3D-Consistent Multi-View Editing by Diffusion Guidance

Josef Bengtson[1] David Nilsson[1] Dong In Lee[2] Fredrik Kahl[1]

[1]Chalmers University of Technology, [2]Korea University

{bjosef,david.nilsson,fredrik.kahl}@chalmers.se, dilee99@korea.ac.kr

Abstract

Recent advancements in diffusion 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 diffusion framework that enforces multi-view consistency during the image editing process. The key assumption is that corresponding points in the unedited images should undergo similar transformations after editing. To achieve this, we introduce a consistency loss that guides the diffusion sampling 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/

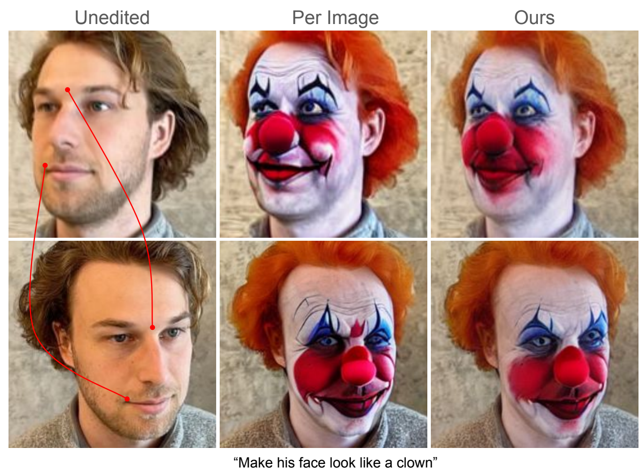

Figure 1. Image editing methods applied independently to multi-view images often produce inconsistent edits across views, as shown here where corresponding regions differ between edits. Our method improves multi-view consistency by guiding the diffusion process under the assumption that matching points in the unedited images, as shown by the red lines, should be edited similarly.

1. Introduction

Text-based image editing has become increasingly powerful by utilizing generative models, allowing for the 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, it remains difficult to use such methods to edit multiple images of the same scene, or to edit 3D models, in a consistent manner.

Given the recent work on photorealistic rendering of 3D models using NeRFs [35] or Gaussian splatting [25], a natural question is whether we can edit such 3D models using image editing methods. If we edit the training views used to train 3D models independently using image editing methods 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 shape of eyebrows or painting on the cheeks are inconsistent for different views. For 3D editing this has the implication that the edited training views do not depict the same 3D scene, making editing difficult. This motivates modifying the image editing method so that multiple views can be edited in a multi-view consistent way.

Early recent work on 3D editing such as Instruct-NeRF2NeRF [18] attempted to solve the multi-view issue by iteratively alternating between editing all images and updating the 3D model until convergence. Other recent works such as EditSplat [31] and DGE [11] modify the generation of the edited images using multi-view constraints or modified attention weights to make them more consistent in an attempt to directly update the 3D model and bypass the costly iterative process of regenerating edited images and updating the 3D model multiple times.

We present a method to directly guide the diffusion process of image editing methods so that the edited images are multi-view consistent, as shown in Fig. 1. Our assumption is that if we match points between two unedited images, then these points should be edited in a similar way. We introduce a simple consistency loss to measure multi-view consistency based on this assumption. During the image generation, we enforce this by using training-free guidance which modifies the diffusion sampling 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 the Gaussian splat training using the edited images using the standard Gaussian splat training process [25]. We show that the edited images are sufficiently multi-view consistent to get an edited Gaussian splat model with sharp and realistic details. Our method also enables consistent editing of sparse views, and we show that it works with two different image editing methods.

In summary our main contributions are:
- We introduce a training-free method to guide diffusion models to generate multi-view consistent images based on the assumption that matching points in the unedited images should be edited in a similar way.
- We show that the generated images can be used to directly refine a Gaussian splat model and accurately reconstruct fine details.
- We experimentally compare our method to recent work and show that we can improve the multi-view consistency of the generated images prior to the Gaussian splat refinement, and we obtain edited Gaussian splat models that are as faithful to the given text-prompts with clear details. This is shown by extensive video comparisons against the baseline methods included on the project page.

2. Related Work

2.1. 2D Image Editing

The development of powerful generative models [17, 42] has led to advancements in the capabilities of image editors, initially mainly through Generative Adversarial Networks (GANs) [17, 51, 67, 68] and more recently through models based on diffusion [24, 28, 43, 54] and flow matching [22, 27, 29]. InstructPix2Pix (IP2P) [7] finetunes Stable Diffusion [42] using image pairs generated by prompt-to-prompt (P2P) [20] by following instructions generated by GPT-3 [8], enabling instruction-based editing. One limiting factor has been long edit times because of a high number of diffusion steps, so there has been work to enable editing using few-step [1, 13, 16, 34, 57] and one-step [30, 37, 38, 66] diffusion. Our work focus on how these methods can be made 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 enable directly editing a 3D representation [6, 32, 56, 60], but such data is expensive to acquire, limiting training to single objects and synthetic scenes. Another popular approach is to leverage image editing methods for 3D editing, but these 2D methods have no consistency guarantees across views. Some works use score distillation sampling (SDS) [40] to update the 3D representation [23, 33, 46, 69]. Instruct-NeRF2NeRF [18] introduced the iterative dataset update, which leverages a 3D model's consistency to refine edits by alternately updating 2D views and the 3D representation, which has been widely adopted in subsequent works [12, 14, 36, 49, 52].

The iterative process is compute-intense and several approaches strive to improve consistency and therefore reduce reliance on iterative dataset updates. Improvements range from incorporating correspondence regularization into the diffusion process [47], using geometric information from the 3D representation to guide the editing [31, 55] and allowing the attention process to consolidate information across views [11, 52, 53, 55]. Several of these recent approaches [11, 52] rely on an iterative refinement on top of the proposed multi-view consistent editing. In contrast our method gives further consistency improvements and does not require iteratively updating the images. Our method can handle sparse view editing, while several of the baseline methods are limited to dense views, since they either require geometric information from an existing 3D representation of the scene [31, 55] or requires rendering a smooth camera trajectory to use ideas from video editing methods [11].

2.3. Diffusion Model Guidance

There exists different approaches for adapting a diffusion model for a specific task. One approach is to fine-tune the model [21, 64], which requires significant amounts of training data and compute. A contrasting approach is to guide the sampling process of an existing diffusion model without additional training, by introducing a loss function to be minimized. This is done in training-free guidance methods [3, 19, 39, 48, 59, 62] that for each step in the diffusion process use the current noisy sample to predict the clean sample to compute a per-step correction. Another approach is to optimize the initial noise for the diffusion process [5, 44, 45], which allows for a direct optimization guided by the used loss function by computing gradients through all diffusion steps. Guidance can leverage diverse loss functions, including 2D image models (e.g. classifiers) and losses designed for 3D consistency [5, 61]. In this work we use training-free guidance applied on image editing methods to improve their multi-view consistency.

