3D Geometry-aware Deformable Gaussian Splatting for Dynamic View Synthesis

Supplementary Material

This supplementary material provides additional implementation details and experimental results. First, we provide the implementation details of our proposed method. Then, we provide additional experimental results in the form of visualization and discuss the limitations and impacts of our method. We conclude with discussions on future work. The source code, network model, and results will be released.

1. Implementation Details

1.1. Loss Function

We apply the photometric loss and regularization for our optimization:

$$L_{total} = L_{photo} + \omega L_{motion}, \tag{1}$$

$$L_{photo} = (1-\lambda)L_{rgb} + \lambda L_{D-SSIM}, \tag{2}$$

where L_{rgb} is the L_1 loss and L_{SSIM} is the structural similarity loss between the rendered image $\hat{\mathbf{C}}_t$ and ground truth image $\mathbf{C}_t$. Generally, within a dynamic scene, the proportion of dynamic points is much smaller than that of the static points. Thus the motion amplitude at dynamic points is not too large. We proposed to exploit this fact by introducing the motion regularization term $L_{motion} = \|\Delta \mathbf{x_t}\|_1$. In our experiments, we set $\lambda = 0.2$ and $\omega = 0.01$.

1.2. Network Architecture

Here, we introduce the network architecture adopted in our method. The Gaussian Canonical Field consists of two branches: the geometric branch and the identity branch. As shown in Fig. 1, the geometric branch takes the position of voxel points as input and outputs the geometrical features f_{geo}. It is roughly composed of three parts, namely DownVoxelBlock, ResidualBlock, and UpVoxelBlock. The specific structures of these three parts are shown in Fig. 2. For the identity branch, we use a simple MLP to get the embedding features $f_{identity}$, which maintains the independence of point features. Then we concatenate the features from the geometric branch and the identity branch, and pass them into another MLP to get fused features $\mathbf{F}_{\text{fuse}}$. Finally, we take the fused features $\mathbf{F}_{\text{fuse}}$, position of Gaussians $\gamma(x)$ and time $\gamma(t)$ into a decoder to get the deformations of position, rotation, and scale from the canonical space to time space. In Fig. 4, we demonstrate the specific structure of MLPs. Additionally, the intermediate hidden layers are shown in blue, the number inside each block signifies the vector's dimension. All layers are standard fully-connected layers, black arrows between layers indicate the ReLU activations. $\gamma(\cdot)$ is a positional encoding function, we use $L = 10$ for position, and $L = 6$ for timestamp. Similar to NeRF [10], we use a skip connection that concatenates the input to the third layer.

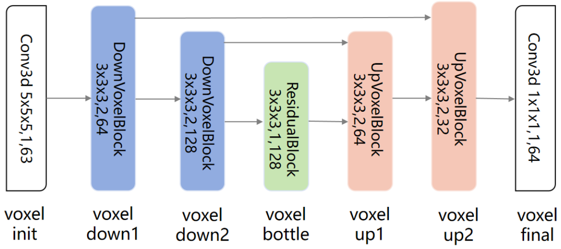

Figure 1. Overall architecture of the geometric branch, which captures local geometric features using a 3D U-Net.

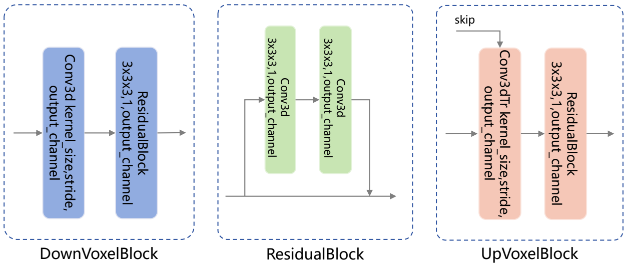
Conv3d kernel_size,stride, output_channel
ResidualBlock 3x3x3,1,output_channel
Conv3d 3x3x3,1,output_channel
Conv3d 3x3x3,1,output_channel
skip
Conv3dTr kernel_size,stride, output_channel
ResidualBlock 3x3x3,1,output_channel
DownVoxelBlock
ResidualBlock
UpVoxelBlock

Figure 2. Detailed structure of DownVoxelBlock, ResidualBlock, and UpVoxelBlock.

