3DitScene: Editing Any Scene via Language-guided Disentangled Gaussian Splatting

Anonymous authors
Paper under double-blind review

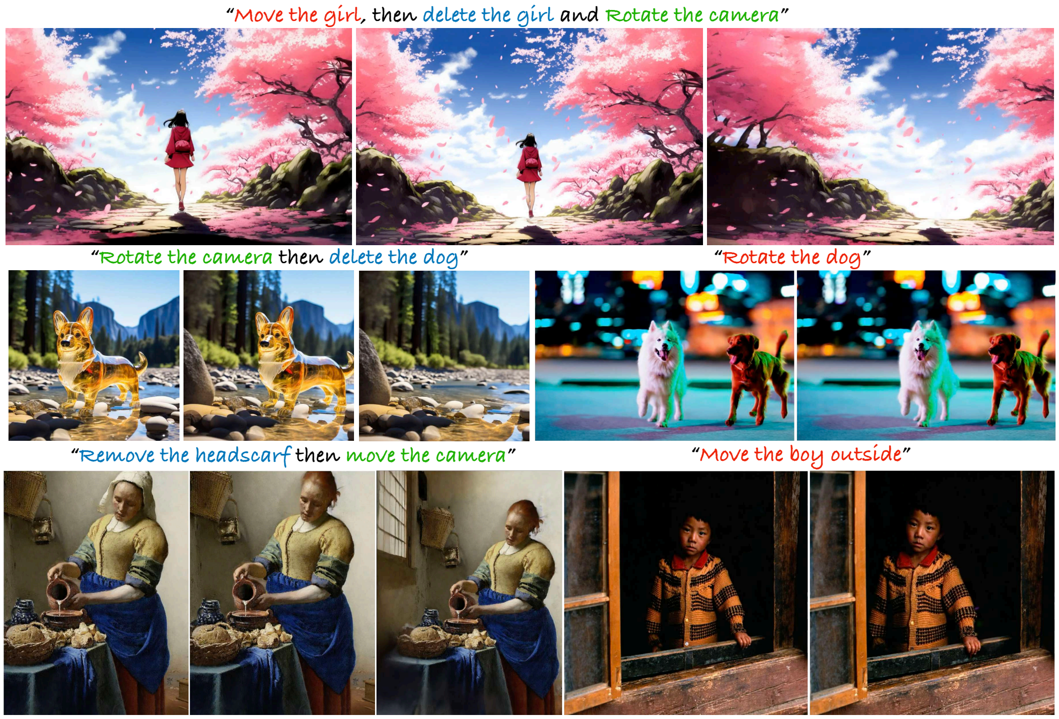

"Move the girl, then delete the girl and Rotate the camera"
"Rotate the camera then delete the dog"
"Rotate the dog"
"Remove the headscarf then move the camera"
"Move the boy outside"

Figure 1: **Image pairs edited by `3DitScene`.** Our method is capable of simultaneously handling various types of edits in both 2D and 3D spaces.

Abstract

Scene image editing is crucial for entertainment, photography, and advertising design. Existing methods solely focus on either 2D individual object or 3D global scene editing. This results in a lack of a unified approach to effectively control and manipulate scenes at the 3D level with different levels of granularity. In this work, we propose `3DitScene`, a novel and unified scene editing framework leveraging language-guided disentangled Gaussian Splatting that enables seamless editing from 2D to 3D, allowing precise control over scene composition and individual objects. We first incorporate 3D Gaussians that are refined through generative priors and optimization techniques. Language features from CLIP then introduce semantics into 3D geometry for object disentanglement. With the disentangled Gaussians, `3DitScene` allows for manipulation at both the global and individual levels, revolutionizing creative expression and empowering control over scenes and objects. Experimental results demonstrate the effectiveness and versatility of `3DitScene` in scene image editing. Code and models will be made publicly available.

1 Introduction

Editing scene images is of great importance in various fields, ranging from entertainment, professional photography and advertising design. Content editing allows to create immersive and captivating experiences for audiences, convey the artistic vision effectively and achieve the desired aesthetic outcomes. With the rapid development of deep generative modeling, many attempts have been made to edit an image effectively. However, they have encountered limitations that hindered their potential.

Previous methods primarily concentrate on scene editing in 2D image space. They commonly rely on generative priors, such as GANs and Diffusion Models (DM), and employ techniques like modification of cross-attention mechanisms (Hertz et al., 2022; 2023), and optimization of network parameters (Kim et al., 2022; Kawar et al., 2023; Ruiz et al., 2023; Gal et al., 2022; Chen et al., 2023b) to edit the appearance and object identity within scene images. While some efforts have been made to extend these methods to 3D editing, they ignore 3D cues and pose a challenge in maintaining 3D consistency, especially when changing the camera pose. Moreover, these approaches typically focus on global scenes and lack the ability to disentangle objects accurately, resulting in limited control over individual objects at the 3D level.

In order to edit any scene images and enable 3D control over both scene and its individual objects, we propose `3DitScene`, a novel scene editing framework which leverage a new scene representation, language-guided disentangled Gaussian Splatting. Concretely, the given image is first projected into 3D Gaussians which are further refined and enriched through 2D generative prior (Rombach et al., 2022; Poole et al., 2022). We thus obtain a comprehensive 3D scene representation that naturally enables novel view synthesis for a given image. In addition, language features from CLIP are distilled into the corresponding 3D Gaussians to introduce semantics into 3D geometry. These semantic 3D Gaussians help disentangle individual objects out of the entire scene representation, resulting in language-guided disentangled Gaussians for scene decomposition. They also allow for a more user-friendly interaction *i.e.,* users could query specific objects or interest via text. To this end, our `3DitScene` enables seamless editing from 2D to 3D and allow for modifications at both the global and individual levels, empowering creators to have precise control over scene composition, and object-level edits.

We dub our pipeline as `3DitScene`. Different from previous works that focus on addressing a single type of editing, `3DitScene` integrates editing requirements within a unified framework. Our teaser figure demonstrates the versatility of `3DitScene` by showcasing its application to diverse scene images. We have conducted evaluations of `3DitScene` under various settings, and the results demonstrate significant improvements over baseline methods.

2 Related work

Image Editing with Generative Models. The field of 2D image synthesis has advanced significantly with the development of generative models such as GANs (Karras et al., 2021; 2019) and diffusion models (Rombach et al., 2022; Song et al., 2020; Ho et al., 2020). Many studies capitalize on the rich prior knowledge embedded in generative models for image editing. Some endeavors utilize GANs for various image editing tasks, including image-to-image translation, latent manipulation (Shen et al., 2020; Yang et al., 2021; Zhu et al., 2020; Xu et al., 2021; Jahanian et al., 2019), and text-guided manipulation (Patashnik et al., 2021). However, due to limitations in training on large-scale data, GANs often struggle to perform well on real-world scene images. As diffusion models make notable progress, the community is increasingly focusing on harnessing the potent text-to-image diffusion model for real image editing (Kim et al., 2022; Kawar et al., 2023; Ruiz et al., 2023; Gal et al., 2022; Chen et al., 2023b; Hertz et al., 2022; 2023; Meng et al., 2021b; Su et al., 2022). However, these methods are confined to the 2D domain and are limited in editing objects within a 3D space. Concurrently, other research efforts (Yenphraphai et al., 2024a) attempt to address 3D-aware image editing, but they introduces inconsistency in the editing process, and cannot change the camera perspective of the entire scene. Wang et al. (2024a); Chen et al. (2024a); Ye et al. (2023); Palandra et al. (2024); Wu et al. (2024a); Wang et al. (2024b); Jaganathan et al. (2024) focus on editing with a given 3DGS scene, but is limited in the types of edits they support. In contrast, our model leverages an explicit 3D Gaussian to convert 2D images into 3D space while disentangling

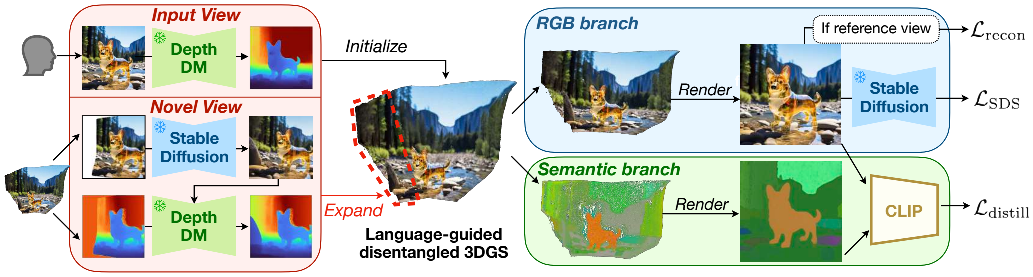


Figure 2: **3DitScene training pipeline.** Given input view, we first initialize 3DGS by lifting pixels to 3D space and then expand it over novel views by RGB and depth inpainting. Semantic features are then distilled into 3D Gaussians to achieve object-level disentanglement.

