PrEditor3D: Fast and Precise 3D Shape Editing
- Supplementary Document -

Ziya Erkoç[1] Can Gümeli[1] Chaoyang Wang[2] Matthias Nießner[1] Angela Dai[1]
Peter Wonka[2,3] Hsin-Ying Lee[2] Peiye Zhuang[2]
[1]Technical University of Munich [2]Snap Inc [3]KAUST

1. Appendix

We present additional details about PrEditor3D in this appendix. We start by explaining some of the implementation details in Sec. 1.1. In Sec. 1.2, we discuss automatic masking, an alternative to user-brushed masking. Sec. 1.3 follows this discussion with the effect of mask granularity on the editing process. Finally, we explain the directional CLIP metrics we used for baseline comparison in Sec. 1.4. We present additional qualitative results in Sec. 1.5.

1.1. Implementation Details

We used the official implementation and checkpoint of MVDream as our multi-view diffusion model. It has 256 x 256 resolution and it can generate four views by default. In all of our generations, we set the classifier-free guidance scale of the diffusion process to 10. Official DDPM inversion [2] implementation only handles single-image but we modified it to handle our four view renderings. The inversion process takes 9 seconds on RTX 3090. With the inverted latents, we ran our inference for 41 steps, which takes around 12 seconds on an RTX 3090. For the segmentation, we calculate bounding boxes using Grounding DINO [3] for all views and add these as constraints to SAM 2 [5] tracking. That is to help SAM 2 with the segmentation, we constrain each frame separately. For merging and reconstruction, we modify GTR [7], which is a feed-forward reconstruction model. GTR mainly operates on triplanes but just before reconstruction, those triplanes are converted into a voxel grid. We manipulated its voxel grids to merge two shapes.

1.2. Automatic Masking

In addition to user-brushed masks, we can also generate and operate on automatically generated masks. Even though they limit the editing region, when compared to user-brushed masks; they can be practically used as a starting point for user-brushed masking.

We leverage our segmentation approach to replace masks given by the user. We use an input prompt from the user to detect the target region using Grounding DINO [3] and SAM 2 [5]. This segmentation method gives us a mask restricted only to the sword. As a result, the generation process cannot go beyond that region. However, when we accept input from user masks, user can explicitly show their intention with the mask and can generate a *”viking axe”*, as shown in Fig. 2.

We want to reiterate that although the user-brushed masks are too coarse and not 3D-consistent, our method can generate impressive results without modifying the original parts of the shape. That is, a quickly drawn mask is enough for our method to work.

1.3. Mask Granularity

We experimented with different granularity levels for the input masks. We started with a mask that we detected automatically using Grounding DINO [3] and SAM 2 [5]. As shown in Fig. 1. If we use the original segmentation, then the generation is restricted to that certain region and the model cannot have room to add ”cat” features. That is, it tries to follow the shape of the original chicken. As we add more dilation, it tries to add features like cat ears. This shows the trade-off between loyalty to input and flexibility. Based on this observation, we gave coarse masks as input and allowed the model to edit flexibly. Thanks to our merging approach, we could still combine the edited region with the original shape to keep the rest intact.

1.4. Directional CLIP Metrics

In Sec. 4.1-4.2 of the main paper, we discuss directional CLIP score metrics [1, 4, 6] to evaluate 3D editing fidelity, to complement other metrics that measure the quality of the output shape. We report directional CLIP scores of different methods in Tab. 3 of the main paper. In this section, we formally define and discuss the reported metrics.

$$\text{CLIP}_{\text{dir}} = \frac{1}{N}\sum_{i=1}^{N} < F^i_{IE} - F^i_{II}, F_{TE} - F_{TI} >, \qquad (1)$$

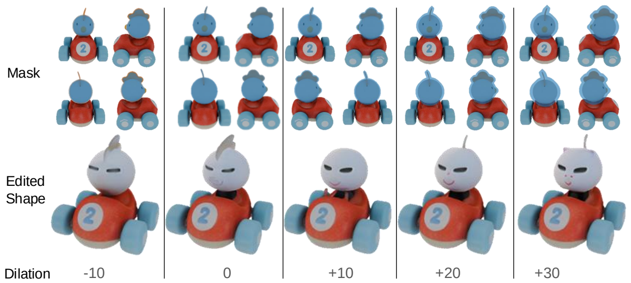

Figure 1. **Different granularity of masking**. Too fine-grained masks can over-constrain the generation process since they only point to the region to be replaced but do not include the user's intention. More dilation increases flexibility but can also edit more regions than intended (e.g., the region underneath the cat). Negative dilation means erosion.

