GaussCtrl: Multi-View Consistent Text-Driven 3D Gaussian Splatting Editing

Jing Wu*[1], Jia-Wang Bian *[2], Xinghui Li[1], Guangrun Wang[1], Ian Reid[2], Philip Torr[1], and Victor Adrian Prisacariu[1]

[1] University of Oxford
[2] Mohamed bin Zayed University of Artificial Intelligence
{jing.wu, philip.torr}@eng.ox.ac.uk, {jiawang.bian, ian.reid}@mbzuai.ac.ae, {xinghui, victor}@robots.ox.ac.uk, wanggrun@gmail.com

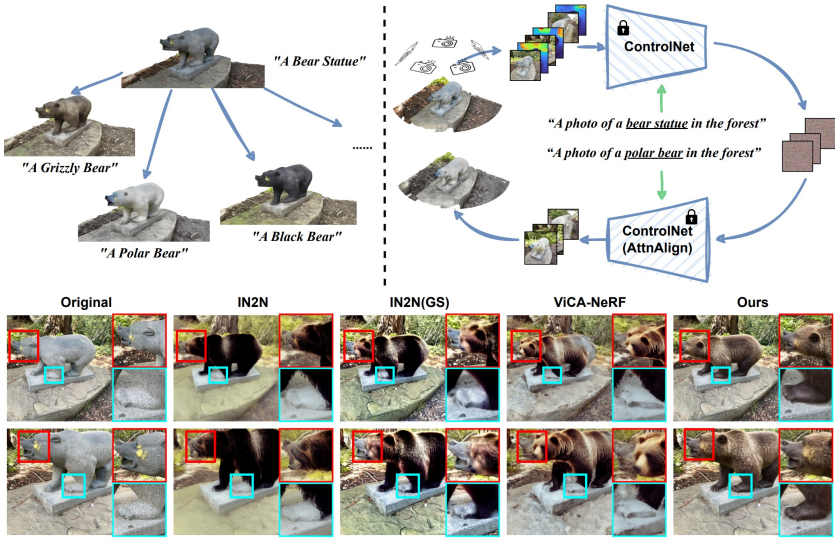

Fig. 1: GaussCtrl. Our method edits a 3D Gaussian Splatting (3DGS) scene by modifying its descriptive prompt (Upper Left). This is achieved by editing the rendered images of 3DGS and re-training the 3D model (Upper Right). Our contribution is a depth-conditioned multi-view consistent editing framework, which substantially improves the blurry or unreasonable 3D results caused by inconsistent editing in previous work (Bottom).

Abstract. We propose GaussCtrl, a text-driven method to edit a 3D scene reconstructed by the 3D Gaussian Splatting (3DGS). Our method first renders a collection of images by using the 3DGS and edits them by using a pre-trained 2D diffusion model (ControlNet) based on the input prompt, which is then used to optimise the 3D model. Our key

* Equal contributions

contribution is multi-view consistent editing, which enables editing all images together instead of iteratively editing one image while updating the 3D model as in previous works. It leads to faster editing as well as higher visual quality. This is achieved by the two terms: (a) **depth-conditioned editing** that enforces geometric consistency across multi-view images by leveraging naturally consistent depth maps. (b) **attention-based latent code alignment** that unifies the appearance of edited images by conditioning their editing to several reference views through self and cross-view attention between images' latent representations. Experiments demonstrate that our method achieves faster editing and better visual results than previous state-of-the-art methods. Project website: `https://gaussctrl.active.vision/`

Keywords: 3D Editing · Diffusion Models · Gaussian Splatting · Neural Radiance Fields

1 Introduction

Neural representations such as Neural Radiance Field (NeRF) [26] and 3D Gaussian Splatting (3DGS) [16] have demonstrated an ability to create a 3D reconstruction that can guarantee remarkably high-quality novel view rendering. However, the ability to edit these resulting 3D representations for the creation of 3D assets remains a crucial yet underexplored aspect. The first pipeline was recently proposed by Instruct NeRF2NeRF [9] (IN2N) to edit a NeRF scene using text-based instructions. It leverages an image-conditioned diffusion model (Instruct Pix2Pix [2]) to iteratively edit the images rendered from the NeRF while updating the underlying NeRF-based reconstruction. Follow-ups [5, 43] follow the same idea of editing 3D from 2D and show encouraging results.

While IN2N transforms the 3D editing problem into a more manageable 2D editing task, consistency across multiple views is not encouraged. Such consistency is critical to the final 3D editing to prevent visual artefacts such as blurring and inconsistent appearance at different viewpoints. However, this is challenging for modern 2D diffusion models to achieve since their input is a single image and they do not enforce geometric consistency across different views. IN2N edits one image at a time while optimising the 3D NeRF, generating an averaged result. This process leads to some geometric consistency, however, the convergence is slow, and it requires many full NeRF optimisations. Although its follow-ups yield better visual quality with their own innovations, inconsistencies across rendered views remain—See Fig. 1 (Bottom).

We propose GaussCtrl to address the challenge of inconsistency. Our method edits 3D scenes with text instructions, as illustrated in Fig. 1 (Upper Left & Right). Distinguishing from IN2N, we propose a depth-controlled editing framework and a latent code alignment module to explicitly enforce the multi-view consistency, enabling editing all images together and updating the 3D model only once. Specifically, **first**, as depth maps of the scene are naturally geometry-consistent, we condition image editing on them by employing ControlNet [49].

Images are inverted to their respective latent codes together with their depth maps via DDIM inversion [41] and then are edited by the denoising process with edited prompts. By doing so, the edited images inherit the consistency from depth maps, avoiding the abrupt and incoherent geometric changes that may present in other editing methods [11,27]. **Second**, we propose an attention-based latent code alignment module to encourage appearance consistency during the editing process. Specifically, we choose several reference views and align all other views' latent codes to these reference views during the denoising process, through the use of both self and cross-view attention. This greatly improves high-level semantic consistency over the entire dataset.

