CTRL-D: Controllable Dynamic 3D Scene Editing with Personalized 2D Diffusion

Supplementary Material

In the supplementary material, we provide additional implementation details (Appendix A), additional experiments (Appendix B), and further discussion of our methods (Appendix C).

A. Implementation Details

Personalization of InstructPix2Pix. We first sample 200 images using the original InstructPix2Pix (IP2P) [4] for prior preservation loss. Next, we fine-tune the model with the image resolutions of 512×384 (monocular scenes) or 384×512 (multi-camera scenes). For data augmentation, we apply the affine transformation function from PyTorch [48] to images with rotation degree $\theta = 15°$, translation $t = (0.1, 0.1)$, and shear $s = 10$. We set the prior preservation weight at $\lambda = 1$. The number of fine-tuning iterations depends on the editing regions; in most cases, 4000 iterations with a batch size of 1 and a learning rate of 10^{-6} are sufficient for effective personalization.

Optimization of Dynamic 3D Gaussians. To update the dataset images, we use our personalized IP2P to edit the images. The IP2P takes three inputs: the image condition C_I, the text condition C_T, and a noisy input z_t. Specifically, we use an original image I_0^v from the dataset as image condition C_I. We render an image I_i^v in the optimization step i, using the current Gaussian model from the same point of view as C_I. Next, let $z_0 = \mathcal{E}(I_i^v)$ where $\mathcal{E}$ is the VAE encoder. For the diffusion model, we perform 20 denoising steps, with an image guidance scale of $s_I = 1.5$ and a text guidance scale of $s_T = 7.5$.

For scene optimization, since the process does not have a single convergence point, we subjectively set the optimization iterations. In our experiments, for monocular scenes, we set the total iteration to 20000 for scenes in which the editing regions are small and 30000 for scenes with large editing regions; for multi-camera scenes, the scenes will converge much sooner, so we set the iterations to 6000. We apply the temporal loss every 10 iterations.

Comparison with Instruct 4D-to-4D [43]. Since Instruct 4D-to-4D (IN4D) employs a parallel optimization approach, which requires two GPUs for optimization, we adopt a similar strategy for our method to compare the running time. Specifically, we implement a parallel version of our method that separates scene optimization and dataset updates across two GPUs. We evaluate both methods on two NVIDIA A40 GPUs. Our method requires approximately 20 minutes for IP2P fine-tuning, 20 minutes for scene optimization on multi-camera scenes, and 40 minutes for monocular scenes on average. We set the maximum iterations for IN4D to 20000, which is consistent with the default setting in their codebase.

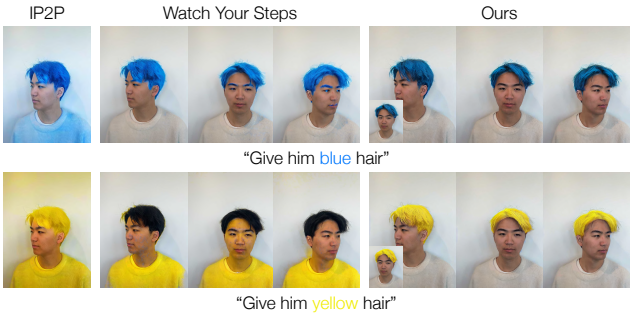

Figure 9. Qualitative comparison with Watch Your Steps [42]. The left-most results are generated by the original IP2P. The reference 2D images for our method appear in the bottom-left corner of our results. Our results demonstrate superior controllability and stability for local editing.

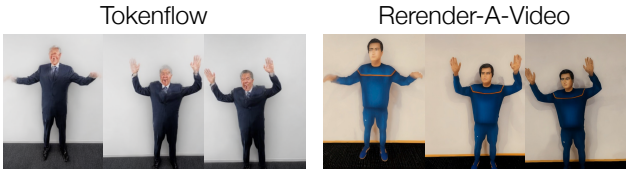

Figure 10. Qualitative comparison with TokenFlow [19] and Rerender-A-Video [69]. Our results are shown in Fig. 5. Our methods perform more precise and stable editing results.

B. Additional Experiments

Comparison with Watch Your Steps [42]. As mentioned in the Introduction, a previous work *Watch Your Steps* (WYS) determines the editing regions based on noise differences. However, this approach is unstable and has several limitations. We compare the results of the original IP2P, WYS, and our personalized IP2P in Fig. 9. In the first row, although WYS focuses more on the desired editing regions, it still edits other areas, such as the eyebrows. In the second row, WYS fails to target the correct editing region, which is the hair. In contrast, our personalized model edits only the hair region, demonstrating the effectiveness of our method in achieving precise control and stability for local editing.

Comparison with TokenFlow [19] and Rerender-A-Video [69]. We conduct the same comparison as in Fig. 5

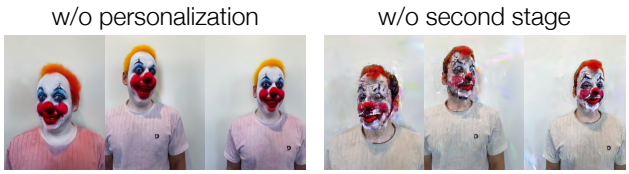

Figure 11. Additional ablation study. The results of our full method are shown in Fig. 4. The results show that our personalization and designed second stage significantly improve the consistency and editing quality.

to evaluate TokenFlow and Rerender-A-Video. The results are shown in Fig. 10. TokenFlow produces poor results, particularly on the face and background. Rerender-A-Video fails to add a suit, focusing mainly on style transfer. These comparisons demonstrate that our approach surpasses methods that edit videos first and then lift them to 3D, as their video editing quality remains suboptimal.

Additional ablation study. We present two ablation studies in Fig. 11. The results of our full method are shown in Fig. 4. Without personalization, edits affect undesired areas such as the T-shirt and cause inconsistent hair colors, as shown on the left of Fig. 11. On the right side, the results without the specifically designed second stage for optimization demonstrate that relying solely on the first stage produces poor results because deformable Gaussians cannot perfectly model the motion.

C. Discussion

Additional Limitation. As mentioned in the main paper, our approach inherits several limitations of the pre-trained IP2P model. Moreover, while the original IP2P performs well in 512×512, its performance degrades when handling images with higher resolutions. This limitation makes it challenging to edit complex scenes from datasets such as the N3DV dataset [33], at higher resolutions, such as 1024×768. At the same time, fine-tuning IP2P on such high resolutions is difficult due to the GPU memory constraints of a single GPU.

Similar to our baseline, Instruct 4D-to-4D, our method also fails to edit sequences with locally complex movements, such as facial expressions. Fine-tuning the model with specific data, e.g., paired data on facial expressions with original and edited sequences, may help the model learn the details better.

Potential Ethical Implications. Leveraging pre-trained generation methods, such as InstructPix2Pix, introduces biases inherent in their training data. While their user-friendliness and ability to produce high-quality results are notable strengths, these features also pose potential risks of misuse. This underscores the need for future work to address ethical concerns through effective bias mitigation strategies and comprehensive reviews. Additionally, while our editing method enables detailed editing of humans and scenes, it raises concerns about potential risks, including creating misleading or deceptive content. To mitigate these risks, future research should carefully review and select training data and model outputs.