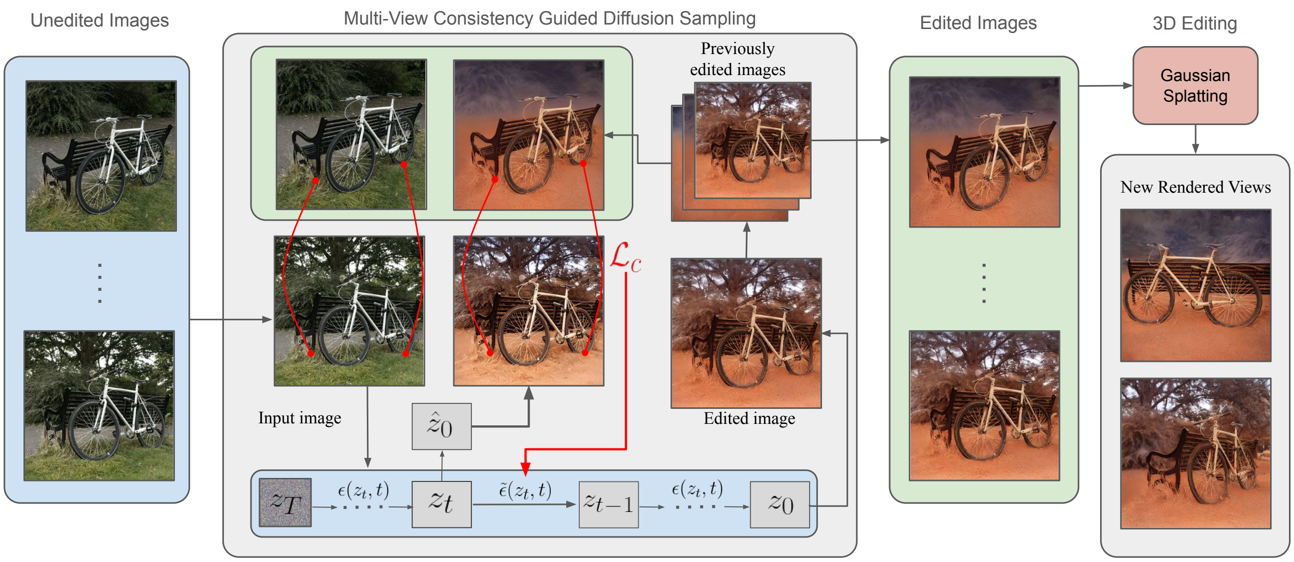

Figure 2. Overview of our method. Given a set of input images, each view is edited sequentially by guiding the diffusion process based on the previously edited images. The guidance is based on the assumption that matching points in the *unedited* images should be edited similarly. During the diffusion process the noise estimate $\epsilon(z_t, t)$ is modified according to a consistency loss $\mathcal{L}_c$ resulting in multi-view consistent edits. In turn, these edited images can be used to update a Gaussian splat model.

3. Method

In this section we present our method. We start with an overview, followed by our consistency loss. We then show how to optimize a set of images to get multi-view consistent edited images and 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. We base our method on 2D image editing methods [7, 38] where images are edited based on a text prompt. Applying such 2D editing independently per image results in inconsistent edits. Our method guides the diffusion process such that the edits for the different images are consistent. The criteria used for the diffusion guidance is based on the assumption that matching points in the unedited images should be edited similarly and have similar appearance also in the edited images. Our edited images can be used directly to edit a Gaussian splat model of the scene.

3.2. Consistency Loss

Our main optimization criterion for image consistent editing is based on the assumption that if we find matching points in the unedited images I_1 and I_2, then those points should be edited in a similar way in the edited images I'_1 and I'_2. We define the consistency loss $\mathcal{L}_c$ between two edited images 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 $\mathcal{M} = \mathcal{M}(I_1, I_2)$ is the set of matches computed between the *unedited* images I_1 and I_2 using an image matcher, and the perceptual loss f_{LPIPS} [65] is computed by extracting 64×64 sized patches around $I'_1(x)$ and $I'_2(y)$. We used $\lambda = 2$. We used the RoMa [15] matcher which gives dense matches with certainty measures. We only used matches with a certainty over 0.05 and included a maximum of 50000 matches. While the assumption that matching points should be similarly edited does not hold for edits with changes in the scene geometry, we found this to work well in practice also for small modifications in geometry.

3.3. Consistency Guided Image Editing

In this section we describe how to use our consistency loss to adapt a diffusion sampling process to generate a set of multi-view consistent edited images. Our method is based on diffusion models for image editing, and we employ two different training-free methods to guide the diffusion process. For InstructPix2Pix [7] we use universal guidance [3] and for the single-step model pix2pix-turbo [38] we use a version of SeedSelect [45], similar to what has previously been used to improve geometric consistency of diffusion models for single-image novel view synthesis [5].

In universal guidance [3], the sampling is steered towards low values of a specific loss function $\mathcal{L}$, which in our case will be the consistency to the previous edits. The noise estimation $\epsilon(z_t, t)$ is modified to include a correction based on the 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

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

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 were obtained using $\lambda_t = \lambda \mathbf{1}(t < N_g)$, so that we activated the guidance only for the last $N_g = 700$ steps of the diffusion process, after first running the diffusion process without any guidance. We also used backward guidance [3] where we optimize a correction $\Delta z_0 = \arg\min_\Delta \mathcal{L}_c(\hat{z}_0 + \Delta)$ 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 also used our consistency loss for the one-step method pix2pix-turbo [38], where the diffusion process is reduced to a single step. Such models can be formulated as $x = f(z)$ where z is the starting noise and x is the generated image, and we optimize $\mathcal{L}_c(f(z))$ with respect to z. This optimization resembles SeedSelect [45] where the starting noise was optimized to constrain the sampling process.

An important consideration when editing multiple images of the same scene is how to select which previously edited images to use when computing the consistency loss $\mathcal{L}_c$. We found that editing one image at a time worked well in practice, and that when we generated a new image we used the matches between that image and two of the previously edited images. We selected the images with the most matching points to the image currently edited. We found that there was no performance improvement when using more than two images, as can be seen in Table 3.

3.4. 3D Gaussian Splat Editing

For editing of 3D Gaussian splat models, we started with a trained Gaussian splat model obtained using the unedited views, and then resumed the training using the edited images for 20 epochs. Additional details can be found in the appendix C.

4. Experiments

In this section we present experimental results. 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 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 [31], including real-world scenes from IN2N [18], Mip-NeRF360 [4] and Blended-MVS [58]. For validation we use 2 scenes from Mip-NeRF360 [4], 1 scene from IN2N [18] 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 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. Additional details can be found in appendix B.

Baselines. We compare our method to two recent methods for 3D Gaussian splat editing, namely EditSplat [31] and DGE [11]. Both these methods use the image editing model InstructPix2Pix [7], so for fairness we also evaluate our method using it. We also test our method on the one-step model pix2pix-turbo [38]. 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 main 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], which compares DINO [10] embeddings of matches obtained via Dust3R [50]. The rendering metrics PSNR, SSIM and LPIPS are computed by training a Gaussian splat model [25] 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 the CLIP [41] embeddings of images and text. Given a reference image I, an edited image I' and text descriptions[1] of the unedited and edited images T and T' respectively, then CLIPdir is the cosine distance between $\text{CLIP}(I') - \text{CLIP}(I)$ and $\text{CLIP}(T') - \text{CLIP}(T)$, CLIPsim is the cosine distance between $\text{CLIP}(I')$ and $\text{CLIP}(T')$, that both measure how well the edited im-

[1]Similar to previous work [31], 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.