We evaluate our approach on a variety of scenes with different text prompts, ranging from forward-facing scenes to challenging 360-degree object-centred scenes. We also perform an ablation study on different components of our method to validate their effectiveness. Experiments demonstrate that our method significantly improves the visual quality of editing and greatly reduces processing time. We summarise our contribution as follows:

1. We propose GaussCtrl, to enable efficient editing of 3DGS scenes with text instructions.
2. We employ depth guidance and the attention-based latent code alignment module to encourage multi-view consistent editing.
3. The proposed method demonstrates more realistic editing and achieves higher visual quality than previous work on a variety of 3D editing scenes.

2 Related Works

2.1 2D Editing with Diffusion Models

Diffusion models [13,14,25,30,40,41] have gained popularity in image generation due to their ability to produce highly realistic images. By training on billions of image-text pairs, these models not only offer the flexibility to customise generation through textual prompts [4, 12, 35, 36, 38] but also enable various forms of editing. Most editing methods leverage pre-trained Stable Diffusion [36]. Starting with inverting the latent representation of the to-be-edited image to its corresponding noise by DDIM inversion, the editing is achieved through the denoising process. One form of editing [6, 28, 29, 32] allows users to label several anchor points and drag them to target positions. DragDiffusion [39] optimises the latent code of an image by minimising feature differences between initial and target positions. SDE-Drag [31] replaces the optimisation with copy-and-paste of features by switching to DDPM scheduler [13]. Another form is to edit images through textual prompts, which is more relevant to our work. Delta denoising score (DDS) [10] directly optimises the image latent representation by forcing the similarity between noises predicted using the original and edited texts. Prompt-to-Prompt (P2P) [11] achieves editing by manipulating the cross attention between image and text. Null-text inversion [27] tackles artefacts at DDIM inversion when using classifier-free guidance [12] and integrates their method

to P2P. While significant progress has been made in 2D image editing, none of them considers multi-view consistency, which may lead to artefacts in editing. Our method, tailored for 3D editing, ensures such consistency through initial latent code control and newly proposed attention-based latent code alignment.

2.2 3D Editing in NeRF and Gaussian Splatting

NeRF and 3DGS are two of the most popular 3D models for neural novel view synthesis. NeRF implicitly encodes the geometry and colours of a scene in a Multi-Layer Perceptron (MLP), whereas 3DGS explicitly expresses the scene as Gaussian ellipse point clouds. Although they exhibit promising results in 3D reconstruction, their editing remains challenging. Current attempts can be largely categorised into two main categories: 3D Style Transfer, and Text-driven Editing.

3D Style Transfer: Similar to 2D style transfer [8], 3D style transfer aligns the style of a 3D scene to the style of a provided reference image. Notable examples include StyleRF [22], StylizedNeRF [15], ARF [48], and PaletteNeRF [18]. However, this line of work fails to edit the local details of the scene, and the reference image is not always available.

Text-driven Editing: Instruct NeRF2NeRF (IN2N) [9] is the first NeRF editing work that edits 3D models with text instructions. This method effectively transforms the 3D editing challenge into a 2D image editing task. By rendering images from the 3D scene and editing them using Instruct Pix2Pix (IPix2Pix) [2], IN2N iteratively updates the 3D scene until convergence. As there is no guarantee of consistent editing of multi-view images, this method suffers from instability, slow processing speeds, and notable artefacts, particularly evident in 360-degree scenes. ViCA-NeRF [5], following a similar idea to IN2N, selects several reference images from the dataset of the scene, edits them by IPix2Pix, and edits the rest of the dataset as blended results of the projection of reference images to alleviate the inconsistency. However, the blending does not fully address the consistency issue and suffers from blurry editing. DreamEditor [50] converts NeRF to mesh and directly optimises the mesh with SDS loss [34] and DreamBooth [37]. Our method shares similarities with IN2N and ViCA-NeRF but aims to address their limitations and offer superior consistency and visual quality in the editing results.

3 Method

We propose GaussCtrl, a novel approach to edit a 3D Gaussian Splatting (3DGS) model using textual prompts. Given a collection of images and their reconstructed 3D model, our method first re-renders each dataset image to the required resolution and renders their respective depth maps. Then, we employ ControlNet [49] to conduct depth-conditioned editing for all images supplemented by attention-based latent code alignment to encourage geometry and appearance consistency. Finally, we optimise the original 3D model using the edited images to obtain the new edited 3D model. Optionally, a mask generated

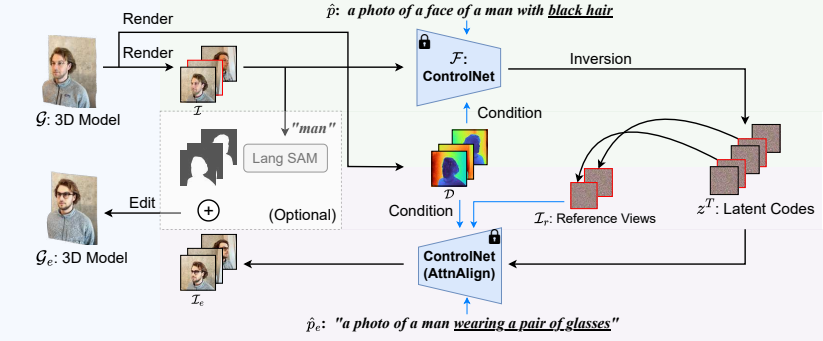

Fig. 2: GaussCtrl pipeline. Given a 3DGS scene and text instructions, our method renders images using the 3DGS and edits the rendered images with text instructions, which are then used to optimise the original 3DGS. Our key contribution is multi-view consistent editing. Towards this, we propose (1) depth-conditioned editing based on ControlNet for geometry consistency; and (2) attention-based latent code alignment for improving consistency during editing.

by Language-based Segment Anything (Lang SAM) [17] is applied to filter the background for better quality when editing local objects. The comprehensive pipeline is illustrated in Fig. 2. In the following, we commence by reviewing the background of 3DGS and ControlNet in Sec. 3.1, followed by the introduction of our proposed methods, including depth-conditioned image editing in Sec. 3.2, and attention-based latent code alignment in Sec. 3.3.

