3D-Consistent Multi-View Editing by Correspondence Guidance

Josef Bengtson[1], David Nilsson[1], Dong In Lee[2], Yaroslava Lochman[1], and Fredrik Kahl[1]

[1] Chalmers University of Technology
{bjosef,david.nilsson,lochman,fredrik.kahl}@chalmers.se
[2] Korea University
dilee99@korea.ac.kr

Abstract. Recent advancements in diffusion and flow models have greatly improved text-based image editing, yet methods that edit images independently often produce geometrically and photometrically inconsistent results across different views of the same scene. Such inconsistencies are particularly problematic for editing of 3D representations such as NeRFs or Gaussian splat models. We propose a training-free guidance framework that enforces multi-view consistency during the image editing process. The key idea is that corresponding points should look similar after editing. To achieve this, we introduce a consistency loss that guides the denoising process toward coherent edits. The framework is flexible and can be combined with widely varying image editing methods, supporting both dense and sparse multi-view editing setups. Experimental results show that our approach significantly improves 3D consistency compared to existing multi-view editing methods. We also show that this increased consistency enables high-quality Gaussian splat editing with sharp details and strong fidelity to user-specified text prompts. Please refer to our project page for video results: https://3d-consistent-editing.github.io/

Keywords: Multi-View Editing · Diffusion models · 3D Consistency

1 Introduction

Text-based image editing has become increasingly powerful with generative models, allowing modification of an image with a prompt such as "turn the person into a clown" or "make it foggy". While these methods show impressive results when editing single images, using them to edit multiple images of the same scene, or to edit 3D models, in a consistent manner remains difficult.

Given recent work on photorealistic rendering of 3D models using NeRFs [40] or Gaussian splatting [27], a natural question is whether we can edit such 3D models using image editing. If the training views used to train 3D models are edited independently, it has the issue that different images are edited with different characteristics. For example, editing to a clown face as in Fig. 1 can look realistic for each image but different characteristics such as the nose size or

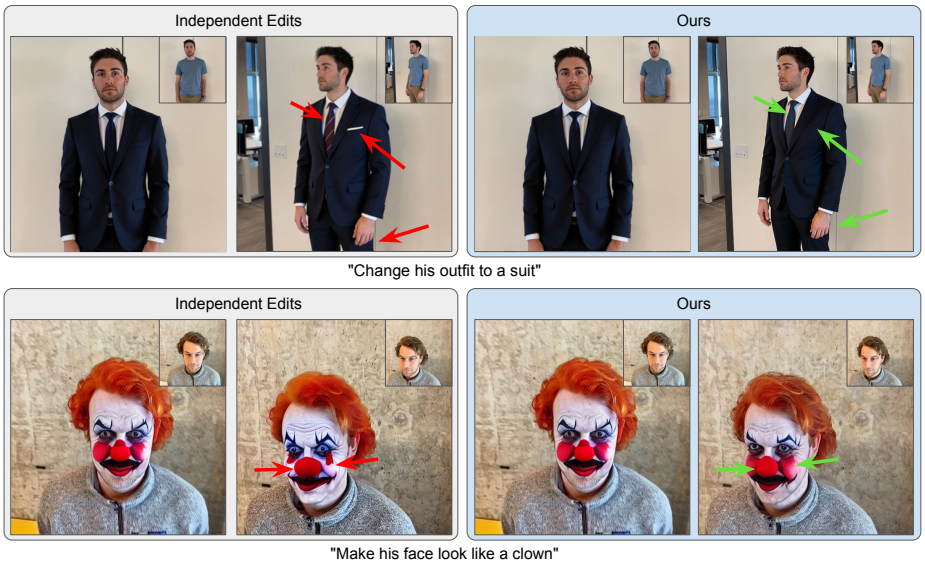

Fig. 1: Image editing methods applied independently to multi-view images often produce inconsistent edits across views, e.g. in the tie pattern and the location and shape of the red dots on the clown's cheek (red arrows). Our method improves multi-view consistency (green arrows) by guiding the denoising process so that corresponding points are edited similarly. Object poses also better match the original images.

painting on the cheeks are inconsistent for different views. For 3D editing this implies that the edited training views do not depict the same 3D scene, making editing difficult. This motivates modifying the image editing so that multiple views can be edited in a multi-view consistent way. Early work on 3D editing such as Instruct-NeRF2NeRF [20] addresses the multi-view issue by alternating between editing all images and updating the 3D model until convergence. Recent works such as EditSplat [33] and DGE [12] modify image generation using multi-view constraints or attention to improve consistency and directly update the 3D model, bypassing the costly iterative regeneration of edited images.

We present a method to directly guide the denoising process of diffusion and flow-matching image editing methods so that the edited images are multi-view consistent, as shown in Fig. 1. Our key idea is that corresponding points across images should look similar after editing. Based on this idea, we introduce a simple guidance loss that measures multi-view consistency. One way to approach this is by matching points between *unedited* images and forcing these points to be edited in a similar way. This approach does not allow for larger geometry changes. While the primary focus of our work is on *near-rigid* edits, we also find that using the same loss with matches between *edited* images improves multi-view consistency in *non-rigid* edits, i.e. edits with larger geometry changes. During image generation, we enforce the consistency by using training-free guidance

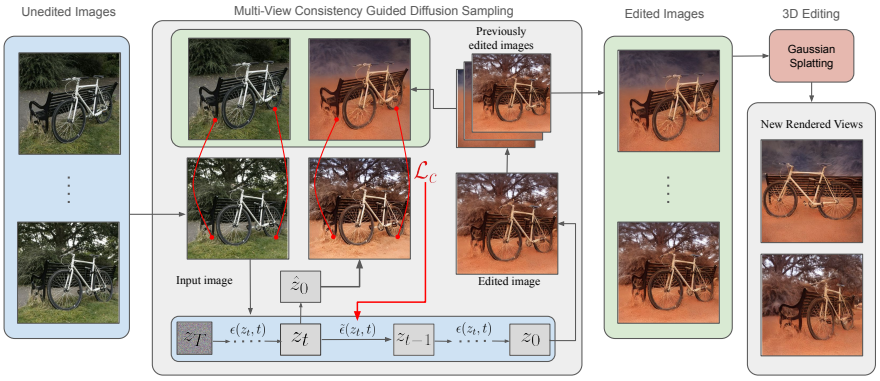

Fig. 2: Overview of our method. Given input images, each view is edited sequentially by guiding the denoising process using previously edited images. The guidance assumes corresponding points should look similar in the edited images. During denoising, the noise estimate $\epsilon(z_t, t)$ is modified according to a consistency loss $\mathcal{L}_c$ producing multi-view consistent edits. Edited images are then used to update a Gaussian splat model.

which modifies the denoising process so that it is guided towards samples where our consistency loss has a low value.

As an application of our multi-view consistent image editing, we show how to edit 3D Gaussian splat models. We simply resume training based on the edited images which are now sufficiently multi-view consistent to get a Gaussian splat model with sharp and realistic details. Our method also enables consistent editing of sparse views, and it works with different image editing methods.

In summary, our main contributions are:

- We introduce a flexible training-free method able to guide widely varying image editing models to generate multi-view consistent images.
- Our method is based on the key idea that corresponding points should look similar in the edited images. This is enforced by optimizing the proposed consistency loss on corresponding points across views.
- We show that the generated images can be used to refine a Gaussian splat model directly and accurately reconstruct fine details.
- We compare our method to recent work and show improved multi-view consistency of the generated images prior to Gaussian splat refinement, and we obtain edited Gaussian splat models faithful to the given text prompts with clear details, demonstrated through extensive video comparisons.

2 Related Work

2.1 2D Image Editing

The development of powerful generative models [19, 47] has led to large advancements in image editing, initially mainly through Generative Adversarial

Networks (GANs) [19, 56, 73, 74] and more recently through models based on diffusion [26, 30, 48, 59] and flow matching [24, 29, 31]. InstructPix2Pix [9] finetunes Stable Diffusion [47] using image pairs generated by prompt-to-prompt (P2P) [22] following instructions generated by GPT-3 [10], enabling instruction-based editing. However, its limiting factor is the long editing time due to the large number of denoising steps. Subsequent methods therefore employ few-step [1, 14, 18, 39, 62] or one-step [32, 42, 43, 72] denoising. Our work focuses on how to make such methods multi-view consistent, enabling direct 3D editing.

2.2 3D Scene Editing

One approach to 3D scene editing is to utilize paired 3D data to directly edit a 3D representation [8, 34, 61, 65], but such data is expensive to acquire, limiting training to single objects and synthetic scenes. Another popular approach leverages existing image editing methods for 3D editing, but these 2D methods provide no consistency guarantees across views. Some works use score distillation sampling (SDS) [45] to update 3D representation [25, 35, 51, 75]. Instruct-NeRF2NeRF [20] uses consistency from a 3D representation to achieve consistent edits by iteratively editing views and updating 3D representation. This idea, known as iterative dataset update, is widely used in follow-up works [13, 15, 41, 54, 57].