2. Results and Discussions

2.1. Results on Neural 3D Video dataset

We further evaluated our method on Neural 3D Video dataset [9], which includes several videos captured with synchronized fixed GoPro camera system. We have evaluated our method in the following four scenarios: Cook Spinach, Cut Roast Beef, Flame Steak and Sear Steak, each scene includes from 17 to 20 cameras for training and one central camera for evaluation. Following previous works, we downsample the images to 1352 × 1014 and report the per-scene PSNR, SSIM and LPIPS for each method, as shown in Table 1. We find our method is struggling in these long-time series. Although our method maintains high fidelity restoration in static regions, its capability is severely limited in dynamic regions.

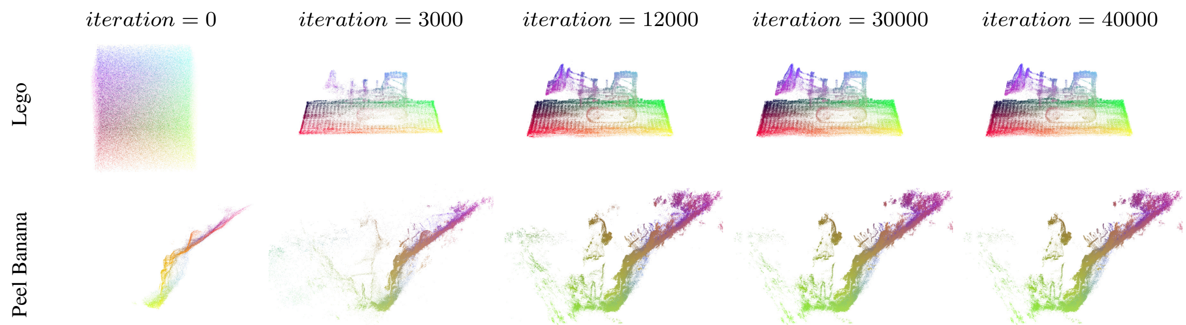

Figure 3. **Visualization of Canonical Point Cloud.** We show the evolution of point clouds in the canonical space with respect to the number of iterations.

Table 1. Quantitative results on scenes from the Neural 3D Video Synthesis

Scene	Cook Spinach			Cut Roast Beef			Flame Steak			Sear Steak		
Method	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
MixVoxels [15]	31.39	0.931	0.113	31.38	0.928	0.111	30.15	0.938	0.108	30.85	0.940	0.103
K-Planes [4]	31.23	0.926	0.114	31.87	0.928	0.114	31.49	0.940	0.102	30.28	0.937	0.104
Hexplanes‡ [2]	31.05	0.928	0.114	30.83	0.927	0.115	30.42	0.939	0.104	30.00	0.939	0.105
Hyperreel [1]	31.77	0.932	**0.090**	**32.25**	0.936	**0.086**	31.48	0.939	**0.083**	31.88	0.942	**0.080**
NeRFPlayer† [14]	30.58	0.929	0.113	29.35	0.908	0.144	31.93	0.950	0.088	29.13	0.908	0.138
StreamRF [8]	30.89	0.914	0.162	30.75	0.917	0.154	31.37	0.923	0.152	31.60	0.925	0.147
SWAGS [13]	**31.96**	0.946	0.094	31.84	**0.945**	0.099	**32.18**	0.953	0.087	32.21	0.950	0.092
Ours	31.39	**0.947**	0.144	29.87	0.944	0.156	31.35	**0.954**	0.129	**32.62**	**0.955**	0.130

