A. Introduction

The content of our supplementary material is organized as follows:

- Firstly, we provide more qualitative results in Section B.
- Secondly, we demonstrate WebUI for GaussianEditor in Section. C along with specifically tailored algorithem for WebUI 3D editing.
- We attach the video of using our WebUI, including tracing, editing, deleting, and adding objects.

B. More Results

In Fig. 3, we demonstrate more results of GaussianEditor . Our method provides controllable, diverse, high-resolution 3D editing, needing only 2-7 minutes.

C. WebUI

Although works in the neural radiance fields (NeRF) [3] domain also incorporate WebUIs, the slow rendering speeds of NeRF mean that users are confined to low resolutions and very low frame rates when using the WebUI, resulting in a subpar user experience. Fortunately, thanks to our adoption of Gaussian Splatting [1] in GaussianEditor , a method known for its rapid rendering capabilities, our WebUI can comfortably support usage at 2K resolution and 60fps. Besides We leverage the interactivity of the webUI to enhance both semantic tracing and object incorporation applications, which will be discussed in the following two subsections.

C.1. Semantic Tracing with Point-base Prompts

Interactive WebUI applications with user interfaces are extremely important. In practical scenarios, users often intend to edit only specific areas of a complete scene, a task that can be challenging to specify solely through text prompts. For instance, it becomes difficult to determine through text which object a user wants to edit when multiple objects of the same type are present in a scene, and the user wishes to change just one of them. To address this issue, we propose semantic tracing with point-based prompts.

Semantic tracing with point-based prompts requires users to click on the screen to add 2D points from a specific view. Specifically, when the user clicks a point on the screen, we back-project this point into a spatial point based on the intrinsic and extrinsic parameters of the current viewpoint camera:

$$[x, y, z]^T = [\mathbf{R}|\mathbf{t}]z(\boldsymbol{p})\mathbf{K}^{-1}[p_x, p_y, 1]^T, \quad (1)$$

where $[\mathbf{R}|\mathbf{t}]$ and $\mathbf{K}$ denote the extrinsic and intrinsic of the current camera, $\boldsymbol{p}$, $z(\boldsymbol{p})$ and $[x, y, z]^T$ refer to the user-clicked pixel, its corresponding depth, and the spatial point, respectively.

Subsequently, in other views, we re-project these spatial points onto the camera's imaging plane, identifying the pixels corresponding to these 3D points. We use the projection points of these 3D points in the reference views as the point prompts for semantic segmentation with SAM [2]. Then, we unproject these semantic segmentation maps back to Gaussians, as demonstrated in the main text.

As can be seen in Fig. 1, with only about five points indicated by the users, semantic tracing with point-based prompts enables finer granularity control over the areas to be tracked.

C.2. Object Incorporation with WebUI

As detailed in the main paper, our proposed method for 3D inpainting with object incorporation allows for the addition of objects specified by text in designated areas. The webUI facilitates users in easily drawing 2D masks to define these areas. Moreover, in this method of 3D inpainting for object incorporation, depth information is crucial for seamlessly integrating new objects into the original Gaussian Scene. Current methods for monocular depth estimation, however, can't always provide completely accurate depth maps, leading to imprecise alignment. Therefore, as depicted in Fig. 2, we utilize the webUI to modify the scale of the estimated depth, enabling users to achieve a more accurate alignment of the objects.

