C^{3}Editor: Achieving Controllable Consistency in 2D Model for 3D Editing

Zeng Tao[1*] Zheng Ding[2] Zeyuan Chen[2] Xiang Zhang[2] Leizhi Li[2] Zhuowen Tu[2]
[1]Fudan University [2]UC San Diego

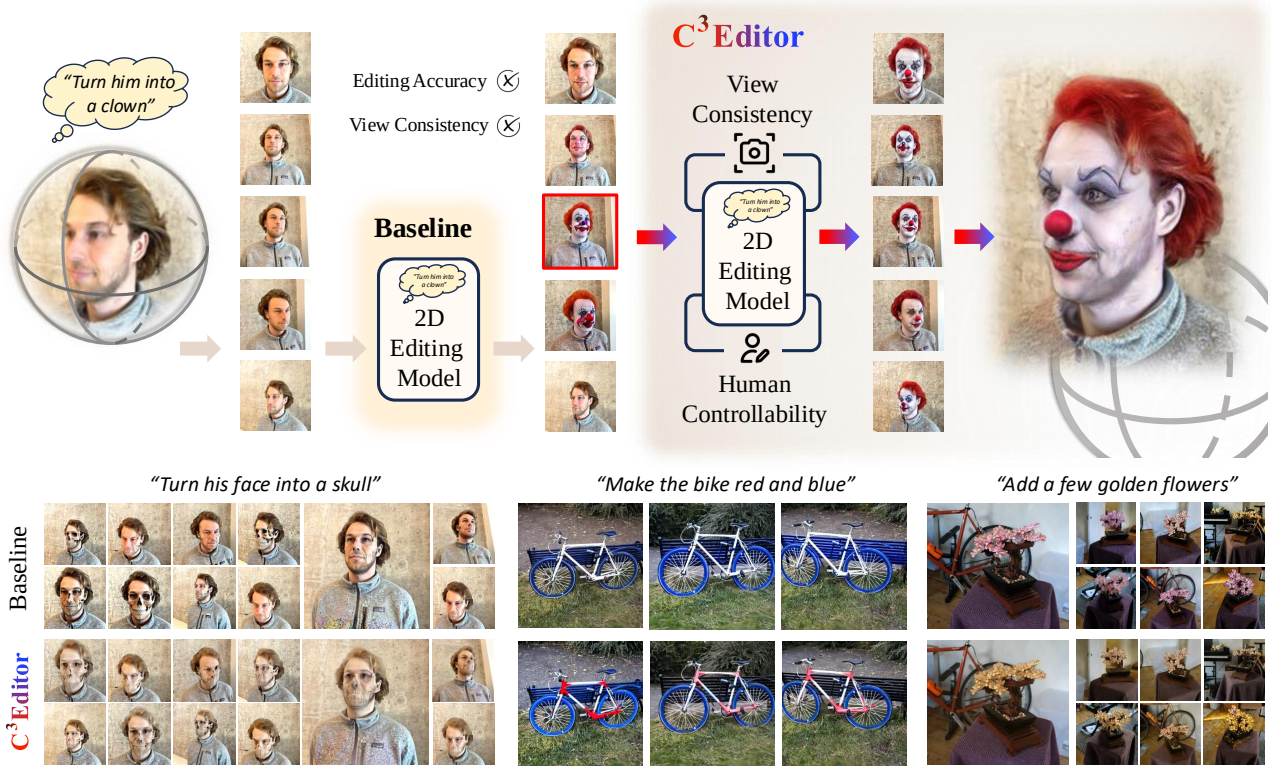

Figure 1. **C^{3}Editor: Controllable Consistent 2D Model for 3D Editing**. **Top:** Our C^{3}Editor method generates consistent 2D editing results across different views by following the original 3D scene, editing text, and user guidance, thereby supporting improved 3D editing performance. **Bottom:** Comparison of 2D and 3D editing results between baseline and C^{3}Editor.

Abstract

Existing 2D-lifting-based 3D editing methods often encounter challenges related to inconsistency, stemming from the lack of view-consistent 2D editing models and the difficulty of ensuring consistent editing across multiple views. To address these issues, we propose C³Editor, a controllable and consistent 2D-lifting-based 3D editing framework. Given an original 3D representation and a text-based editing prompt, our method selectively establishes a view-consistent 2D editing model to achieve superior 3D editing results. The process begins with the controlled selection of a ground truth (GT) view and its corresponding edited image as the optimization target, allowing for user-defined manual edits. Next, we fine-tune the 2D editing model within the GT view and across multiple views to align with the GT-edited image while ensuring multi-view consistency. To meet the distinct requirements of GT view fitting and multi-view consistency, we introduce separate LoRA modules for targeted fine-tuning. Our approach delivers more consistent and controllable 2D and 3D editing results than existing 2D-lifting-based methods, outperforming them in both qualitative and quantitative evaluations.

[*]Work done during internship at UC San Diego.

1. Introduction

The remarkable success of 2D generative models [2–4, 8, 18, 23, 41, 43, 45] has spurred rapid advancements in the field of generation, leading to successful applications in related areas such as editing tasks [1, 6, 16, 21, 29, 34]. Leveraging the superior performance of 2D models as priors also has become a popular approach in 3D tasks [12, 25, 30, 39, 44, 46]. Given the scarcity of real-world 3D data and the high cost of training, utilizing pretrained 2D models as guidance offers a promising solution. For example, 2D-lifting-based 3D editing methods [15] use a 2D editing model [6] to obtain edited images from different viewpoints, which are then used to update the original 3D representation.