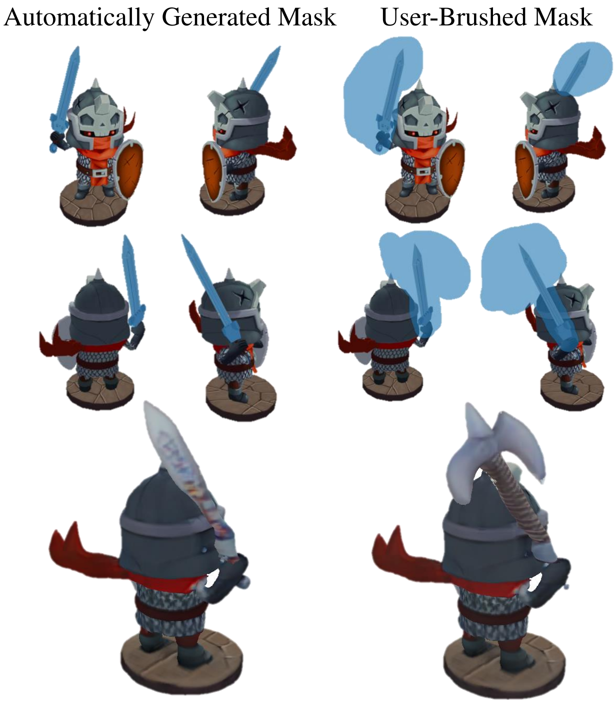

Figure 2. **Comparing automatically generated mask to user-generated mask**. Users may want to do specific editing such as replacing the *"sword"* with *"a viking axe"*. If we only rely on automatic masking, the result may not follow the user's intention since the automatically generated mask can limit the editing to a certain region. However, when we rely on explicit masking, we can get the specific shape requested by the user.

where $< .,. >$ refers to an inner product, F^i_{IE}, F^i_{II} are the normalized CLIP image embeddings over rendered images of input and edited shapes, indexed by i, and F_{TE}, F_{TI} are the corresponding normalized text embeddings of edited and input prompts. i indexes a particular frame, while N is the total number of rendered frames. In our directional CLIP evaluations, we use $N = 70$ views rendered over a 360° trajectory, significantly larger than the four input views we use for our method and the baseline methods.

We also introduce additional metrics inspired by CLIP$_{\text{dir}}$, but aim to fix some of its problems. First, we define

$$\text{CLIP}_{\text{dir-cos}} = \frac{1}{N}\sum_{i=1}^{N} \text{C}(F^i_{IE} - F^i_{II}, F_{TE} - F_{TI}), \quad (2)$$

where C$(.,.)$ is the cosine distance.

We also introduce two modified versions of these metrics, namely

$$\text{CLIP}_{\text{dir-avg}} =< \frac{1}{N}\sum_{i=1}^{N} F^i_{IE} - F^i_{II}, F_{TE} - F_{TI} > \quad (3)$$

$$\text{CLIP}_{\text{dir-avg-cos}} = \text{C}(\frac{1}{N}\sum_{i=1}^{N} F^i_{IE} - F^i_{II}, F_{TE} - F_{TI}) \quad (4)$$

that compute the same metrics over the average image embeddings instead of averaging scores to ensure further robustness.

We also propose two similarity change error metrics, CLIP$_{\text{diff-edit}}$ and CLIP$_{\text{diff-noedit}}$

$$\text{CLIP}_{\text{diff-edit}} = \frac{1}{N}\sum_{i=1}^{N} |\text{C}(F^i_{II}, F_{TW}) - \text{C}(F^i_{IE}, F_{TW})|_{\text{rel}} \quad (5)$$

$$\text{CLIP}_{\text{diff-noedit}} = \frac{1}{N}\sum_{i=1}^{N} |\text{C}(F^i_{II}, F_{TG}) - \text{C}(F^i_{IE}, F_{TG})|_{\text{rel}}. \quad (6)$$

Here, $|x - y|_{\text{rel}} = \frac{|x-y|}{\max(x,y)}$, F_{TW} is the text embedding of the edited word or phrase, and F_{TG} represents the "generic" text. For instance, when the prompt "a chicken riding a bike" becomes "cat riding a bike", F_{TW} embeds the text "cat" and F_{TG} embeds the text "object riding a bike". By measuring similarity differences of rendered images to F_{TW} and F_{TG}, we aim to measure the preservation of the object and context semantics, respectively.