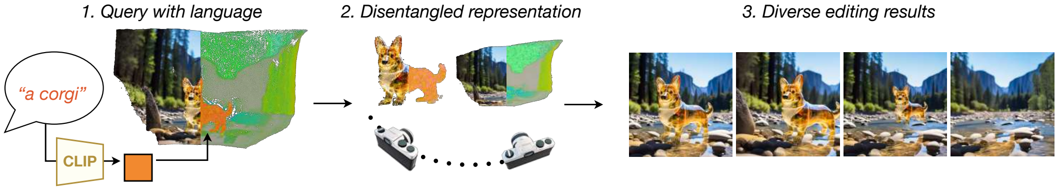


Figure 3: **3DitScene Inference pipeline.** User can query object of interest via language prompt. Enabled by the disentangled 3D representation, user can change camera viewpoint, and manipulate the object of interest in a flexible manner.

objects with language guidance. This approach enables our model not only to consistently perform 3D-aware object editing but also facilitates scene-level novel-view synthesis.

Single-view 3D Scene Synthesis. Among 3D scenes generation (Zhang et al., 2023b; Höllein et al., 2023; Chung et al., 2023; Chen et al., 2023a;c; Mao et al., 2023; Epstein et al., 2024), conditional generation on a single-view presents an unique challenge. Previous approaches address this challenge by training a versatile model capable of inferring a 3D representation of a scene based on a single input image (Wiles et al., 2020; Tucker & Snavely, 2020; Hu et al., 2021; Han et al., 2022; Flynn et al., 2019; Li et al., 2021; Hong et al., 2023; Yu et al., 2021). However, these methods demand extensive datasets for training and tend to produce blurry textures when confronted with significant changes in camera viewpoints. Recently, several works have embraced diffusion priors (Liu et al., 2023; Chan et al., 2023; Xu et al., 2023; Gu et al., 2023; Tang et al., 2023; Qian et al., 2023; Chen et al., 2024b) to acquire a probabilistic distribution for unseen views, leading to better synthesis results. Nevertheless, these methods often concentrate on object-centric scenes or lack 3D consistency. Our approach connect 2D images and 3D scenes with explicit 3D Gaussians and incorporate diffusion knowledge, which overcome the mentioned challenges.

3 Method

Our target is to propose a 3D-aware scene image editing framework (Fig. 2) that allows simultaneous control over the camera and objects. To accomplish this, Sec. 3.1 introduces a novel scene representation called language-guided disentangled Gaussian splatting. In order to achieve object-level control, Sec. 3.2 further distills language features into the Gaussian splatting representation, achieving disentanglement at the object level. We elaborate the optimization process in Sec. 3.3 and demonstrate the flexible user control enabled by our framework during inference in Sec. 3.4.

3.1 3D Gaussian Splatting from Single Image

Preliminary. 3D Gaussian Splatting (3DGS) (Kerbl et al., 2023) has been proved effective in both reconstructive (Luiten et al., 2023; Yang et al., 2023) and generative setting (Zou et al., 2023; Tang et al., 2023). It represents a 3D scene via a set of explicit 3D Gaussians. Each 3D Gaussian describes its location by a center vector $\mathbf{x} \in \mathbb{R}^3$, a scaling factor $\mathbf{s} \in \mathbb{R}^3$, a rotation quaternion $\mathbf{q} \in \mathbb{R}^4$, and also stores an opacity value $\alpha \in \mathbb{R}$ and spherical harmonics (SH) coefficients $\mathbf{c} \in \mathbb{R}^k$ (k represents the degrees of freedom of SH) for volumetric rendering. All the above parameters can be denoted as $\Theta = \{\mathbf{x}_i, \mathbf{s}_i, \mathbf{q}_i, \alpha_i, \mathbf{c}_i | i \in [0, \cdots, N-1]\}$, where N is the number of 3D Gaussians. A tile-based rasterizer is used to render these Gaussians into 2D image.

Image-to-3DGS initialization. Given an input image $\mathbf{I} \in \mathbb{R}^{3\times H\times W}$, an off-the-shelf depth prediction model is applied to estimate its depth map $\mathbf{D} \in \mathbb{R}^{H\times W}$. Then, we could transform image pixels into 3D space, forming the corresponding 3D point clouds:

$$\mathcal{P} = \phi_{2\rightarrow 3}(\mathbf{I}, \mathbf{D}, \mathbf{K}, \mathbf{T}), \tag{1}$$

where $\mathbf{K}$ and $\mathbf{T}$ are camera intrinsic and extrinsic matrices respectively. Such point clouds $\mathcal{P}$ are then used to initialize the 3DGS by directly copying the location and color values, with other GS-related parameters randomly initialized. To refine the 3DGS's appearance, we adopt a reconstruction loss:

$$\mathcal{L}_{\text{recon}} = \|\mathbf{I} - f(\mathcal{P}, \mathbf{K}, \mathbf{T})\|_2^2, \tag{2}$$

where f is the rendering function.

We further enhance the rendered quality by leveraging prior knowledge from image generative foundation model, namely Stable Diffusion (Rombach et al., 2022). It provides update direction to the images rendered by the current 3DGS in the form of Score Distillation Sampling (Poole et al., 2022) loss, denoted as $\mathcal{L}_{\text{SDS}}$.

3DGS expansion by inpainting. When camera perspectives changes, rendered views will contain holes due to occlusion or new region outside the original view frustum. We use Stable Diffusion to inpaint the uncovered regions. Then, the newly added pixels need to be accurately transformed into 3D space to align seamlessly with the existing 3D Gaussians.

Previous methods (Chung et al., 2023; Höllein et al., 2023; Yu et al., 2023) first predict the depth values, and then use heuristic methods to adjust the values to align with the existing 3D structure. However, relying on heuristic methods often overlooked various scenarios, leading to artifacts such as depth discontinuities or shape deformations.

Instead, we propose a novel method to lifted novel contents to 3D while ensuring seamless alignment without any heuristic procedures. The key insight is to treat the problem as an image inpainting task, and utilize state-of-the-art diffusion-based depth estimation models (Ke et al., 2024; Fu et al., 2024; Yang et al., 2024) as a prior to solve the task. During denoising steps, rather than using models to predict the noise over the entire image, we employ the forward diffusion process to determine the value of fixed areas (Meng et al., 2021a). This approach guarantees the final result, after denoising, adheres to the depth of original fixed parts, ensuring smooth expansion.

After smooth 3DGS expansion via depth inpainting, we take the imagined novel views as reference views, and apply reconstruction loss $\mathcal{L}_{\text{recon}}$ to supervise the updated 3DGS. SDS loss $\mathcal{L}_{\text{SDS}}$ is adopted for views rendered from camera perspectives that are interpolated between the user-provided viewpoint and the newly imagined views.