3.1 Background

3D Gaussian Splatting: Gaussian Splatting [16] is an explicit 3D representation based on point clouds. A set of 3D Gaussians is modelled to represent the scene. Each Gaussian ellipse has a colour and an opacity and is defined by its centred position x (mean), and a full covariance matrix Σ: $G(x) = e^{-\frac{1}{2}x^T\Sigma^{-1}x}$. When projecting 3D Gaussians to 2D for rendering, a method of splatting [51] is used to position the Gaussians on 2D planes, which involves a new covariance matrix Σ' in camera coordinates defined as $\Sigma' = JW\Sigma W^T J^T$, where W denotes a given viewing transformation matrix and J is the Jacobian of the affine approximation of the projective transformation. To enable differentiable optimisation, Σ is further decomposed into s scaling matrix S and a rotation matrix R: $\Sigma = RSS^T R^T$.

ControlNet: ControlNet [49] is an end-to-end spatial conditional generation model built on top of Stable Diffusion (SD) [36]. It implants additional U-Net encoders to SD, which enables image generation controlled by various kinds of extra information, *e.g.*, depth, normals, edges, or hand-drawn priors. We employ ControlNet for its ability of depth-controlled generation.

3.2 Depth-conditioned Image Editing

Previous work [9] employs Instruct-Pix2Pix [2] for image editing, resulting in visually appealing individual images. However, ensuring multi-view consistency between these images remains a challenge, often resulting in visual artefacts and unstable editing outcomes. To this end, we conduct depth-conditioned image editing by employing ControlNet $\mathcal{F}$, comprising a U-Net block $\mathcal{F}_U$ and a ControlNet block $\mathcal{F}_C$. As the depths $\mathcal{D}$ are extracted from the 3D model, they are naturally consistent across multiple views. By conditioning image editing on these consistent depth maps, our method effectively promotes consistency in 3D geometry across all edited images.

Given a to-be-edited image $\mathcal{I}$ and its corresponding description prompt $\hat{p}$, we begin by computing its latent code z^0, using the VAE encoder of the ControlNet. We then iteratively invert it to its corresponding Gaussian noise z^T via DDIM inversion. Mathematically, the inversion can be described as follows:

$$\epsilon^t = \mathcal{F}_U(z^t, t, \hat{p}, \mathcal{F}_C(z^t, t, \hat{p}, \mathcal{D})) \tag{1}$$

$$z^{t+1} = \sqrt{\alpha_{t+1}} \frac{z^t - \sqrt{1-\alpha_t} \cdot \epsilon^t}{\sqrt{\alpha_t}} + \sqrt{1-\alpha_{t+1}} \epsilon^t \tag{2}$$

where t is the time step of the diffusion process and α_t is the scheduling coefficient in DDIM scheduler. After reaching z^T, we replace the original prompt $\hat{p}$ with the edited prompt $\hat{p}_e$ containing changed content, and obtain the edited latent code z_e^0 through the denoising process:

$$\epsilon_p^t = \mathcal{F}_U(z_e^t, t, \hat{p}_e, \mathcal{F}_C(z_e^t, t, \hat{p}_e, \mathcal{D})) \tag{3}$$

$$\epsilon_\emptyset^t = \mathcal{F}_U(z_e^t, t, \emptyset, \mathcal{F}_C(z_e^t, t, \emptyset, \mathcal{D})) \tag{4}$$

$$\epsilon^t = \epsilon_\emptyset^t + \omega \cdot (\epsilon_p^t - \epsilon_\emptyset^t) \tag{5}$$

$$z_e^{t-1} = \sqrt{\alpha_{t-1}} \frac{z_e^t - \sqrt{1-\alpha_t} \cdot \epsilon^t}{\sqrt{\alpha_t}} + \sqrt{1-\alpha_{t-1}} \epsilon^t \tag{6}$$

where z_e^t denotes the latent code of the edited images ($z_e^T = z^T$), $\emptyset$ is an empty prompt and Eq. (5) is the classifier-free guidance [12] to improve the fidelity of the editing to the edited prompt $\hat{p}_e$. We obtain the final image $\mathcal{I}_e$ by decoding z_e^0 using the VAE decoder of the ControlNet.

Discussion on DDIM inversion: The original ControlNet operates as a generative model, typically accepting randomly initialised noise z^T as input, thus yielding diverse results. However, for editing tasks, we adopt a different approach by reversing the to-be-edited images into noise and utilising them as input for ControlNet. By doing so, the output is conditioned on the original image, and more importantly, multi-view consistency is improved during editing. This is because the original images have naturally consistent colour and geometry, where we use DDIM inversion to obtain consistent initial noises for all to-be-edited images towards consistent editing.

3.3 Attention-based Latent Code Alignment

While our depth-conditioned editing approach enhances geometric consistency, individual images are still edited independently, posing challenges for appearance consistency. Despite sharing the same editing prompt, edited images may exhibit discrepancies in colours or produce peculiar results, particularly at challenging viewpoints. Previous studies [3, 11] have identified a relationship between the appearance of images generated by diffusion models and the key-value pairs in the image self-attention mechanism of the U-Net. Inspired by this fact, we propose an attention-based latent code alignment module that explicitly aligns the appearance of images to selected reference views during editing. Therefore, images are no longer edited independently; instead, their appearances are unified to a common standard. This ensures greater consistency across edited images and mitigates issues related to appearance discrepancies.