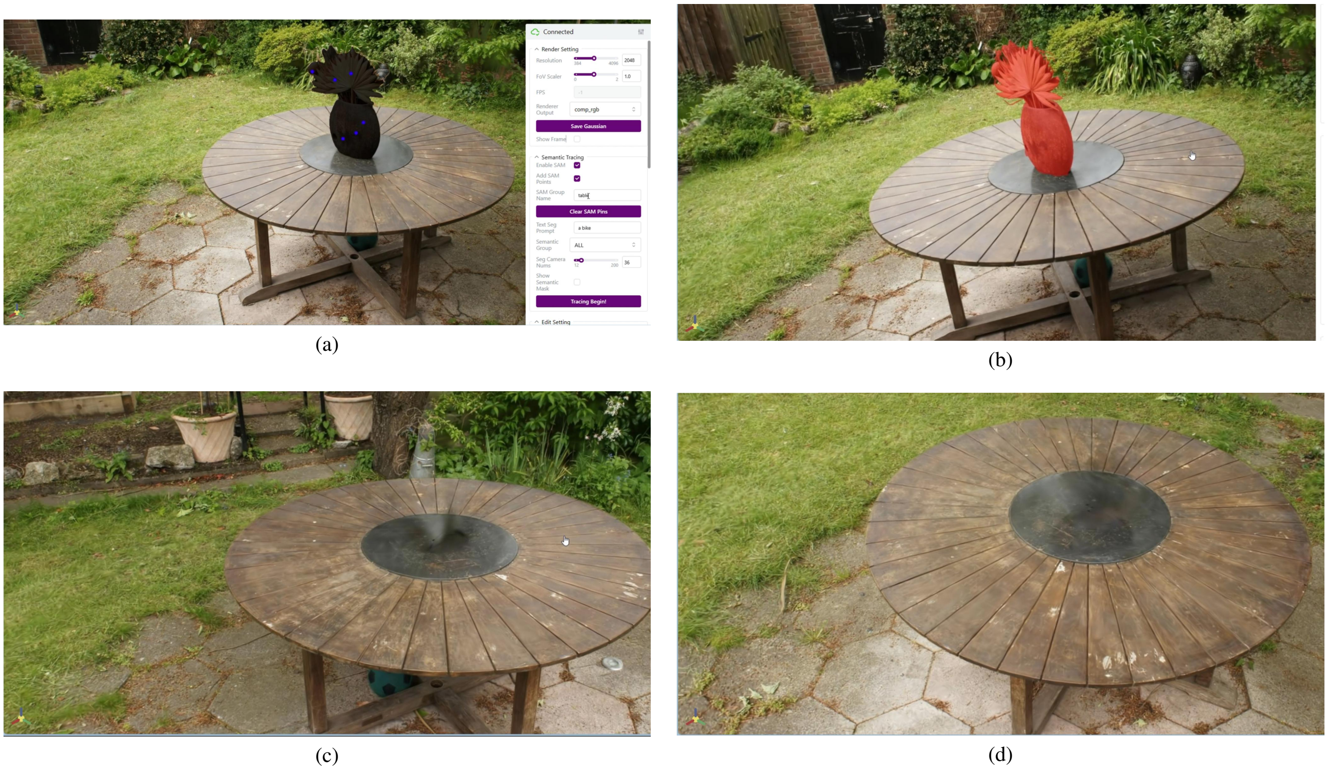
To be more specific, users control the Gaussian scale by sliding a slider. After obtaining a new depth scale, we update the position and size of the added objects according to the new depth scale. Since the entire process involves only minor adjustments to the position and scale parameters of a few Gaussians, real-time scaling can be achieved.

References

[1] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (ToG)*, 42(4): 1–14, 2023. 1

[2] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 1

[3] 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. In *ECCV*, 2020. 1



(a) (b) (c) (d)

Figure 1. Semantic Tracing with Point-based Prompts. In (a), users provide key points on a view by clicking the screen with the mouse. In (b), we segment the target object based on these points. (c) and (d) depict the results after removing the segmented objects. It can be seen from the above that our point-based tracing method offers high precision and interactivity.

(a) (b) (c) (d)

Figure 2. Object Incorporation with WebUI. Empowered by our interactive WebUI, the depth scale addresses the limitation of monocular depth estimation, which cannot guarantee precise depth map predictions. As can be seen in (a), inaccurate results can lead to failures when aligning generated objects with Gaussian scenes. We leverage the interactive nature of our WebUI to dynamically adjust the estimated depth in real-time, thus resolving this issue, demonstrated in other images.

Figure 3. **More results of GaussianEditor.** GaussianEditor allows for fast, versatile, high-resolution 3D editing, requiring only 2-7 minutes and 10-20GB of GPU memory on a single A6000 GPU. Please note that the background of face editing scenes remains unchanged.