3.2 Language-guided Disentangled Gaussian Splatting

Based on the 3DGS built from single input image, users can generate novel views. In this section, we further distill CLIP (Radford et al., 2021) language feature to 3D Gaussians. This introduce semantics into 3D geometry, which helps disentangle individual objects out of the entire scene representation.

Language feature distillation. We augment each 3D Gaussian with a language embedding $\mathbf{e} \in \mathbb{R}^C$, where C denotes the number of the channels. Similar to RGB image $\mathbf{I}$, a 2D semantic feature map $\mathbf{E} \in \mathbb{R}^{C\times H\times W}$ can also be rendered by the rasterizer. To learn the embedding, we first use Segment Anything Model (SAM) (Kirillov et al., 2023; Zhang et al., 2023a) to get semantic masks $\mathbf{M}_i$. Then, we can obtain embedding of each object $\mathbf{I} \odot \mathbf{M}_i$ and supervise the corresponding region on rendered feature map $\mathbf{E}$, according to the distillation loss:

$$\mathcal{L}_{\text{distill}} = \sum_i \left\| \left(\mathbf{E} - g\left(\mathbf{I} \odot \mathbf{M}_i\right) \right) \odot \mathbf{M}_i \right\|_2^2, \tag{3}$$

Table 1: **User study result.** We report the percentage of favorite users for the consistency and quality of images edited by each method

		AnyDoor	Ojbect 3DIT	Image Scuplting	Ours
Consistency	Human	5.1	16.8	12.7	**65.4**
	GPT4-v	0.0	6.7	31.3	**62.0**
Quality	Human	10.4	0.5	25.1	**64.0**
	GPT4-v	6.7	13.3	39.2	**40.8**

where g is the CLIP's image encoder, and $\odot$ denotes element-wise multiplication. Following LangSplat (Qin et al., 2024), we additionally train an autoencoder to compress the embedding space to optimize the memory consumption of language embedding $\mathbf{e}$.

Scene decomposition. After distillation, we can decompose the scene into different objects. This enables user to query and ground specific object, and perform editing over single object (*e.g.* translation, rotation, removal, re-stylizing).

It is worth noting that such scene decomposition property not only enables more flexible edits during inference stage, but also provides augmentation over scene layouts during the optimization process. Since now we can query and render each object independently, we apply random translation, rotation, and removal over objects. This augmentation over the scene layout leads to a significant improvement in the appearance of occluded regions, ultimately enhancing the overall quality of the edited views (see Sec. 4.4).

3.3 Training

The overall training objective can be expressed as:

$$\mathcal{L} = \lambda_{\text{recon}}\mathcal{L}_{\text{recon}} + \lambda_{\text{SDS}}\mathcal{L}_{\text{SDS}} + \lambda_{\text{distill}}\mathcal{L}_{\text{distill}}, \tag{4}$$

where λ_{recon}, λ_{SDS} and λ_{distill} are coefficients that balance each loss term.

3.4 Inference

Due to the disentangled nature of our representation, users can now interact with and manipulate objects in a flexible manner. Here, we mainly discuss prompting objects via two different modalities:

Text prompt. Users can query an object through text prompts as shown in Fig. 3. Following LERF (Kerr et al., 2023) and LangSplat (Qin et al., 2024), we calculate the relevancy score `score` between the language embedding $\mathbf{e}$ in the 3D Gaussians and the embedding of the text prompt $\mathbf{e}_l$ as:

$$\texttt{score} = \min_i \frac{\exp(\mathbf{e}\cdot\mathbf{e}_l)}{\exp(\mathbf{e}\cdot\mathbf{e}_l) + \exp(\mathbf{e}\cdot\mathbf{e}^i_{\text{canon}})}, \tag{5}$$

where $\mathbf{e}^i_{\text{canon}}$ is the CLIP embeddings of canonical phrases including "*object*", "*things*", "*stuff*", and "*texture*". Gaussians that have relevance scores below a predefined threshold are excluded. The remaining part is identified as the object of user interest.

Bounding box. Users can also select an object by drawing an approximate bounding box around it on the input image. We group 3D Gaussians within the bounding box by K-Means clustering, and discard clusters whose number of Gaussians does not exceed a threshold proportion.

In the meantime, user can also adjust the camera by specifying intrinsic and extrinsic parameters.

4 Experiments

4.1 Settings

Implementation details. To lift an image to 3D, we use GeoWizard (Fu et al., 2024) to estimate its relative depth. Stable Diffusion (Rombach et al., 2022)'s inpainting pipeline is adopted to

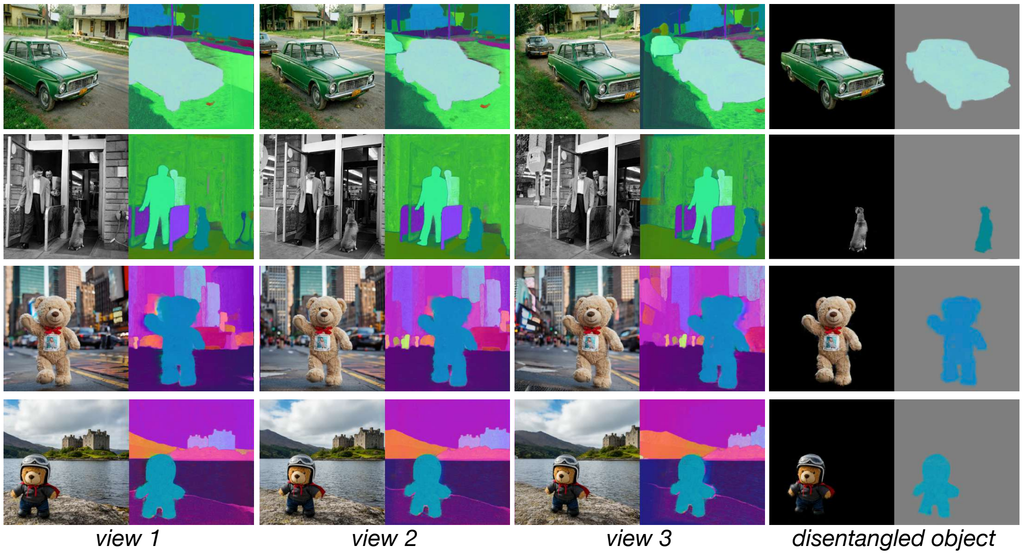


Figure 4: **Visualization of rendered images and feature maps.** For each sample, we show three views of rendered images and feature maps. To demonstrate the disentangled scene representation, we use the language embedding to select a foreground object and render it exclusively.

generate new content for 3DGS's expansion. We leverage MobileSAM (Zhang et al., 2023a) and OpenCLIP (Ilharco et al., 2021) to segment and compute rendered views' feature maps, which are further leveraged to supervise the language embedding of 3D Gaussians. We use Stable Diffusion to perform Score Distillation Sampling (Poole et al., 2022) during optimization. Given the already decent image quality at the start of optimization benefited from explicit 3DGS initialization, we adopt a low classifier-free guidance (Ho & Salimans, 2022) scale.

Baselines. We compare our method with following scene image editing works: (1) AnyDoor (Chen et al., 2023b) is a 2D diffusion-based model that can teleport target objects into given scene images. It leverages Stable Diffusion's powerful image generative prior by finetuning upon it. (2) Object 3DIT (Michel et al., 2024) is designed for 3D-aware object-centric image editing via language instructions. It finetunes Stable Diffusion over a synthetic dataset containing pairs of original image, language instruction, and edited image. (3) Image Sculpting (Yenphraphai et al., 2024b) is also designed for 3D-aware object-centric image editing. It estimates a 3D model from an object in the input image to enable precise 3D control over the geometry. It also uses Stable Diffusion to refine the edited image quality. (4) AdaMPI (Han et al., 2022) focuses on the control over camera perspective. It leverages monocular depth estimation and color inpainting with learned adaptive layered depth representations. (5) LucidDreamer (Chung et al., 2023) tackles novel view synthesis by querying Stable Diffusion's inpainting pipeline with dense camera trajectories.

