluating edit quality.

## D. Method Comparison Visualization

As outlined in the main paper, we provide additional visual examples comparing our method with other approaches. These visualizations are organized by different scenes and are presented in Figures [VIII](#), [IX](#), [X](#), [XI](#), and [XII](#).

## E. Evaluation Setup and Details

In this section, we outline the scenes and prompts used for evaluating our method. Our evaluation follows the same setup as DGE [7], applied to three common scenes, along with an additional three scenes from other datasets. The evaluated scenes include: *Face*, *Bear*, and *Person* from IN2N [17]; *Garden* and *Stump* from MIP-NeRF360 [4]; and *Dinosaur* from BlendedMVS [63]. The editing prompts, tailored for both IP2P and ControlNet [5, 66], as well as the source and target prompts used for metrics evaluation, are detailed in Tab. [IV](#).

## F. User Study and Evaluation

As described in the main paper, we conducted a user study to evaluate the subjective quality of the results. To illustrate the structure of the study, we provide an example question used during the evaluation in Figure [XIII](#). This example highlights how participants were asked to compare and assess the quality of edits generated by different methods.

## G. 2DGS and 3DGS Warping

In our paper, we chose to use 2DGS [20] over 3DGS [26] due to its superior geometric accuracy. Figure [XIV](#) shows the source view, the validity mask highlighting reliable depth regions, and the warping results using 3DGS and 2DGS. While 3DGS produces noticeable artifacts, 2DGS demonstrates better alignment and accuracy. This improvement is particularly evident in regions defined as valid by the mask, further supporting the decision to use 2DGS in our method.

## H. Code and Implementation Details

We extend our gratitude to NeRFStudio [50], whose infrastructure served as the foundation for our implementation, providing the necessary tools and framework for developing our method. The code specific to our method (i.e., not including the base code of NeRFStudio) is included in this supplementary material. We also acknowledge IGS2GS [52] for additional reference.



Figure IV. The figure presents a comparison of different style edits based on various source image editing approaches, using different random seeds. Each method is evaluated with three different seeds. For each part, the top row displays the edited source image, while the two rows below show novel views generated from the edited model. GC=GaussCtrl, CN=ControlNet.

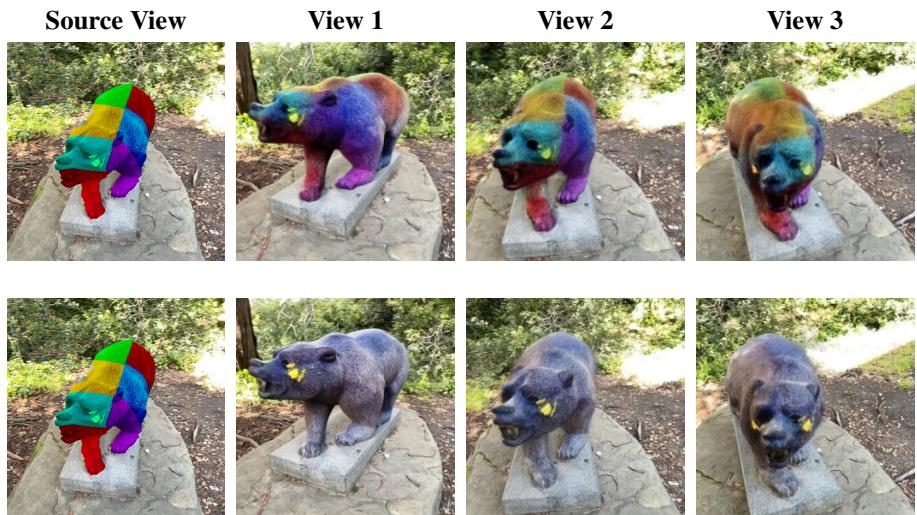


Figure V. User-generated edits comparison between our method and the DGE method. The first row shows the user-provided edited image followed by three novel views generated using our method. The second row displays the same using the DGE method. This comparison highlights the differences in edit quality and consistency between the two approaches.