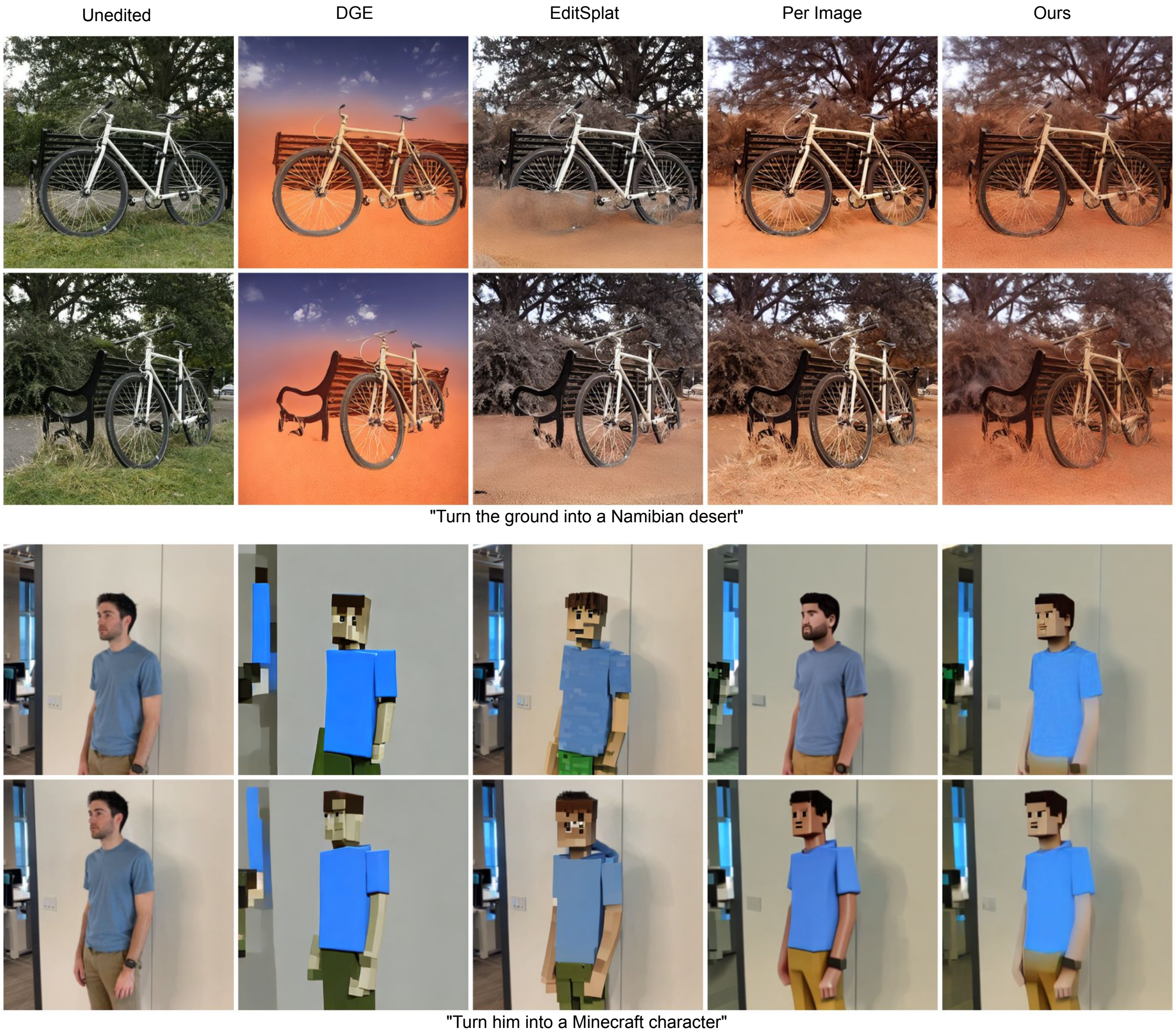

Figure 3. Qualitative example of multi-view consistent image editing using methods all based on the image editing method IP2P. We note that EditSplat and DGE edit the image more than our method, which is more similar to the unedited images, as can be seen e.g. in the shape of the face or the texture of the grass next to the bench. We also see that our method produces more consistent edits for the different views as can be seen in the face or arms of the person, and also for the bicycle wheels.

age aligns with the desired edit. Finally CLIPimage is the cosine distance between CLIP(I) and CLIP(I'), measuring similarity of the edited image with the input image. There is often a trade-off between CLIPimage and CLIPdir/CLIPsim since increasing how closely the edited image corresponds to the prompt will reduce it's similarity to the input image.

4.2. Image Consistency Evaluation

In this section we compare how multi-view consistent the edited images are. For fair comparison, we compare to the other methods by extracting 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 the edited image has a lower value of CLIPimage than IP2P per image, which

Method	MEt3R ↓	PSNR ↑	SSIM ↑	LPIPS ↓	CLIPdir ↑	CLIPsim ↑	CLIPimage ↑
Unedited	0.183	27.36	0.815	0.190	-	0.199	1.0
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
IP2P per image	0.243	21.19	0.679	0.271	0.153	0.249	0.813
IP2P+Ours	**0.212**	**23.46**	**0.716**	**0.247**	0.152	0.249	**0.817**
pix2pix-turbo per image	0.291	18.29	0.564	0.429	**0.124**	**0.254**	0.766
pix2pix-turbo+Ours	**0.226**	**24.73**	**0.696**	**0.339**	0.116	0.252	**0.768**

Table 1. Evaluation for multi-view consistent editing. We see that our method significantly improves the consistency with both IP2P [7] and pix2pix-turbo [38]. 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.