4.2 Quantitative results

We conduct a user study to compare the edited results by our method with the established baselines. We generate 20 samples for each method and request users to vote for their preferred method based on consistency with the original image and quality for each sample. We collect feedback from 25 users, and report the result in Tab. 1. Our method consistently outperforms previous baselines in terms of both consistency and image quality. As recommended in a previous study (Wu et al., 2024b), GPT4-v has the ability to evaluate 3D consistency and image quality. Therefore, we include GPT-4v as an additional criterion. The preference of GPT4-v is well aligned with human preference, which once again demonstrates the superiority of `3DitScene`.

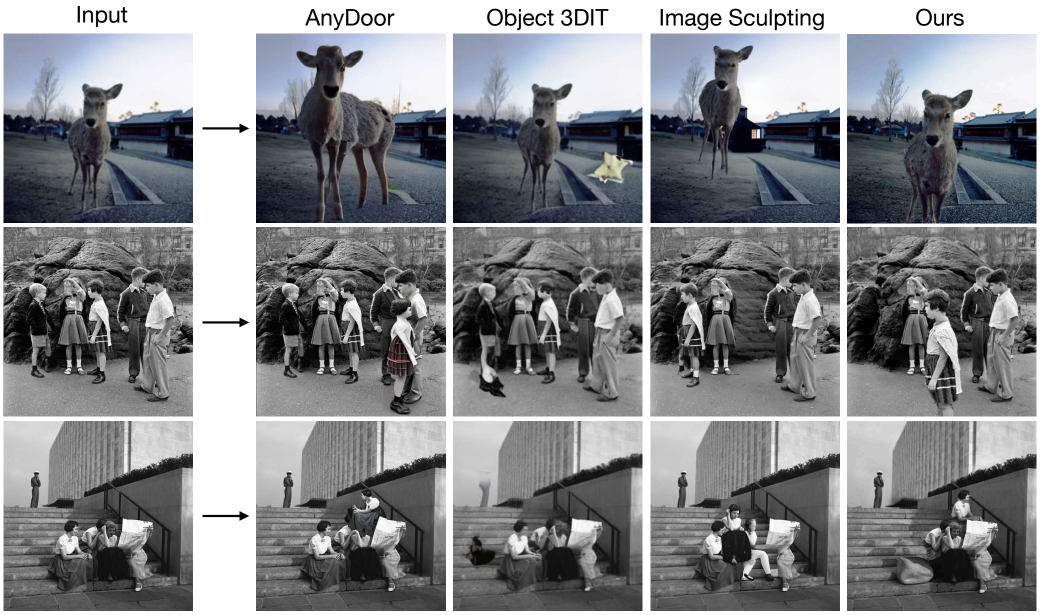


Figure 5: **Comparison results of object-centric manipulation.** We apply translation, resizing, and removal over foreground objects.

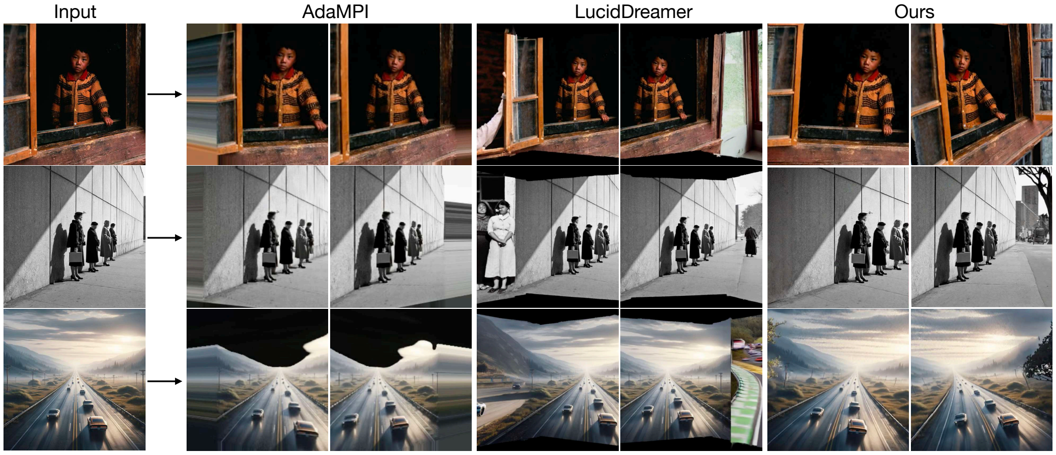


Figure 6: **Comparison results of camera control.** We show two views with different camera perspectives for each method.

4.3 Qualitative results

Fig. 4 showcases the generated novel views with their respective feature maps produced by our framework. The feature maps demonstrate remarkable accuracy in capturing the semantic content of the images. This ability to distinctly separate semantic information plays a crucial role in achieving precise object-level control In the following, we demonstrate flexible editing over scene images enabled by our framework, and also compare with baseline methods.

Object manipulation. Since different methods define object manipulation, particularly translation operations, in different coordinate systems[1], it becomes challenging to evaluate them under a unified and fair setting. Therefore, we evaluate each method under its own specific setting to achieve the

[1] AnyDoor, Object 3DIT and Image Sculpting respectively employs 2D masks, language prompts, and image coordinates for control. We use coordinates in 3D space instead.

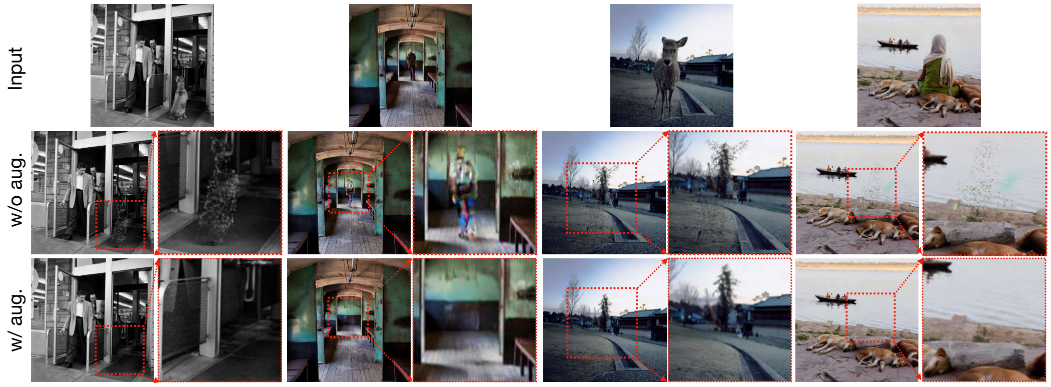


Figure 7: **Ablation results for layout augmentation during optimization.** To evaluate the degree of object-level disentanglement, we conduct object removal for each sample. The top row displays the input image, while the next two rows showcase the edited scene

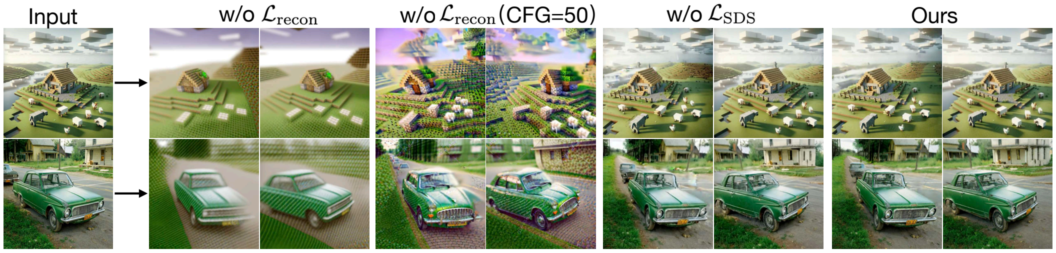

Input
w/o $\mathcal{L}_{\text{recon}}$
w/o $\mathcal{L}_{\text{recon}}$(CFG=50)
w/o $\mathcal{L}_{\text{SDS}}$
Ours

Figure 8: **Ablation results for loss terms.** We show rendered novel views under different loss settings. The left column lists the input image. In right columns, two views are shown for each configuration. The quality degrades when reconstruction or SDS loss term is discarded

best possible result. As shown in Fig. 5, AnyDoor struggles to maintain object identity and 3D consistency when manipulating object layouts, primarily due to the absence of 3D cues. Object 3DIT, trained on synthetic datasets, exhibits limited generalization ability to real images. By leveraging a 3D model derived from the input image, Image Sculpting achieves better results. Nonetheless, it encounters issues with inconsistency when manipulating objects.

