

-Supplementary Material- GaussianEditor: Editing 3D Gaussians Delicately with Text Instructions

Junjie Wang*, Jiemin Fang*, Xiaopeng Zhang, Lingxi Xie, Qi Tian
Huawei Inc.

{is.wangjunjie, jaminfong, 198808xc, zxphistory}@gmail.com tian.qil@huawei.com



Figure A1. GaussianEditor demonstrates excellent extension capabilities. It can be seamlessly integrated with the 3D generative model, such as GaussianDreamer [6].

A. Appendix

A.1. Additional Implementation Details

GaussianEditor takes a 3D scene reconstructed by 3D Gaussian Splatting [3] as input. Learning each scene takes 30,000 iterations. Images wider than 512 pixels are resized to 512. Similar to Instruct NeRF2NeRF (IN2N) [2], GaussianEditor also uses Instruct Pix2Pix (IP2P) [1] to edit 2D pictures. The classifier-free diffusion guidance weights are set as follows:

- 1) Fig. 1: $s_I \in [1.4, 1.5]$, $s_T \in [7.0, 12.0]$,
- 2) Fig. 4 Bicycle: $s_I = 1.2$, $s_T = 12.0$,
- 3) Fig. 4 Bear: $s_I = 1.5$, $s_T = 6.5$,
- 4) Fig. 5: $s_I = 1.2$, $s_T = 8.0$,
- 5) Fig. 6: $s_I \in [1.2, 1.5]$, $s_T \in [7.5, 12.0]$,
- 6) Fig. 7: $s_I = 1.3$, $s_T = 12.0$,

where s_I is the weight for image guidance and s_T is the weight for text guidance.

GaussianEditor implements 3D editing based on the 2D diffusion model. Due to the instability of 2D editing, scenes tend to become blurry as the number of iterations increases. Therefore, we observe the current rendering results during the training process and limit the editing rounds, generally within 200 rounds.

*Equal contributions.

†Corresponding author.

A.2. Quantitative Evaluation

Quantitative Evaluation Based on CLIP. In Tab. A1, we present the quantitative evaluation results. The scenes in Fig. 5 are used for this test. We follow the metrics used in Instruct NeRF2NeRF (IN2N) [2], including the CLIP [4] text-image direction similarity and image-image similarity between the original scene and the edited scene. The quantitative results indicate that our method achieves a comparable CLIP text-image direction similarity score with IN2N, while image-image similarity has improved a lot. We would like to analyze the limitations of the used metric as follows.

Table A1. Results of CLIP Text-Image Direction Similarity and Image-Image Similarity between the original scene and edited scene. Test scene is shown in Fig. 5

	CLIP Text-Image Direction Similarity \uparrow	Image-Image Similarity \uparrow
IN2N [2]	0.12	0.86
Ours	0.11	0.94

Limitation of The CLIP-based Metric. Although we provide quantitative analysis based on CLIP. However, we find that the current CLIP-based metrics are not reliable enough. For example, CLIP has problems with color discrimination. As shown in Fig. A2, we use CLIP to calculate the similarity between solid color images, which are white and yellow respectively, and the text descriptions, *i.e.*



		
“This is white”	0.192	0.231
“This is yellow”	0.176	0.272

Figure A2. Similarity scores between the text and image features encoded by CLIP [4]. Pure white images consistently have lower scores¹.

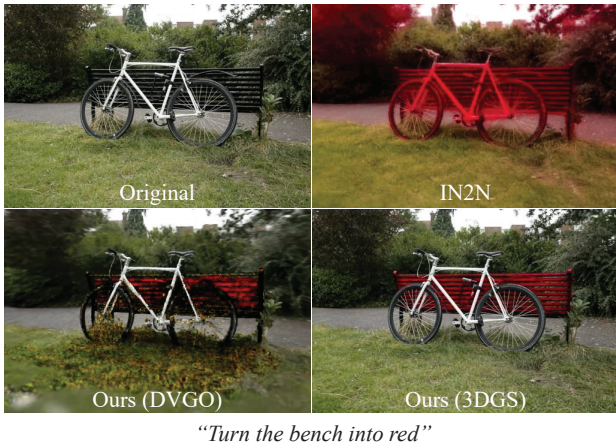


Figure A3. Visualization result of Tab. 1.

“This is white” or “This is yellow”. The results show that yellow images consistently achieve higher matching scores. This is one of the reasons why our CLIP text-image direction similarity does not show an evident advantage. Therefore, we believe that a more reliable evaluation metric for text-guided editing tasks is one of the important future research directions.