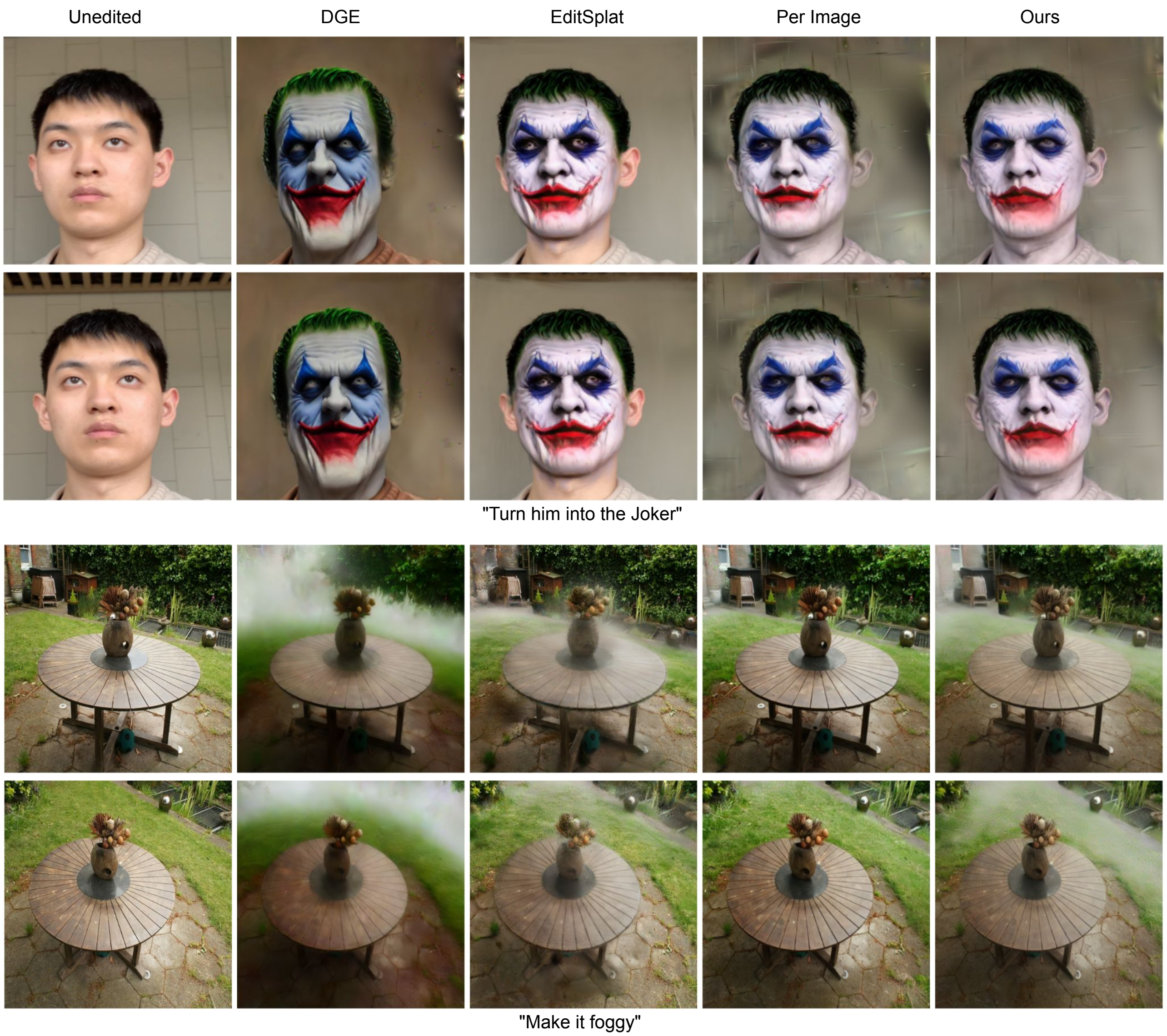

Figure 4. We show renderings from edited 3D Gaussian splat models. For the face we note that the per image edits sometimes cause blurriness, as seen e.g. in the ears which are sharper for ours. EditSplat uses a segmentation mask, leading to the edit being localized only to the face and hair of the person. For DGE we see that the fidelity to the text prompt is high, but the face has drastically altered appearance. For the garden our method gives a more clear edit than per image edits and EditSplat, as seen by more visible fog and clearer details preserved on the object on the table.

Method	CLIPdir ↑	CLIPsim ↑	CLIPimage ↑	MEt3R ↓
EditSplat	0.123	0.238	0.832	**0.215**
DGE	**0.146**	**0.242**	0.752	0.242
IP2P per image	0.126	0.239	**0.833**	0.216
IP2P+Ours	0.121	0.237	0.830	**0.215**
pix2pix-turbo per image	**0.067**	**0.224**	**0.845**	**0.210**
pix2pix-turbo+Ours	**0.067**	0.223	0.823	**0.210**

Table 2. Evaluation of 3D Gaussian splat editing. We see that 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.

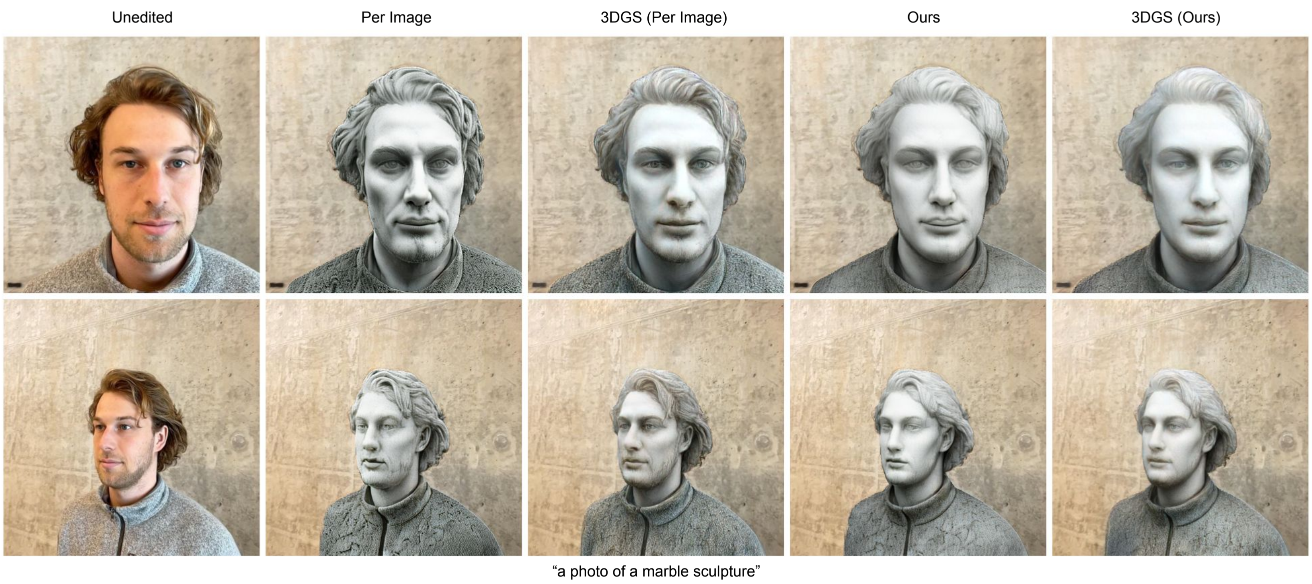

Figure 5. We show our multi-view consistent editing using the image editing method pix2pix-turbo. We see that the per image edits can be inconsistent and that there is a loss of detail when editing a 3D Gaussian splat model using these inconsistent images. In contrast our edits are more consistent and the details are more accurately recovered by the Gaussian splat renderings.

indicates that the multi-view edited images aggregate edits that are stronger than the baseline edit method. For the text metrics there is no clear 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 can see that our edits are significantly more multi-view consistent. Comparing to EditSplat or DGE we see that while they are more consistent than the per-image edits there are inconsistencies in the images, e.g. the face of the person or the wheels of the bicycle.

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 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

Ablation		PSNR ↑	SSIM ↑	LPIPS ↓	MEt3R ↓	CLIPdir ↑	CLIPsim ↑	CLIPimage ↑	Edit 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

Table 3. Ablation of our method using InstructPix2Pix. We test how many of the previously edited images we should use to compute the consistency loss, and also how many many optimization steps to use to compute the backward guidance.

	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 4. Evaluation of final renderings for the sparse setting where 3-4 views are given as input. Our method combined with pix2pix-turbo is used to edit the images, and ViewCrafter 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), and the renderings metrics are computed with respect to a few of the generated images used as a test set. We see that the consistency is significantly higher for our method compared to using independent edits per image.