In contrast, our method delivers satisfactory 3D-aware object-level editing results. It maintains accurate 3D consistency of edited objects after rearranging their layout. Additionally, it preserves occlusion relationships within the scene, such as moving the girl to be partially occluded by a foreground object in the last row example.

Camera control. We compare our methods with AdaMPI and LucidDreamer for camera control. As illustrated in Fig. 6, AdaMPI only focuses on scenarios where the camera zooms in, and does not consider novel view synthesis. Therefore, this approach is not suitable for 3D-aware image editing when large camera control is required. LucidDreamer also leverages Stable Diffusion's inpainting capacity for novel view synthesis. However, it suffers from sudden transitions in the content within the frame (see sample in the bottom line). It also requires dense camera poses. In contrast, our method only needs as few as three camera poses and enables smooth transitions from the input view to novel views, enhancing user control over the camera perspective.

4.4 Ablation study

Layout augmentation during optimization. As our representation disentangles at object level, we could perform layout augmentation during optimization. Here, we investigate whether disentanglement property benefits the optimization process. We use the task of removing objects to evaluate the degree of disentanglement.

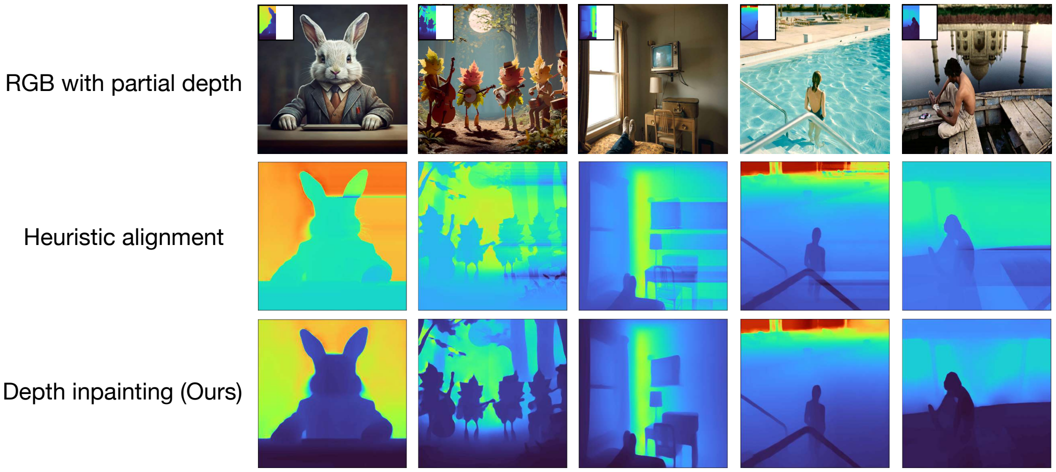


Figure 9: **Ablation results for depth inpainting.** The first row shows images with their corresponding depth maps (available on the left half). The second and third row display the depth map predicted by heuristic alignment, and our depth inpainting method respectively.

As illustrated in Fig. 7, when layout augmentation is disabled during optimization, floating artifacts can be observed. We discover that these Gaussians lie inside the object. They are occluded by Gaussians at the surface. As they do not contribute to the rendering result, they are consequently not updated by gradient descent during optimization, leaving their language embeddings unsupervised.

In contrast, when applying layout augmentation during optimization, such Gaussians will be exposed when the foreground object is moved away, and hence updated. With this ablation, it is concluded that the disentanglement property of the proposed representation not only enables more flexible inference, but also contributes to the optimization process.

Loss terms. During optimization, we adopt three loss terms: $\mathcal{L}_{\text{recon}}$, $\mathcal{L}_{\text{SDS}}$, and $\mathcal{L}_{\text{distill}}$. $\mathcal{L}_{\text{distill}}$ plays a critical role in distilling language embedding into 3D. The remaining two terms focus on enhancing the visual quality of images. As illustrated in Fig. 8, the image quality degrades severely without $\mathcal{L}_{\text{recon}}$ or $\mathcal{L}_{\text{SDS}}$. Without $\mathcal{L}_{\text{recon}}$, the image is only refined by the SDS loss, which creates discrepancies with the original image. When the CFG value is set low, 5 as default, the image appears lacking in details and exhibits unusual texture patterns. Increasing the CFG value introduces more details, yet leads to inconsistencies with the original image, while the issue of strange texture patterns persists. Additionally, only applying $\mathcal{L}_{\text{recon}}$ results to floating artifacts and blurriness across the entire image. In conclusion, both SDS and reconstruction loss are crucial for achieving decent image quality.

Depth inpainting. When expanding 3DGS at novel views, we need to estimate the depth map of unseen regions. Here, we compare our inpainting-based depth estimation with heuristic-based method. Fig. 9 show images with depth map available in the left part. The task is to predict the depth map of the right part. Method relying on heuristic alignment results to artifacts like depth discontinuity. In contrast, our proposed method is capable of producing accurate depth maps that align well with the left known part.

5 Conclusion and Discussion

We present a novel framework, `3DitScene`, for scene image editing. Our primary objective is to facilitate 3D-aware editing of both objects and the entire scene within a unified framework. We achieve this by leveraging a new scene representation, language-guided disentangled scene representation. This representation is learnt by distilling CLIP's language feature into 3D Gaussians. The semantic 3D Gaussians effectively disentangle individual objects out of the entire scene, , thereby enabling localized object editing. We test `3DitScene` under different settings and prove its superiority compared to previous methods.

References

Eric R Chan, Koki Nagano, Matthew A Chan, Alexander W Bergman, Jeong Joon Park, Axel Levy, Miika Aittala, Shalini De Mello, Tero Karras, and Gordon Wetzstein. Generative novel view synthesis with 3d-aware diffusion models. *ICCV*, 2023.

Dave Zhenyu Chen, Haoxuan Li, Hsin-Ying Lee, Sergey Tulyakov, and Matthias Nießner. Scenetex: High-quality texture synthesis for indoor scenes via diffusion priors. *arXiv preprint arXiv:2311.17261*, 2023a.

Xi Chen, Lianghua Huang, Yu Liu, Yujun Shen, Deli Zhao, and Hengshuang Zhao. Anydoor: Zero-shot object-level image customization. *arXiv preprint arXiv:2307.09481*, 2023b.

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*, pp. 21476–21485, 2024a.

Zhaoxi Chen, Guangcong Wang, and Ziwei Liu. Scenedreamer: Unbounded 3d scene generation from 2d image collections. *arXiv preprint arXiv:2302.01330*, 2023c.

Zilong Chen, Feng Wang, Yikai Wang, and Huaping Liu. Text-to-3d using gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21401–21412, 2024b.

Jaeyoung Chung, Suyoung Lee, Hyeongjin Nam, Jaerin Lee, and Kyoung Mu Lee. Luciddreamer: Domain-free generation of 3d gaussian splatting scenes. *arXiv preprint arXiv:2311.13384*, 2023.

Dave Epstein, Ben Poole, Ben Mildenhall, Alexei A Efros, and Aleksander Holynski. Disentangled 3d scene generation with layout learning. *arXiv preprint arXiv:2402.16936*, 2024.