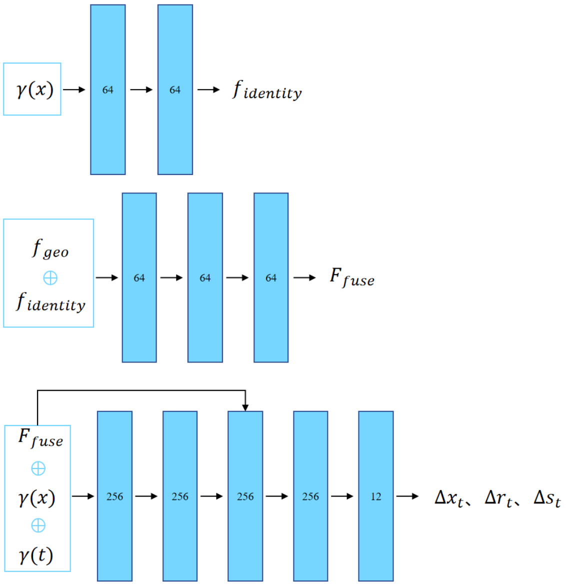

Figure 4. Detailed structure of MLPs we have used in our method.

2.2. More Visualization Results

Point Cloud For the D-NeRF synthetic scenes [12], we randomly initialize 150000 points as the initial point cloud. We visualize the point cloud of the scene in the canonical space with different iterations. In Fig. 3, it can be observed that we can reconstruct the scene even from a random point cloud. Moreover, in complex scenes such as Peel Banana in the HyperNeRF dataset [11], we can also reconstruct the scene even if there are no dynamic parts in the input point clouds, as shown in Fig. 3. Our supplementary video also presents the trajectory of the scene's point cloud as it evolves over time. Our supplementary video is available at our homepage: https://npucvr.github.io/GaGS/.

Quantitative Results We show more qualitative comparisons in Fig. 6 and Fig. 7 for D-NeRF synthetic dataset [12] and HyperNeRF dataset [11]. In our supplementary video, we also showcase the temporal interpolation capability of our method when maintaining a fixed camera viewpoint while time evolves. Additionally, we demonstrate the ability to synthesize novel viewpoints while keeping the time fixed and observing the scene from arbitrary viewpoints.

Temporal Interpolation We show the temporal interpolation ability of our method. In Fig. 8 and Fig. 9, we fix the camera viewpoint and show the results for temporal changes of the D-NeRF synthetic dataset [12] and HyperNeRF dataset [11]. Our method shows great temporal interpolation abilities for both synthetic and real datasets. More results are presented in our homepage.