1.5. Additional Qualitative Results

We provide additional qualitative results to further explore what our method can achieve. We illustrate the results in Figure 3. PrEditor3D can operate on various shapes such as human, animal, car and apartments. We also showcase additional baseline results in Figure 4.

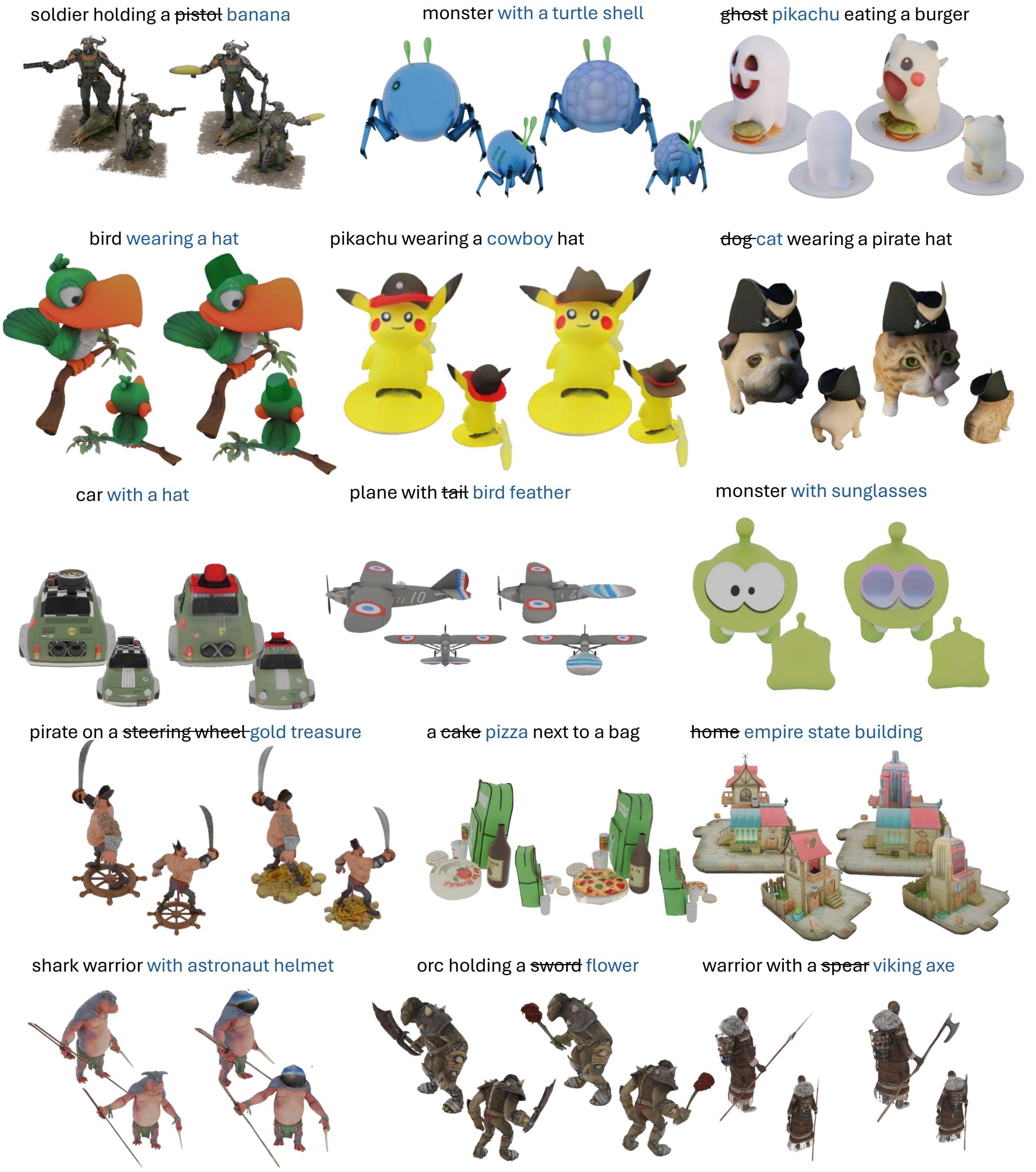
soldier holding a ~~pistol~~ banana
monster with a turtle shell
~~ghost~~ pikachu eating a burger
bird wearing a hat
pikachu wearing a cowboy hat
~~dog~~ cat wearing a pirate hat
car with a hat
plane with ~~tail~~ bird feather
monster with sunglasses
pirate on a ~~steering wheel~~ gold treasure
a ~~cake~~ pizza next to a bag
~~home~~ empire state building
shark warrior with astronaut helmet
orc holding a ~~sword~~ flower
warrior with a ~~spear~~ viking axe