However, directly transferring 2D priors to the 3D domain presents certain challenges, such as the issue of viewpoint consistency [32]. Since 2D models lack view information and 3D awareness, conflicts between views may arise when applied to 3D tasks. In 3D editing tasks, directly using edited 2D images that lack consistency across views can lead to errors in 3D editing. Some approaches attempt to address this by constructing external datasets [26, 27, 37]. However, addressing the view inconsistency problem in 3D editing remains challenging. The training process requires datasets containing consistent editing text, original 2D images, and edited 2D images across multiple views, which are difficult to obtain. One viable approach is to designate the edited result of one specific view as the ground truth (GT) and, leveraging the generalization ability of the 2D model, gradually adapt other viewpoints to match this viewpoint, achieving internal consistency across views.

Additionally, text-based editing inherently supports diversity, but current 2D-lifting-based 3D editing methods suppress this diversity by uniformly processing different 2D editing results [10, 13–15]. Our goal is to allow for the controllability of optimization directions, enabling the 3D editing results to express more possibilities and better align with human intent.

In response, we propose C^{3}Editor, a controllable and consistent 2D-lifting-based 3D editing method. Given an original 3D scene and an editing text prompt, we aim to obtain a view-consistent 2D editing model selectively, thereby achieving improved 3D editing results. By selecting a GT view and its corresponding edited image as the optimization target, our approach stabilizes the GT view's editing results and then progressively enforces consistency across different views through view propagation. Furthermore, we introduce separate LoRA modules to fine-tune the model, addressing the unique requirements of GT view fitting and multi-view consistency separately. This structured approach ensures that the 2D editing model achieves cohesive 3D editing results across all views, enhancing both visual consistency and user controllability.

In summary, our contributions are as follows:

- We develop a view-consistent 2D editing model based on the original 3D representation and an editing text prompt, facilitating enhanced 3D editing outcomes. This approach effectively bridges the gap between 2D and 3D, as well as between original and edited representations.
- Our controllable 3D editing method allows users to select a ground truth (GT) edited image and manually adjust it to produce consistent 3D editing results across views.
- Our C^{3}Editor method mainly focuses on two aspects: Intra-GT and Inter-view. We specifically design GT selection and intra-GT Loss methods to ensure stable GT fitting, followed by view propagation and inter-view loss for view consistency. Different LoRAs serve separate consistent purposes. Qualitative and quantitative experiments demonstrate the effectiveness of C^{3}Editor.

2. Related Work

2.1. Diffusion Model and Fine-tuning

Diffusion models [18, 35] have become powerful tools in generative tasks due to their unique approach of iteratively refining data from noise, allowing for precise control over the generation process. These models learn data distributions through a diffusion process that gradually adds and then reverses noise, effectively modeling complex data patterns in images, audio, and even text. Because of their robust performance, diffusion models are widely applied in tasks [12, 25, 30, 38–40, 42] such as image synthesis, inpainting, super-resolution, and conditional generation, where they can generate or manipulate visual content based on additional inputs, such as text prompts, segmentation maps, or depth maps. This versatility makes them particularly valuable for tasks requiring high-quality, detailed outputs and subtle adjustments.

Fine-tuning diffusion models is essential for adapting them to specific tasks or datasets. Through targeted fine-tuning, diffusion models can be optimized to perform controlled edits, match stylistic demands, or generalize to new domains beyond their original training data. Techniques such as low-rank adaptation (LoRA) [20] and other parameter-efficient tuning methods [19, 24, 28] allow for effective customization by focusing on updating key parts of the model while keeping the core structure intact. This approach is especially useful when integrating diffusion models as priors in cross-domain applications, where maintaining high fidelity across varying views is critical. Fine-tuning thus enables diffusion models to meet specialized generative requirements, ensuring they maintain both visual quality and flexibility across diverse tasks.

2.2. Diffusion-based 2D Editing

Diffusion-based 2D editing techniques [1, 6, 16, 21, 29, 34] have revolutionized the field of image manipulation by leveraging the denoising diffusion process to transform noise into structured visual representations. In these models, editing is performed iteratively, where each step refines the image by reversing the noise and generating realistic features, allowing for adequate control over the level and type of modifications applied.

The key advantage of diffusion-based 2D editing lies in its ability to use conditional inputs, like text prompts or segmentation maps, to guide the editing process. For example, Instruct-Pix2Pix [6] can interpret prompts to modify colors, add textures, or alter structures while maintaining the coherence of the image. These models can learn data distributions that align with specific editing goals, making them versatile across diverse applications. By fine-tuning or adjusting model parameters, diffusion models can also be specialized for specific editing tasks, allowing them to adapt to particular styles or constraints required by the user. This combination of iterative refinement, conditional control, and adaptability has made diffusion-based 2D editing a powerful tool in modern image generation and editing tasks.

2.3. 2D-lifting-based 3D Editing

Recent advancements in 3D editing have increasingly integrated diffusion-based 2D editing models, leveraging their established capabilities to enhance 3D workflows [7, 9, 10, 13–15]. These models, originally designed for detailed image modifications, contribute to 3D editing by transferring their proficiency in nuanced, high-quality adjustments to three-dimensional representations. Methods like Neural Radiance Fields (NeRF) [31] and 3D Gaussian Splatting (3DGS) [22] incorporate 2D editing models to improve the consistency and detail of 3D content.

By using 2D diffusion models as priors, recent approaches enhance the fidelity and stylistic consistency of 3D edits, especially in maintaining coherence across multiple views. Some works, such as DGE [11], combine images from different viewpoints into videos for processing. A primary challenge in this domain remains ensuring multi-view consistency, as traditional 2D-based edits applied to 3D models often lead to discrepancies between perspectives. Some methods, such as ConsistDreamer [10], model 3D-aware consistency by means of constraints like neural feature alignment or volume-based feature consistency. This has provided inspiration for our work. However, since it is not open-sourced, it is impossible to make a comparison for now. We compare our method with NeRF-based Instruct-NeRF-to-NeRF [15], ViCA-NeRF [14], and GS-based GaussianEditor [13].