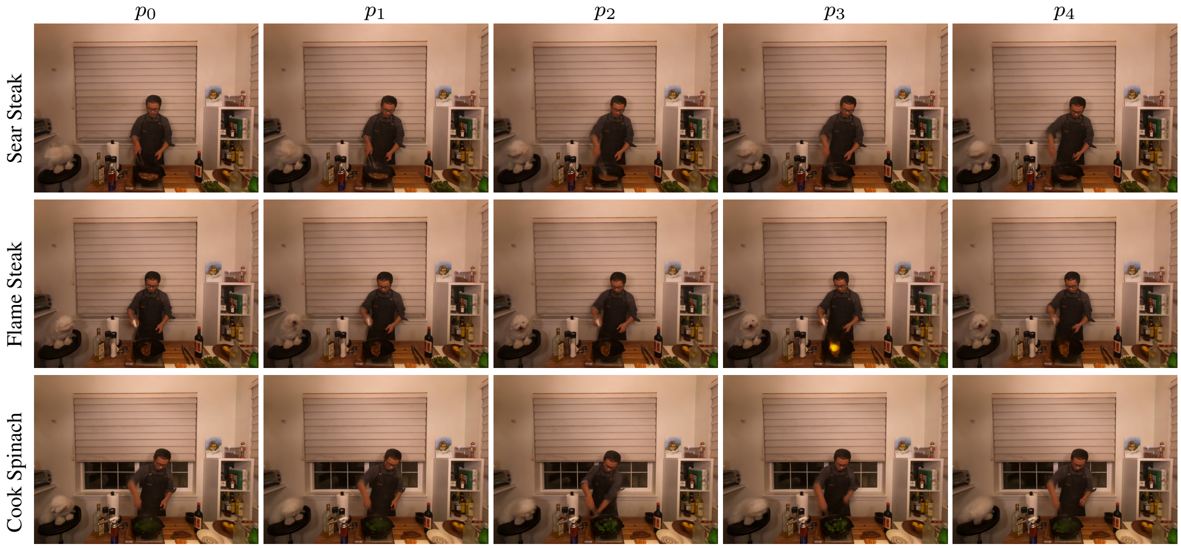

Figure 5. Results on Neu3DV dataset.

2.3. Limitations and Impacts

Limitations First, our proposed method represents the deformation of Gaussians from the canonical space to time space. However, it can only chronicle a point within the scene from start to finish, lacking the capability to depict a point that abruptly emerges or disappears in the scene at a specific moment. Second, our proposed method essentially describes the motion and deformation of points in the canonical space. It necessitates acquiring precise camera poses in advance. However, in the context of dynamic scene modeling, obtaining accurate camera poses is inherently very challenging. Our approach is also constrained by this limitation. Last, our method struggles to describe excessively complex motions and long time videos, such as rapid movements of objects within the scene. This challenge results in the network facing difficulties in estimating point motions, ultimately leading to failures, as shown in Fig. 5, we provide some cases in the test camera on Neu3DV dataset [9]. Due to the lack of explicit modeling of motion, our method exhibits insufficient capability in capturing fine-grained movements over long temporal sequences. However, it still maintains the ability to describe general motions, such as the swinging of curtains and human body movements.

Broader Impacts Our proposed method can be applied to various industries, including visual effects synthesis in the film industry, game modeling, autonomous driving simulation, and more. For the film industry and game modeling, dynamic scenes can be synthesized by our method. In autonomous driving simulation, our proposed method can provide more data from different viewpoints, which will contribute to the advancement of autonomous driving.

2.4. Future Work

In the future, we plan to exploit the motion mask to distinguish the dynamic points and static points of the scene, which will decrease the computing resource by only estimating the deformation of dynamic points. Also, we will investigate explicit motion modeling by exploiting the foreground and background motion segmentation cues.

References

[1] Benjamin Attal, Jia-Bin Huang, Christian Richardt, Michael Zollhoefer, Johannes Kopf, Matthew O'Toole, and Changil Kim. HyperReel: High-fidelity 6-dof video with ray-conditioned sampling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[2] Ang Cao and Justin Johnson. HexPlane: A fast representation for dynamic scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[3] Jiemin Fang, Taoran Yi, Xinggang Wang, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Matthias Nießner, and Qi Tian. Fast dynamic radiance fields with time-aware neural voxels. In *Proceedings of the Conference on Computer Graphics and Interactive Techniques in Asia (SIGGRAPH ASIA)*, 2022. 4, 5

[4] Sara Fridovich-Keil, Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa. K-Planes: Explicit radiance fields in space, time, and appearance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[5] Xiang Guo, Guanying Chen, Yuchao Dai, Xiaoqing Ye, Jiadai Sun, Xiao Tan, and Errui Ding. Neural deformable voxel grid for fast optimization of dynamic view synthesis.