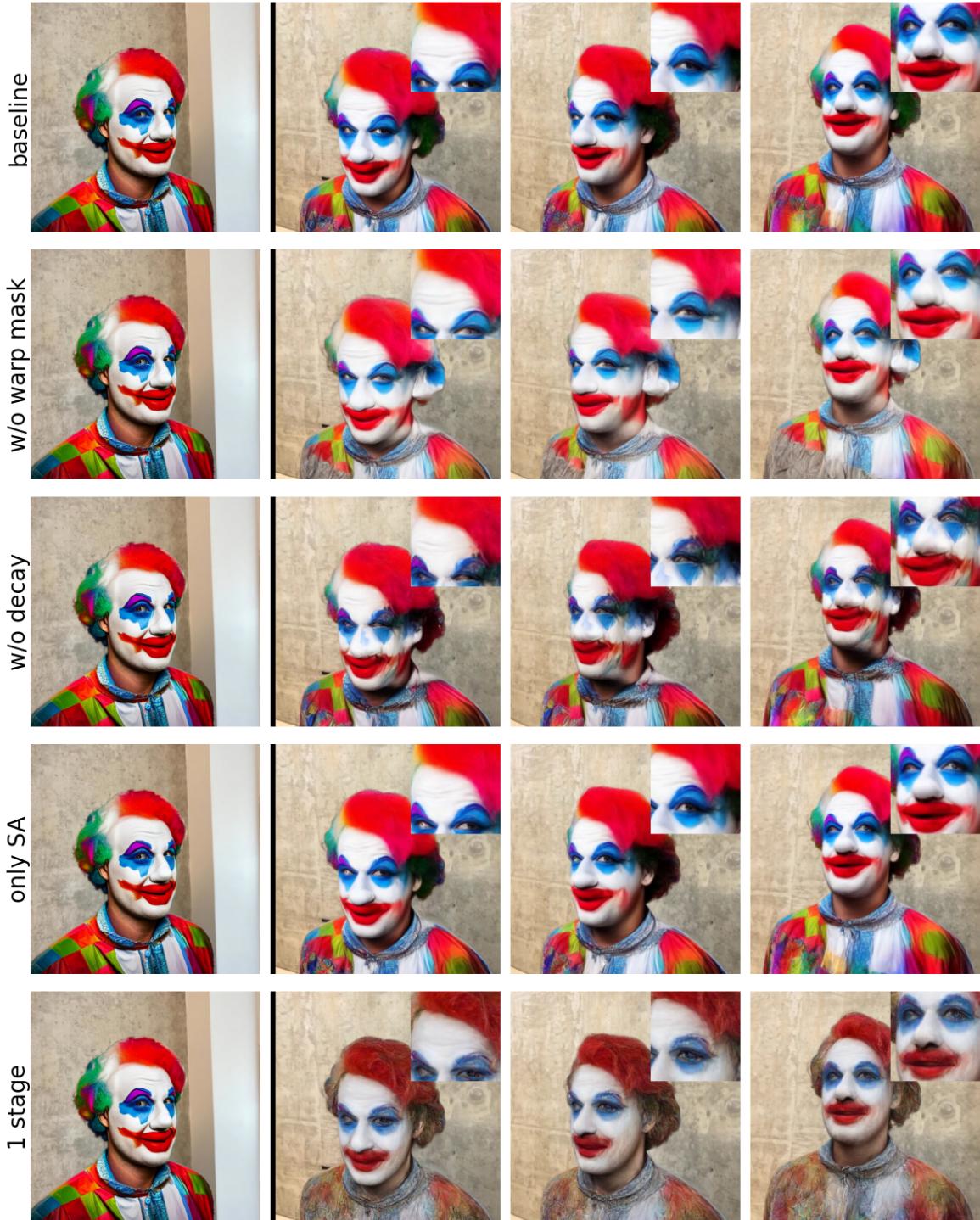


Figure VI. Ablation Study Overview. The figure shows how key components—warping mask, blending decay, attention types, and stage count—affect performance. The leftmost column is the edited source image for reference, with each row highlighting their impact on output quality. The first row represents the baseline, which is our full method, showcasing the effectiveness of all components combined.