The iterative process is compute-intense and several approaches make the editing process more consistent to reduce dataset update iterations. Improvements range from incorporating correspondence regularization into the denoising process [4, 52], scene-specific finetuning [69], utilizing geometric information from the 3D representation or monocular depth estimation to guide the editing [17, 28, 33, 60] and allowing the attention process in the editing method to consolidate information across views [12, 17, 57, 58, 60]. Several of these recent approaches [12, 57] rely on iterative refinement on top of multi-view consistent editing. In contrast, our method provides further consistency improvements and does not require updating the images. Our method can also handle sparse view editing, while several baseline methods are limited to dense views, since they require geometric information from an existing 3D representation [33, 60] or rendering a smooth camera trajectory, inspired by video editing methods [12].

2.3 Training-Free Guidance

There exist different approaches for adapting a diffusion or flow model to a specific task. One approach is to fine-tune the model [23, 70], which requires significant amounts of training data and compute. A contrasting approach is to guide the sampling process of an existing model without additional training. This does not require training data but instead a loss function to minimize. In training-free guidance methods, each step in the denoising process uses the current noisy sample to predict the clean sample and compute a per-step correction [5, 21, 44, 53, 64, 67]. Another approach is to optimize the initial noise for the denoising process [3, 7, 49, 50], allowing direct optimization guided by the loss function by computing gradients through all denoising steps. Any loss function

can be used as guidance, e.g. from an image classifier or segmentation model. There are also methods that use losses for 3D consistency for single-image novel view synthesis [7, 66], or solve image inpainting based on geometric correlation to a reference image [37]. In this work, we apply training-free guidance to image editing methods to improve their multi-view consistency.

3 Method

We start with an overview and follow by introducing our consistency loss. We then show how to optimize a set of images for multi-view consistent edits. Finally, we describe how these images can be used to edit a Gaussian splat model.

3.1 Overview

We show an overview of our method in Fig. 2. We propose a method to edit a set of images in a 3D consistent way. Our method can be seen as an extension to the 2D image editing methods like [9, 31, 43], where images are edited based on a text prompt. Applying 2D editing methods on each image independently results in inconsistent edits. Our method guides the denoising process such that the edits are consistent across the views. We use the fact that corresponding points in the edited images should have similar perceptual features. Our edited images can be used directly to edit a Gaussian splat model of the scene.

3.2 Consistency Loss on Correspondences

Our main optimization criterion for 3D-consistent editing is based on the assumption of similar appearances for corresponding points. In particular, (1) if an edit preserves geometry, corresponding points in the unedited images I_1 and I_2 should be edited in a similar way in the edited images I'_1 and I'_2, and (2) if an edit changes geometry, the corresponding points in I'_1 and I'_2 should have similar features. We define the consistency loss $\mathcal{L}_c$ as

$$\mathcal{L}_c = \sum_{(x,y)\in\mathcal{M}} \left(\|I'_1(x) - I'_2(y)\|_1 + \lambda f_{\text{LPIPS}}(I'_1(x), I'_2(y))\right), \tag{1}$$

where f_{LPIPS} is the perceptual loss [71] (with λ set to 2) applied to patches centered around the matches, and $\mathcal{M}$ is the set of corresponding points. To obtain corresponding points, we match images from either the unedited pair (I_1, I_2) or the edited pair (I'_1, I'_2). Using matches between unedited images ensures that the geometric details are preserved, which is beneficial in rigid and near-rigid editing, such as changing textures, materials, colors and weather conditions. On the other hand, for nonrigid editing (like changing objects or adding new objects), using matches between edited images additionally forces consistency in the areas where geometry is changed. For this to work, it is important to use a robust matcher that has an understanding of global context and semantics. We use the robust dense matcher RoMa [16]. More details on hyperparameters can be found in appendix Sec. B.

3.3 Consistency Guided Image Editing

Here we describe how to use our consistency loss to adapt a denoising process to generate a set of multi-view consistent edits. Our method is based on pre-trained image editing models, and we employ different training-free methods to guide the denoising process, as is detailed below.

To use our consistency loss with the diffusion model InstructPix2Pix [9], we employ universal guidance [5], where the sampling is steered towards low values of a loss function $\mathcal{L}(z)$, which in our case is the consistency w.r.t. previous edits. The noise estimation $\epsilon(z_t, t)$ is modified to include a correction based on gradients of the loss function $\mathcal{L}$ to obtain a new noise estimate $\tilde{\epsilon}(z_t, t)$ that guides the latent to low values of the loss. The modified noise is computed as

$$\tilde{\epsilon}(z_t, t) = \epsilon(z_t, t) + \lambda_t \nabla_{z_t} \mathcal{L}(\hat{z}_0(z_t)) \tag{2}$$

where $\hat{z}_0(z_t)$ is a one-step prediction of the denoised latent $\hat{z}_0(z_t) = \frac{1}{\sqrt{\alpha_t}}(z_t - \sqrt{1-\alpha_t}\epsilon(z_t, t))$, and λ_t is the weight of the guidance. For our application, the best results are obtained using $\lambda_t = \lambda \mathbf{1}(t < N_g)$, so that we activate the guidance only for the last $N_g = 700$ steps of the denoising process, after first running the denoising process without any guidance. We also use backward guidance [5], where we optimize a correction $\Delta z_0 = \text{argmin}_\Delta \mathcal{L}(\hat{z}_0 + \Delta)$ for N_b gradient descent steps, and update the noise prediction as $\tilde{\epsilon}'(z_t, t) = \tilde{\epsilon}(z_t, t) - \sqrt{\alpha_t/(1-\alpha_t)}\Delta z_0$.

We can also use our consistency loss with the one-step method pix2pix-Turbo [43], where the denoising process is reduced to a single step. Such a model can be formulated as $x = f(z)$, where z is the starting noise and x is the generated image, and we optimize $\mathcal{L}(f(z))$ with respect to z. This optimization resembles SeedSelect [50] where the starting noise is optimized to constrain the sampling process. This is similar to what has previously been used to improve geometric consistency of diffusion models for single-image novel view synthesis [7].

Our consistency loss can also be used to guide flow matching models. We utilize an efficient optimization strategy proposed by [3] that uses a linear approximation of the denoising trajectories to reformulate the optimization objective as $\mathcal{L}([f(z) - z]_{\text{sg}} + z)$, where $f(z)$ is the output of the flow matching process, z is the initial noise and $[\cdot]_{\text{sg}}$ is the stop-gradient operation. The difference $f(z) - z$ is therefore seen as constant throughout the denoising process. This removes the expensive unrolling related to computing gradients through the denoising process with many steps and enables efficient guiding of flow matching models with our consistency loss. More details regarding this can be found in appendix Sec. B.

Image Ordering An important consideration when editing multiple images of the same scene is how to select which previously edited image to use when computing the consistency loss $\mathcal{L}_c$. We found that editing one image at a time works well in practice, and when we generate a new image, we use the matches between that image and two of the previously edited images. We select the images with the most matching points to the currently edited image. We found that there is no performance improvement when using more than two images (see Sec. 4.6).

Fig. 3: Qualitative comparison using InstructPix2Pix. EditSplat and DGE edit images more drastically. Our method preserves the scene better, as can be seen e.g. in the texture of the grass next to the bench, and in shape of the face, gaze, preserved watch. Our method also produces more consistent edits between different views as can be seen on the bicycle wheels, or on the face or arms of the person.

3.4 3D Gaussian Splat Editing

For editing 3D Gaussian splat models, we start with a trained Gaussian splat model obtained from the unedited views and then resume training using the consistently edited images for 20 epochs. It is also possible to use only a sparse set of (e.g., 3-4) images. In that case a multi-view diffusion method like ViewCrafter [68] can generate novel views of the edited scene, using poses interpolated between the edited views. The edited images and generated views can then be used to train a Gaussian splat model representing the edited scene. Additional details are in appendix Sec. C.

Fig. 4: Renderings from edited 3D Gaussian splat models. For the top scene, per-image edits sometimes cause blurriness, as seen e.g. in the ears which are sharper for ours. EditSplat uses a segmentation mask, leading to the edit being localized only on the face and hair. For DGE, the fidelity to the text prompt is high, but the face geometry has drastically changed. For the bottom scene, our method gives a clearer edit than all other methods, as seen by more visible fog and sharp details preserved on the objects.

4 Experiments

We evaluate our method in two main ways, namely for multi-view consistent image editing and 3D editing of Gaussian splat models. We also provide ablation studies and results when editing just a sparse set of views. In addition to the results presented here we on the project page (https://3d-consistent-editing.github.io/) provide video results showing comparisons against the baseline methods for all the scenes.

4.1 Setup

Implementation Details To evaluate our model, we use the same 8 test scenes as in EditSplat [33], including real-world scenes from IN2N [20], Mip-NeRF360 [6]

and BlendedMVS [63]. For validation we use 2 scenes from Mip-NeRF360 [6], 1 scene from IN2N [20] and 1 self-captured scene containing images of a person's head, comparable to the IN2N "Face" scene. For the test scenes we use a total of 21 different edit prompts and for the validation scenes we use a total of 7 different edit prompts. The exact prompts are provided in appendix Sec. D. The scenes are of varying size, containing 65-350 images per scene. All experiments are done on a single A100 GPU. The complete editing process for the "Face" scene from IN2N takes about 22 minutes for our method when using InstructPix2Pix and about 17 minutes when using pix2pix-Turbo.