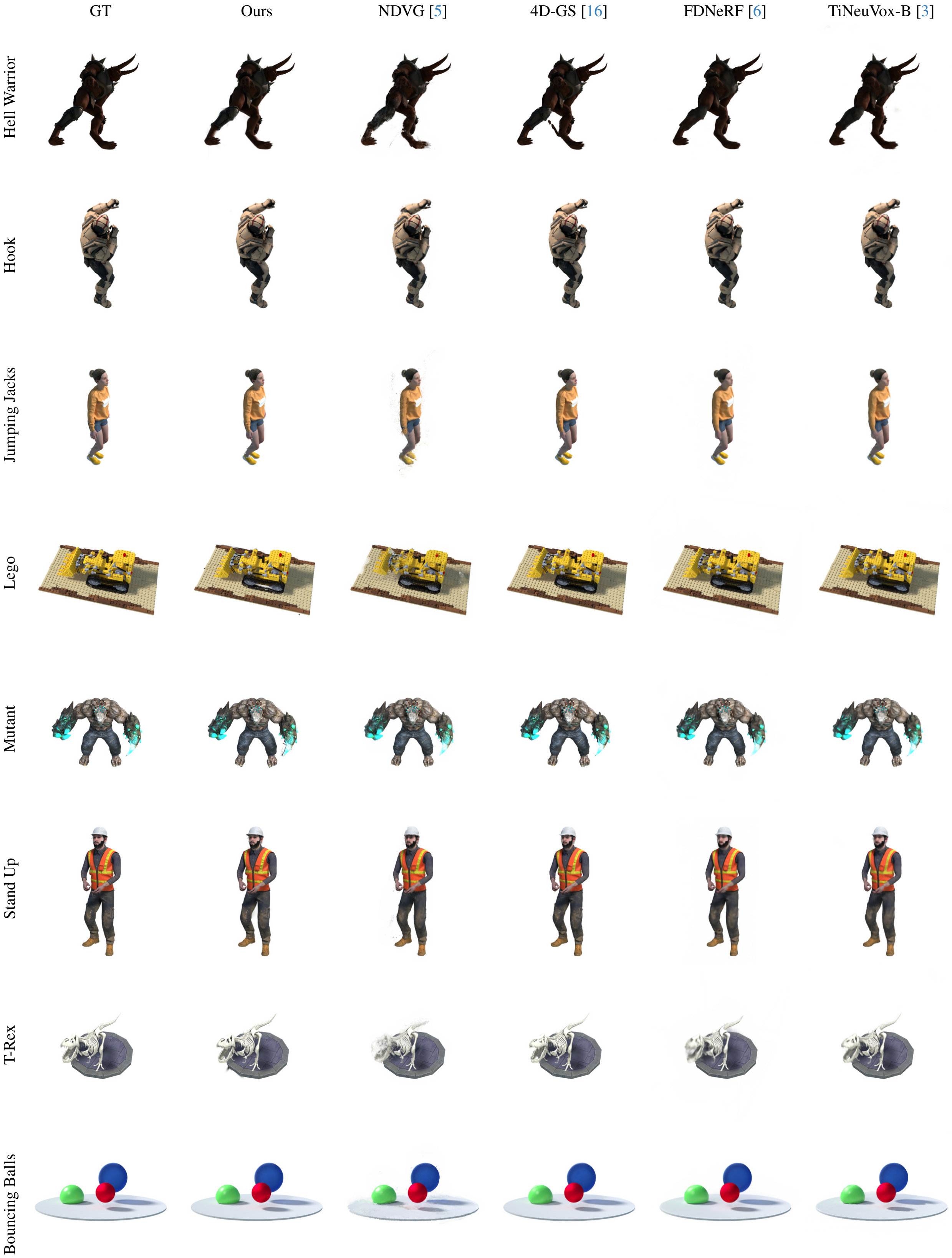

Figure 6. **Qualitative comparison on the D-NeRF synthetic dataset.** We show synthesized images on the D-NeRF synthetic dataset of our method and other competing methods.