3. Method

3.1. Overview

Given a 3D representation Φ (*e.g.*, 3D GS), a text prompt for editing y, and the original 2D editing model Θ_O (like Instruct-Pix2Pix [6]), the goal of our method is to process Θ_O to obtain Θ_C that is related to y and ensures multi-view consistency, thereby achieving improved 3D editing results. In Sec. 3.2, the ground truth (GT) view v_{gt} and GT edited image $I^e_{v_{\mathrm{gt}}}$ are manually selected from the 2D editing results I^e_v of different views v, rendered by Θ_O, which serve as the optimization target. In Sec. 3.3, we optimize Θ_O with a specifically designed intra-GT loss to fit $I^e_{v_{\mathrm{gt}}}$. We maintain global consistency through the view propagation method and inter-view loss described in Sec. 3.4. In Sec. 3.5, we introduce different LoRAs for different fine-tuning objectives to separately fine-tune the diffusion model.

Using Θ_C obtained, each view v in V undergoes a complete editing process, producing consistent view-editing results I^e_v. Considering the gradient storage issues in fine-tuning the diffusion model, each complete editing process includes 5 diffusion denoising steps, which achieves a good trade-off between GPU memory limits and editing quality. The final 3D editing result is obtained by updating the original 3D representation Φ with the edited results of all views I^e_v. We adopt 3D Gaussian Splatting (3D GS) as our 3D representation due to its efficient training speed and excellent rendering quality. We adopt widely used Instruct-Pix2Pix [6] as our diffusion-based pre-trained 2D editing model, for its outstanding performance in 2D editing tasks. The method for updating 3D GS is consistent with that in GaussianEditor. The detailed process is illustrated in Fig. 2.

3.2. Controllable Optimization Direction

The independent 2D editing processes of different views v lead to different editing results. To avoid view conflicts in 3D editing, we select the editing result $I^e_{v_{\mathrm{gt}}}$ from a specific view v_{gt} as the optimization direction. In subsequent operations, the 2D editing model will use this GT as a reference to edit images from other views, thereby preventing conflicts in the 3D editing process.

As shown in Fig. 2 Phase 1, for each view v, an independent editing process is performed, resulting in different editing outcomes I^e_v. User then selects a specific view and its corresponding edited result as the GT view v_{gt} and GT edited image $I^e_{v_{\mathrm{gt}}}$, setting the target optimization direction. Different choices of view and edited results lead to different optimization directions, and consequently, varying final 3D editing outcomes, which are shown in Sec. 4.3. Therefore, this selection should follow certain guidelines, such as choosing results of higher editing quality and selecting a more central view. Based on these guidelines, the user can choose their desired optimization direction.

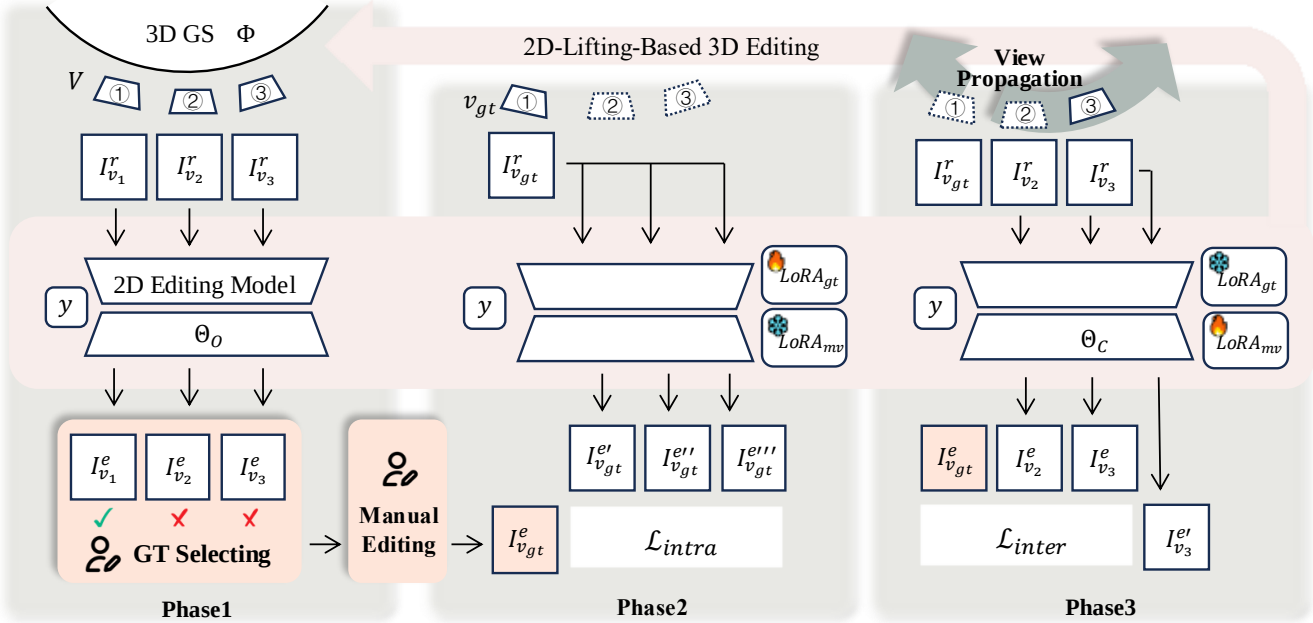

Figure 2. **C^{3}Editor Method Pipeline**. Given a 3D representation Φ, a text prompt for editing y, and the original 2D editing model Θ_O, our method aims to process Θ_O to obtain Θ_C that is related to y and ensures multi-view consistency, thereby achieving improved 3D editing results. **Phase 1**: Controllable optimization direction selecting and manual editing in Sec. 3.2. **Phase 2**: Intra-GT prior fitting in Sec. 3.3 to fit the GT information. **Phase 3**: View propagation and inter-view consistent construcing in Sec. 3.4. Details of LoRA modules for separate fine-tuning are in Sec. 3.5.