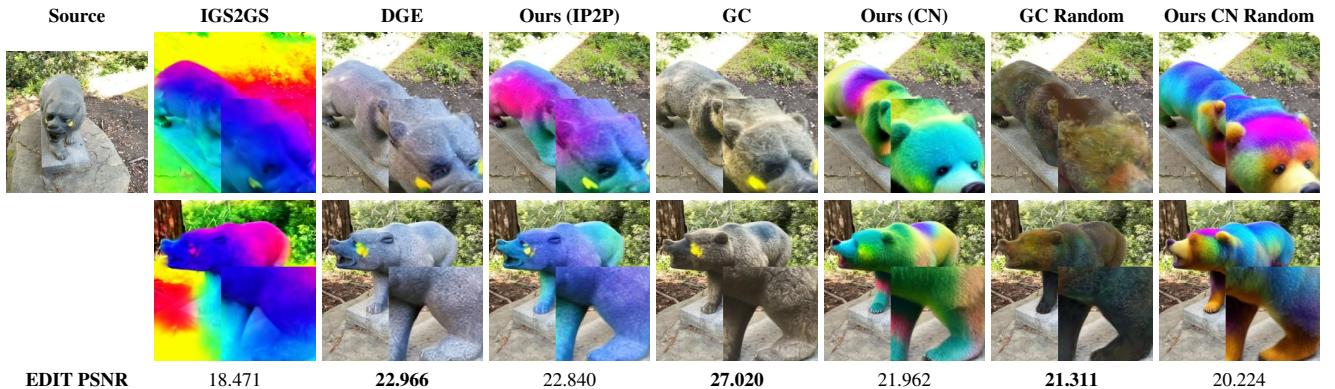


Figure VII. This figure presents a comparison of Edit PSNR against edit quality. For each method, we show two different views along with a center zoom-in to enable a more precise evaluation of image quality. The left column present the source image.