John Flynn, Michael Broxton, Paul Debevec, Matthew DuVall, Graham Fyffe, Ryan Overbeck, Noah Snavely, and Richard Tucker. Deepview: View synthesis with learned gradient descent. In *CVPR*, 2019.

Xiao Fu, Wei Yin, Mu Hu, Kaixuan Wang, Yuexin Ma, Ping Tan, Shaojie Shen, Dahua Lin, and Xiaoxiao Long. Geowizard: Unleashing the diffusion priors for 3d geometry estimation from a single image. *arXiv preprint arXiv:2403.12013*, 2024.

Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618*, 2022.

Jiatao Gu, Alex Trevithick, Kai-En Lin, Joshua M Susskind, Christian Theobalt, Lingjie Liu, and Ravi Ramamoorthi. Nerfdiff: Single-image view synthesis with nerf-guided distillation from 3d-aware diffusion. In *ICML*, 2023.

Yuxuan Han, Ruicheng Wang, and Jiaolong Yang. Single-view view synthesis in the wild with learned adaptive multiplane images. In *ACM SIGGRAPH Conference Proceedings*, 2022.

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.

Amir Hertz, Andrey Voynov, Shlomi Fruchter, and Daniel Cohen-Or. Style aligned image generation via shared attention. *arXiv preprint arXiv:2312.02133*, 2023.

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

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. 2020.

Lukas Höllein, Ang Cao, Andrew Owens, Justin Johnson, and Matthias Nießner. Text2room: Extracting textured 3d meshes from 2d text-to-image models. *arXiv preprint arXiv:2303.11989*, 2023.

Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. Lrm: Large reconstruction model for single image to 3d. *arXiv preprint arXiv:2311.04400*, 2023.

Ronghang Hu, Nikhila Ravi, Alexander C Berg, and Deepak Pathak. Worldsheet: Wrapping the world in a 3d sheet for view synthesis from a single image. In *ICCV*, 2021.

Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, July 2021. URL `https://doi.org/10.5281/zenodo.5143773`. If you use this software, please cite it as below.

Vishnu Jaganathan, Hannah Hanyun Huang, Muhammad Zubair Irshad, Varun Jampani, Amit Raj, and Zsolt Kira. Ice-g: Image conditional editing of 3d gaussian splats. *arXiv preprint arXiv:2406.08488*, 2024.

Ali Jahanian, Lucy Chai, and Phillip Isola. On the" steerability" of generative adversarial networks. *arXiv preprint arXiv:1907.07171*, 2019.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019.

Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. 2021.

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 *CVPR*, 2023.

Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9492–9502, 2024.

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.

Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. In *CVPR*, pp. 19729–19739, 2023.

Gwanghyun Kim, Taesung Kwon, and Jong Chul Ye. Diffusionclip: Text-guided diffusion models for robust image manipulation. In *CVPR*, 2022.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.

Jiaxin Li, Zijian Feng, Qi She, Henghui Ding, Changhu Wang, and Gim Hee Lee. Mine: Towards continuous depth mpi with nerf for novel view synthesis. In *ICCV*, 2021.

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

Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. *arXiv preprint arXiv:2308.09713*, 2023.

Weijia Mao, Yan-Pei Cao, Jia-Wei Liu, Zhongcong Xu, and Mike Zheng Shou. Showroom3d: Text to high-quality 3d room generation using 3d priors. *arXiv preprint arXiv:2312.13324*, 2023.

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. *arXiv preprint arXiv:2108.01073*, 2021a.

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. *arXiv preprint arXiv:2108.01073*, 2021b.

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

Francesco Palandra, Andrea Sanchietti, Daniele Baieri, and Emanuele Rodolà. Gsedit: Efficient text-guided editing of 3d objects via gaussian splatting. *arXiv preprint arXiv:2403.05154*, 2024.

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

Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.

Guocheng Qian, Jinjie Mai, Abdullah Hamdi, Jian Ren, Aliaksandr Siarohin, Bing Li, Hsin-Ying Lee, Ivan Skorokhodov, Peter Wonka, Sergey Tulyakov, et al. Magic123: One image to high-quality 3d object generation using both 2d and 3d diffusion priors. *arXiv preprint arXiv:2306.17843*, 2023.

Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. Langsplat: 3d language gaussian splatting. In *CVPR*, 2024.

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*, pp. 8748–8763. PMLR, 2021.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *CVPR*, 2023.

Yujun Shen, Ceyuan Yang, Xiaoou Tang, and Bolei Zhou. Interfacegan: Interpreting the disentangled face representation learned by gans. *IEEE TPAMI*, 2020.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Xuan Su, Jiaming Song, Chenlin Meng, and Stefano Ermon. Dual diffusion implicit bridges for image-to-image translation. *arXiv preprint arXiv:2203.08382*, 2022.

Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. *arXiv preprint arXiv:2309.16653*, 2023.

Richard Tucker and Noah Snavely. Single-view view synthesis with multiplane images. In *CVPR*, 2020.

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*, pp. 20902–20911, 2024a.

Yuxuan Wang, Xuanyu Yi, Zike Wu, Na Zhao, Long Chen, and Hanwang Zhang. View-consistent 3d editing with gaussian splatting. *arXiv preprint arXiv:2403.11868*, 20, 2024b.

Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. Synsin: End-to-end view synthesis from a single image. In *CVPR*, 2020.

Jing Wu, Jia-Wang Bian, Xinghui Li, Guangrun Wang, Ian Reid, Philip Torr, and Victor Adrian Prisacariu. Gaussctrl: multi-view consistent text-driven 3d gaussian splatting editing. *arXiv preprint arXiv:2403.08733*, 2024a.

Tong Wu, Guandao Yang, Zhibing Li, Kai Zhang, Ziwei Liu, Leonidas Guibas, Dahua Lin, and Gordon Wetzstein. Gpt-4v (ision) is a human-aligned evaluator for text-to-3d generation. *arXiv preprint arXiv:2401.04092*, 2024b.

Yinghao Xu, Yujun Shen, Jiapeng Zhu, Ceyuan Yang, and Bolei Zhou. Generative hierarchical features from synthesizing images. In *CVPR*, 2021.

Yinghao Xu, Hao Tan, Fujun Luan, Sai Bi, Peng Wang, Jiahao Li, Zifan Shi, Kalyan Sunkavalli, Gordon Wetzstein, Zexiang Xu, et al. Dmv3d: Denoising multi-view diffusion using 3d large reconstruction model. *arXiv preprint arXiv:2311.09217*, 2023.

Ceyuan Yang, Yujun Shen, and Bolei Zhou. Semantic hierarchy emerges in deep generative representations for scene synthesis. *IJCV*, 2021.

Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. *arXiv preprint arXiv:2401.10891*, 2024.

Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. *arXiv preprint arXiv:2309.13101*, 2023.

Mingqiao Ye, Martin Danelljan, Fisher Yu, and Lei Ke. Gaussian grouping: Segment and edit anything in 3d scenes. *arXiv preprint arXiv:2312.00732*, 2023.

Jiraphon Yenphraphai, Xichen Pan, Sainan Liu, Daniele Panozzo, and Saining Xie. Image sculpting: Precise object editing with 3d geometry control. *arXiv preprint arXiv:2401.01702*, 2024a.

Jiraphon Yenphraphai, Xichen Pan, Sainan Liu, Daniele Panozzo, and Saining Xie. Image sculpting: Precise object editing with 3d geometry control. *arXiv preprint arXiv:2401.01702*, 2024b.

Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *CVPR*, 2021.

Hong-Xing Yu, Haoyi Duan, Junhwa Hur, Kyle Sargent, Michael Rubinstein, William T Freeman, Forrester Cole, Deqing Sun, Noah Snavely, Jiajun Wu, et al. Wonderjourney: Going from anywhere to everywhere. *arXiv preprint arXiv:2312.03884*, 2023.