Baselines Since our main goal is consistent multi-view editing, we wanted to compare with other methods that also aim to achieve this, therefore we chose EditSplat [33] and DGE [12] since they are state-of-the-art methods for 3D editing that claim multi-view consistency during the 2D editing process. Another recent method that claims improved multi-view consistency is InterGSEdit [58], but it does not have code available. Both EditSplat and DGE use the image editing model InstructPix2Pix [9], so for fairness we also evaluate our method using it. We also test our method on the one-step model pix2pix-Turbo [43] and flow-matching based model FLUX.1 [31]. InstructPix2Pix and pix2pix-Turbo have limited ability in changing geometry, therefore we use near-rigid edits for these methods and optimize the consistency loss based on matches between unedited images. When we evaluate image consistency, we extract the images *prior* to updating the Gaussian splats, which for EditSplat is the output from their Multi-view Fusion Guidance (MFG) and for DGE after one iteration of their Multi-View Consistent Editing. We also compute metrics for the unedited images to get upper bounds on the consistency. As a baseline, we also edit per image without any consideration of multi-view consistency.

Metrics There are two aspects of the generated images we evaluate, namely multi-view consistency and fidelity to the text prompt. For multi-view consistency, we evaluate MEt3R [2] that compares DINO [11] embeddings of matches obtained via Dust3R [55]. The rendering metrics PSNR, SSIM and LPIPS are computed by training a Gaussian splat model [27] from scratch using only the edited images. The edited images are split into training and test sets, and the metrics are computed on the edited images set aside as test images. Inconsistent views yield blurrier renderings and thus lower reconstruction metrics. We also evaluate how well the edited images match the given text prompts by comparing CLIP [46] embeddings of images and text. Given a reference image I, an edited image I' and text descriptions[3] of the unedited and edited images T and T', respectively, CLIPdir measures the cosine distance between $\text{CLIP}(I') - \text{CLIP}(I)$ and $\text{CLIP}(T') - \text{CLIP}(T)$, CLIPsim is the cosine distance between $\text{CLIP}(I')$

[3] Similar to previous work [33], we have three prompts per scene: a description of the unedited image, a description of the edited image and the edit prompt, for instance "a photo of a park", "a photo of a Namibian desert", and "Turn the ground into a Namibian desert", respectively.

Method	MEt3R↓	PSNR↑	SSIM↑	LPIPS↓	CLIPdir↑	CLIPsim↑	CLIPimage↑
Unedited	0.183	27.36	0.815	0.190	-	0.199	1.0
Based on InstructPix2Pix							
EditSplat	0.329	20.20	0.641	0.389	0.159	0.252	0.768
DGE	0.224	21.58	0.705	0.256	**0.173**	**0.257**	0.757
Per image	0.243	21.19	0.679	0.271	0.153	0.249	0.813
Ours	**0.212**	**23.46**	**0.716**	**0.247**	0.152	0.249	**0.817**
Based on pix2pix-Turbo							
Per image	0.291	18.29	0.564	0.429	**0.124**	**0.254**	0.766
Ours	**0.226**	**24.73**	**0.696**	**0.339**	0.116	0.252	**0.768**

Table 1: Evaluation for multi-view consistent editing. Our method significantly improves the consistency with both InstructPix2Pix [9] and pix2pix-Turbo [43]. For DGE and EditSplat we note decreased consistency metrics, but also that CLIPsim and CLIPdir are improved and CLIPimage decreases which indicates larger edits to the images than our method and the per image baseline.

and CLIP(T'). Both measure how well the edited image aligns with the desired edit. Finally, CLIPimage is the cosine distance between CLIP(I) and CLIP(I'), measuring similarity between the input and the edited image. There is a trade-off between CLIPimage and CLIPdir/CLIPsim since increasing how closely the edited image corresponds to the prompt reduces its similarity to the input image.

4.2 Image Consistency Evaluation

Here we evaluate the multi-view consistency of the edits. For fair comparison, we extract the images *prior* to training Gaussian splats, in contrast to the results of Gaussian splat editing in Sec. 4.3. The image editing consistency is presented in Table 1. For the consistency metrics, we can see that our method obtains the best values of all methods, showing that our edited images are more multi-view consistent than both the per image baseline as well as for the two methods we compare to. We note that for both EditSplat and DGE, CLIPimage is lower than for InstructPix2Pix per image, indicating that the multi-view edited images aggregate edits that are stronger than the baseline edit method. For the text metrics there is no clear overall improvement of our method compared to the per-image edits, which is expected since we optimize for multi-view consistency only. We show qualitative results in Fig. 3. Comparing our method to the per-image edits, we see that our edits have significantly improved multi-view consistency. While EditSplat or DGE are more consistent than the per-image edits, some inconsistencies still remain, e.g. on the bicycle wheels or on the face of the person.

4.3 Gaussian Splat Editing

As an application of our multi-view consistent image editing, we show how to use the edited images to update 3D Gaussian splats. We show the results in

	CLIPdir↑	CLIPsim↑	CLIPimage↑	MEt3R↓
Based on InstructPix2Pix				
EditSplat	0.123	0.238	0.832	**0.215**
DGE	**0.146**	**0.242**	0.752	0.242
Per image	0.126	0.239	**0.833**	0.216
Ours	0.121	0.237	0.830	**0.215**
Based on pix2pix-Turbo				
Per image	**0.067**	**0.224**	**0.845**	**0.210**
Ours	**0.067**	0.223	0.823	**0.210**

Table 2: Evaluation of 3D Gaussian splat editing. The renderings from the edited Gaussian splat models are similarly accurate with respect to the text prompts as when using the per-image edits. We note that DGE gets the highest values of CLIPdir and CLIPsim but the lowest for CLIPimage indicating more substantial edits of the scene. The Met3R score here is evaluated on the rendered views, and measures how consistent the renders are across different viewing directions.

Table 2. We note that CLIPsim and CLIPdir are comparable for our method and EditSplat, both achieving similar values as the per-image edits. DGE shows improved values for CLIPdir and CLIPsim but worse on CLIPimage which indicates stronger edits that are less similar to the unedited images. Note that CLIPsim and CLIPdir measure how well the images fit the text prompt, and only loosely measure other relevant aspects such as sharpness of the images. The increased MEt3R score for DGE shows that the renderings for this method are less consistent than the others.

We show qualitative examples in Fig. 4. We notice that updating Gaussian splats with independently edited images can cause blurry results compared to using multi-view consistent editing. This can be seen e.g. at the ear of the person or the fog over the grass which is only present in a few of the independently edited images and only weakly seen in the Gaussian splat model. These cases are handled better by our consistency guided denoising. We also note the DGE can result in edits where the scene content is modified significantly, e.g. the shape of the face is modified so that it no longer resembles the person in the input image. We provide several more examples in the videos on the project page, where we see that our method can produce clear edits with preserved details.

To show that our method is not limited to InstructPix2Pix, we also use it with pix2pix-Turbo [43] and present results in Fig. 5. The per-image edits are inconsistent in several places, e.g. colors around the eyes. Edits using our correspondance guidance are significantly more consistent which can also be seen for the Gaussian splat renderings.

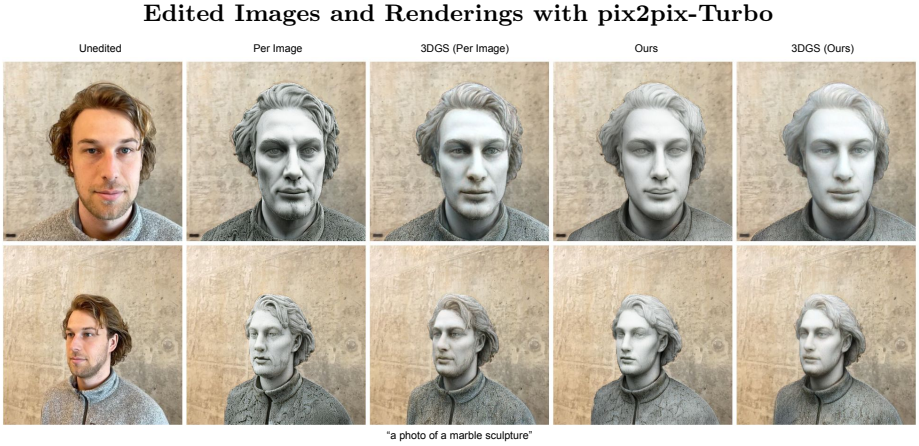

Fig. 5: Multi-view editing using the image editing method pix2pix-Turbo. Per-image edits can be inconsistent and there is a loss of detail when editing a 3D Gaussian splat model using these inconsistent images. In contrast, our edits are more consistent and the details are more accurately recovered by the Gaussian splat renderings.

	PSNR↑	SSIM↑	LPIPS↓	CLIPsim↑	MEt3R↓
Per Image	22.20	0.687	0.290	0.259	0.404
Ours	**24.49**	**0.725**	**0.281**	**0.263**	**0.364**

Table 3: Evaluation of final renderings for the sparse setting where 3-4 views are given as input. Our method combined with pix2pix-Turbo [43] is used to edit the images, and the multi-view diffusion model ViewCrafter [68] is used to generate additional edited views that are then used to train a 3DGS representation. We evaluate on four scenes (a total of 6 prompts). The metrics are computed w.r.t. a few of the generated images used as a test set. The consistency is significantly higher for our method compared to using independent edits per image.