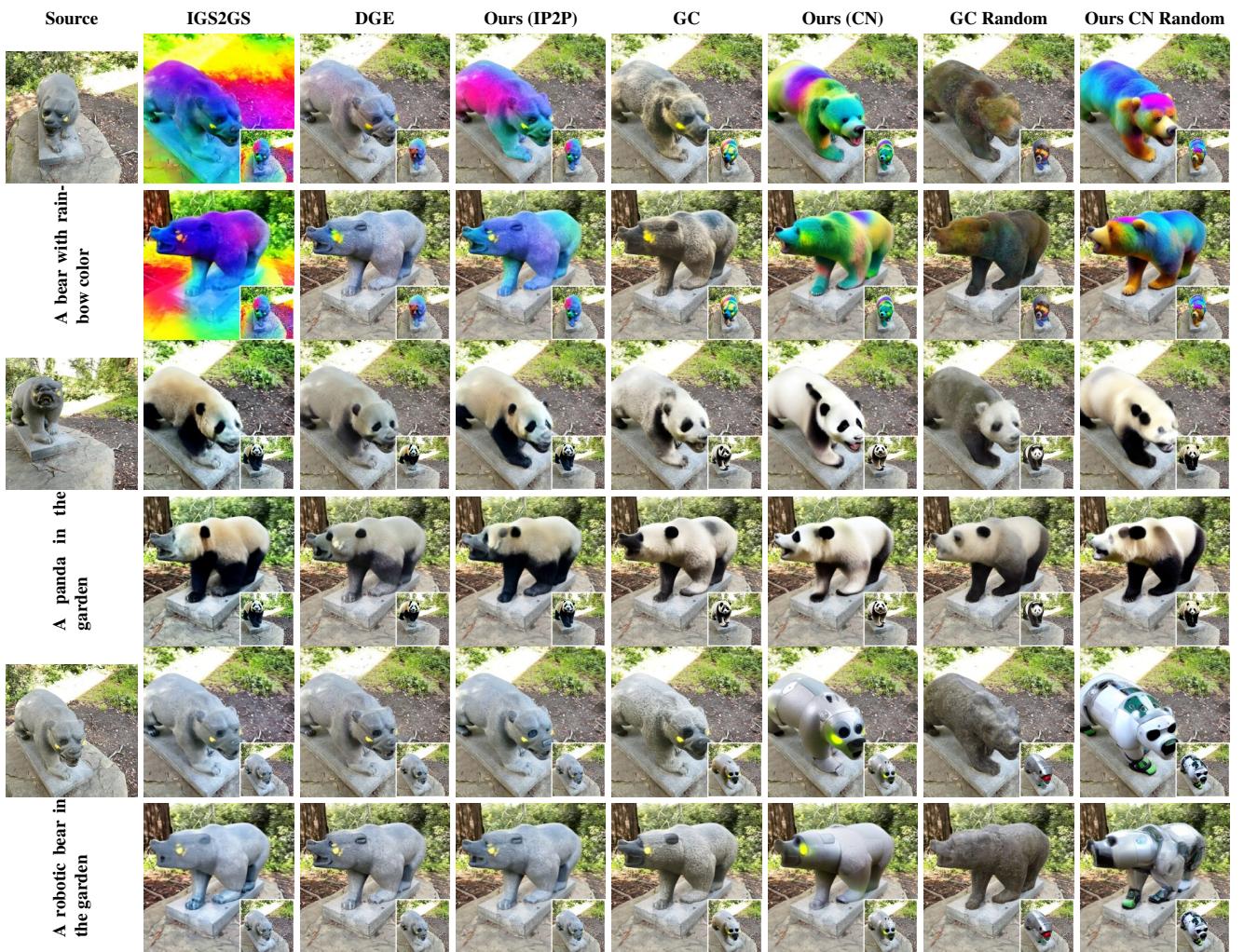


Figure VIII. A comparison of scene editing methods of *bear* scene. Each sample shows two views, with the modified source image shown as an inset.