Specifically, we first define the attention between two latent codes z_i and z_j as:

$$\text{Attn}_{i,j} = \text{Softmax}(\frac{W_q(z_i)W_k(z_j)^\top}{\sqrt{c}})W_v(z_j), \tag{7}$$

where $W_q(\cdot)$, $W_k(\cdot)$, and $W_v(\cdot)$ are linear projections to obtain query, key, and value for the attention operation, and c is a scaling factor. Given latent codes of N_r reference images $z^t_{r,i}$, where $i = 1, 2, ..., N_r$ and the latent code of the to-be-edited image z^t_e at the time step t, our alignment module is defined as:

$$\text{AttnAlign}_e = \lambda \cdot \text{Attn}_{e,e} + (1-\lambda) \cdot \frac{1}{N_r}\sum_{i=1}^{N_r} \text{Attn}_{e,i} \tag{8}$$

where $\lambda \in [0, 1]$. This module blends the self-attention of z^t_e with the cross-view attention between z^t_e and each reference view $z^t_{r,i}$. The cross-view attention aligns the appearance of all edited images to the reference views while the self-attention helps each edited image retain its distinctiveness. We find that this design significantly improves the appearance consistency and minimises anomalies at challenging viewpoints. We replace all image self-attention modules with the proposed alignment module in the U-Net block $\mathcal{F}_U$ and the ControlNet block $\mathcal{F}_C$. Therefore, Eq. (3) and Eq. (4) becomes:

$$\epsilon^t_p = \mathcal{F}_U(z^t_e, z^t_r, t, \hat{p}_e, \mathcal{F}_C(z^t_e, z^t_r, t, \hat{p}_e, \mathcal{D})) \tag{9}$$

$$\epsilon^t_\emptyset = \mathcal{F}_U(z^t_e, z^t_r, t, \emptyset, \mathcal{F}_C(z^t_e, z^t_r, t, \emptyset, \mathcal{D})) \tag{10}$$

4 Experiments

In this section, we first introduce the setup of experiments, including the testing data, method baselines, evaluation metrics, and implementation details. We then provide both qualitative and quantitative results of our method, followed by ablation studies on each proposed component and a limitation discussion.

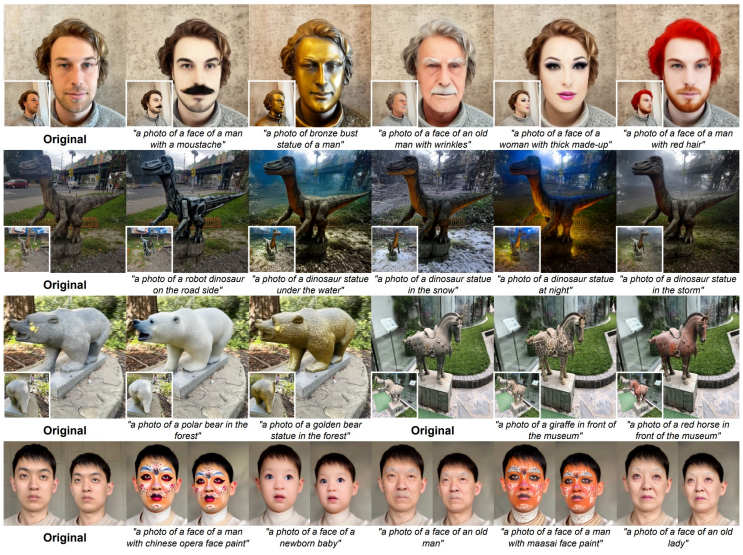

Fig. 3: Qualitative results. We show diverse results of text-guided editing in various scenes, ranging from editing objects to adjusting environments, *e.g.*, changing the appearance and age of the target human, and modifying the environment.

4.1 Experiment Setup

Dataset: To validate the effectiveness of GaussCtrl, we collect a variety of scenes from multiple existing datasets for evaluation. Specifically, we collected four 360-degree scenes from IN2N [9], Mip-NeRF360 [1], and BlendedMVS [45] datasets, and two forward-facing scenes from IN2N [9] and NeRF-Art [44] dataset. For each scene, we evaluate our method on multiple text-based instructions.

Baselines: We mainly compare GaussCtrl with two state-of-the-art methods: Instruct-GS2GS [43], a very recent update of Instruct NeRF2NeRF (IN2N) [9] that replaces the NeRF in IN2N with a 3DGS model and ViCA-NeRF [5] for the similarities shared between them. We denote Instruct-GS2GS as IN2N(GS) in the following paragraphs for it being a variant of IN2N. Both IN2N(GS) and VICA-NeRF employ Instruct-Pix2Pix, which takes an instruction-like prompt to edit images. Our method is based on Stable Diffusion, using a description-like prompt. Therefore, to edit a scene, we use editing instructions in IN2N(GS) and VICA-NeRF while modifying the original scene description in our method. To ensure a fair comparison, we preprocess all dataset images to 512×512 resolution and evaluate all methods on the same data. We provide more visual comparison results, including a comparison with IN2N in our supplementary.

Evaluation Criteria: Following previous methods [2, 7, 9], we use CLIP Text-Image Directional Similarity (CLIP_{dir}) to evaluate the alignment of the 3D edit with text instructions. However, we notice that this metric may not always reflect the true visual quality of editing, on which we will elaborate in later paragraphs.

Fig. 4: Qualitative comparison on 360-degree scenes. Our method generates more consistent and higher-quality images than previous state-of-the-art methods.

Implementation Details: Our method is implemented by using the PyTorch library. We use the "splatfacto" model, an improved implementation of 3D Gaussian Splatting, in NeRFStudio [42] library for 3D reconstruction. We employ Stable Diffusion v1.5 and its corresponding ControlNet for 2D image editing using the Huggingface library [33]. For reference views, we sample $N_r = 4$ views from the dataset images randomly. We set λ in Eqn. 8 as 0.6. We apply Language-based Segment Anything (Lang SAM) [17,20,23,24,46] as our mask segmentation backbone. Our method takes around 9 minutes to edit one scene on an NVIDIA RTX A5000 with a GPU memory of 24GB.