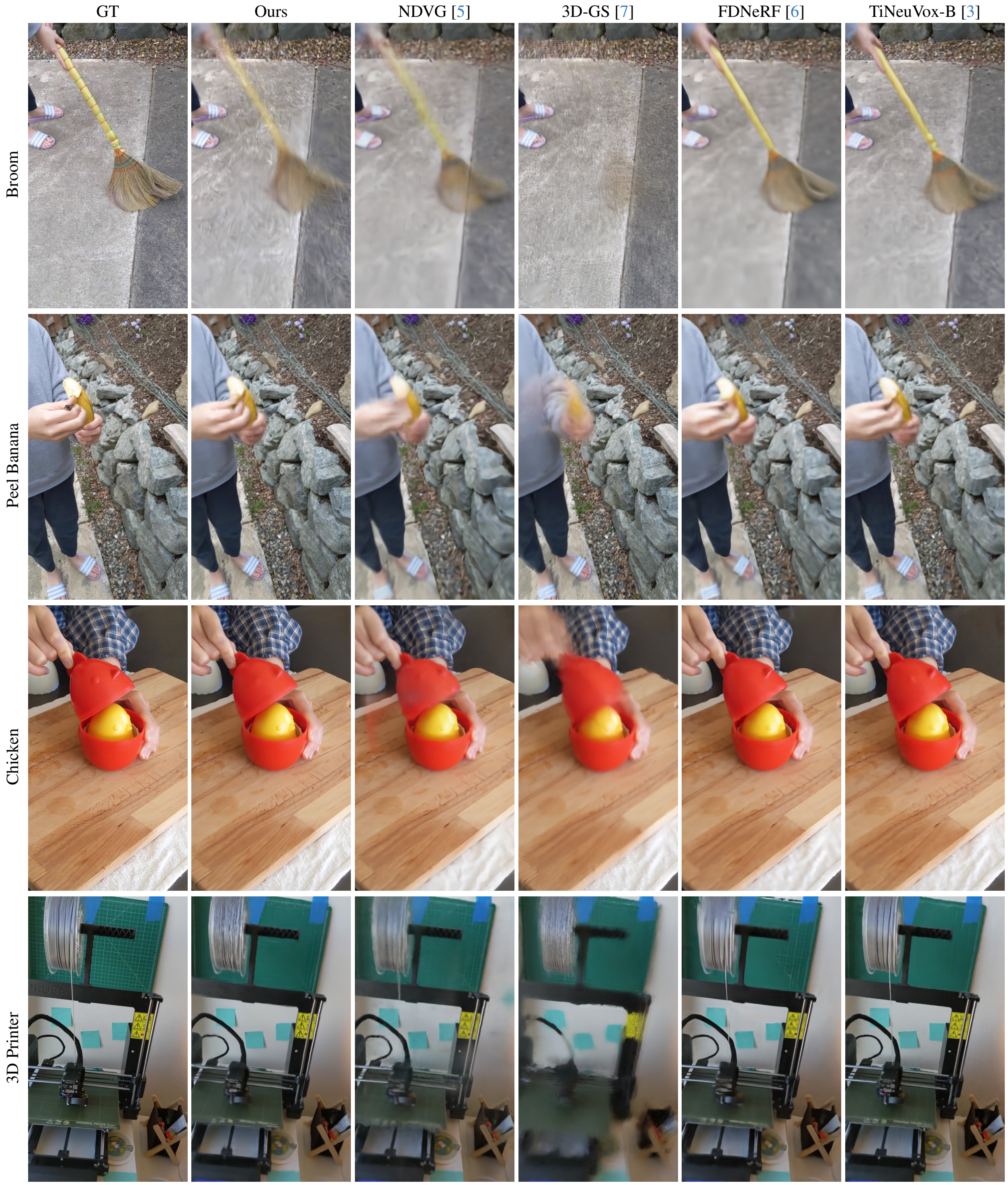

Figure 7. **Qualitative comparison on the HyperNeRF dataset.** We show synthesized images on the HyperNeRF dataset of our method and other competing methods.