User Study. Here are more details of the user study shown in Sec. 4.3. 4 human editing results in Fig. 5 and 3 bear editing results in Fig. 4 are chosen for the user study, forming 7 questions for the questionnaire. In every question, we showcase the original scene, the text instructions for editing, and the editing results of IN2N [2] and GaussianEditor. For equality, the editing results in the question are randomly named using the letter A or B. Users are required to choose the better one. After 21 users submit their questionnaires, 147 votes (21 users \times 7 questions) are collected. GaussianEditor gets 128 votes for all questions and IN2N gets 19 votes, accounting for 87.07% and 12.93%, respectively.

¹The red border is to make it easier for readers to see the white image. The actual image input to the CLIP does not have this border.

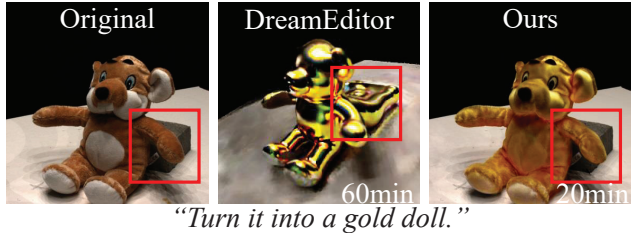


Figure A4. Comparison to DreamEditor on DTU dataset.

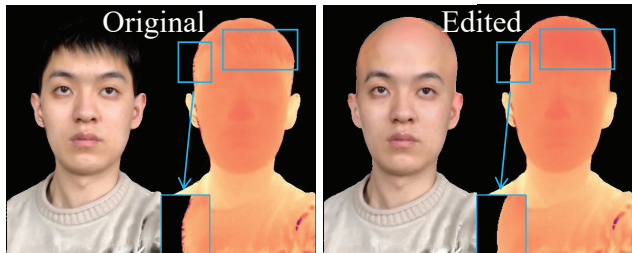


Figure A5. Depth map of hair editing in Fig. 1.

A.3. Qualitative Evaluation

Comparison with IN2N [2] and Different Backbones. In Fig. A3, we show the qualitative result of IN2N and GaussianEditor with different backbones. This scene is also used in Tab. 1. IN2N fails in this task and turns the bicycle, bench, and tree all red. Besides, the backbone using DVGO [5] also has difficulty in localizing the bench precisely and produces worse rendering results, while GaussianEditor grounds the bench precisely and turns it red.

Comparison with DreamEditor [7]. In Fig. A4, we show the qualitative result of DreamEditor and GaussianEditor. GaussianEditor delicately edits the doll and retains the hair details, while DreamEditor wipes the hair and changes the back box. Besides, GaussianEditor gets the wanted editing result using less time.

Depth Map of Geometric Editing. In Fig. A5, we show the depth map of the hair editing result shown in Fig. 1. The depth map indicates that GaussianEditor possesses a certain level of geometric editing capability. The task of handling drastic geometric editing changes is left for future work.

A.4. Extension

GaussianEditor demonstrates excellent extension abilities. For instance, it can be seamlessly integrated with the 3D generative model GaussianDreamer [6], resulting in enhanced editing effects. Specifically, as shown in Fig. A1, upon obtaining the Gaussian RoI, the Gaussians within the RoI are saved individually and utilized as the initialization for the 3D-generation model. Simultaneously, the text description of the edited scene is fed into the pipeline of the 3D generation model. Eventually, the edited new object is merged into the original scene to form an edited 3D scene.

References

- [1] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *CVPR*, 2023. [1](#)
- [2] Ayaan Haque, Matthew Tancik, Alexei Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-nerf2nerf: Editing 3d scenes with instructions. In *CVPR*, 2023. [1](#), [2](#)
- [3] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (ToG)*, 2023. [1](#)
- [4] Alec Radford, Jong Wook Kim, Chris Hallacy, A. Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021. [1](#), [2](#)
- [5] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *CVPR*, 2022. [2](#)
- [6] Taoran Yi, Jiemin Fang, Guanjun Wu, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Qi Tian, and Xinggang Wang. Gaussian-dreamer: Fast generation from text to 3d gaussian splatting with point cloud priors. *arxiv:2310.08529*, 2023. [1](#), [2](#)
- [7] Jingyu Zhuang, Chen Wang, Lingjie Liu, Liang Lin, and Guanbin Li. Dreameditor: Text-driven 3d scene editing with neural fields. *arXiv preprint arXiv:2306.13455*, 2023. [2](#)