4.2 Qualitative Evaluation

Fig. 3 illustrates various editing results of our method, including in both 360-degree and forward-facing scenes. The text instructions range from editing objects to adjusting environments. Fig. 4 and Fig. 5 show the qualitative comparison of our method with previous SOTA alternatives in 360-degree and forward-facing scenes, respectively. In Fig. 7, we sample 10 different views of a scene to

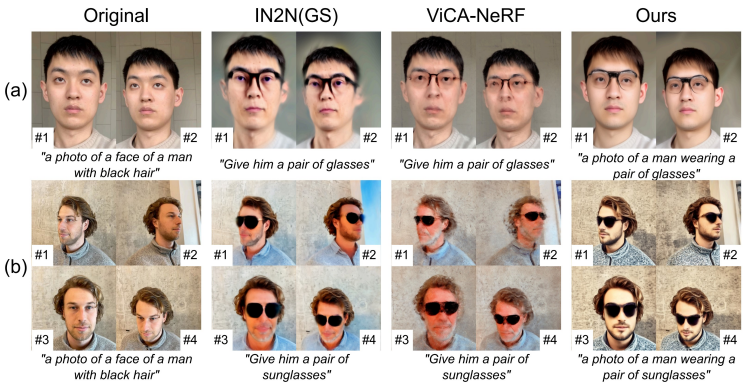

Fig. 5: Qualitative results on forward-facing scenes. Our method generates more realistic results with better quality, consistency, and less artefact.

visually compare the editing consistency of different methods. We make more detailed analyses below.

Fig. 3 shows that our method can perform realistic edits globally (*"a photo of a dinosaur in the snow"*) and locally (*"a photo of a polar bear in the forest"*) with high quality and consistency. Also, it can change the colour of a specific area (*"a photo of the face of a man with red hair"*) and texture of the object (*"a photo of a bronze bust statue of a man"*). Our method also shows good consistency for complicated texture editing such as *"a photo of a face of a man with Chinese opera face paint"*.

Fig. 4 shows qualitative comparisons between our method and baselines in 360-degree scenes. Specifically, when editing objects, IN2N(GS) suffers from incomplete editing shown by the face area indicated by view #2,4, the feet area in view #1,2,4 of scene (a) and the neck area in view # 1, 3 of scene (b). These show the inconsistent editing from using Instruct Pix2Pix when editing local objects. When editing scene's environments, IN2N(GS) demonstrates relatively better quality than editing objects as shown by scenes (c) and (d) VICA-NeRF suffers from blurry results in both local object and scene environment editing. We attribute these artefacts to their blending of inconsistently edited reference images. Thanks to the depth-conditioned editing and proposed latent code alignment module, our method demonstrates sharper, more consistent and realistic results in both object and environment editing, indicating the superiority of our method over previous alternatives.

Fig. 5 shows qualitative comparisons in forward-facing scenes. Compared with 360-degree scenes, IN2N(GS) and ViCA-NeRF show better consistency in the forward-facing setting. This is attributed to that the variation of image viewpoints in the forward-facing setting is not as extreme as in the 360-degree setting, meaning that individual image editing may retain certain consistency. However, compared with our method, IN2N(GS) and ViCA-NeRF still suffer from artefacts such as blurry boundaries. Our method also generates more realistic results.

Table 1: Quantitative Evaluation. CLIP_{dir}: CLIP Text-Image Direction Similarity

	Scene	IN2N		IN2N(GS)		ViCA-NeRF		**Ours**	
		CLIP_{dir}	Time	CLIP_{dir}	Time	CLIP_{dir}	Time	CLIP_{dir}	Time
360	Bear Statue	0.1019	∼1.5h	0.1165	∼13.5min	0.1104	∼38.5min	**0.1388**	∼9min
	Dinosaur	0.1466		0.1490		0.0723		**0.1584**	
	Garden	**0.3027**		0.1663		0.2903		0.2891	
	Stone Horse	0.1654		0.1947		0.1926		**0.2268**	
Forward	Fangzhou	0.1598		**0.2032**		0.1809		0.1887	
	Face	0.1332		0.1357		0.1119		**0.1503**	

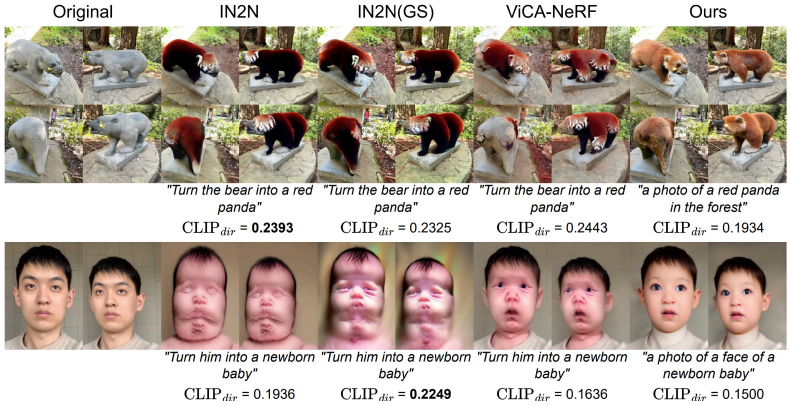

Fig. 6: Failure cases of CLIP_{dir}. CLIP_{dir} reflects the alignment between text instructions and editing results but ignores the editing quality. In the top row, our red panda has a lighter colour, making it score lower. Other methods have wrong face geometry but score higher than ours. In the bottom row, previous methods are closer to *newborn baby*, making them score higher. However, the first two methods have terrible results, and ViCA's result is unnatural in the baby's eye areas.