Chaoning Zhang, Dongshen Han, Yu Qiao, Jung Uk Kim, Sung-Ho Bae, Seungkyu Lee, and Choong Seon Hong. Faster segment anything: Towards lightweight sam for mobile applications. *arXiv preprint arXiv:2306.14289*, 2023a.

Qihang Zhang, Chaoyang Wang, Aliaksandr Siarohin, Peiye Zhuang, Yinghao Xu, Ceyuan Yang, Dahua Lin, Bolei Zhou, Sergey Tulyakov, and Hsin-Ying Lee. Scenewiz3d: Towards text-guided 3d scene composition. *arXiv preprint arXiv:2312.08885*, 2023b.

Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. In-domain gan inversion for real image editing. In *ECCV*, 2020.

Zi-Xin Zou, Zhipeng Yu, Yuan-Chen Guo, Yangguang Li, Ding Liang, Yan-Pei Cao, and Song-Hai Zhang. Triplane meets gaussian splatting: Fast and generalizable single-view 3d reconstruction with transformers. *arXiv preprint arXiv:2312.09147*, 2023.

Appendix

A Implementation Details

Scene initialization. We first utilize GeoWizard (Fu et al., 2024) to estimate the relative depth of the input image. Next, we lift the image into 3D space based on this depth and perform 300 SDS steps. The camera azimuth angle is randomly sampled from $[-15°, 15°]$. After 300 steps, we employ Stable Diffusion (Rombach et al., 2022)'s inpainting pipeline to generate new content for the expansion of 3DGS. Specifically, we inpaint the rendeorange views at azimuth angles of $-15°$ and $15°$. Finally, we use GeoWizard (Fu et al., 2024) again to estimate the depth of the newly added regions and lift them into 3D space.

SDS optimization. We perform 1500 SDS steps to optimize the whole scene. We randomly sample the diffusion time step from $[l, r]$, where $l = 0.02$, and r starts at 0.5 and gradually decreases to 0.2 by the 1000th step. We use guidance strength of 5 for classifier-free guidance.

Coefficients. In Eq. (4), we choose $\lambda_{\text{recon}} = 1000$, $\lambda_{\text{SDS}} = 0.01$, and $\lambda_{\text{distill}} = 1$.

B Editing Pipeline and User Interface

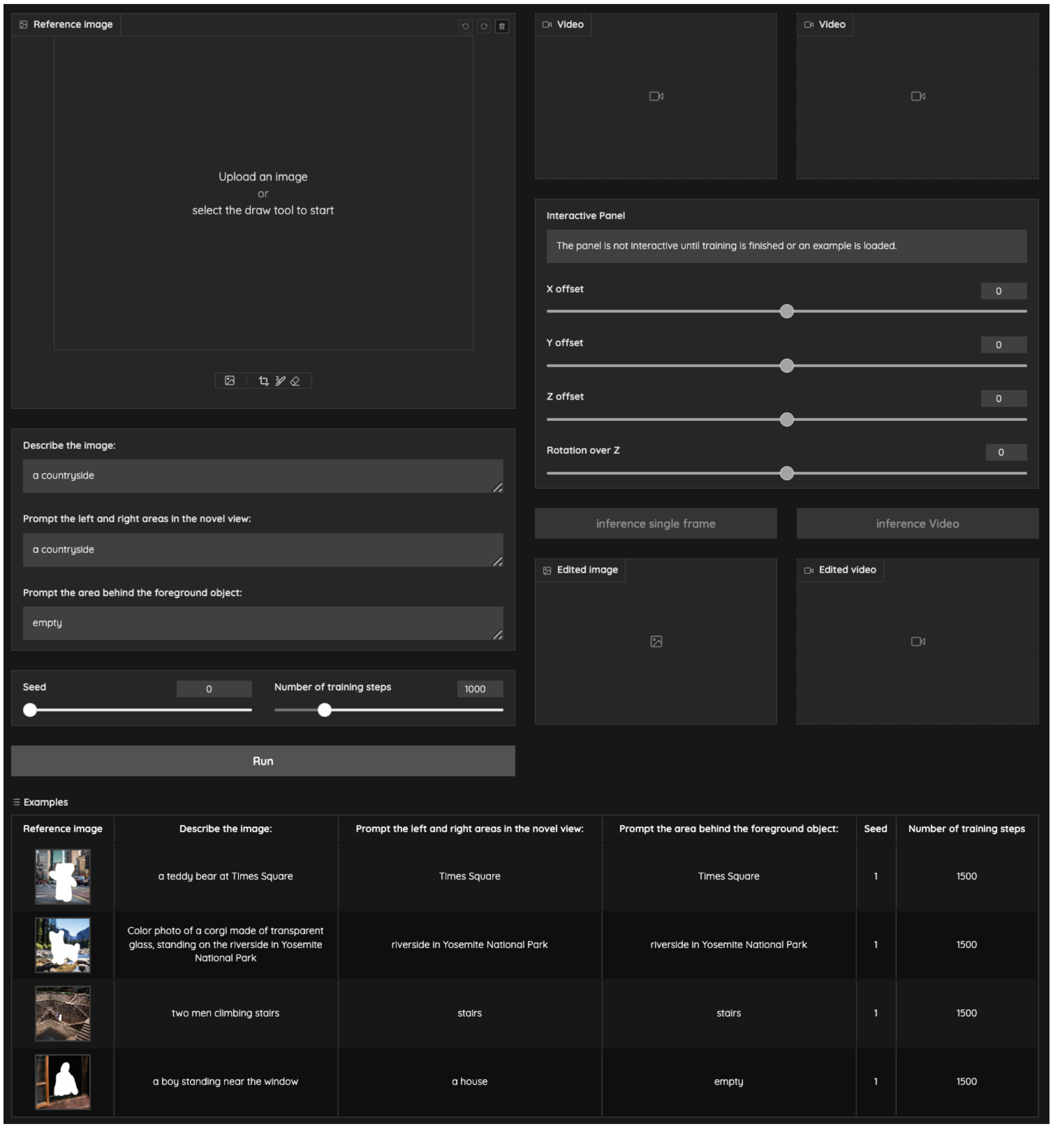
To enhance the user experience, we explore two types of editing frameworks:

1. **Large language model (LLM) based.** We prompt the LLM to take the user's editing request as input and parse it into porangeefined components, including camera movement angles, descriptions of the object of interest, and the transformation matrix for that object. Our editing algorithm then uses these components as input to optimize the 3D scene and carry out the necessary edits.

   However, relying solely on the LLM as the interface for our editing algorithm has its drawbacks; it makes it difficult for users to interactively manipulate and edit the optimized 3D scene. To address this, we have also developed a user interface that facilitates these interactions.

2. **User interface (UI) based.** We developed an user interface for editing. We have implemented a web-based interactive panel that allows users to control the editing process. Fig. A1 provides an overview of the user interface. To use it, users need to upload an image, specify the object of interest, and provide text prompts, as illustrated in Fig. A2. They can then visualize the optimization process in real time. After the optimization is complete, users can use draggable sliders for various edits, including moving, removing, and rotating objects, as shown in Fig. A3, Fig. A4, and Fig. A5.

Both types of editing workflows do not require users to interact with the code during the entire process.



Reference image

Upload an image
or
select the draw tool to start

Describe the image:

a countryside

Prompt the left and right areas in the novel view:

a countryside

Prompt the area behind the foreground object:

empty

Seed: 0

Number of training steps: 1000

Interactive Panel

The panel is not interactive until training is finished or an example is loaded.