4.4 Sparse Editing

As another application of our multi-view consistent editing, we also investigate editing just a sparse set of 3-4 images and interpolating with the multi-view diffusion model ViewCrafter [68] to generate additional edited views that can then be used to train a Gaussian splat model representing the edited scene (see Sec. 3.4), instead of editing 40-125 images. Similarly to the image consistency evaluation, we use a few of the views as test views to evaluate the Gaussian splat model with. We show the results in Fig. 6 and Table 3. We note that our sparsely edited views are more consistent, which makes the rendering significantly more consistent and sharp.

Ablation		PSNR↑	SSIM↑	LPIPS↓	MEt3R↓	CLIPdir↑	CLIPsim↑	CLIPimage↑	Time (s)
Matched images	1	22.97	0.670	0.294	0.394	0.152	0.254	0.856	**28**
	2	**23.15**	**0.675**	**0.290**	**0.385**	**0.153**	**0.255**	**0.857**	36
	3	23.13	0.673	0.295	0.392	0.152	0.254	0.856	44
Backward steps	0	21.71	0.654	0.294	0.396	0.164	0.259	**0.868**	**16**
	3	23.15	0.675	**0.290**	0.385	**0.153**	**0.255**	0.857	36
	6	**23.51**	**0.678**	0.304	**0.382**	0.145	0.251	0.850	57
LPIPS patch loss weight (λ)	0	22.27	0.674	0.300	0.403	**0.165**	**0.260**	0.856	**21**
	2	**23.15**	**0.675**	**0.290**	**0.385**	0.153	0.255	**0.857**	36
	5	22.56	0.674	0.313	0.409	0.146	0.254	0.855	36

Table 4: Ablation of our method using InstructPix2Pix. We evaluate how many previously edited images to use for the consistency loss, the number of optimization steps for backward guidance, and the weight for the LPIPS patch loss.

4.5 Flexibility of Consistency Guidance

To showcase the flexibility of our consistency-based guidance approach, we present the qualitative results using the recently published method FLUX.1 [31].

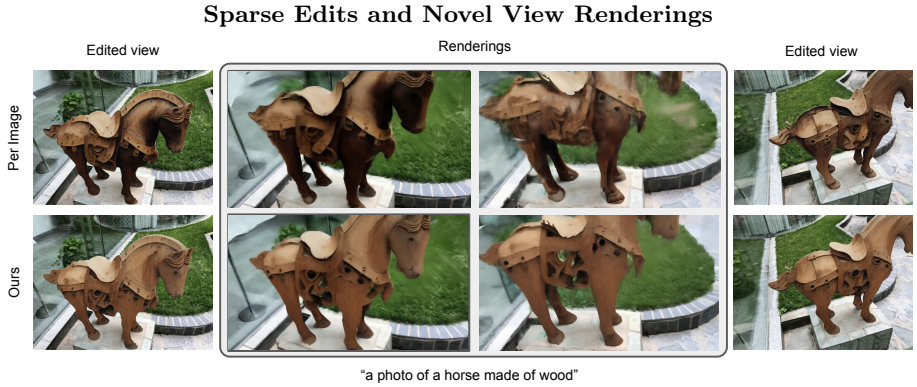

Fig. 6: Example of editing a few sparse views and using the multi-view diffusion model ViewCrafter [68] to interpolate. We note that inconsistencies in the edited views lead to blurry renderings and view inconsistencies in the interpolated images. When using our edited images the interpolated views are both more consistent and contain sharper details.

It is a flow matching model for image generation and editing that demonstrates impressive quality and performs a wide range of edits, including non-rigid edits such as object insertion or replacement (which InstructPix2Pix and pix2pix-Turbo cannot reliably do). It retains persistentcharacter/object features through the edits. However, consistency in 3D pose, color, and texture remains limited, leading to inconsistencies across views. The improvements using our guidance are shown in Fig. 7. We use the same optimization of the consistency

Fig. 7: Qualitative comparison using FLUX.1. The loss is computed between each image and an anchor image. On the left we show rigid and near-rigid edits, and on the right we show non-rigid edits. The multi-view edits are more consistent with our approach, e.g. in the following details: (a) makeup paint colors, face pose, gaze; (b) snow pattern on the table and ground, middle reflective part of the table; (c) hat and (d) plate colors.

loss in (1), where for rigid and near-rigid edits (left part of Fig. 7) we use matches between unedited images, and for non-rigid edits (right part of Fig. 7) we use matches between edited images. Note that our guidance works well with both small and wide baselines and for a diverse range of prompts.

4.6 Ablation Studies

We present ablation studies in Table 4 on important factors affecting performance and run-time. The ablations are provided using InstructPix2Pix [9]. We found that the number of images used to compute the matching loss (1) saturated when using more than two images. We also found that using 3 backward steps in universal guidance provided the best trade-off between consistency and edit time, with 6 steps giving similar consistency metrics but at an increased computational cost. Additionally the ablation shows that including the patch-based LPIPS loss gives improved consistency, while increasing its weight above $\lambda = 2$ did not lead to additional improvements.

5 Conclusion

We present a flexible method for 3D consistent image editing. It guides the denoising process of a pre-trained single-image editing model by optimizing a

correspondence-based objective so that matching points in different views are edited in a coherent way. We can improve the multi-view consistency for both rigid or near-rigid edits as well as non-rigid edits depending on which correspondences are used. Furthermore, we show that we can improve the multi-view consistency using a range of different image editing methods, namely diffusion models, one-step models and flow matching models. We experimentally show that the consistency of the edited images is better compared to existing methods. We also demonstrate the possibility of editing a Gaussian splat model directly using both sparse and dense views.

Acknowledgments

This work was supported by the Wallenberg AI, Autonomous Systems, and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. Computational resources were provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS) at Chalmers Centre for Computational Science and Engineering (C3SE), partially funded by the Swedish Research Council under grant agreement no. 2022-06725, and by the Berzelius resource, provided by the Knut and Alice Wallenberg Foundation at the National Supercomputer Centre.

References

1. Alimohammadi, A., Mikaeili, A., Nag, S., Hassanpour, N., Tagliasacchi, A., Mahdavi-Amiri, A.: Cora: Correspondence-aware image editing using few step diffusion. In: Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers. SIGGRAPH Conference Papers '25, Association for Computing Machinery, New York, NY, USA (2025). https://doi.org/10.1145/3721238.3730650, https://doi.org/10.1145/3721238.3730650 4
2. Asim, M., Wewer, C., Wimmer, T., Schiele, B., Lenssen, J.E.: Met3r: Measuring multi-view consistency in generated images. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 6034–6044 (2025) 9
3. Baek, S., Dong, E., Namazifard, S., Matthews, M.J., Yi, K.M.: Sonic: Spectral optimization of noise for inpainting with consistency (2025), https://arxiv.org/abs/2511.19985 4, 6, 21
4. Bai, Q., Ouyang, H., Xu, Y., Wang, Q., Yang, C., Cheng, K.L., Shen, Y., Chen, Q.: Edicho: Consistent image editing in the wild. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 15277–15287 (October 2025) 4
5. Bansal, A., Chu, H.M., Schwarzschild, A., Sengupta, R., Goldblum, M., Geiping, J., Goldstein, T.: Universal guidance for diffusion models. In: The Twelfth International Conference on Learning Representations (2024) 4, 6
6. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. Proceedings of the Computer Vision and Pattern Recognition Conference (2022) 8, 9

