3D-Consistent Multi-View Editing by Correspondence Guidance
(Supplementary Material)

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

In this supplementary material we show video results and additional qualitative results of our consistent multi-view image editing (Sec. 1), technical details related to both our multi-view editing (Sec. 2) and the 3D Gaussian splat editing (Sec. 3), and details of the dataset and prompts that were used (Sec. 4).

We encourage the reader to watch video results on the 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.

1 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 [2] in Fig. 1, 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 [11], we see in Fig. 3 that the overall colors and backgrounds are more consistent for our method and in Fig. 4 we note that the texture on the bear is more consistent with our editing. Finally, we show additional results for FLUX.1 [7] in Fig. 5, 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 On the project page we provide additional video results showing Gaussian splat renderings for both the dense and sparse view setups. Additionally, we show our method and the methods we compare to for all 21 scene-prompts pairs in our test set, using both InstructPix2Pix [2] and pix2pix-Turbo [11].

Sparse 3DGS Editing We show an additional example of the sparse editing in Fig. 6, where we see that the renderings of the Gaussian splat model are more consistent when using our edited images compared to the per image edits.

2 Multi-View Image Editing

Image Editing Methods We test our method together with two different diffusion based image editing methods, InstructPix2Pix [2] and the single-step model pix2pix-Turbo [11], as well as the flow-matching based method FLUX.1 [7]. 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 a 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 a 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 [7] 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 [1]. This assumption of linear denoising trajectories should in theory be a perfect approximation [9,10], but in practice an approximation error still exists. We find that, while using this approximation, we are still able to stably converge to initial seeds that improve consistency.

Image Matching Details We use the robust dense matcher RoMa [4], 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 [13] 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. 2. For the sparse editing we edit 3-4 images and then utilize the multi-view diffusion model ViewCrafter [12] 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 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