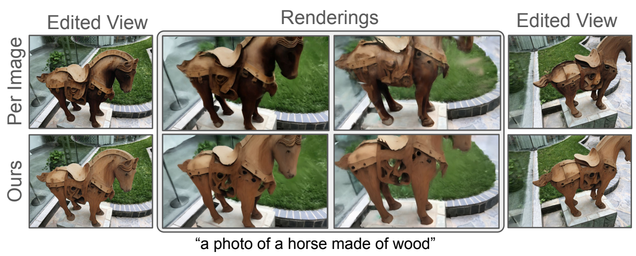

Figure 6. Example of editing a few sparse views and using ViewCrafter to interpolate. We note that inconsistencies in the edited views lead to blurry renderings and view inconsistencies in the interpolated images. When using our edited images the interpolated views are both more consistent and contain sharper details.

in the Gaussian splat model. These cases are handled better by our consistency guided diffusion sampling. 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 show an example of our method using pix2pix-turbo [38] in Fig. 5. We see that the per-image edits are inconsistent in several places, e.g. colors around the eyes, while our edits are significantly more consistent which can also be seen for the Gaussian splat renderings.

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 instead of editing 40-125 images. After editing the images we use ViewCrafter [63] to generate novel views of the edited scene, using poses that are interpolated between the poses of the edited views. The edited images and the generated novel views are then used to train a Gaussian Splat model. 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 Table 4 and Fig. 6. We note that our sparsely edited views are more consistent, which makes the rendering significantly more consistent and sharp.

4.5. Ablation Studies

We provide ablation studies in Table 3 of important aspects for the performance and run-time of our method. The ablations are provided using InstructPix2Pix [7]. 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.

5. Conclusion

We presented a method for multi-view consistent editing of a set of images. Our method was built on the assumption that matching points in the unedited images should be edited in a similar way. This criteria was used to guide the diffusion sampling of pre-trained image editing methods capable of editing single images. We experimentally showed that the consistency of the edited images was higher than for existing work, and that it was possible to directly edit a Gaussian splat model 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] Amirhossein Alimohammadi, Aryan Mikaeili, Sauradip Nag, Negar Hassanpour, Andrea Tagliasacchi, and Ali Mahdavi-Amiri. 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*, New York, NY, USA, 2025. Association for Computing Machinery. 2

[2] Mohammad Asim, Christopher Wewer, Thomas Wimmer, Bernt Schiele, and Jan Eric Lenssen. Met3r: Measuring multi-view consistency in generated images. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 6034–6044, 2025. 4

[3] Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Roni Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024. 2, 3, 4

[4] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2022. 4

[5] Josef Bengtson, David Nilsson, and Fredrik Kahl. 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*, pages 6399–6408, 2025. 2, 3

[6] Josef Bengtson, David Nilsson, Che-Tsung Lin, Marcel Büsching, and Fredrik Kahl. Adjustable visual appearance for generalizable novel view synthesis. In *Pattern Recognition and Artificial Intelligence*, pages 157–171, Singapore, 2025. Springer Nature Singapore. 2

[7] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2023. 2, 3, 4, 6, 8, 12

[8] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *NeurIPS*, 33:1877–1901, 2020. 2

[9] John Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, 1986. 12

[10] Mathilde Caron, Hugo Touvron, Ishan Misra, Herv'e J'egou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9630–9640, 2021. 4

[11] Minghao Chen, Iro Laina, and Andrea Vedaldi. Dge: Direct gaussian 3d editing by consistent multi-view editing. In *European Conference on Computer Vision*, pages 74–92. Springer, 2024. 1, 2, 4, 12

[12] 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, 2023. 2

[13] Gilad Deutch, Rinon Gal, Daniel Garibi, Or Patashnik, and Daniel Cohen-Or. Turboedit: Text-based image editing using few-step diffusion models. In *SIGGRAPH Asia 2024 Conference Papers*, New York, NY, USA, 2024. Association for Computing Machinery. 2

[14] Jiahua Dong and Yu-Xiong Wang. Vica-nerf: View-consistency-aware 3d editing of neural radiance fields. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 2

[15] Johan Edstedt, Qiyu Sun, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg. Roma: Robust dense feature matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19790–19800, 2024. 3

[16] Yiming Gong, Zhen Zhu, and Minjia Zhang. Instantedit: Text-guided few-step image editing with piecewise rectified flow, 2025. 2

[17] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 2

[18] 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. 1, 2, 4, 12

[19] Yutong He, Naoki Murata, Chieh-Hsin Lai, Yuhta Takida, Toshimitsu Uesaka, Dongjun Kim, Wei-Hsiang Liao, Yuki Mitsufuji, J Zico Kolter, Ruslan Salakhutdinov, and Stefano Ermon. Manifold preserving guided diffusion. In *The Twelfth International Conference on Learning Representations*, 2024. 2

[20] 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

[21] 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. In *International Conference on Learning Representations*, 2022. 2

[22] Vincent Tao Hu, Wei Zhang, Meng Tang, Pascal Mettes, Deli Zhao, and Cees Snoek. 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 Press, 2024. 2

[23] Hiromichi Kamata, Yuiko Sakuma, Akio Hayakawa, Masato Ishii, and Takuya Narihira. Instruct 3d-to-3d: Text instruction guided 3d-to-3d conversion. *arXiv preprint arXiv:2303.15780*, 2023. 2

[24] 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 *Conference on Computer Vision and Pattern Recognition 2023*, 2023. 2

[25] 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. 1, 2, 4, 12

[26] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4015–4026, 2023. 12

[27] Vladimir Kulikov, Matan Kleiner, Inbar Huberman-Spiegelglas, and Tomer Michaeli. Flowedit: Inversion-free text-based editing using pre-trained flow models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19721–19730, 2025. 2

[28] Gihyun Kwon and Jong Chul Ye. Diffusion-based image translation using disentangled style and content representation, 2023. 2

[29] Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, Kyle Lacey, Yam Levi, Cheng Li, Dominik Lorenz, Jonas Müller, Dustin Podell, Robin Rombach, Harry Saini, Axel Sauer, and Luke Smith. Flux.1 kontext: Flow matching for in-context image generation and editing in latent space, 2025. 2

[30] Zhixin Lai, Keqiang Sun, Fu-Yun Wang, Dhritiman Sagar, and Erli Ding. Instantportrait: One-step portrait editing via diffusion multi-objective distillation. In *The Thirteenth International Conference on Learning Representations*, 2025. 2

[31] Dong In Lee, Hyeongcheol Park, Jiyoung Seo, Eunbyung Park, Hyunje Park, Ha Dam Baek, Sangheon Shin, Sangmin Kim, and Sangpil Kim. 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*, pages 11135–11145, 2025. 1, 2, 4, 12, 13

[32] Lin Li, Zehuan Huang, Haoran Feng, Gengxiong Zhuang, Rui Chen, Chunchao Guo, and Lu Sheng. Voxhammer: Training-free precise and coherent 3d editing in native 3d space. *arXiv preprint arXiv:2508.19247*, 2025. 2

[33] Yuhan Li, Yishun Dou, Yue Shi, Yu Lei, Xuanhong Chen, Yi Zhang, Peng Zhou, and Bingbing Ni. Focaldreamer: text-driven 3d editing via focal-fusion assembly. In *Proceedings of the 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 Press, 2024. 2

[34] Kangfu Mei, Mauricio Delbracio, Hossein Talebi, Zhengzhong Tu, Vishal M Patel, and Peyman Milanfar. 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. 2