To highlight the improvement in multi-view consistency, we render 10 views from different angles around the object for each method in Fig. 7. As shown in view #1,2,4,8,10, both IN2N(GS) and ViCA-NeRF lose many details and are blurry on side views of the polar bear, a direct result of inconsistent editing. Second, ViCA-NeRF loses most of the details of the bear's face as indicated in view #1,2,3,8,9,10. What's more, it can be observed in view #6 that IN2N(GS) and ViCA both suffer from the face-on-the-back problem, which is caused by Instruct Pix2Pix forcefully producing a polar bear and fitting it to the image layout. Our approach greatly mitigates this problem through our latent code alignment module by conditioning editing on reference views. More detailed comparisons of editing consistency and quality are included in the supplementary material.

4.3 Quantitative Evaluation

We calculate the average CLIP_{dir} over text instructions for each scene and summarize the results in Tab. 1. Our method outperforms other approaches in four out of six scenes. However, we notice that CLIP_{dir} may not always reflect the

Fig. 7: Editing consistency comparison on the bear scene (Polar bear). IN2N(GS) and ViCA both suffer from editing inconsistency (View #1,2,4,8,10), which results in artefacts and blurry of the bear's face. Additionally, they are affected by the face-on-the-back problem. Our method improves on both problems.

editing quality, as it measures more about the global similarity between the text prompt and the edited images, ignoring the majority of local details. We illustrate two failure cases of this metric in Fig. 6. Our method generates better visual results but gets lower scores than previous methods. Therefore, we include as many examples as we can in the paper and the supplementary material to reflect the true visual quality of our method. We additionally provide the editing time for each method, and our method is the fastest among them.

4.4 Ablation Study

We conduct ablation studies on our method to demonstrate the effectiveness of each proposed component. We selected the scene "bear statue", a 360-degree scene, as our subject because the 360-degree scene can better illustrate the effect of multi-view consistency. To ensure a fair comparison, the guidance scale λ is set to 7.5 in all cases. Lang-SAM [19,21,47] is not applied either to highlight the effect of each component on the scene's environment. The result is provided in Fig. 8, where we show the original images in (a).

One-time Instruct Pix2Pix Edit (b): The most significant limitation of Instruct Pix2Pix (IPix2Pix) is that it fails to edit images in challenging viewpoints. For example, IPix2Pix fails to produce noticeable effects in challenging views #4, 6, 7, and 8, where the bear statue is viewed from behind, and only partially alters views #2, 3, and 5. This limitation prompts IN2N to choose iterative editing over one-time editing for the unstable performance of IPix2Pix. Even in relatively simpler views #9,10, IPix2Pix still exhibits artefacts around the face of the bear, limiting the performance of IN2N and VICA-NeRF.

ControlNet with Random Noise (c): When ControlNet operates with random noise instead of inverted latent codes, it performs a generation task instead of an editing task. While the generated images maintain consistency in geometry, owing to the incorporation of depth maps, their overall style diverges significantly from that of the original images. Moreover, at challenging viewpoints such

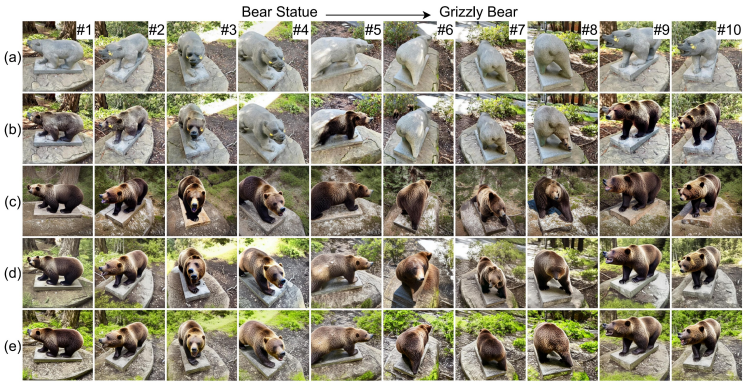

Fig. 8: Ablation studies on the consistent editing. (a): Sampled images from the original dataset. (b): Editing results using Instruct Pix2Pix [2]. (c): Our proposed depth-conditioned editing, which uses ControlNet with the randomly initialised latent codes. (d): Consistent initial latent code is applied by using DDIM inversion. (e): Attention-based latent code alignment is added based on (d).

as views #6,7,8, ControlNet forcefully generates front-facing views of a bear with the geometry of the bottom of the bear, which damages the eventual quality of 3D editing. Additionally, similar to IPix2Pix, it also suffers from artefacts when editing local details, such as the face area in view #10.

ControlNet with Inverted Latent Codes (w/o AttnAlign) (d): When employing latent codes inverted from the original images, we observe a significant improvement in the general style alignment compared to using random noise. Additionally, texture and colour consistency are notably enhanced. We attribute these enhancements to consistent initial latent codes brought by latent inversion. However, artefacts experienced by random noise still exist, such as the forceful front-facing views of the bear in view #6,7,8 and artefacts around the facial area in view #10.

ControlNet with Inverted Latent Codes & AttnAlign (e): After applying our proposed latent code alignment module, the artefacts presented in (c) and (d) are notably mitigated. By conditioning the editing on reference views using cross-view attention, the model searches for extra information about the to-be-edited image from the reference views. Therefore, it has a better understanding of the semantics and geometry of the image, avoiding producing forceful results. Such conditioning also unifies the appearances of edited images to a common ground, which further improves the quality of the final 3D editing.