For the obtained GT image $I^e_{v_{\text{gt}}}$, users can directly use it as is. However, if there are any unsatisfactory elements, users can make manual edits according to their preferences. They can utilize image editing tools such as Photoshop to modify the image content, then set the edited image as the GT image $I^e_{v_{\text{gt}}}$ for the model. The GT view and edited result are then used to guide the subsequent 2D editing model fine-tuning process, ensuring that the 2D model can achieve controllable editing results across all views.

3.3. Intra-GT Prior Fitting

With the optimization direction established, the next step is to train the 2D diffusion model to fit the GT image. Adjustments to the 2D diffusion model are divided into two parts: intra-GT and inter-view adaptation. In this section, we need to make the 2D model fit the chosen optimization direction $I^e_{v_{\text{gt}}}$ on the GT view v_{gt}, aiming to establish a foundational intra-view editing stability on the GT view v_{gt}.

As shown in Fig. 2 Phase 2, we freeze the 3D representation and add LoRA modules to the diffusion model for fine-tuning. The edited image $I^e_{v_{\text{gt}}}$ is used as the GT. An independent, complete editing process is performed on the rendered image $I^r_{v_{\text{gt}}}$ of v_{gt} to obtain an edited image $I^{e'}_{v_{\text{gt}}}$ that differs from the GT image $I^e_{v_{\text{gt}}}$. Compute the loss between $I^{e'}_{v_{\text{gt}}}$ and $I^e_{v_{\text{gt}}}$, back-propagate, and update the LoRA. The loss $\mathcal{L}_{\text{intra}}$ consists of two parts: the L_1 loss and perceptual loss between $I^{e'}_{v_{\text{gt}}}$ and $I^e_{v_{\text{gt}}}$.

$$\mathcal{L}_{\text{intra}} = \lambda_1 L_1(I^{e'}_{v_{\text{gt}}}, I^e_{v_{\text{gt}}}) + \lambda_2 L_{\text{Perceptual}}(I^{e'}_{v_{\text{gt}}}, I^e_{v_{\text{gt}}}) \quad (1)$$

Through multiple iterations of this process, the fine-tuned 2D diffusion model acquires a certain fitting capability for the GT image, while also improving editing stability for the same view, achieving similar results across different editing processes.

3.4. View Propagation and Inter-view Consistency

After Sec. 3.3, the 2D editing model can only fit $I^e_{v_{\text{gt}}}$ on v_{gt}, with limited generalization ability, and has limited consistency in editing effects for views that differ significantly. If the current 2D model is used directly as the prior, it can only maintain consistent editing for v_{gt} and views nearby, while its performance on more distant views remains uncertain. Therefore, in this section, we introduce additional methods to ensure consistent editing across all views. We leverage the interrelations between viewpoints to enable the 2D diffusion model to achieve consistent editing across all views gradually.

As shown in Fig. 2 Phase 3, specifically, we sort the views as a sequence S by their distance of camera center points from v_{gt}, from closest to farthest, and then perform fine-tuning of the 2D diffusion model on each view in S