7. Bengtson, J., Nilsson, D., Kahl, F.: Geometric consistency refinement for single image novel view synthesis via test-time adaptation of diffusion models. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 6399–6408 (2025) 4, 5, 6
8. Bengtson, J., Nilsson, D., Lin, C.T., Büsching, M., Kahl, F.: Adjustable visual appearance for generalizable novel view synthesis. In: Wallraven, C., Liu, C.L., Ross, A. (eds.) Pattern Recognition and Artificial Intelligence. pp. 157–171. Springer Nature Singapore, Singapore (2025) 4
9. Brooks, T., Holynski, A., Efros, A.A.: Instructpix2pix: Learning to follow image editing instructions. In: Proceedings of the Computer Vision and Pattern Recognition Conference (2023) 4, 5, 6, 9, 10, 14, 21
10. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. NeurIPS **33**, 1877–1901 (2020) 4
11. Caron, M., Touvron, H., Misra, I., J'egou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. 2021 IEEE/CVF International Conference on Computer Vision (ICCV) pp. 9630–9640 (2021), https://api.semanticscholar.org/CorpusID:233444273 9
12. Chen, M., Laina, I., Vedaldi, A.: Dge: Direct gaussian 3d editing by consistent multi-view editing. In: European Conference on Computer Vision. pp. 74–92. Springer (2024) 2, 4, 9, 23
13. Chen, Y., Chen, Z., Zhang, C., Wang, F., Yang, X., Wang, Y., Cai, Z., Yang, L., Liu, H., Lin, G.: Gaussianeditor: Swift and controllable 3d editing with gaussian splatting (2023) 4
14. Deutch, G., Gal, R., Garibi, D., Patashnik, O., Cohen-Or, D.: Turboedit: Text-based image editing using few-step diffusion models. In: SIGGRAPH Asia 2024 Conference Papers. SA '24, Association for Computing Machinery, New York, NY, USA (2024). https://doi.org/10.1145/3680528.3687612, https://doi.org/10.1145/3680528.3687612 4
15. 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) 4
16. Edstedt, J., Sun, Q., Bökman, G., Wadenbäck, M., Felsberg, M.: Roma: Robust dense feature matching. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19790–19800 (2024) 5, 23
17. Fujiwara, H., Mukuta, Y., Harada, T.: Improved 3D Scene Stylization via Text-Guided Generative Image Editing with Region-Based Control. In: Christie, M., Han, P.H., Lin, S.S., Pietroni, N., Schneider, T., Tsai, H.R., Wang, Y.S., Zhang, E. (eds.) Pacific Graphics Conference Papers, Posters, and Demos. The Eurographics Association (2025). https://doi.org/10.2312/pg.20251277 4
18. Gong, Y., Zhu, Z., Zhang, M.: Instantedit: Text-guided few-step image editing with piecewise rectified flow (2025), https://arxiv.org/abs/2508.06033 4
19. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks. Communications of the ACM **63**(11), 139–144 (2020) 3, 4
20. Haque, A., Tancik, M., Efros, A.A., Holynski, A., Kanazawa, A.: Instruct-nerf2nerf: Editing 3d scenes with instructions. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 19740–19750 (2023) 2, 4, 8, 9, 23
21. He, Y., Murata, N., Lai, C.H., Takida, Y., Uesaka, T., Kim, D., Liao, W.H., Mitsufuji, Y., Kolter, J.Z., Salakhutdinov, R., Ermon, S.: Manifold preserving guided

diffusion. In: The Twelfth International Conference on Learning Representations (2024), https://openreview.net/forum?id=o3BxOLoxm1 4
22. Hertz, A., Mokady, R., Tenenbaum, J., Aberman, K., Pritch, Y., Cohen-Or, D.: Prompt-to-prompt image editing with cross attention control. arXiv preprint arXiv:2208.01626 (2022) 4
23. 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 4
24. Hu, V.T., Zhang, W., Tang, M., Mettes, P., Zhao, D., Snoek, C.: Latent space editing in transformer-based flow matching. In: Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence. AAAI'24/IAAI'24/EAAI'24, AAAI Press (2024). https://doi.org/10.1609/aaai.v38i3.27998, https://doi.org/10.1609/aaai.v38i3.27998 4
25. Kamata, H., Sakuma, Y., Hayakawa, A., Ishii, M., Narihira, T.: Instruct 3d-to-3d: Text instruction guided 3d-to-3d conversion. arXiv preprint arXiv:2303.15780 (2023) 4
26. Kawar, B., Zada, S., Lang, O., Tov, O., Chang, H., Dekel, T., Mosseri, I., Irani, M.: Imagic: Text-based real image editing with diffusion models. In: Conference on Computer Vision and Pattern Recognition 2023 (2023) 4
27. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Trans. Graph. **42**(4), 139–1 (2023) 1, 9, 23
28. Koh, E., Hyun, S., Lee, M., Chung, J., Seo, K., Heo, J.P.: Diffusion feature field for text-based 3d editing with gaussian splatting. In: The Thirty-ninth Annual Conference on Neural Information Processing Systems (2025) 4
29. Kulikov, V., Kleiner, M., Huberman-Spiegelglas, I., Michaeli, T.: Flowedit: Inversion-free text-based editing using pre-trained flow models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 19721–19730 (2025) 4
30. Kwon, G., Ye, J.C.: Diffusion-based image translation using disentangled style and content representation (2023), https://arxiv.org/abs/2209.15264 4
31. Labs, B.F., Batifol, S., Blattmann, A., Boesel, F., Consul, S., Diagne, C., Dockhorn, T., English, J., English, Z., Esser, P., Kulal, S., Lacey, K., Levi, Y., Li, C., Lorenz, D., Müller, J., Podell, D., Rombach, R., Saini, H., Sauer, A., Smith, L.: Flux.1 kontext: Flow matching for in-context image generation and editing in latent space (2025), https://arxiv.org/abs/2506.15742 4, 5, 9, 13, 21
32. Lai, Z., Sun, K., Wang, F.Y., Sagar, D., Ding, E.: Instantportrait: One-step portrait editing via diffusion multi-objective distillation. In: The Thirteenth International Conference on Learning Representations (2025), https://openreview.net/forum?id=ZkFMe3OPfw 4
33. Lee, D.I., Park, H., Seo, J., Park, E., Park, H., Baek, H.D., Shin, S., Kim, S., Kim, S.: Editsplat: Multi-view fusion and attention-guided optimization for view-consistent 3d scene editing with 3d gaussian splatting. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 11135–11145 (2025) 2, 4, 8, 9, 23, 24
34. Li, L., Huang, Z., Feng, H., Zhuang, G., Chen, R., Guo, C., Sheng, L.: Voxhammer: Training-free precise and coherent 3d editing in native 3d space. arXiv preprint arXiv:2508.19247 (2025) 4

35. Li, Y., Dou, Y., Shi, Y., Lei, Y., Chen, X., Zhang, Y., Zhou, P., Ni, B.: Focaldreamer: text-driven 3d editing via focal-fusion assembly. In: Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence. AAAI'24/IAAI'24/EAAI'24, AAAI Press (2024). https://doi.org/10.1609/aaai.v38i4.28113, https://doi.org/10.1609/aaai.v38i4.28113 4
36. Lipman, Y., Chen, R.T.Q., Ben-Hamu, H., Nickel, M., Le, M.: Flow matching for generative modeling. In: The Eleventh International Conference on Learning Representations (2023), https://openreview.net/forum?id=PqvMRDCJT9t 21
37. Liu, K.H., Yang, C.K., Chen, M.H., Liu, Y.L., Lin, Y.Y.: Corrfill: Enhancing faithfulness in reference-based inpainting with correspondence guidance in diffusion models. In: 2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 1618–1627. IEEE (2025) 5
38. Liu, X., Gong, C., Liu, Q.: Flow straight and fast: Learning to generate and transfer data with rectified flow. In: The Eleventh International Conference on Learning Representations (2023), https://openreview.net/forum?id=XVjTT1nw5z 21
39. Mei, K., Delbracio, M., Talebi, H., Tu, Z., Patel, V.M., Milanfar, P.: Codi: Conditional diffusion distillation for higher-fidelity and faster image generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2024) 4
40. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM **65**(1), 99–106 (2021) 1
41. Mirzaei, A., Aumentado-Armstrong, T., Brubaker, M.A., Kelly, J., Levinshtein, A., Derpanis, K.G., Gilitschenski, I.: Watch your steps: Local image and scene editing by text instructions. In: ECCV (2024) 4
42. Nguyen, T.T., Nguyen, Q., Nguyen, K., Tran, A., Pham, C.: Swiftedit: Lightning fast text-guided image editing via one-step diffusion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (June 2025) 4
43. Parmar, G., Park, T., Narasimhan, S., Zhu, J.Y.: One-step image translation with text-to-image models. arXiv preprint arXiv:2403.12036 (2024) 4, 5, 6, 9, 10, 11, 12, 21
44. Patel, M., Wen, S., Metaxas, D.N., Yang, Y.: Steering rectified flow models in the vector field for controlled image generation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2025) 4
45. Poole, B., Jain, A., Barron, J.T., Mildenhall, B.: Dreamfusion: Text-to-3d using 2d diffusion. In: ICLR (2023) 4
46. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PmLR (2021) 9
47. 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) 3, 4
48. Saharia, C., Chan, W., Chang, H., Lee, C., Ho, J., Salimans, T., Fleet, D., Norouzi, M.: Palette: Image-to-image diffusion models. In: ACM SIGGRAPH 2022