[35] 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. 1

[36] Ashkan Mirzaei, Tristan Aumentado-Armstrong, Marcus A. Brubaker, Jonathan Kelly, Alex Levinshtein, Konstantinos G. Derpanis, and Igor Gilitschenski. Watch your steps: Local image and scene editing by text instructions. In *ECCV*, 2024. 2

[37] Trong-Tung Nguyen, Quang Nguyen, Khoi Nguyen, Anh Tran, and Cuong Pham. Swiftedit: Lightning fast text-guided image editing via one-step diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. 2

[38] Gaurav Parmar, Taesung Park, Srinivasa Narasimhan, and Jun-Yan Zhu. One-step image translation with text-to-image models. *arXiv preprint arXiv:2403.12036*, 2024. 2, 3, 4, 6, 8, 12

[39] Maitreya Patel, Song Wen, Dimitris N. Metaxas, and Yezhou Yang. Steering rectified flow models in the vector field for controlled image generation. *arXiv preprint arXiv:2412.00100*, 2024. 2

[40] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *ICLR*, 2023. 2

[41] 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. 4

[42] 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

[43] Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 Conference Proceedings*, New York, NY, USA, 2022. Association for Computing Machinery. 2

[44] Dvir Samuel, Rami Ben-Ari, Nir Darshan, Haggai Maron, and Gal Chechik. Norm-guided latent space exploration for text-to-image generation. 2023. 2

[45] Dvir Samuel, Rami Ben-Ari, Simon Raviv, Nir Darshan, and

Gal Chechik. Generating images of rare concepts using pre-trained diffusion models. 2024. 2, 3, 4

[46] Etai Sella, Gal Fiebelman, Peter Hedman, and Hadar Averbuch-Elor. Vox-e: Text-guided voxel editing of 3d objects. In *ICCV*, 2023. 2

[47] Liangchen Song, Liangliang Cao, Jiatao Gu, Yifan Jiang, Junsong Yuan, and Hao Tang. Efficient-nerf2nerf: Streamlining text-driven 3d editing with multiview correspondence-enhanced diffusion models. *arXiv preprint arXiv:2312.08563*, 2023. 2

[48] Zhicheng Sun, Zhenhao Yang, Yang Jin, Haozhe Chi, Kun Xu, Kun Xu, Liwei Chen, Hao Jiang, Yang Song, Kun Gai, and Yadong MU. RectifID: Personalizing rectified flow with anchored classifier guidance. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 2

[49] Binglun Wang, Niladri Shekhar Dutt, and Niloy J. Mitra. Proteusnerf: Fast lightweight nerf editing using 3d-aware image context. *Proc. ACM Comput. Graph. Interact. Tech.*, 7(1), 2024. 2

[50] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20697–20709, 2024. 4

[51] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2018. 2

[52] Yuxuan Wang, Xuanyu Yi, Zike Wu, Na Zhao, Long Chen, and Hanwang Zhang. View-consistent 3d editing withnbsp;gaussian splatting. In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29 – October 4, 2024, Proceedings, Part XXXV*, page 404–420, Berlin, Heidelberg, 2024. Springer-Verlag. 2

[53] Minghao Wen, Shengjie Wu, Kangkan Wang, and Dong Liang. Intergsedit: Interactive 3d gaussian splatting editing with 3d geometry-consistent attention prior. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025. 2

[54] Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, Hang Zhang, Hao Meng, Hu Wei, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. Qwen-image technical report, 2025. 2

[55] Jing Wu, Jia-Wang Bian, Xinghui Li, Guangrun Wang, Ian Reid, Philip Torr, and Victor Prisacariu. GaussCtrl: Multi-View Consistent Text-Driven 3D Gaussian Splatting Editing. *ECCV*, 2024. 2

[56] Ruihao Xia, Yang Tang, and Pan Zhou. Towards scalable and consistent 3d editing, 2025. 2

[57] Sihan Xu, Yidong Huang, Jiayi Pan, Ziqiao Ma, and Joyce Chai. Inversion-free image editing with natural language. 2024. 2

[58] Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2020. 4

[59] Haotian Ye, Haowei Lin, Jiaqi Han, Minkai Xu, Sheng Liu, Yitao Liang, Jianzhu Ma, James Zou, and Stefano Ermon. TFG: Unified training-free guidance for diffusion models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 2

[60] Junliang Ye, Shenghao Xie, Ruowen Zhao, Zhengyi Wang, Hongyu Yan, Wenqiang Zu, Lei Ma, and Jun Zhu. Nano3d: A training-free approach for efficient 3d editing without masks, 2025. 2

[61] Meng YOU, Zhiyu Zhu, Hui LIU, and Junhui Hou. NVS-solver: Video diffusion model as zero-shot novel view synthesizer. In *The Thirteenth International Conference on Learning Representations*, 2025. 2

[62] Jiwen Yu, Yinhuai Wang, Chen Zhao, Bernard Ghanem, and Jian Zhang. Freedom: Training-free energy-guided conditional diffusion model. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 2

[63] Wangbo Yu, Jinbo Xing, Li Yuan, Wenbo Hu, Xiaoyu Li, Zhipeng Huang, Xiangjun Gao, Tien-Tsin Wong, Ying Shan, and Yonghong Tian. Viewcrafter: Taming video diffusion models for high-fidelity novel view synthesis. *arXiv preprint arXiv:2409.02048*, 2024. 8, 12

[64] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. 2

[65] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2018. 3

[66] Xuanjia Zhao, Jian Guan, Congyi Fan, Dongli Xu, Youtian Lin, Haiwei Pan, and Pengming Feng. Fastdrag: Manipulate anything in one step. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 2

[67] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017. 2

[68] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. *NIPS*, 2017. 2

[69] Jingyu Zhuang, Chen Wang, Lingjie Liu, Liang Lin, and Guanbin Li. Dreameditor: Text-driven 3d scene editing with neural fields. In *SIGGRAPH*, 2023. 2

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).

A. Additional Qualitative Results

We provide multiple additional examples of our multi-view consistent image editing in this section, using both InstructPix2Pix [7] and pix2pix-turbo [38]. We show results for InstructPix2Pix in Fig. 8 where we note that e.g. the eyes and the surrounding regions are more consistent for our method, and in Fig 9 where we see that e.g. details on the saddle are better preserved across views when using our method. For pix2pix-turbo 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. We also show an additional example of the sparse editing in Fig. 12 where we see that the renderings of the Gaussian splat model is more consistent using our edited images compared to the per image edits.

We also provide video results on our project page (https://3d-consistent-editing.github.io/) showing the Gaussian splat renderings. We show our method and the methods we compare to for all 21 scene-prompts pairs in our test set. We show results using both InstructPix2Pix [7] and pix2pix-turbo [38]. We also show renderings of Gaussian splats from our sparse view setup.

B. Multi-View Image Editing

Image Editing Methods. We test our method together with two different diffusion based image editing methods, both InstructPix2Pix [7] and the single-step model pix2pix-turbo [38]. For InstructPix2Pix we use 20 diffusion 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 [9] 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 from SAM [26] to ensure that only the desired object is changed and that the rest of the image remains unchanged.

Sparse Editing. We show an overview of the sparse editing in Fig. 7. For the sparse editing we edit 3-4 images and then utilize the multi-view diffusion model ViewCrafter [63] 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 5000 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 [31] and DGE [11] 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 [25], 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 [18, 31].