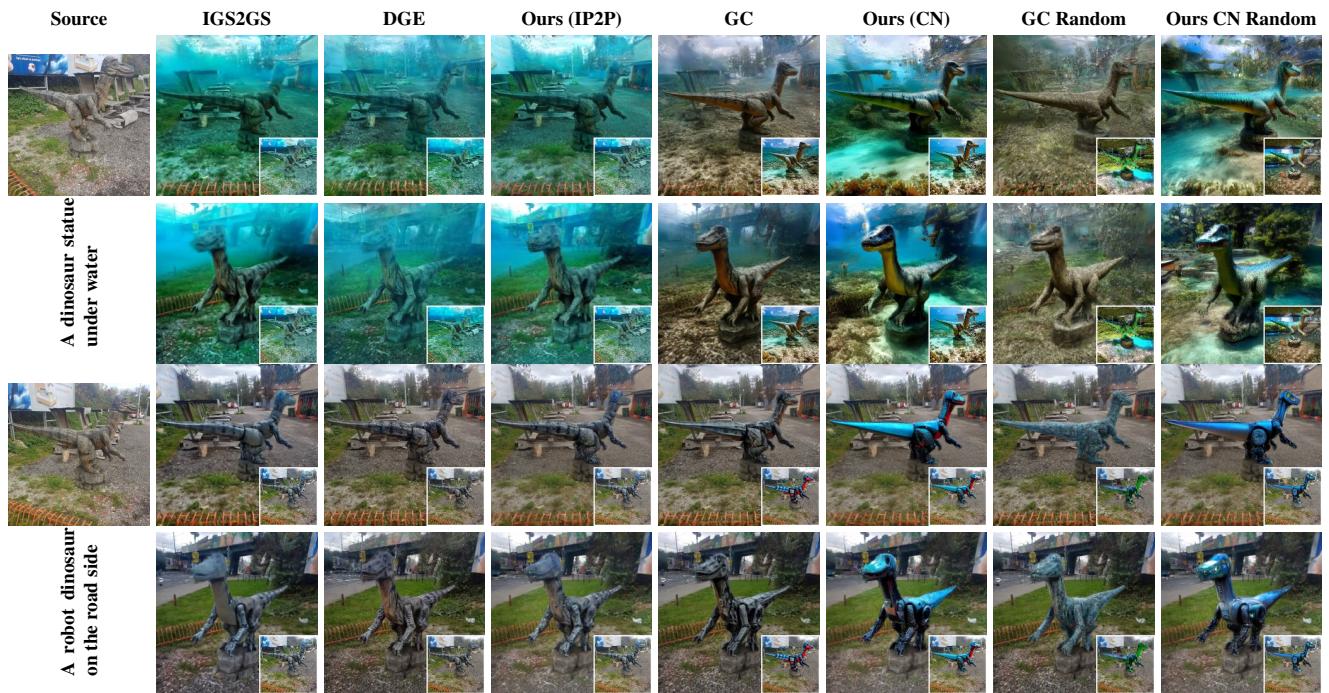


Figure IX. A comparison of scene editing methods of *dinosaur* scene. Each sample shows two views, with the modified source image shown as an inset.

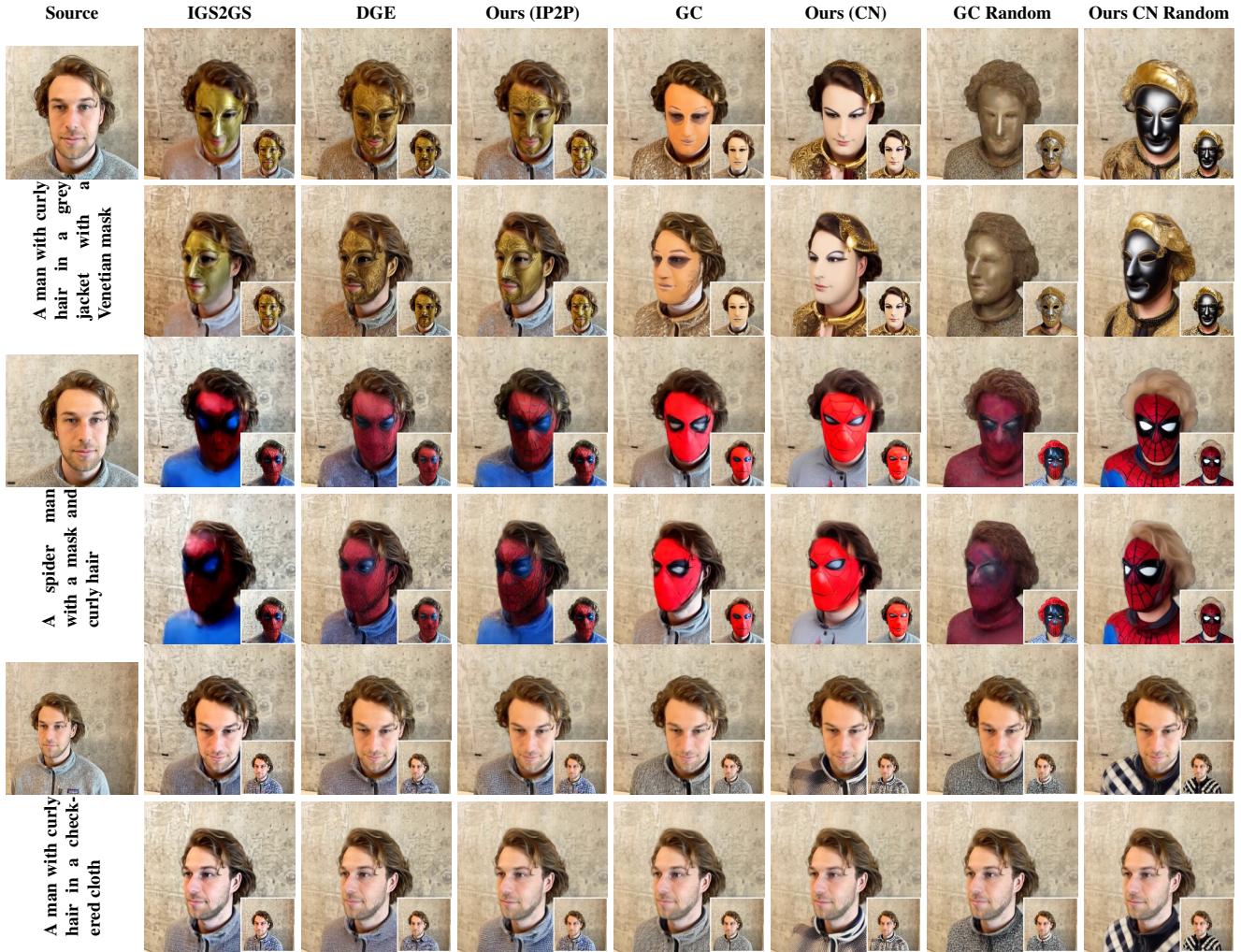


Figure X. A comparison of scene editing methods of *face* scene. Each sample shows two views, with the modified source image shown as an inset.