X offset: 0

Y offset: 0

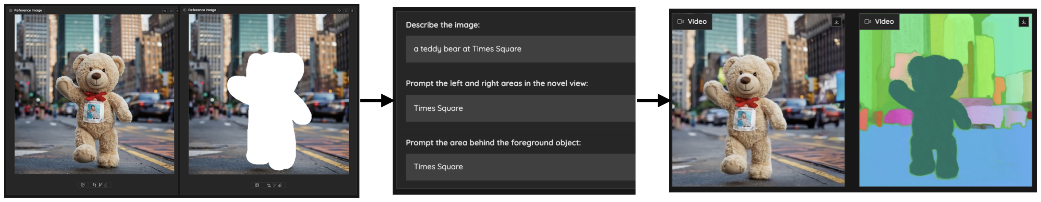
Z offset: 0

Rotation over Z: 0

Examples

Reference image	Describe the image:	Prompt the left and right areas in the novel view:	Prompt the area behind the foreground object:	Seed	Number of training steps
	a teddy bear at Times Square	Times Square	Times Square	1	1500
	Color photo of a corgi made of transparent glass, standing on the riverside in Yosemite National Park	riverside in Yosemite National Park	riverside in Yosemite National Park	1	1500
	two men climbing stairs	stairs	stairs	1	1500
	a boy standing near the window	a house	empty	1	1500

Figure A1: **Overview of the user interface.** To provide better user experience, we have developed a web-based interface. Users simply need to upload the input image and enter a text prompt. They can then visualize the optimization process in real time and edit the scene using sliders.



Describe the image:

a teddy bear at Times Square

Prompt the left and right areas in the novel view:

Times Square

Prompt the area behind the foreground object:

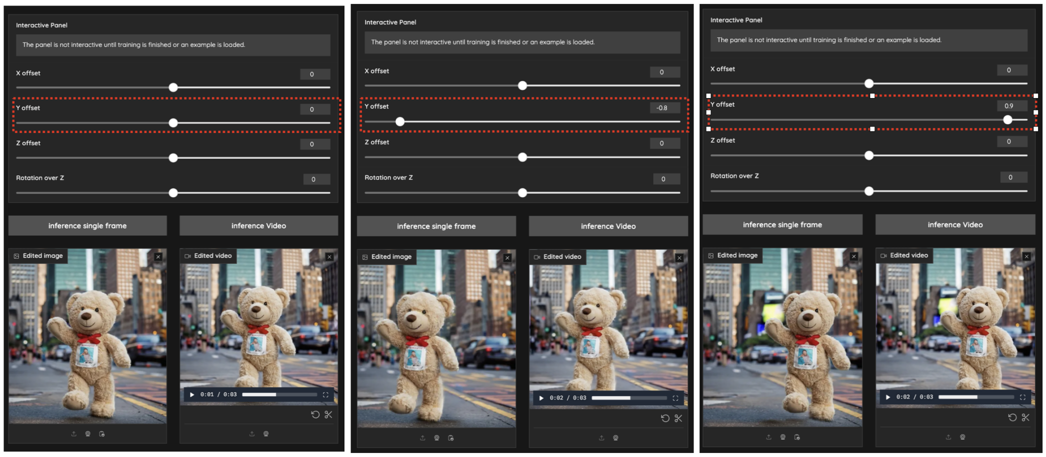
Times Square

Upload image, and specify object of interest

Optional text prompt

Visualize the optimization

Figure A2: **Procedure to optimize a scene via the user interface.** To optimize a 3D scene for 3D-aware image editing, users need to upload an image, specify the object of interest, and provide text prompts. They can then visualize the optimization process in real time.



Interactive Panel

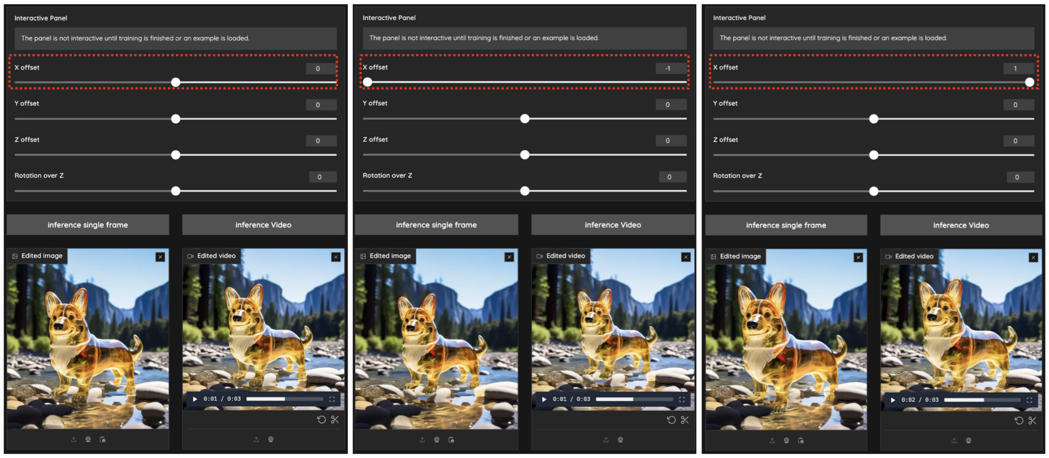
The panel is not interactive until training is finished or an example is loaded.

X offset 0 | Y offset 0 | Z offset 0 | Rotation over Z 0

X offset 0 | Y offset -0.8 | Z offset 0 | Rotation over Z 0

X offset 0 | Y offset 0.9 | Z offset 0 | Rotation over Z 0

Figure A3: **Interactive editing via the user interface.** In this sample, we demonstrate that by sliding the bar, users can adjust the object's offset along the Y coordinate.



Interactive Panel

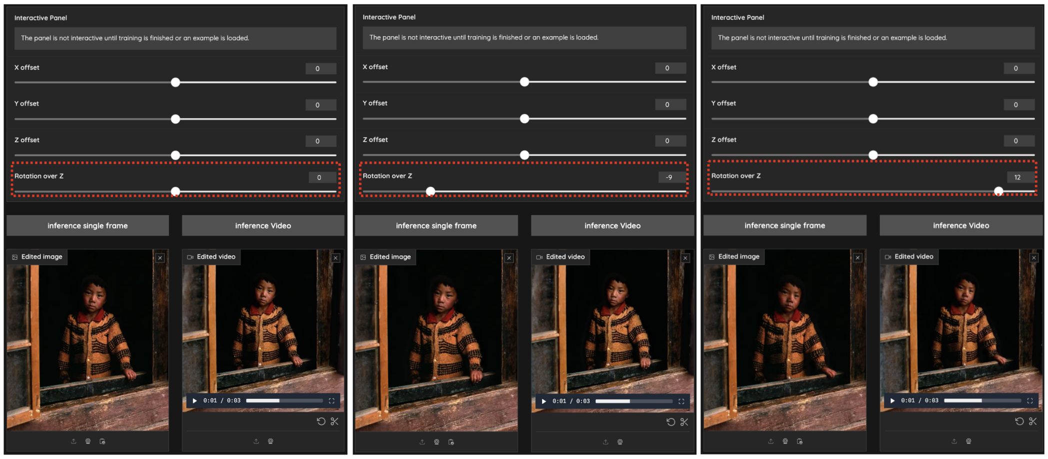
The panel is not interactive until training is finished or an example is loaded.

X offset 0 | Y offset 0 | Z offset 0 | Rotation over Z 0

X offset -1 | Y offset 0 | Z offset 0 | Rotation over Z 0

X offset 1 | Y offset 0 | Z offset 0 | Rotation over Z 0

Figure A4: **Interactive editing via the user interface.** In this sample, we demonstrate that by sliding the bar, users can adjust the object's offset along the X coordinate.



Interactive Panel

The panel is not interactive until training is finished or an example is loaded.

X offset 0 | Y offset 0 | Z offset 0 | Rotation over Z 0

X offset 0 | Y offset 0 | Z offset 0 | Rotation over Z -9

X offset 0 | Y offset 0 | Z offset 0 | Rotation over Z 12

Figure A5: **Interactive editing via the user interface.** In this sample, we demonstrate that by sliding the bar, users can adjust the object's rotation.