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.

DGE. DGE also initially performs an multi-view consis-

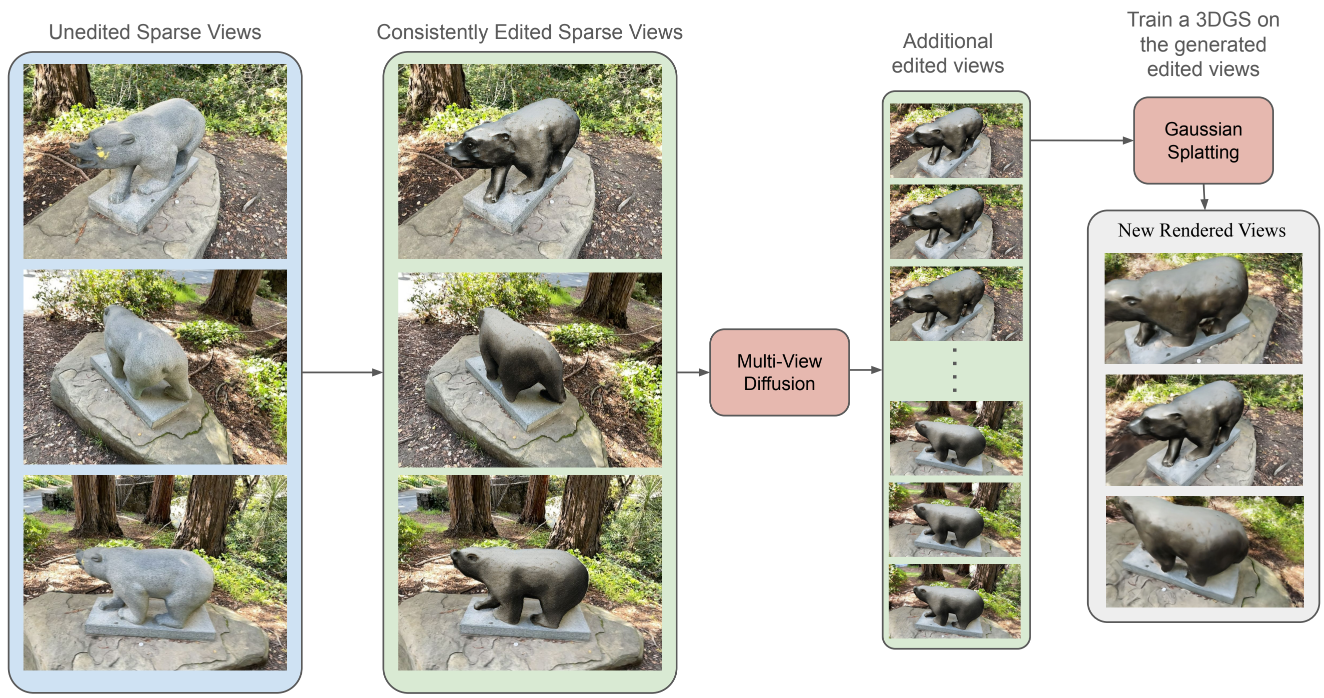

Figure 7. Overview of the sparse editing pipeline, that from a few sparse views generates a Gaussian splat model of the edited scene. Our consistent editing method is used to edit the given sparse views and a Multi-View Diffusion network is then used to generate additional edited images of the same scene. A Gaussian splat model can then be trained on all these views, and we can render novel views using the Gaussian splat model.

tent 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 Gaussian 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 [31]. We additionally show all pairs in the validation set in Table 6. Note that different scenes are used in the validation and test sets.

Figure 8. Additional Qualitative example of multi-view consistent image editing using methods all based on the image editing method IP2P. We note that our method is able to preserve both details better than the other methods, as seen in e.g. the eyes and the area around the eyes.

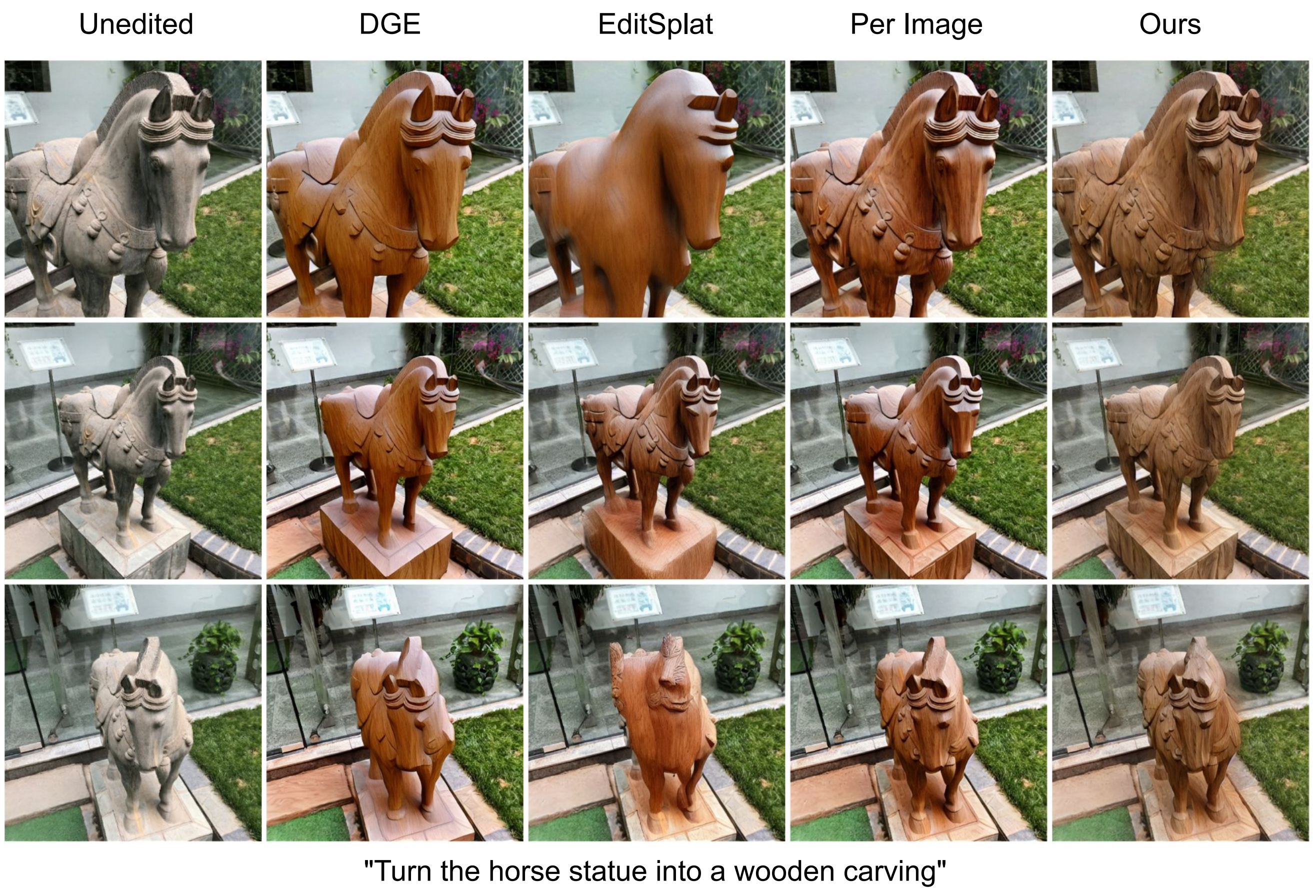

Figure 9. Additional Qualitative example of multi-view consistent image editing using methods all based on the image editing method IP2P. We note that our method is able to preserve detailed textures than the other methods, as can be seen e.g. on the saddle.