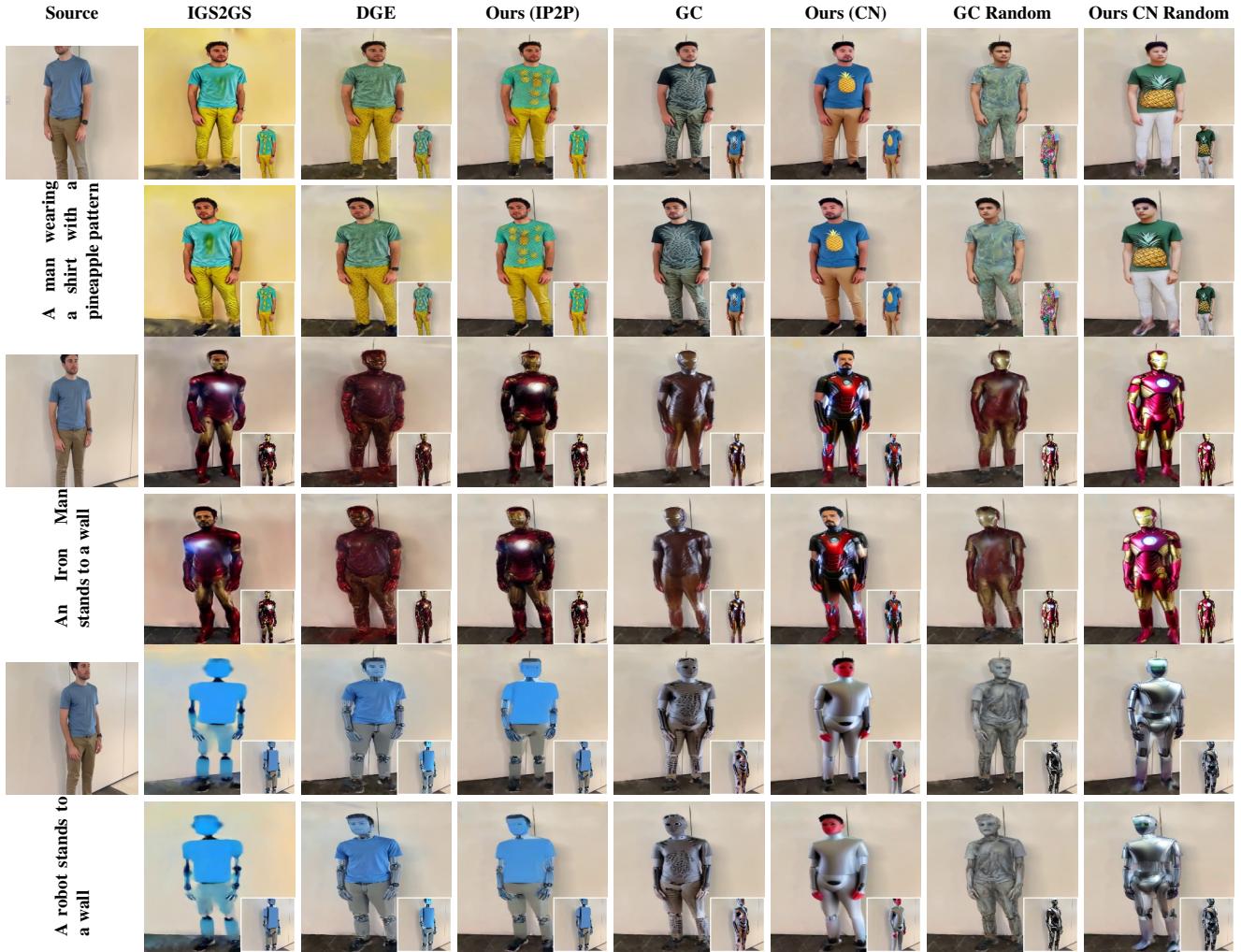


Figure XI. A comparison of scene editing methods of *person* scene. Each sample shows two views, with the modified source image shown as an inset.