Conference Proceedings. SIGGRAPH '22, Association for Computing Machinery, New York, NY, USA (2022). https://doi.org/10.1145/3528233.3530757, https://doi.org/10.1145/3528233.3530757 4
49. Samuel, D., Ben-Ari, R., Darshan, N., Maron, H., Chechik, G.: Norm-guided latent space exploration for text-to-image generation (2023) 4
50. Samuel, D., Ben-Ari, R., Raviv, S., Darshan, N., Chechik, G.: Generating images of rare concepts using pre-trained diffusion models (2024) 4, 6
51. Sella, E., Fiebelman, G., Hedman, P., Averbuch-Elor, H.: Vox-e: Text-guided voxel editing of 3d objects. In: ICCV (2023) 4
52. Song, L., Cao, L., Gu, J., Jiang, Y., Yuan, J., Tang, H.: Efficient-nerf2nerf: Streamlining text-driven 3d editing with multiview correspondence-enhanced diffusion models. arXiv preprint arXiv:2312.08563 (2023) 4
53. Sun, Z., Yang, Z., Jin, Y., Chi, H., Xu, K., Xu, K., Chen, L., Jiang, H., Song, Y., Gai, K., MU, Y.: RectifID: Personalizing rectified flow with anchored classifier guidance. In: The Thirty-eighth Annual Conference on Neural Information Processing Systems (2024), https://openreview.net/forum?id=KKrj1vCQaG 4
54. Wang, B., Dutt, N.S., Mitra, N.J.: Proteusnerf: Fast lightweight nerf editing using 3d-aware image context. Proc. ACM Comput. Graph. Interact. Tech. **7**(1) (may 2024). https://doi.org/10.1145/3651290, https://doi.org/10.1145/3651290 4
55. Wang, S., Leroy, V., Cabon, Y., Chidlovskii, B., Revaud, J.: Dust3r: Geometric 3d vision made easy. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20697–20709 (2024) 9
56. Wang, T.C., Liu, M.Y., Zhu, J.Y., Tao, A., Kautz, J., Catanzaro, B.: High-resolution image synthesis and semantic manipulation with conditional gans. In: Proceedings of the Computer Vision and Pattern Recognition Conference (2018) 4
57. Wang, Y., Yi, X., Wu, Z., Zhao, N., Chen, L., Zhang, H.: View-consistent 3d editing with gaussian splatting. In: Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29 – October 4, 2024, Proceedings, Part XXXV. p. 404–420. Springer-Verlag, Berlin, Heidelberg (2024). https://doi.org/10.1007/978-3-031-72761-0_23, https://doi.org/10.1007/978-3-031-72761-0_23 4
58. Wen, M., Wu, S., Wang, K., Liang, D.: Intergsedit: Interactive 3d gaussian splatting editing with 3d geometry-consistent attention prior. Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2025) 4, 9
59. Wu, C., Li, J., Zhou, J., Lin, J., Gao, K., Yan, K., ming Yin, S., Bai, S., Xu, X., Chen, Y., Chen, Y., Tang, Z., Zhang, Z., Wang, Z., Yang, A., Yu, B., Cheng, C., Liu, D., Li, D., Zhang, H., Meng, H., Wei, H., Ni, J., Chen, K., Cao, K., Peng, L., Qu, L., Wu, M., Wang, P., Yu, S., Wen, T., Feng, W., Xu, X., Wang, Y., Zhang, Y., Zhu, Y., Wu, Y., Cai, Y., Liu, Z.: Qwen-image technical report (2025), https://arxiv.org/abs/2508.02324 4
60. Wu, J., Bian, J.W., Li, X., Wang, G., Reid, I., Torr, P., Prisacariu, V.: GaussCtrl: Multi-View Consistent Text-Driven 3D Gaussian Splatting Editing. ECCV (2024) 4
61. Xia, R., Tang, Y., Zhou, P.: Towards scalable and consistent 3d editing (2025), https://arxiv.org/abs/2510.02994 4
62. Xu, S., Huang, Y., Pan, J., Ma, Z., Chai, J.: Inversion-free image editing with natural language (2024) 4
63. 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. Proceedings of the Computer Vision and Pattern Recognition Conference (2020) 9

64. Ye, H., Lin, H., Han, J., Xu, M., Liu, S., Liang, Y., Ma, J., Zou, J., Ermon, S.: TFG: Unified training-free guidance for diffusion models. In: The Thirty-eighth Annual Conference on Neural Information Processing Systems (2024), https://openreview.net/forum?id=N8YbGX98vc 4
65. Ye, J., Xie, S., Zhao, R., Wang, Z., Yan, H., Zu, W., Ma, L., Zhu, J.: Nano3d: A training-free approach for efficient 3d editing without masks (2025), https://arxiv.org/abs/2510.15019 4
66. YOU, M., Zhu, Z., LIU, H., Hou, J.: NVS-solver: Video diffusion model as zero-shot novel view synthesizer. In: The Thirteenth International Conference on Learning Representations (2025), https://openreview.net/forum?id=zDJf7fvdid 5
67. Yu, J., Wang, Y., Zhao, C., Ghanem, B., Zhang, J.: Freedom: Training-free energy-guided conditional diffusion model. Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2023) 4
68. Yu, W., Xing, J., Yuan, L., Hu, W., Li, X., Huang, Z., Gao, X., Wong, T.T., Shan, Y., Tian, Y.: Viewcrafter: Taming video diffusion models for high-fidelity novel view synthesis. arXiv preprint arXiv:2409.02048 (2024) 7, 12, 13, 23
69. Zeng, T., Ding, Z., Chen, Z., Zhang, X., Li, L., Tu, Z.: C3Editor: Achieving controllable consistency in 2d model for 3d editing. In: ICCV Wild3D Workshop (2025) 4
70. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models (2023) 4
71. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the Computer Vision and Pattern Recognition Conference (2018) 5, 23
72. Zhao, X., Guan, J., Fan, C., Xu, D., Lin, Y., Pan, H., Feng, P.: Fastdrag: Manipulate anything in one step. In: The Thirty-eighth Annual Conference on Neural Information Processing Systems (2024) 4
73. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: ICCV (2017) 4
74. Zhu, J.Y., Zhang, R., Pathak, D., Darrell, T., Efros, A.A., Wang, O., Shechtman, E.: Toward multimodal image-to-image translation. NIPS (2017) 4
75. Zhuang, J., Wang, C., Liu, L., Lin, L., Li, G.: Dreameditor: Text-driven 3d scene editing with neural fields. In: SIGGRAPH (2023) 4

Appendix

Overview

In this appendix we show video results and additional qualitative results of our consistent multi-view image editing (Sec. A), technical details related to both our multi-view editing (Sec. B) and the 3D Gaussian splat editing (Sec. C), and details of the dataset and prompts that were used (Sec. D).

We also provide video results on our project page (https://3d-consistent-editing.github.io/) which presents a qualitative comparison across all applications. For full transparency and fair comparison, we also show the results from *all* 21 scene-prompt pairs of the test set we use when evaluating the different methods.

A Additional Qualitative Results

We provide multiple additional examples of our multi-view consistent image editing in this section.

Multi-View Image Editing We show results for InstructPix2Pix [9] in Fig. 8, where we note that, e.g., the eyes and the surrounding regions are more consistent for our method and the details on the saddle are better preserved across views when using our method. For pix2pix-Turbo [43], we see in Fig. 10 that the overall colors and backgrounds are more consistent for our method and in Fig. 11 we note that the texture on the bear is more consistent with our editing. Finally, we show additional results for FLUX.1 [31] in Fig. 12, where it can be seen that our method improves color and texture consistency in the fur of the bear and the grass in front of the bicycle. We also observe consistent shape and appearance for the unicorn statue and improved geometry for the changed hairstyle.

3DGS Editing In "main.mp4" we provide additional video results showing Gaussian splat renderings for both the dense and sparse view setups. Additionally in "all_scenes.mp4" we show our method and the methods we compare to for all 21 scene-prompts pairs in our test set, using both InstructPix2Pix [9] and pix2pix-Turbo [43].

Sparse 3DGS Editing We show an additional example of the sparse editing in Fig. 13, where we see that the renderings of the Gaussian splat model is more consistent using our edited images compared to the per image edits.

B Multi-View Image Editing

Image Editing Methods We test our method together with two different diffusion based image editing methods, InstructPix2Pix [9] and the single-step model pix2pix-Turbo [43], as well as the flow-matching based method FLUX.1 [31]. For *InstructPix2Pix* we use 20 denoising steps and guidance scales $s_T = 7.5$ and $s_I = 1.5$, which are the same choices as in DGE and EditSplat. The single-step model *pix2pix-Turbo* is trained to take a Canny edge map and generate an image based on this and a given text prompt. So the editing process with pix2pix-Turbo is to first convert the image to an Canny edge map that is then used to generate an edited image, which leads to the appearance of the whole image being changed even if an local edit is desired. To address this we utilize a segmentation mask to ensure that only the desired object is changed and that the rest of the image remains unchanged. For the flow-matching based model FLUX.1 [31] we use 28 denoising steps. We also take advantage of the fact that denoising trajectories for flow-matching based models are approximately linear, making it possible to avoid computing gradients through all the denoising steps, as described in [3]. This assumption of linear denoising trajectories should in theory be a perfect approximation [36,38], but in practice an approximation error still exits. We find that while using this approximation we still are able to stably converge to initial seeds that improve consistency.