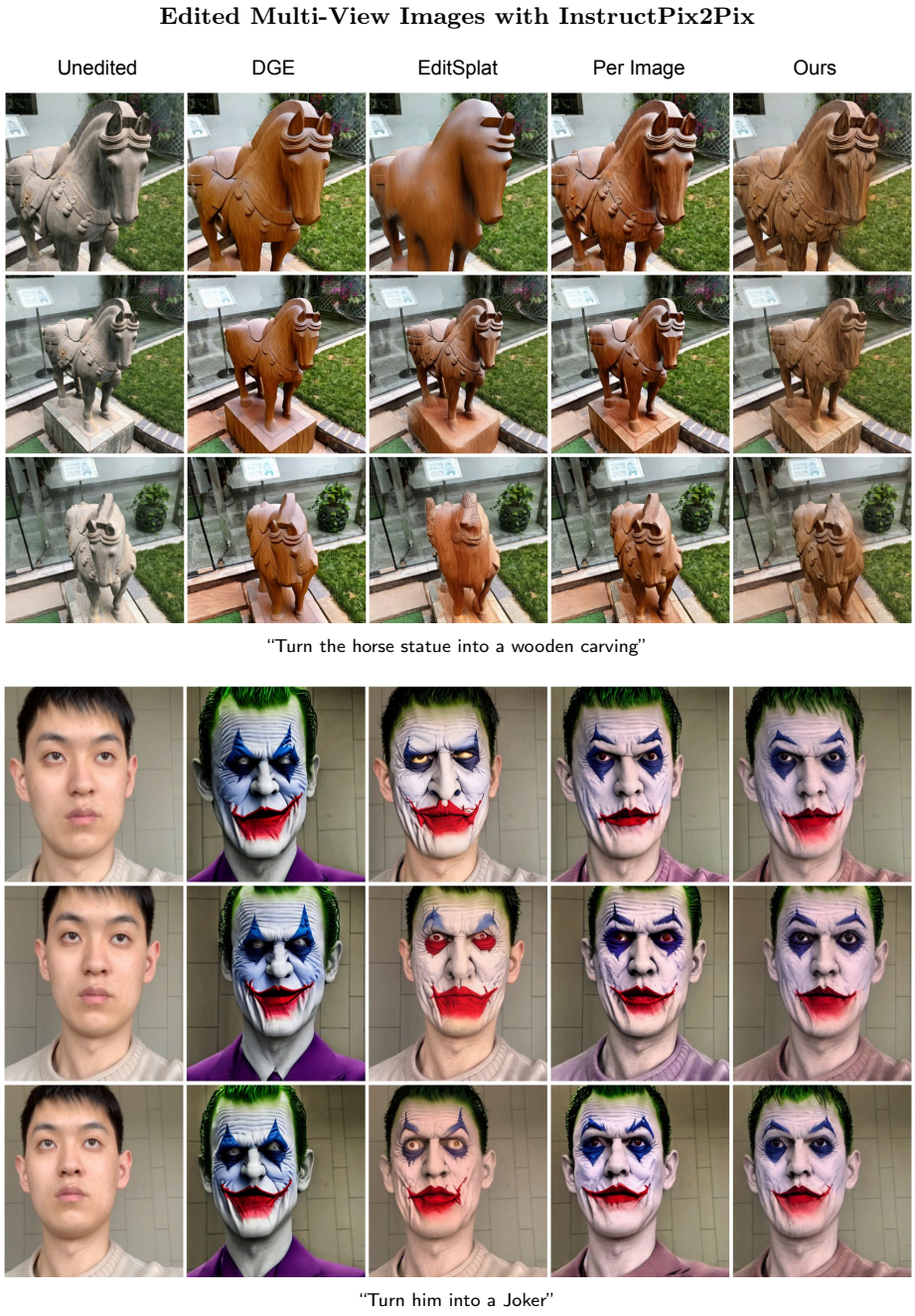

Fig. 1: Additional qualitative example of multi-view consistent image editing using methods all based on the image editing method InstructPix2Pix. We note that our method is able to preserve details better than the other methods, as seen, e.g., on the saddle and the area around the eyes.

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.

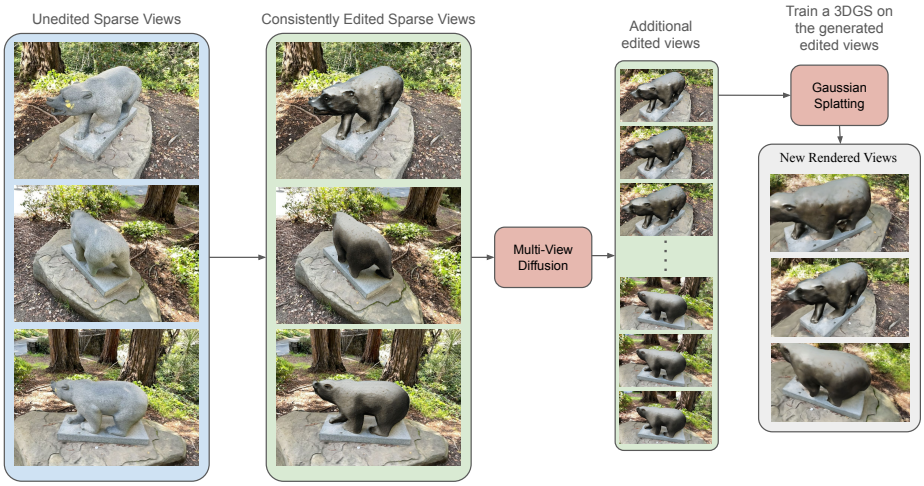

Fig. 2: Overview of the sparse editing pipeline, that from a few sparse views generates a Gaussian splat model of the edited scene. Our consistent editing method is used to edit the given sparse views and a Multi-View Diffusion network is then used to generate additional edited images of the same scene. A Gaussian splat model can then be trained on all these views, and we can render novel views using the Gaussian splat model.

3 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 [8] and DGE [3], 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 [6], 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 [5, 8].