4.5 Limitations

Some may be concerned about its ability to alter the scene's original geometry as we condition the editing on depth maps. However, We find that in the most of 3D editing literature [5, 9, 50], significant changes to the original geometry are not typically required. Instead, editing tasks often involve adjusting object

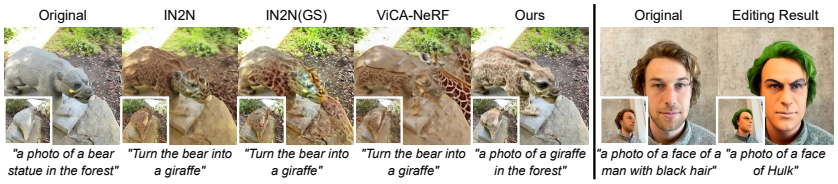

Fig. 9: Failure cases. Left: Due to using depth guidance, our method cannot work well when a significant geometry change is required. However, we find that existing methods also cannot work well in this scenario even though they do not use depths. Right: our method fails when the 2D pre-trained diffusion model doesn't work well. Nevertheless, it shows that our method can still preserve the consistency.

styles, modifying background environments, or adding localized features, such as adding a moustache to a person. For the completeness of the experiments, we include an example that requires geometric changes. As illustrated on the left of Fig. 9, we fail to turn the bear statue into a giraffe. However, the same failure is also observed in IPix2Pix and baseline methods like IN2N and VICA-NeRF. Another limitation is that the final result is not always faithful to the user's intention. As demonstrated in the right of Fig. 9, our method fails to transform the man into the comic character Hulk. We suspect the root of this problem lies in the ControlNet, which does not recognise the word "Hulk", and does not produce the correct result. However, the consistency and sharp results demonstrate the effectiveness of our method.

5 Conclusion

In this paper, we propose an efficient 3D-aware consistency control editing method, GaussCtrl, which greatly mitigates the artefacts and blurry results caused by the inconsistency in 2D editing, especially in 360-degree scenes. Based on a pre-captured Gaussian model, our method controls multi-view consistency by encouraging a consistency in all the stages of editing, *i.e.*, Depth-conditioned Image Editing, and Attention-based Latent Code Alignment. We evaluate the performance of GaussCtrl on diverse scenes, text prompts, and objects. Our method outperforms other state-of-the-art methods through our experiments.

Broader Impact: Our method is one of the 3D editing methods that can be potentially misused for creating deceptive or harmful content, which could erode trust in digital media and exacerbate issues of misinformation and cyberbullying. By generating hyper-realistic alterations to images, videos, or even deepfakes, 3D editing technologies can be utilised to fabricate events, impersonate individuals, or manipulate scenes in ways nearly indistinguishable from reality. This capability not only leads to higher chances of confusion and misinformation but also opens pathways for harassment and defamation. Hence, it is necessary to enhance regulatory frameworks to mitigate these societal risks.

References

1. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. CVPR (2022)
2. Brooks, T., Holynski, A., Efros, A.A.: Instructpix2pix: Learning to follow image editing instructions. In: CVPR (2023)
3. Cao, M., Wang, X., Qi, Z., Shan, Y., Qie, X., Zheng, Y.: Masactrl: Tuning-free mutual self-attention control for consistent image synthesis and editing. arXiv preprint arXiv:2304.08465 (2023)
4. Dhariwal, P., Nichol, A.: Diffusion models beat gans on image synthesis. ArXiv **abs/2105.05233** (2021), https://api.semanticscholar.org/CorpusID:234357997
5. Dong, J., Wang, Y.X.: Vica-nerf: View-consistency-aware 3d editing of neural radiance fields. In: Thirty-seventh Conference on Neural Information Processing Systems (2023)
6. Epstein, D., Jabri, A., Poole, B., Efros, A.A., Holynski, A.: Diffusion self-guidance for controllable image generation (2023)
7. Gal, R., Patashnik, O., Maron, H., Chechik, G., Cohen-Or, D.: Stylegan-nada: Clip-guided domain adaptation of image generators (2021)
8. Gatys, L.A., Ecker, A.S., Bethge, M.: Image style transfer using convolutional neural networks. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2414–2423 (2016). https://doi.org/10.1109/CVPR.2016.265
9. Haque, A., Tancik, M., Efros, A., Holynski, A., Kanazawa, A.: Instruct-nerf2nerf: Editing 3d scenes with instructions. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2023)
10. Hertz, A., Aberman, K., Cohen-Or, D.: Delta denoising score (2023)
11. Hertz, A., Mokady, R., Tenenbaum, J., Aberman, K., Pritch, Y., Cohen-Or, D.: Prompt-to-prompt image editing with cross attention control (2022)
12. Ho, J.: Classifier-free diffusion guidance. ArXiv **abs/2207.12598** (2022)
13. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. arXiv preprint arxiv:2006.11239 (2020)
14. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: LoRA: Low-rank adaptation of large language models. In: International Conference on Learning Representations (2022), https://openreview.net/forum?id=nZeVKeeFYf9
15. Huang, Y.H., He, Y., Yuan, Y.J., Lai, Y.K., Gao, L.: Stylizednerf: Consistent 3d scene stylization as stylized nerf via 2d-3d mutual learning. In: Computer Vision and Pattern Recognition (CVPR) (2022)
16. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics **42**(4) (July 2023), https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/
17. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., Dollár, P., Girshick, R.: Segment anything. arXiv:2304.02643 (2023)
18. Li, B., Weinberger, K.Q., Belongie, S., Koltun, V., Ranftl, R.: Language-driven semantic segmentation. In: International Conference on Learning Representations (2022), https://openreview.net/forum?id=RriDjddCLN
19. Li*, C., Liu*, H., Li, L.H., Zhang, P., Aneja, J., Yang, J., Jin, P., Lee, Y.J., Hu, H., Liu, Z., et al.: Elevater: A benchmark and toolkit for evaluating language-augmented visual models. arXiv preprint arXiv:2204.08790 (2022)