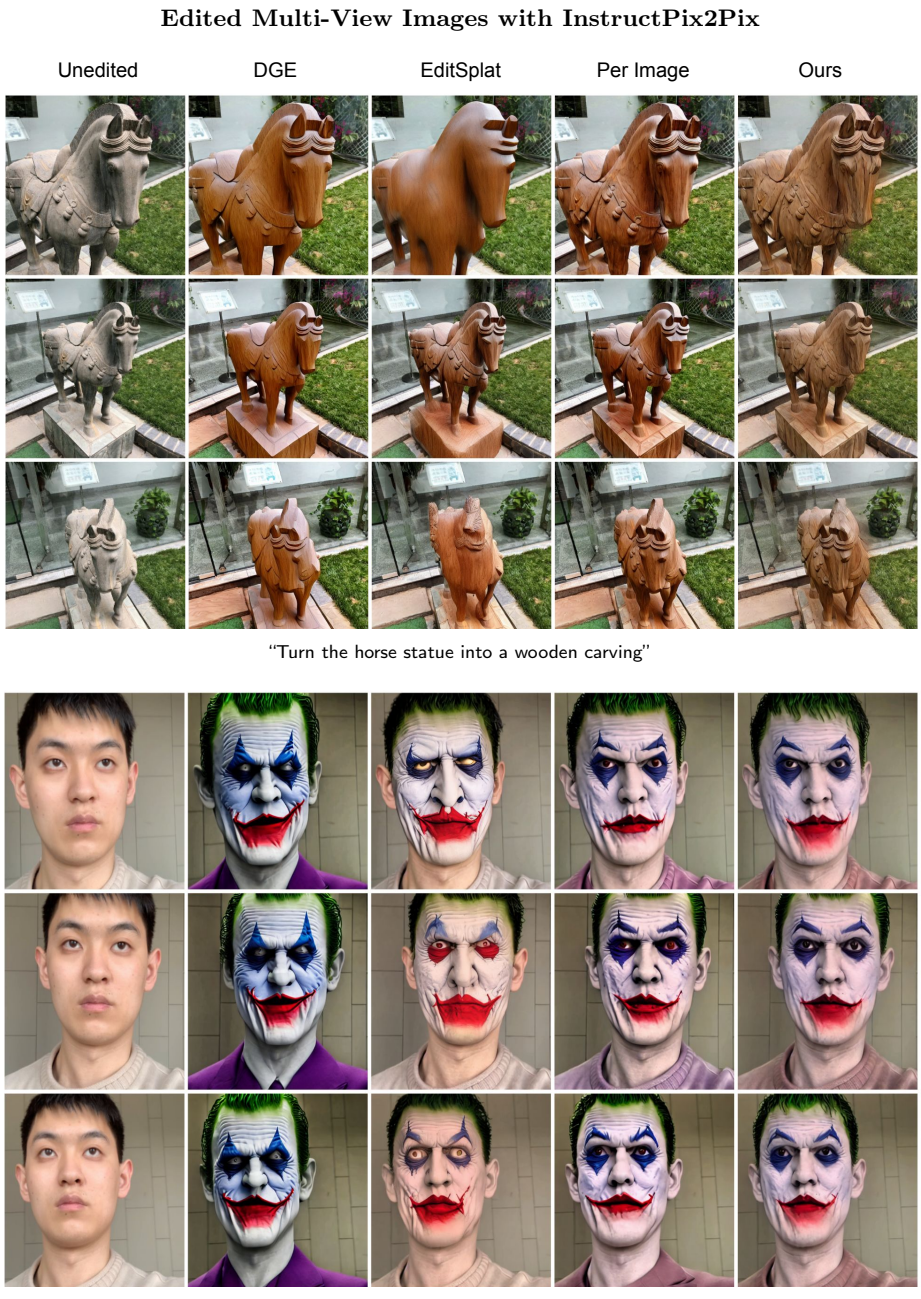

Fig. 8: Additional qualitative example of multi-view consistent image editing using methods all based on the image editing method InstructPix2Pix. We note that our method is able to preserve both details better than the other methods, as seen in e.g. on the saddle and the eyes and the area around the eyes.

Image Matching Details We use the robust dense matcher RoMa [16], which is a robust dense matcher that also returns certainties. We only use matches with a certainty over 0.05 and include a maximum of 50 000 matches. For the LPIPS patch loss we use the perceptual loss [71] applied to patches centered around the matches. We randomly select 1 500 correspondences out of the current matches and extract patches of size 64×64 centered at these positions.

Sparse Editing We show an overview of the sparse editing in Fig. 9. For the sparse editing we edit 3-4 images and then utilize the multi-view diffusion model ViewCrafter [68] to interpolate between these views to generate 50-75 additional views of the scene. These additional edited views can then be used to train a Gaussian model representing the edited scene. In this setup there is no existing Gaussian model of the unedited scene available since we only have a few images available of the scene. We thus train a Gaussian model from scratch based on the generated edited views, using 5 000 iterations. Training from scratch instead of editing a Gaussian splat model makes this setup more sensitive to inconsistencies in the edited views. Since a new Gaussian model needs to be generated from the edited images, and we do not have any prior geometry from the Gaussian splat model of the unedited images, inconsistencies in the images can easily lead to incorrect geometry or strong view-dependent effects. Another reason is that inconsistencies in the edited sparse views can lead to significant appearance change in the additional views generated by ViewCrafter, which again lead to difficulties of training a Gaussian splat model.

C Gaussian Splat Editing

In this section, we describe the differences in the training of the edited Gaussian splat models between our method and the methods we compare to. When we test EditSplat [33] and DGE [12], we use their provided code without any changes. We refer to these papers for full details and provide a brief overview here.

Ours. We use the standard Gaussian splatting training procedure from the original paper [27], except that we limit the training to 20 epochs (800-2500 iterations depending on the number of images in the scene), since we resume the training from a Gaussian splat model of the unedited images. Different from the original Gaussian splat training settings, we used a loss function where the L1 and LPIPS rendering losses have equal weights, similar to earlier editing methods [20, 33].

EditSplat. Similarly to ours, EditSplat also first edits the images and then uses those images to update the Gaussians. There are several techniques used to edit the Gaussians. Based on the prompt and edit model, they compute attention weights over the images, indicating regions where the prompt has a large influence and where the Gaussians should change. The Gaussians are trimmed so that a fixed fraction of the Gaussians with large attention weights are excluded from the editing. The motivation is that retraining excessive source attributes, as indicated by the regions with large attention weights, is detrimental for the optimization to the edited images.

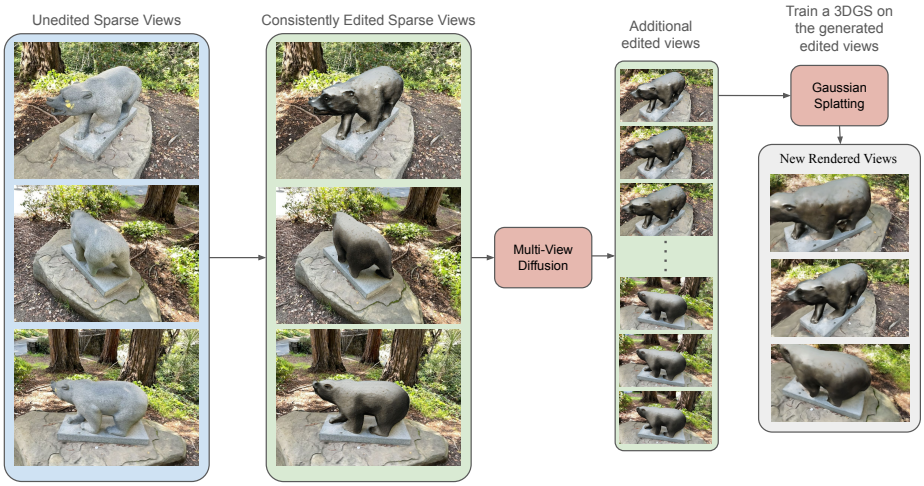

Fig. 9: Overview of the sparse editing pipeline, that from a few sparse views generates a Gaussian splat model of the edited scene. Our consistent editing method is used to edit the given sparse views and a Multi-View Diffusion network is then used to generate additional edited images of the same scene. A Gaussian splat model can then be trained on all these views, and we can render novel views using the Gaussian splat model.

DGE. DGE also initially performs a multi-view consistent editing process that is used to update an existing Gaussian model of the unedited images. But instead of just performing one update step, they perform an iterative refinement where they render images from the updated Gaussians and repeat the editing process, re-updating the 3D model. This process is repeated for a total of 3 iterations, which was the default value in their released code.

D Scene-Prompt Pairs

We show all scene-prompt pairs used in the test set in Table 5. Most of the pairs are the same as used in prior work [33]. We additionally show all pairs in the validation set in Table 6. Note that different scenes are used in the validation and test sets.