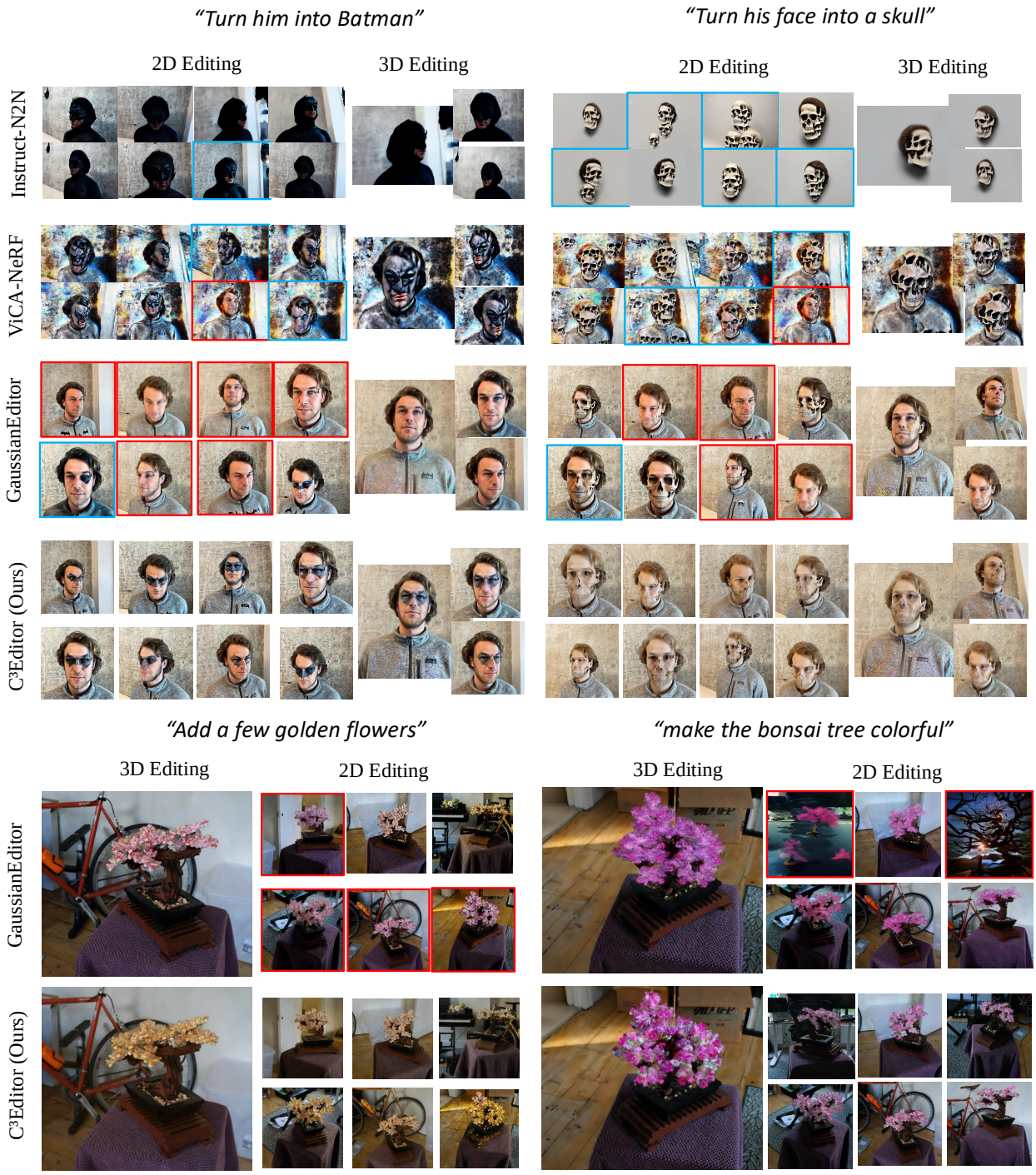

Figure 3. **Comparison of Qualitative Results**. Compared to baseline methods, C^{3}Editor can generate view-consistent 2D images, avoiding inter-view conflicts (highlighted in blue) and erroneous 2D edits (highlighted in red), thereby achieving better 3D editing results.

other than v_{gt}. v_0 in the sequence represents v_{gt}. We perform a 2D editing process on each view v_i with the index $i \in \{1, 2, \ldots, j, i, \ldots, n-1\}$ in the sequence separately. The resulting image $I^e_{v_i}$ serves as the GT image for v_i. Next, an independent 2D editing process is applied to this view, and another edited image $I^{e'}_{v_i}$ is obtained. The loss $\mathcal{L}_{\text{inter}}$ comprises three parts: loss 1 between the edited image $I^{e'}_{v_i}$ and the GT image $I^e_{v_i}$, loss 2 between $I^{e'}_{v_i}$ and $I^e_{v_j}$ of the closest processed view v_j, loss 3 between $I^{e'}_{v_i}$ and $I^e_{v_{\text{gt}}}$. $\mathcal{L}_{\text{inter}}$ is

as follows:

$$\mathcal{L}_{\text{inter}} = \underbrace{\lambda_3 L_1(I^{e'}_{v_i}, I^{e}_{v_i}) + \lambda_4 L_{\text{Perceptual}}(I^{e'}_{v_i}, I^{e}_{v_i})}_{\text{loss 1}} + \underbrace{\lambda_5 L_{\text{Perceptual}}(I^{e'}_{v_i}, I^{e}_{v_j})}_{\text{loss 2}} + \underbrace{\lambda_6 L_{\text{Perceptual}}(I^{e'}_{v_i}, I^{e}_{v_{\text{gt}}})}_{\text{loss 3}} \quad (2)$$

$\mathcal{L}_{\text{inter}}$ is back-propagated, and LoRA is used to fine-tune the diffusion model. After this process, we reverse the sequence S and repeat the above steps until reaching v_{gt}. The generalization capability gradually expands from the v_{gt} to encompass all views.

3.5. Separate Fine-tuning

To prevent the loss of GT information during the inter-view fine-tuning process, we design two LoRAs, each serving different fine-tuning goals. The fine-tuning of the 2D Editing model Θ_O is divided into two main aspects: LoRA_{gt} for fitting the GT view image $I^{e}_{v_{\text{gt}}}$, and LoRA_{mv} for ensuring consistency across different views. The inference process takes place in Sec. 3.2 and Fig. 2 Phase 1, with no trainable model parameters. In Sec. 3.3 and Fig. 2 Phase 2, we use LoRA_{gt} to fine-tune Θ_O while keeping LoRA_{mv} frozen. After this step, LoRA_{gt} helps the Θ_O fit $I^{e}_{v_{\text{gt}}}$. In Sec. 3.4 and Fig. 2 Phase 3, we freeze LoRA_{gt} and use LoRA_{mv} to fine-tune Θ_O. During the separate fine-tuning process, the model uses the GT information obtained by LoRA_{gt} and leverages LoRA_{mv} to achieve global consistency.

4. Experiments

4.1. Implementation Details

Our method builds on the advanced 2D-lifting-based 3D GS Editing Method, GaussianEditor [13]. Specifically, we use 3D GS [22] as the 3D representation and the widely-used Instruct-Pix2Pix [6] as the diffusion-based 2D editing model. All experiments were conducted on a single NVIDIA RTX A6000, with the fine-tuning process taking 1 minute in total. We use MipNeRF-360 [5] and Instruct-NeRF-to-NeRF dataset [15] to measure the performance of our method. The MipNeRF-360 dataset contains 360-degree views of 3D scenes, while the Instruct-NeRF-to-NeRF dataset contains 3D scenes. We use the CLIP-Score [47] (image-text and image-image) as the evaluation metrics. The former measures the similarity between 3D edited results and editing text, while the latter measures the similarity between 2D images produced by the 2D editing process. A higher score indicates greater editing quality and view consistency. We also use the Fréchet Inception Distance (FID) [17, 36] between original rendered images and edited results to evaluate the quality of 3D editing. A lower FID score indicates higher image quality.

For each scene and editing text, we perform unique training to obtain the corresponding 2D editing model. Following the same approach as GaussianEditor, we first use Gaussian Semantic Tracing [13] to generate a mask of the editing target within the 3D GS. We then follow the processing steps outlined in Sec. 3 to obtain a 2D editing model with view consistency. This model serves as the 2D prior, achieving the final 3D editing result. For further details on the 3D editing process, please refer to GaussianEditor [13].