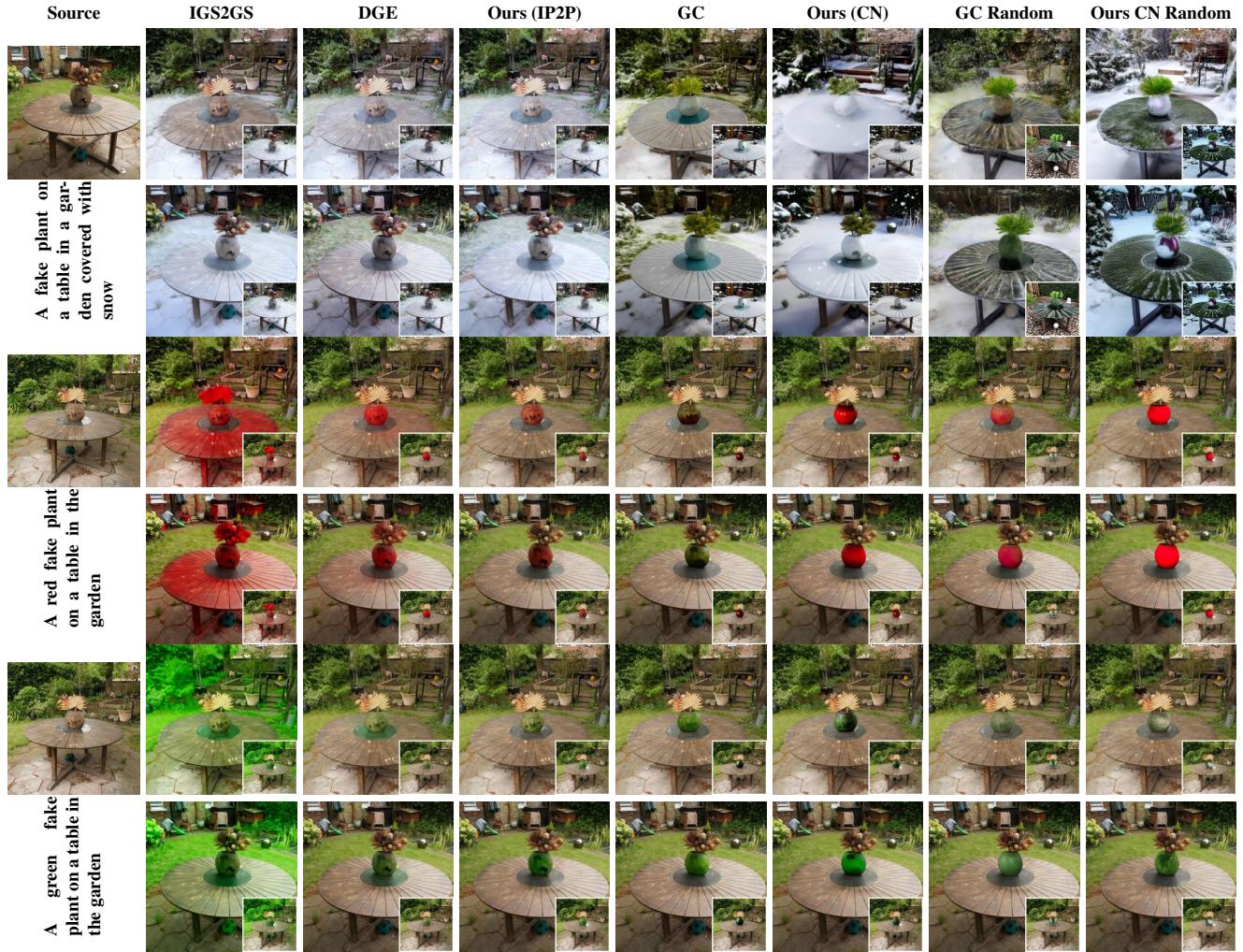


Figure XII. A comparison of scene editing methods of *table* scene. Each sample shows two views, with the modified source image shown as an inset.