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

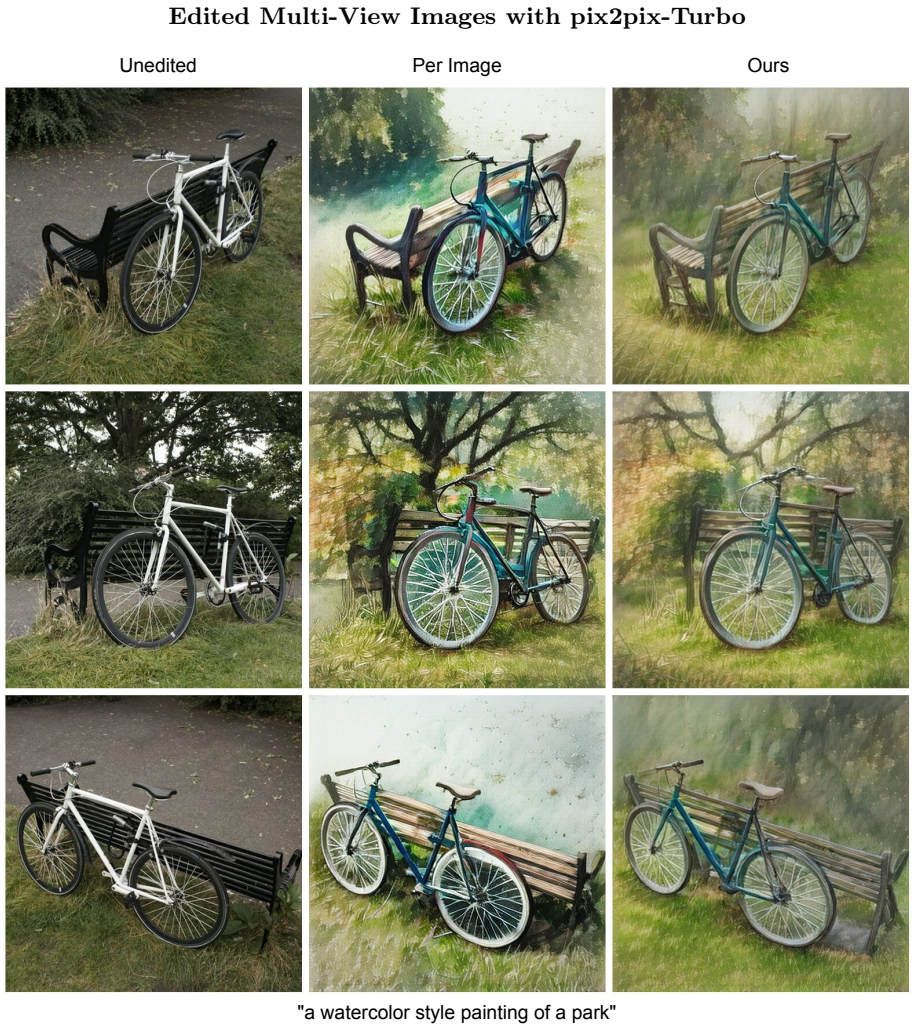

Fig. 3: Additional qualitative example of multi-view consistent image editing with pix2pix-Turbo. We note that our method gives a more consistent overall color, e.g., on the bicycle frame, and that the background is more consistent.

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.

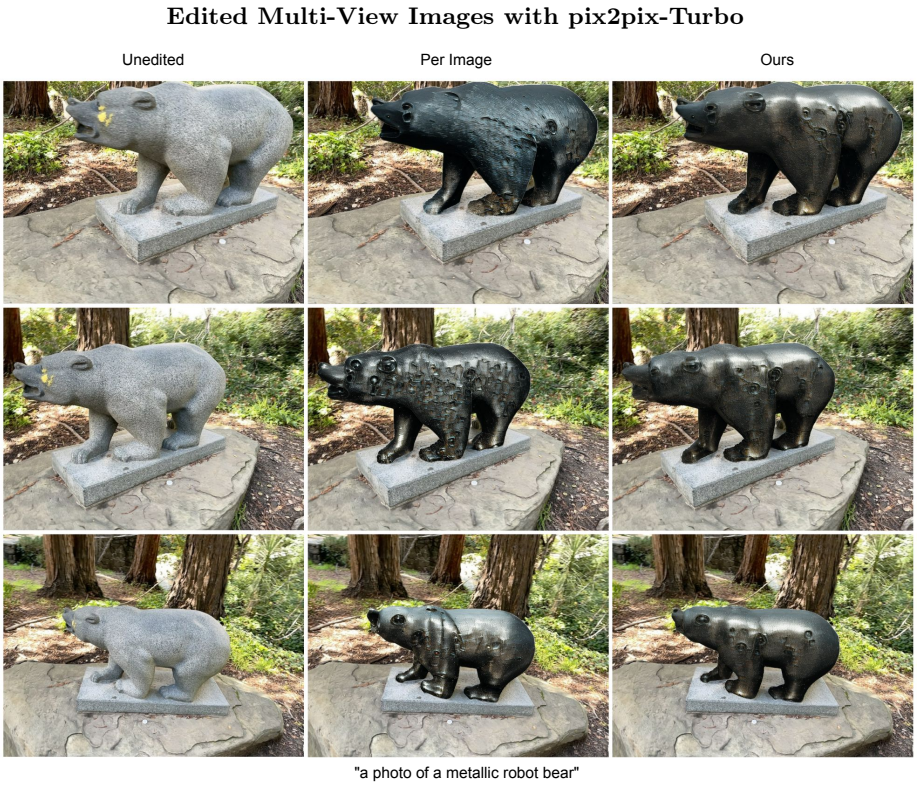

Fig. 4: Additional qualitative example of multi-view consistent image editing with pix2pix-Turbo. We note that with our method the texture on the bear is consistent across the views, and that realistic lighting reflections are captured.