The hyperparameters are set as follows: 30 iterations for $\mathcal{L}_{\text{intra}}$ updates, and 3 iterations for $\mathcal{L}_{\text{inter}}$ updates. We set λ_n=1. For the two LoRA modules, we use the AdamW optimizer with the following settings: $r = 4$, LoRA alpha $=$ 4, init LoRA weights="gaussian", lr $= 10^{-4}$, betas $= (0.9, 0.999)$, weight decay $= 10^{-2}$, eps $= 10^{-8}$.

4.2. Comparison with Baselines

Qualitative Comparison. Fig. 3 illustrates the qualitative results of our method. Compared to NeRF-based Instruct-NeRF-to-NeRF [15], ViCA-NeRF [14], and GS-based GaussianEditor [13] methods, our approach demonstrates superior consistency in both 2D and 3D editing. With the editing texts of the face scene, such as "Turn him into Batman" and "Turn his face into a skull", our method addresses inconsistency issues encountered by baseline methods, achieving accurate edits on facial features. In natural scenes, like the bonsai scene, our approach produces improved editing results compared to baseline methods, excelling in color and other details. This improvement is due to the fact that current 3D editing techniques using 2D editing models as priors often encounter inconsistencies during 2D editing, leading to issues such as inter-view inconsistency (highlighted in the blue boxes) and editing errors (highlighted in the red boxes), which result in inaccurate or incomplete 3D edits. Our method, however, achieves consistent 2D editing results, leading to accurate 3D edits. For more qualitative results, please refer to the following sections and supplementary materials.

Method	GaussianEditor	**C^{3}Editor (Ours)**
Image-Text CLIP-Score (↑)	24.18	**25.21**
Image-Image CLIP-Score (↑)	84.20	**87.46**
FID (↓)	112.21	**89.95**
Time Difference	Avg 56s more than GaussianEditor	

Table 1. **Comparison of Quantitative Results**. Our method surpasses the baseline method on all of the three metrics.

Quantitative Comparison. As shown in Tab. 1, we use CLIP-Score [47] (image-image and image-text) and FID [17, 36] as the metrics for quantitative evaluation.

Specifically, we calculate the CLIP-Score between images from edited 3D results and editing text. A higher score indicates better editing loyalty. Our method achieves a higher CLIP-Score, demonstrating the improved quality of our 3D editing results. We also calculate the CLIP-Score within the 2D images produced by the editing process. A higher score indicates greater similarity between edited 2D images, thus representing stronger view consistency. Our method achieves a higher CLIP-Score, demonstrating the view consistency of our editing approach. The lower FID score of our method indicates better image quality in the 3D editing results. Our method outperforms GaussianEditor in both qualitative and quantitative evaluations, showcasing the effectiveness of our approach in achieving controllable and consistent 3D editing results.

4.3. Controllable Editing

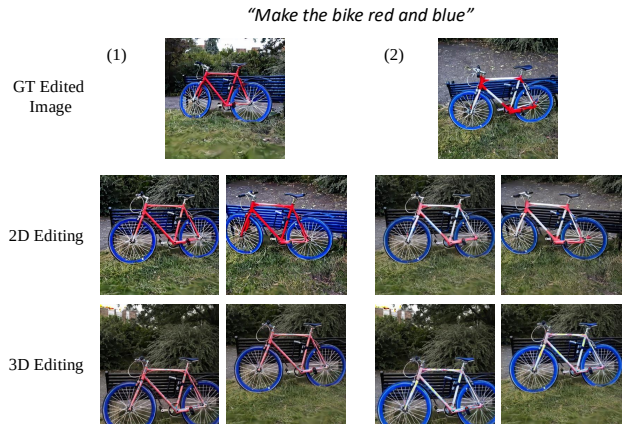

Figure 4. **Controllable Editing Results with Different GT Selections**. In C^{3}Editor, users can decide the optimization direction by selecting the GT edited image they prefer.

Controllable GT Selection. With different selections of the GT edited image, our method can achieve the corresponding editing results. As shown in Fig. 4, given the editing prompt "Make the bike red and blue", the pre-trained 2D editing model can produce different outcomes. In Fig. 4 (1), the wheels are edited to blue while the entire bike frame is edited to red. In Fig. 4 (2), the wheels are also blue, but only part of the frame is edited to red. Using our method, the obtained 2D editing model can edit images from other views to produce the corresponding 3D editing result.

The 2D editing process inherently allows for diverse outcomes, and our controllable GT selection effectively supports this diversity, enabling results that better align with user intentions. By allowing targeted selection of the GT image, our approach minimizes 3D editing errors that may arise from inaccuracies in 2D editing, thus enhancing the precision and stability of the final editing outcome.

Manual GT Editing. Furthermore, users can manually edit the GT image according to their preferences. As shown in Fig. 5, for the editing prompt "Make the bike red and blue", we modify the original edit to turn the front wheel of the bike red. Using this manually edited image as the GT image $I^e_{v_{\text{gt}}}$, we proceed with the subsequent steps. The final 2D and 3D editing results maintain consistency with the GT image and exhibit view consistency. Our method offers users a manual editing option, enabling them to correct 2D editing results and align the model's output with human intent. This feature introduces an additional dimension of controllable generation, allowing for enhanced customization and adaptability in the editing process.

Figure 5. **Controllable Editing Results with Manual Editing**. In C^{3}Editor, users can edit the GT manually and obtain the corresponding 2D and 3D editing results.

4.4. Ablation Study

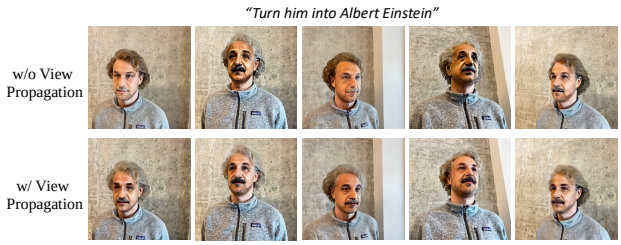

Figure 6. **Ablation Study on View Propagation**. View propagation helps obtain more view-consistent results than the GT view.