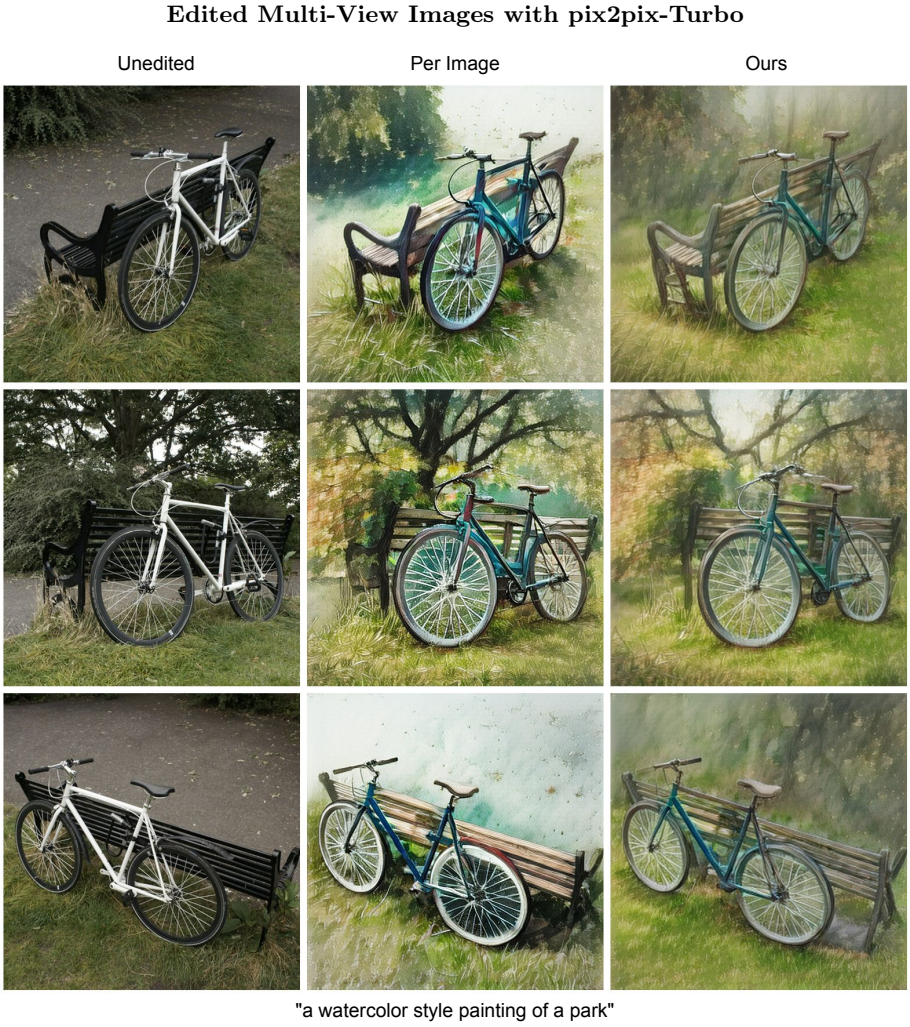

Fig. 10: Additional qualitative example of multi-view consistent image editing with pix2pix-Turbo. We note that our method is able to give a more consistent overall color, and e.g. on the bicycle frame, and that the background is more consistent.

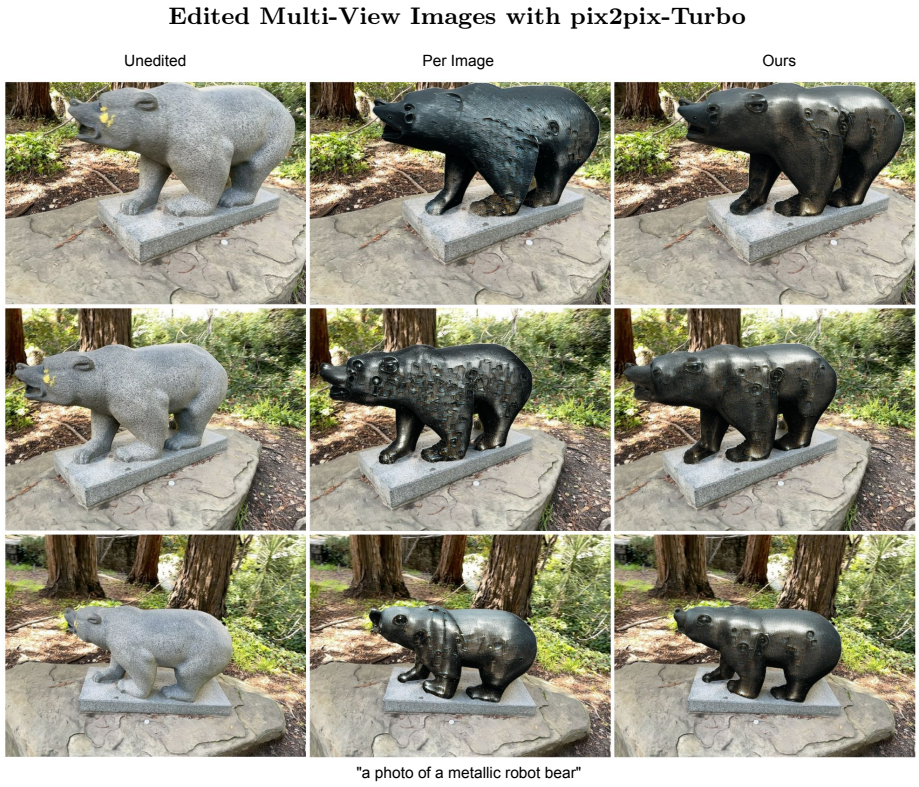

Fig. 11: Additional qualitative example of multi-view consistent image editing with pix2pix-Turbo. We note that with our method the texture on the bear is consistent across the views and is able to capture realistic lighting reflections.

Fig. 12: Additional qualitative comparison using FLUX.1. The loss is computed between each image and an anchor image. On the left we show rigid and near-rigid edits, and on the right we show non-rigid edits. The multi-view edits are more consistent with our approach, e.g. in the following details: (a) fur color and pattern, texture of the platform; (b) appearance of grass and pavement; (c) texture and shape of the statue, and (d) hair length and overall style.

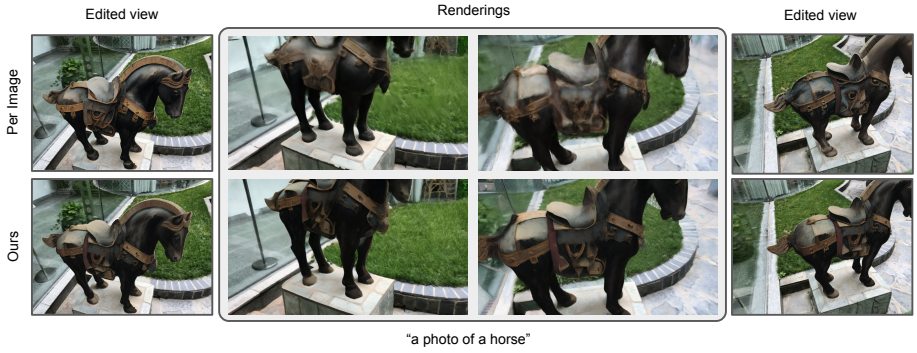

Fig. 13: Additional example of editing a few sparse views and using ViewCrafter to interpolate. We note that the edited views are more consistent for our method compared to per image edits, which can be seen in e.g. the saddlebag in both the edited images and the Gaussian splat renderings.

Scene	Source Description	Target Description	Editing Instruction
Face	a photo of a face of a man	a photo of a face of a clown	Make his face look like a clown
Face	a photo of a face of a man	a photo of a marble sculpture	Make his face resemble that of a marble sculpture
Face	a photo of a face of a man	a Vincent Van Gogh painting	Make him look like a Vincent Van Gogh painting
Fangzhou	a photo of a face of a man	a photo of the face of the Joker	Turn him into the Joker
Fangzhou	a photo of a face of a man	a photo of the face of Steve Jobs	Turn him into Steve Jobs
Fangzhou	a photo of a face of a man	a photo of a face of a man with Maasai face paint	Give him Maasai face paint
Garden	a photo of an outdoor garden	a photo of a foggy outdoor garden	Make it foggy
Garden	a photo of an outdoor garden	a photo of a snowy outdoor garden	Make it snowy
Bicycle	a photo of a park	a photo of a Namibian desert	Turn the ground into a Namibian desert
Bicycle	a photo of a park	a watercolor style painting of a park	Make the entire scene look as if it's painted in watercolor style
Bear	a photo of a bear statue	a photo of a metallic robot bear	Turn the bear statue into a metallic robot
Bear	a photo of a bear statue	a photo of a panda	Turn the bear statue into a panda
Person	a photo of a person	a photo of a person in Minecraft	Turn him into a Minecraft character
Person	a photo of a person	a photo of a person wearing clothes with a pineapple pattern	Make the person wear clothes with a pineapple pattern
Person	a photo of a person	a photo of a person wearing a suit	Make the person wear a suit
Bonsai	a photo of a bonsai	a photo of a snowy bonsai	Make the bonsai snowy
Bonsai	a photo of a bonsai	a photo of a bonsai with yellow petals	Make the bonsai have yellow petals
Bonsai	a photo of a bonsai	a photo of a bonsai made of paper	Change the bonsai to look like it's made of paper, folded into intricate origami shapes
Stone Horse	a photo of a horse statue	a photo of a horse made of wood	Turn the horse statue into a wooden carving
Stone Horse	a photo of a horse statue	a photo of a horse made of jade	Turn the stone horse into a jade carving
Stone Horse	a photo of a horse statue	a photo of a zebra	Make the stone horse a zebra

Table 5: All 21 scene-prompt pairs used as a test set for evaluation.

Scene	Source Description	Target Description	Editing Instruction
Kitchen	a photo of a lego excavator	a Vincent Van Gogh painting of a lego excavator	Turn it into a Vincent Van Gogh painting
MaleFace	a photo of a face of a man	a photo of a face of a man with a bandana	Give him a bandana
MaleFace	a photo of a face of a man	a photo of a face of a very old man	Make him look very old
Campsite	a photo of a campsite	a photo of a campsite in the Sahara desert	Make the ground look like the Sahara desert
Campsite	a photo of a campsite	a photo of a campsite with snow on the ground	Make the ground snowy
Vasedeck	a photo of a vase with flowers	a photo of a vase with some red flowers	Make some of the flowers red
Vasedeck	a photo of a vase with flowers	a photo of a vase with yellow and blue flowers	Make the flowers yellow and blue

Table 6: All 7 scene–prompt pairs used as a validation set when choosing hyperparameters and performing ablation studies for our method.