20. Li, F., Zhang, H., Liu, S., Guo, J., Ni, L.M., Zhang, L.: Dn-detr: Accelerate detr training by introducing query denoising. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13619–13627 (2022)
21. Li*, L.H., Zhang*, P., Zhang*, H., Yang, J., Li, C., Zhong, Y., Wang, L., Yuan, L., Zhang, L., Hwang, J.N., Chang, K.W., Gao, J.: Grounded language-image pre-training. In: CVPR (2022)
22. Liu, K., Zhan, F., Chen, Y., Zhang, J., Yu, Y., Saddik, A.E., Lu, S., Xing, E.: Stylerf: Zero-shot 3d style transfer of neural radiance fields (2023)
23. Liu, S., Li, F., Zhang, H., Yang, X., Qi, X., Su, H., Zhu, J., Zhang, L.: DAB-DETR: Dynamic anchor boxes are better queries for DETR. In: International Conference on Learning Representations (2022), https://openreview.net/forum?id=oMI9PjOb9Jl
24. Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Li, C., Yang, J., Su, H., Zhu, J., et al.: Grounding dino: Marrying dino with grounded pre-training for open-set object detection. arXiv preprint arXiv:2303.05499 (2023)
25. Luo, C.: Understanding diffusion models: A unified perspective. ArXiv **abs/2208.11970** (2022), https://api.semanticscholar.org/CorpusID:251799923
26. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis (2020)
27. Mokady, R., Hertz, A., Aberman, K., Pritch, Y., Cohen-Or, D.: Null-text inversion for editing real images using guided diffusion models. arXiv preprint arXiv:2211.09794 (2022)
28. Mou, C., Wang, X., Song, J., Shan, Y., Zhang, J.: Diffeditor: Boosting accuracy and flexibility on diffusion-based image editing. arXiv preprint arXiv:2402.02583 (2023)
29. Mou, C., Wang, X., Song, J., Shan, Y., Zhang, J.: Dragondiffusion: Enabling drag-style manipulation on diffusion models. arXiv preprint arXiv:2307.02421 (2023)
30. Nichol, A., Dhariwal, P.: Improved denoising diffusion probabilistic models. ArXiv **abs/2102.09672** (2021), https://api.semanticscholar.org/CorpusID:231979499
31. Nie, S., Guo, H.A., Lu, C., Zhou, Y., Zheng, C., Li, C.: The blessing of randomness: Sde beats ode in general diffusion-based image editing. arXiv preprint arXiv:2311.01410 (2023)
32. Pan, X., Tewari, A., Leimkühler, T., Liu, L., Meka, A., Theobalt, C.: Drag your gan: Interactive point-based manipulation on the generative image manifold. In: ACM SIGGRAPH 2023 Conference Proceedings (2023)
33. von Platen, P., Patil, S., Lozhkov, A., Cuenca, P., Lambert, N., Rasul, K., Davaadorj, M., Wolf, T.: Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers (2022)
34. Poole, B., Jain, A., Barron, J.T., Mildenhall, B.: Dreamfusion: Text-to-3d using 2d diffusion. arXiv preprint arXiv:2209.14988 (2022)
35. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with clip latents. ArXiv **abs/2204.06125** (2022), https://api.semanticscholar.org/CorpusID:248097655
36. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10684–10695 (2022)
37. Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., Aberman, K.: Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation.

In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2023)
38. Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E.L., Ghasemipour, S.K.S., Ayan, B.K., Mahdavi, S.S., Lopes, R.G., Salimans, T., Ho, J., Fleet, D.J., Norouzi, M.: Photorealistic text-to-image diffusion models with deep language understanding. ArXiv **abs/2205.11487** (2022), https://api.semanticscholar.org/CorpusID:248986576
39. Shi, Y., Xue, C., Pan, J., Zhang, W., Tan, V.Y., Bai, S.: Dragdiffusion: Harnessing diffusion models for interactive point-based image editing. arXiv preprint arXiv:2306.14435 (2023)
40. Sohl-Dickstein, J., Weiss, E.A., Maheswaranathan, N., Ganguli, S.: Deep unsupervised learning using nonequilibrium thermodynamics. p. 2256–2265. ICML'15, JMLR.org (2015)
41. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. arXiv:2010.02502 (October 2020), https://arxiv.org/abs/2010.02502
42. Tancik, M., Weber, E., Ng, E., Li, R., Yi, B., Kerr, J., Wang, T., Kristoffersen, A., Austin, J., Salahi, K., Ahuja, A., McAllister, D., Kanazawa, A.: Nerfstudio: A modular framework for neural radiance field development. In: ACM SIGGRAPH 2023 Conference Proceedings. SIGGRAPH '23 (2023)
43. Vachha, C., Haque, A.: Instruct-gs2gs: Editing 3d gaussian splats with instructions (2024), https://instruct-gs2gs.github.io/
44. Wang, C., Jiang, R., Chai, M., He, M., Chen, D., Liao, J.: Nerf-art: Text-driven neural radiance fields stylization. arXiv preprint arXiv:2212.08070 (2022)
45. Yao, Y., Luo, Z., Li, S., Zhang, J., Ren, Y., Zhou, L., Fang, T., Quan, L.: Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. Computer Vision and Pattern Recognition (CVPR) (2020)
46. Zhang, H., Li, F., Liu, S., Zhang, L., Su, H., Zhu, J., Ni, L.M., Shum, H.Y.: Dino: Detr with improved denoising anchor boxes for end-to-end object detection (2022)
47. Zhang, H., Zhang, P., Hu, X., Chen, Y.C., Li, L.H., Dai, X., Wang, L., Yuan, L., Hwang, J.N., Gao, J.: Glipv2: Unifying localization and vision-language understanding. arXiv preprint arXiv:2206.05836 (2022)
48. Zhang, K., Kolkin, N., Bi, S., Luan, F., Xu, Z., Shechtman, E., Snavely, N.: Arf: Artistic radiance fields (2022)
49. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models (2023)
50. Zhuang, J., Wang, C., Liu, L., Lin, L., Li, G.: Dreameditor: Text-driven 3d scene editing with neural fields. arXiv preprint arXiv:2306.13455 (2023)
51. Zwicker, M., Pfister, H., van Baar, J., Gross, M.: Ewa volume splatting. In: Proceedings Visualization, 2001. VIS '01. pp. 29–538 (2001). https://doi.org/10.1109/VISUAL.2001.964490