Effectiveness of View Propagation. We conduct ablation experiments to evaluate the effectiveness of view propagation. As shown in Fig. 6, after fine-tuning the 2D diffusion model based on the GT image $I^e_{v_{\text{gt}}}$, we sort one set of views in order based on their camera center distance to the GT view v_{gt} and another set in random order. We then proceed with subsequent steps under these different viewpoint orders. It can be observed that without view propagation, the 2D diffusion model could not achieve fully consistent results. However, with view propagation, the consistency significantly improved. This is because, once the model is fitted to the GT, it gains the ability to produce stable outputs for the GT. It also exhibits a certain level of generalization for viewpoints close to the GT camera position. However, for more distant viewpoints, due to the large gap between input images I^r_v, it is unable to achieve a consistent editing

result.

Method	LoRA	LoRA_{gt} + LoRA_{mv}
Image-Image CLIP-Score (↑)	87.03	**87.46**

Table 2. **Ablation Study on Separate Fine-Tuning**. Using different LoRAs to separately fine-tune the diffusion model can achieve better performance on view consistency.

Effectiveness of Separate LoRA Fine-tuning. We also conduct ablation on the design of LoRA. As shown in Tab. 2, we compare results obtained using only a single LoRA with those achieved using different LoRAs for fine-tuning different parts. We use the image-to-image CLIP-Score as the evaluation metric. It is observed that when using two LoRAs for fine-tuning different components, our method produces better results. This is because fine-tuning the diffusion model on the same LoRA can cause disturbances to the previously acquired GT information during subsequent viewpoint fine-tuning, thereby reducing inter-view editing consistency.

5. Visualization

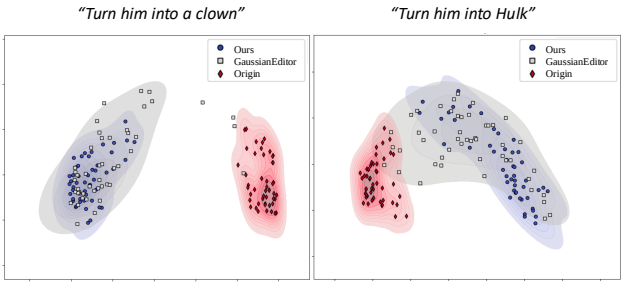

Figure 7. **Visualization of Original and Edited Image Features**. Features of edited images obtained by C^3Editor are more concentrated than the baseline model.

Visualization of Rendered and Edited Image Features. We visualized the image features resulting from our edits in Fig. 7. Each point in the figure represents an image after feature extraction and dimensionality reduction. Each 2D image was feature-extracted using CLIP ViT-B/32 [33], followed by PCA for dimensionality reduction, and these features were plotted as 2D scatter plots and density plots. In the figures, blue represents images generated by our method, gray represents those generated by the baseline method, GaussianEditor, and red represents the original rendered images. The editing prompt on the left is "Turn him into a clown," while on the right, it is "Turn him into Hulk." As shown, the features of the 2D edited images produced by our method are more concentrated than those from the baseline method, indicating that our method achieves stronger view consistency, nearly matching the original images' consistency. Additionally, the features generated by our method show almost no outliers or points that are confused with the original image features, demonstrating that our approach avoids the erroneous edits seen in the baseline method. Our method not only improves the consistency of the generated images but also reduces the occurrence of incorrect edits.

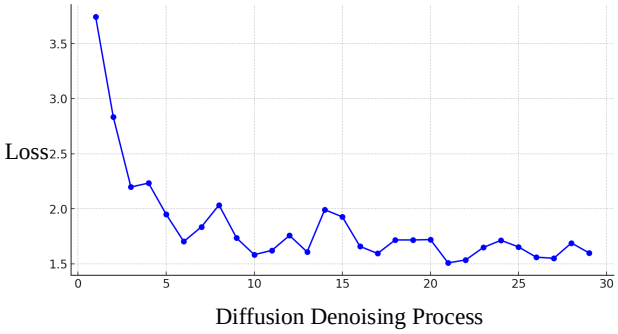

Figure 8. **Visualization of Loss Change During Intra-GT Prior Fitting**. The loss gradually decreases as the iterations progress, indicating that the fine-tuning process effectively stabilizes the editing results for the GT view.

Visualization of Intra-GT Prior Fitting. Fig. 8 illustrates the change in $\mathcal{L}_{\text{intra}}$ during the Intra-GT Prior Fitting phase. In this process, our goal is for each independently performed diffusion denoising process on the GT view to approximate the GT edited image. As shown in the figure, the loss gradually decreases as the iterations progress. This indicates that the fine-tuning process effectively stabilizes the editing results for the GT view, consistently producing outputs close to the GT edited image. At this phase, the 2D editing model increasingly captures the GT information and achieves stable editing on the GT view.

6. Conclusion

In this paper, we propose C^3Editor, a controllable and consistent 2D-lifting-based 3D editing method. Our approach creates the specific 2D editing model to assist in achieving view consistency and controllable 3D editing results. Qualitative and quantitative evaluations demonstrate that our method outperforms baseline methods in both 2D and 3D results.

Limitations. Our method still has certain limitations. For instance, a unique 2D editing model must be trained for each specific scene and editing prompt. Meanwhile, the 3D optimization process may also result in the outcome not being fully consistent with the GT edited image. In the future, we aim to enhance the generalization capabilities of the editing model and move towards developing a truly generic multi-view 3D editing model.

Acknowledgment This work is supported by NSF award IIS-2127544 and NSF award IIS-2433768.

References

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

[2] Omri Avrahami, Ohad Fried, and Dani Lischinski. Blended latent diffusion. *ACM Transactions on Graphics (TOG)*, 42 (4):1–11, 2023. 2

[3] Omri Avrahami, Amir Hertz, Yael Vinker, Moab Arar, Shlomi Fruchter, Ohad Fried, Daniel Cohen-Or, and Dani Lischinski. The chosen one: Consistent characters in text-to-image diffusion models. *arXiv preprint arXiv:2311.10093*, 2023.