Figure 3. **Additional qualitative results.** Our method can edit complex shapes and scenes.

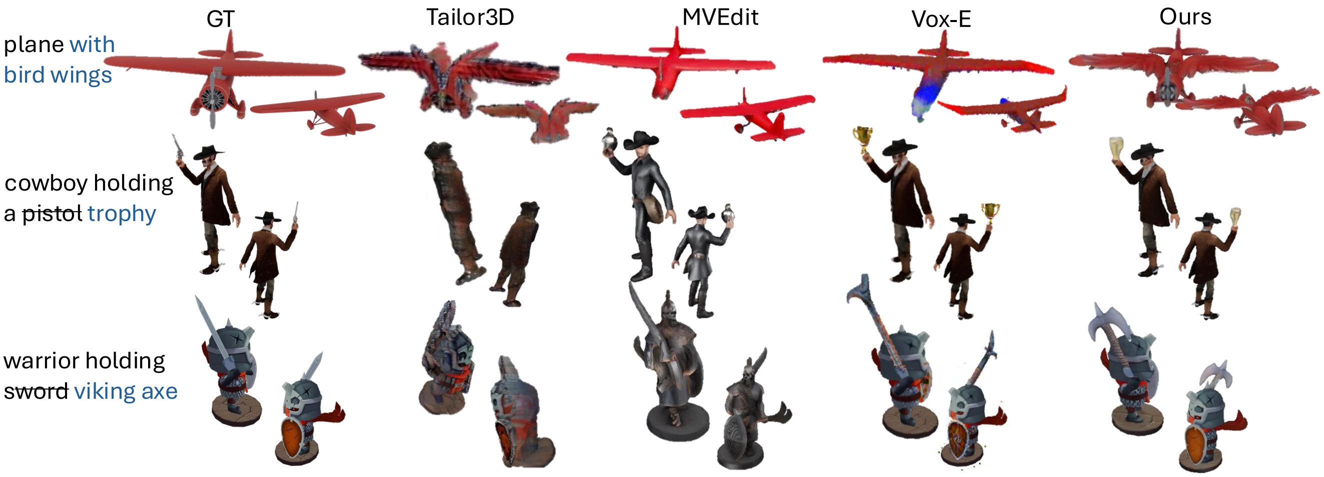
GT
Tailor3D
MVEdit
Vox-E
Ours
plane with bird wings
cowboy holding a ~~pistol~~ trophy
warrior holding ~~sword~~ viking axe

Figure 4. **Additional qualitative baseline comparisons.** Compared to baselines, we can generate more consistent results and can effectively edit complex structures such as a bird wing.

References

[1] Rinon Gal, Or Patashnik, Haggai Maron, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. Stylegan-nada: Clip-guided domain adaptation of image generators. *ACM TOG*, 2022. 1

[2] Inbar Huberman-Spiegelglas, Vladimir Kulikov, and Tomer Michaeli. An edit friendly ddpm noise space: Inversion and manipulations. In *CVPR*, 2024. 1

[3] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023. 1

[4] 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 *ICML*, 2021. 1

[5] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 1

[6] Etai Sella, Gal Fiebelman, Peter Hedman, and Hadar Averbuch-Elor. Vox-e: Text-guided voxel editing of 3d objects. In *ICCV*, 2023. 1

[7] Peiye Zhuang, Songfang Han, Chaoyang Wang, Aliaksandr Siarohin, Jiaxu Zou, Michael Vasilkovsky, Vladislav Shakhrai, Sergey Korolev, Sergey Tulyakov, and Hsin-Ying Lee. Gtr: Improving large 3d reconstruction models through geometry and texture refinement. *arXiv preprint arXiv:2406.05649*, 2024. 1