<b>Scene</b>	<b>Source Prompt</b>	<b>Target Prompt</b>	<b>Edit Prompt (IP2P)</b>	<b>Edit Prompt (CN)</b>
bear	A stone bear in a garden	A bear with rainbow color	Make the color of the bear look like rainbow	a photo a rainbow colored bear in the forest
bear	A stone bear in a garden	A robotic bear in the garden	Make the bear look like a robot	a photo of a robot bear in the forest
bear	A stone bear in a garden	A panda in the garden	Make the bear look like a panda	a photo of a panda bear in the forest
person	A man standing next to a wall wearing a blue T-shirt and brown pants	A man looks like a mosaic sculpture standing next to a wall	Make the man look like a mosaic sculpture	a photo of a man look like a mosaic sculpture
person	A man standing next to a wall wearing a blue T-shirt and brown pants	A robot stands to a wall	Turn the man into a robot	a photo of a robot
person	A man standing next to a wall wearing a blue T-shirt and brown pants	An Iron Man stands to a wall	Turn him into Iron Man	a photo of Iron Man
person	A man standing next to a wall wearing a blue T-shirt and brown pants	A man wearing a shirt with a pineapple pattern	Make the person wear a shirt with a pineapple pattern	a photo of a person wear a shirt with a pineapple pattern
face	A man with curly hair in a grey jacket	A man with curly hair in a grey jacket with a Venetian mask	Give him a Venetian mask	A man wearing a Venetian mask over his face
face	A man with curly hair in a grey jacket	A man with curly hair in a checkered cloth	Give him a checkered jacket	a photo of a man wearing a checkered jacket
face	A man with curly hair in a grey jacket	A spider man with a mask and curly hair	Turn him into spiderman with a mask	a photo of man wearing a Spiderman mask over his face
garden	A fake plant on a table in the garden	A fake plant on a table in a garden covered with snow	Make it snowy	a photo of a fake plant on a table in the garden in the snow
garden	A fake plant on a table in the garden	A red fake plant on a table in the garden	Turn the vase into red	a photo of a red fake plant on a table in the garden
garden	A fake plant on a table in the garden	A green fake plant on a table in the garden	Turn the vase into green	a photo of a green fake plant on a table in the garden
stump	A stump in the forest	A stump in the forest as Monet's painting	Turn it into Monet's painting	a photo of a stump as Monet's painting
stump	A stump in the forest	A stump in the forest on fire	Make it burn with fire	a photo of a stump on fire
dinosaur	A dinosaur statue on the road side	A robot dinosaur on the road side	Turn the dinosaur into a robot	a photo of a robot dinosaur on the road side
dinosaur	A dinosaur statue on the road side	A dinosaur statue under water	Make it underwater	a photo of a dinosaur statue under the water

Table IV. Prompts for different scenes, showing variations across Source Prompts, Target Prompts, and Edit Prompts for both IP2P and ControlNet.

Given the source image and the prompt, choose the better edit.\*



An Iron Man stands to a wall



○



○



○



○



(a) Source View



(b) Validity Mask



(c) 3DGS Warping



(d) 2DGS Warping

Figure XIII. An example question from the user study designed to evaluate the subjective quality of edits. Participants were presented with results from multiple methods and asked to compare and select the edit that best adhered to the given prompt while maintaining visual fidelity.

Figure XIV. A visual comparison of warping results using 2DGS [20] and 3DGS [26] depths. The top-left image shows the source view, the top-right image displays the warp validity mask, the bottom-left image presents warping using 3DGS, and the bottom-right image demonstrates warping using 2DGS. This figure highlights the superior geometric accuracy achieved with 2DGS, particularly in valid regions as indicated by the validity mask. Artifacts present in both methods can largely be attributed to areas outside the valid regions.