Figure 10. Additional Qualitative example of multi-view consistent image editing using methods all based on the image editing method pix2pix-turbo. We note that our method is able to give a more consistent overall color, and also that the background is more consistent.

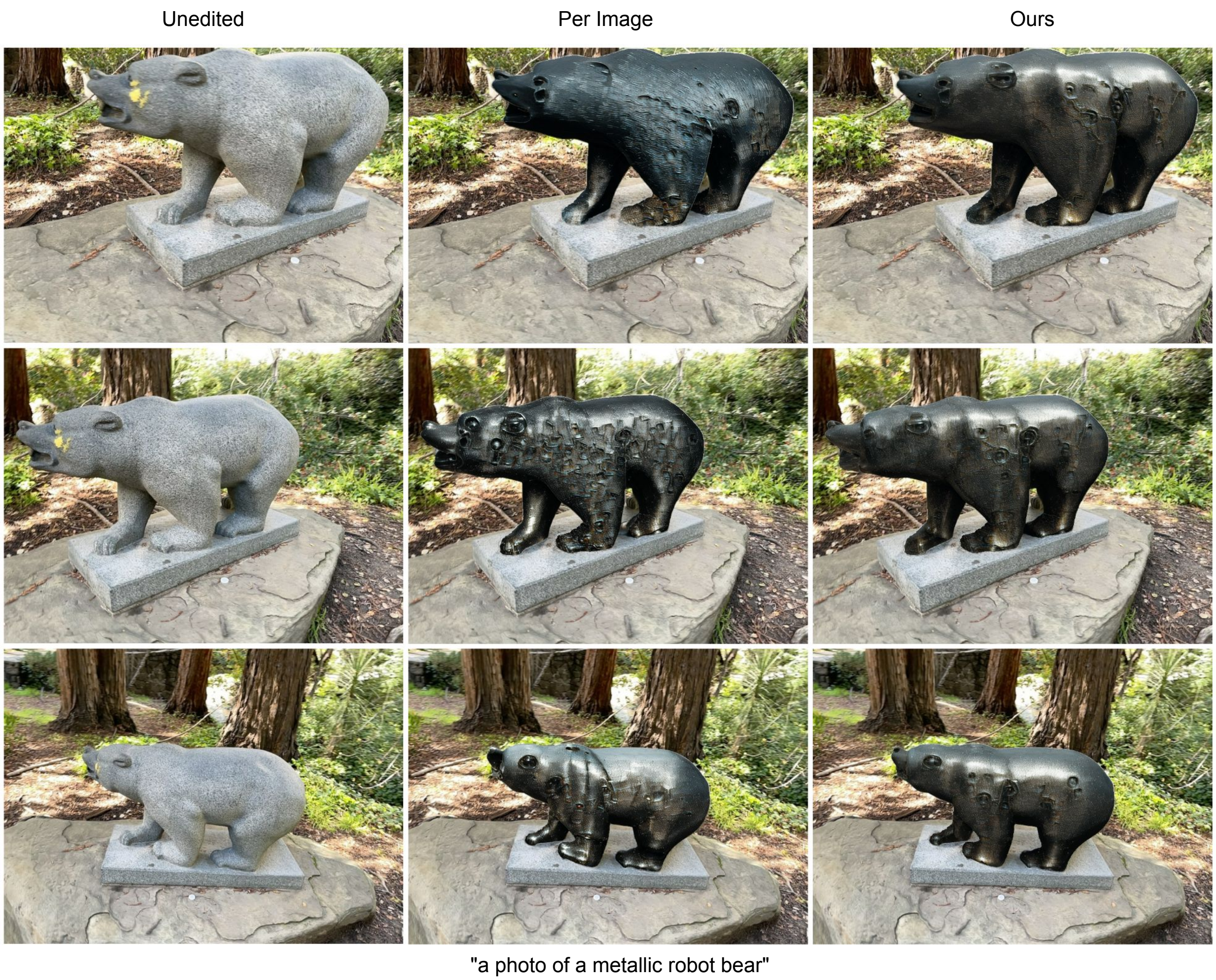

Figure 11. Additional Qualitative example of multi-view consistent image editing using methods all based on the image editing method pix2pix-turbo. We note that with our method the texture on the bear is consistent across the views and is able to capture realistic lighting reflections.

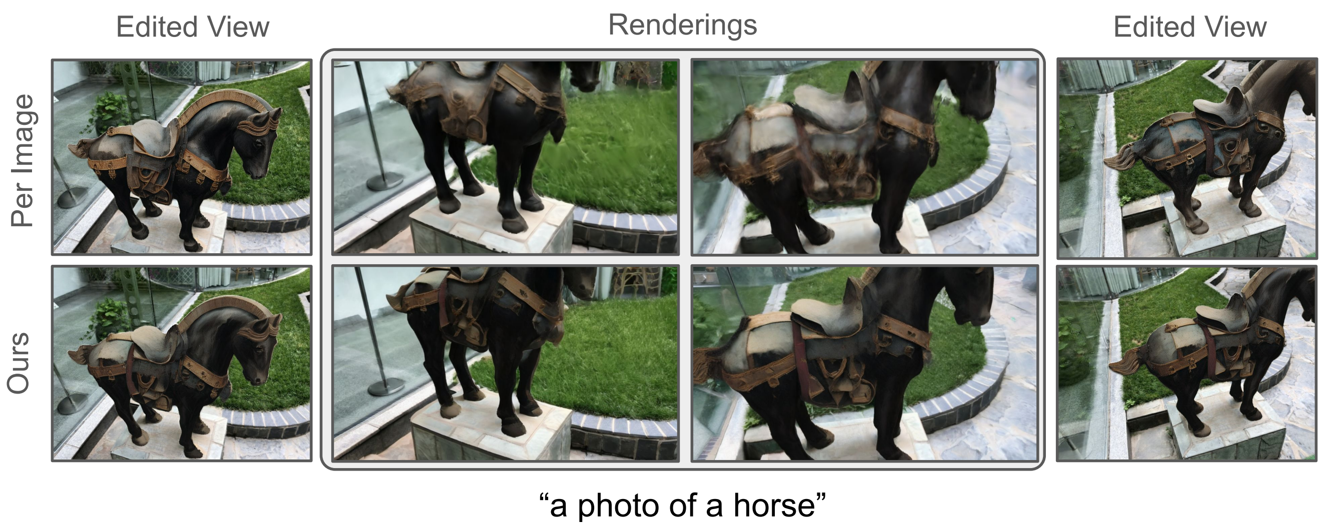

Figure 12. Additional example of editing a few sparse views and using ViewCrafter to interpolate. We note that the edited views are more consistent for out method compared to per image edits, which can be seen in e.g. the saddlebag in both the edited images and the Gaussian splat renderings.

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 when evaluating all methods.

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.