[4] Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Soumyadip Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 843–852, 2023. 2

[5] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5470–5479, 2022. 6

[6] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18392–18402, 2023. 2, 3, 6

[7] Chenjie Cao, Chaohui Yu, Yanwei Fu, Fan Wang, and Xiangyang Xue. Mvinpainter: Learning multi-view consistent inpainting to bridge 2d and 3d editing. *arXiv preprint arXiv:2408.08000*, 2024. 3

[8] Hila Chefer, Yuval Alaluf, Yael Vinker, Lior Wolf, and Daniel Cohen-Or. Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models. *ACM Transactions on Graphics (TOG)*, 42(4):1–10, 2023. 2

[9] Jun-Kun Chen and Yu-Xiong Wang. Proedit: Simple progression is all you need for high-quality 3d scene editing. *arXiv preprint arXiv:2411.05006*, 2024. 3

[10] Jun-Kun Chen, Samuel Rota Bulò, Norman Müller, Lorenzo Porzi, Peter Kontschieder, and Yu-Xiong Wang. Consistdreamer: 3d-consistent 2d diffusion for high-fidelity scene editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21071–21080, 2024. 2, 3

[11] Minghao Chen, Iro Laina, and Andrea Vedaldi. Dge: Direct gaussian 3d editing by consistent multi-view editing. *arXiv preprint arXiv:2404.18929*, 2024. 3

[12] Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia. Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22246–22256, 2023. 2

[13] Yiwen Chen, Zilong Chen, Chi Zhang, Feng Wang, Xiaofeng Yang, Yikai Wang, Zhongang Cai, Lei Yang, Huaping Liu, and Guosheng Lin. Gaussianeditor: Swift and controllable 3d editing with gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21476–21485, 2024. 2, 3, 6

[14] Jiahua Dong and Yu-Xiong Wang. Vica-nerf: View-consistency-aware 3d editing of neural radiance fields. *Advances in Neural Information Processing Systems*, 36, 2024. 3, 6

[15] Ayaan Haque, Matthew Tancik, Alexei A Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-nerf2nerf: Editing 3d scenes with instructions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19740–19750, 2023. 2, 3, 6

[16] 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. 2, 3

[17] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 6

[18] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 2

[19] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pages 2790–2799. PMLR, 2019. 2

[20] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 2

[21] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6007–6017, 2023. 2, 3

[22] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 3, 6

[23] Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-concept customization of text-to-image diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1931–1941, 2023. 2

[24] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021. 2

[25] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. In *Proceedings of the IEEE/CVF*

Conference on Computer Vision and Pattern Recognition, pages 300–309, 2023. 2

[26] Minghua Liu, Chao Xu, Haian Jin, Linghao Chen, Mukund Varma T, Zexiang Xu, and Hao Su. One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization. *Advances in Neural Information Processing Systems*, 36, 2024. 2

[27] 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*, pages 9298–9309, 2023. 2

[28] Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Lam Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv preprint arXiv:2110.07602*, 2021. 2

[29] 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*, 2021. 2, 3

[30] Gal Metzer, Elad Richardson, Or Patashnik, Raja Giryes, and Daniel Cohen-Or. Latent-nerf for shape-guided generation of 3d shapes and textures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12663–12673, 2023. 2

[31] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 3

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

[33] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 8

[34] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1 (2):3, 2022. 2, 3

[35] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 2

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

[37] Yichun Shi, Peng Wang, Jianglong Ye, Mai Long, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. *arXiv preprint arXiv:2308.16512*, 2023. 2

[38] Divyansh Srivastava, Xiang Zhang, He Wen, Chenru Wen, and Zhuowen Tu. Lay-your-scene: Natural scene layout generation with diffusion transformers. *arXiv preprint arXiv:2505.04718*, 2025. 2

[39] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *Advances in Neural Information Processing Systems*, 36, 2024. 2

[40] Haiyang Xu, Yu Lei, Zeyuan Chen, Xiang Zhang, Yue Zhao, Yilin Wang, and Zhuowen Tu. Bayesian diffusion models for 3d shape reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10628–10638, 2024. 2

[41] Xingqian Xu, Jiayi Guo, Zhangyang Wang, Gao Huang, Irfan Essa, and Humphrey Shi. Prompt-free diffusion: Taking" text" out of text-to-image diffusion models. *arXiv preprint arXiv:2305.16223*, 2023. 2

[42] Guanning Zeng, Xiang Zhang, Zirui Wang, Haiyang Xu, Zeyuan Chen, Bingnan Li, and Zhuowen Tu. Yolo-count: Differentiable object counting for text-to-image generation. *arXiv preprint arXiv:2508.00728*, 2025. 2

[43] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023. 2

[44] Xiang Zhang, Zeyuan Chen, Fangyin Wei, and Zhuowen Tu. Uni-3d: A universal model for panoptic 3d scene reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9256–9266, 2023. 2

[45] Xinchen Zhang, Ling Yang, Yaqi Cai, Zhaochen Yu, Jiake Xie, Ye Tian, Minkai Xu, Yong Tang, Yujiu Yang, and Bin Cui. Realcompo: Dynamic equilibrium between realism and compositionality improves text-to-image diffusion models. *arXiv preprint arXiv:2402.12908*, 2024. 2

[46] Qingcheng Zhao, Xiang Zhang, Haiyang Xu, Zeyuan Chen, Jianwen Xie, Yuan Gao, and Zhuowen Tu. Depr: Depth guided single-view scene reconstruction with instance-level diffusion, 2025. 2

[47] SUN Zhengwentai. clip-score: CLIP Score for PyTorch. https://github.com/taited/clip-score, 2023. Version 0.1.1. 6