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.

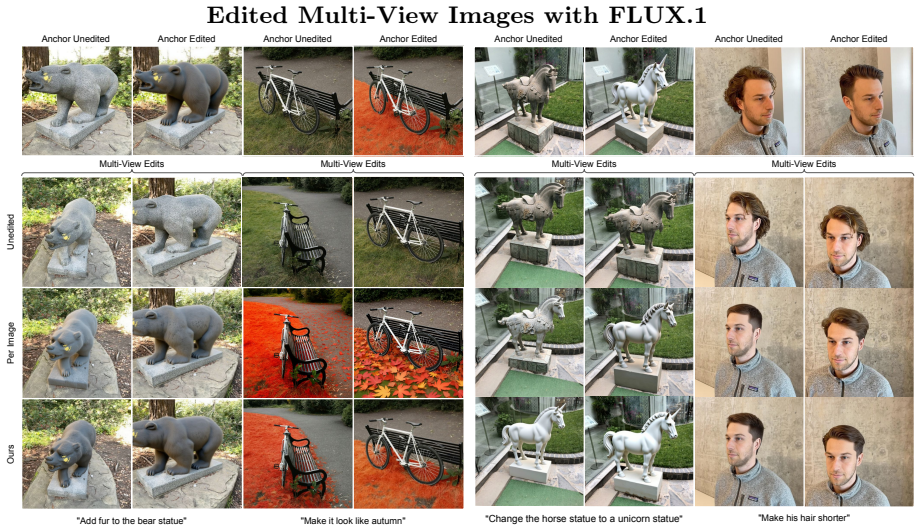

Fig. 5: Additional qualitative comparison using FLUX.1. The loss is computed between each image and an anchor image. On the left we show rigid and near-rigid edits, and on the right we show non-rigid edits. The multi-view edits are more consistent with our approach, e.g., in the following details: (a) 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.

4 Scene-Prompt Pairs

We show all scene-prompt pairs used in the test set in Table 1. Most of the pairs are the same as used in prior work [8]. We additionally show all pairs in the validation set in Table 2. Note that different scenes are used in the validation and test sets.

References

1. 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`
2. 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)
3. 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)
4. 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)

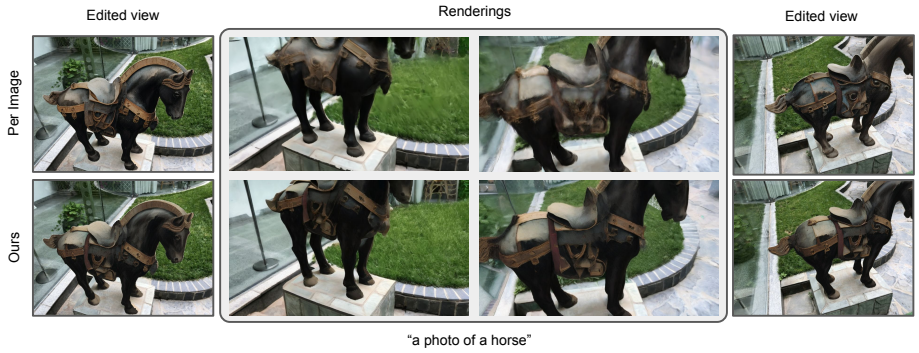

Fig. 6: Additional example of editing a few sparse views and using ViewCrafter to interpolate. We note that the edited views are more consistent for our method compared to per image edits, which can be seen, e.g., in the saddlebag in both the edited images and the Gaussian splat renderings.

5. 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)
6. 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)
7. 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
8. 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)
9. 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
10. 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
11. Parmar, G., Park, T., Narasimhan, S., Zhu, J.Y.: One-step image translation with text-to-image models. arXiv preprint arXiv:2403.12036 (2024)
12. 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)
13. 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)

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 1: 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 2: All 7 scene–prompt pairs used as a validation set when choosing hyperparameters and performing ablation studies for our method.