In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, 2022. 4, 5

[6] Xiang Guo, Jiadai Sun, Yuchao Dai, Guanying Chen, Xiaoqing Ye, Xiao Tan, Errui Ding, Yumeng Zhang, and Jingdong Wang. Forward flow for novel view synthesis of dynamic scenes. In *Proceedings of the IEEE International Confer-*

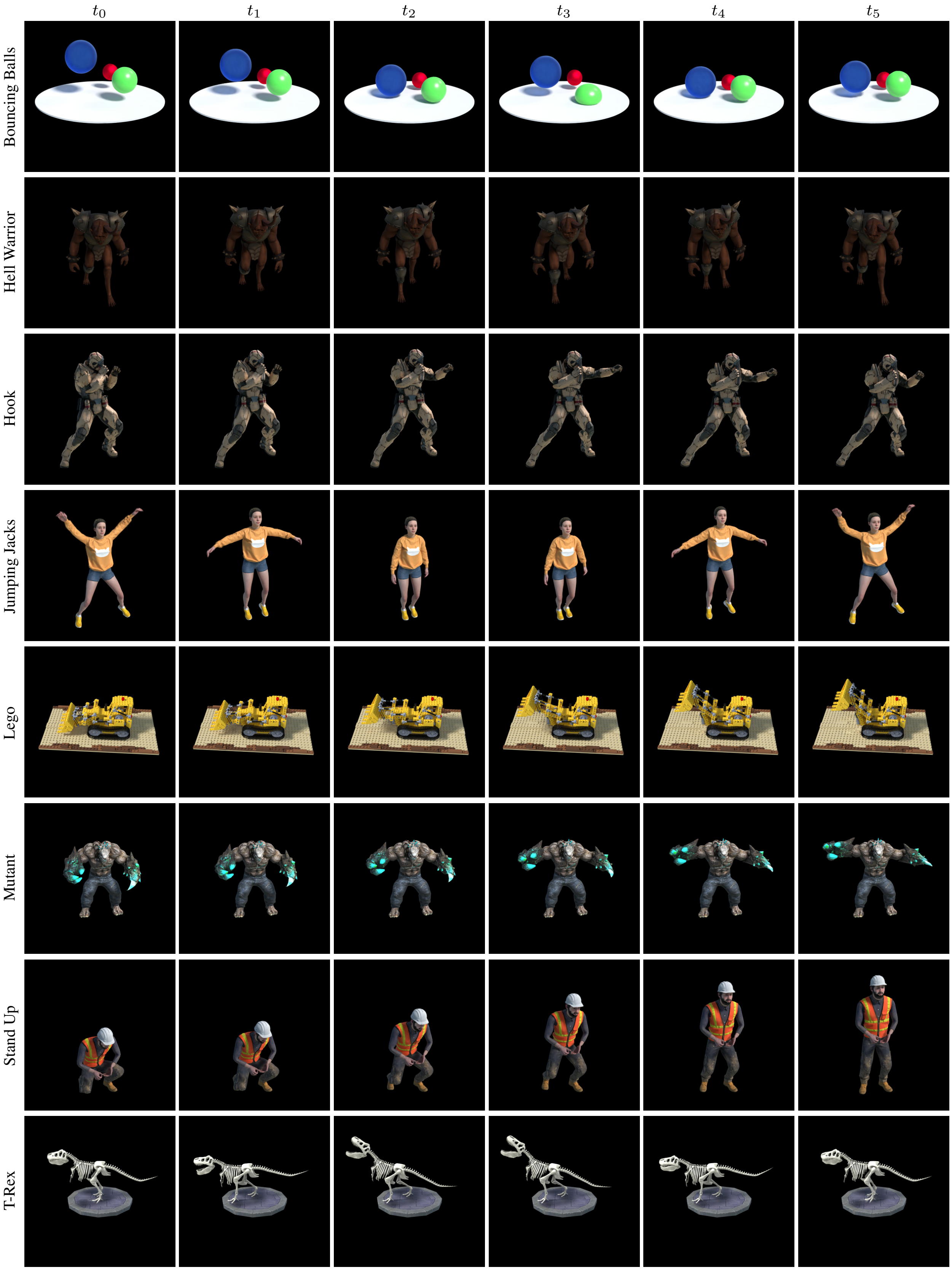

Figure 8. **Temporal Interpolation Capability on the D-NeRF synthetic dataset.** We show the temporal interpolation capabilities of our method. Specifically, we showcase our ability to perform time interpolation by maintaining a fixed camera viewpoint while observing the temporal changes in scene content.

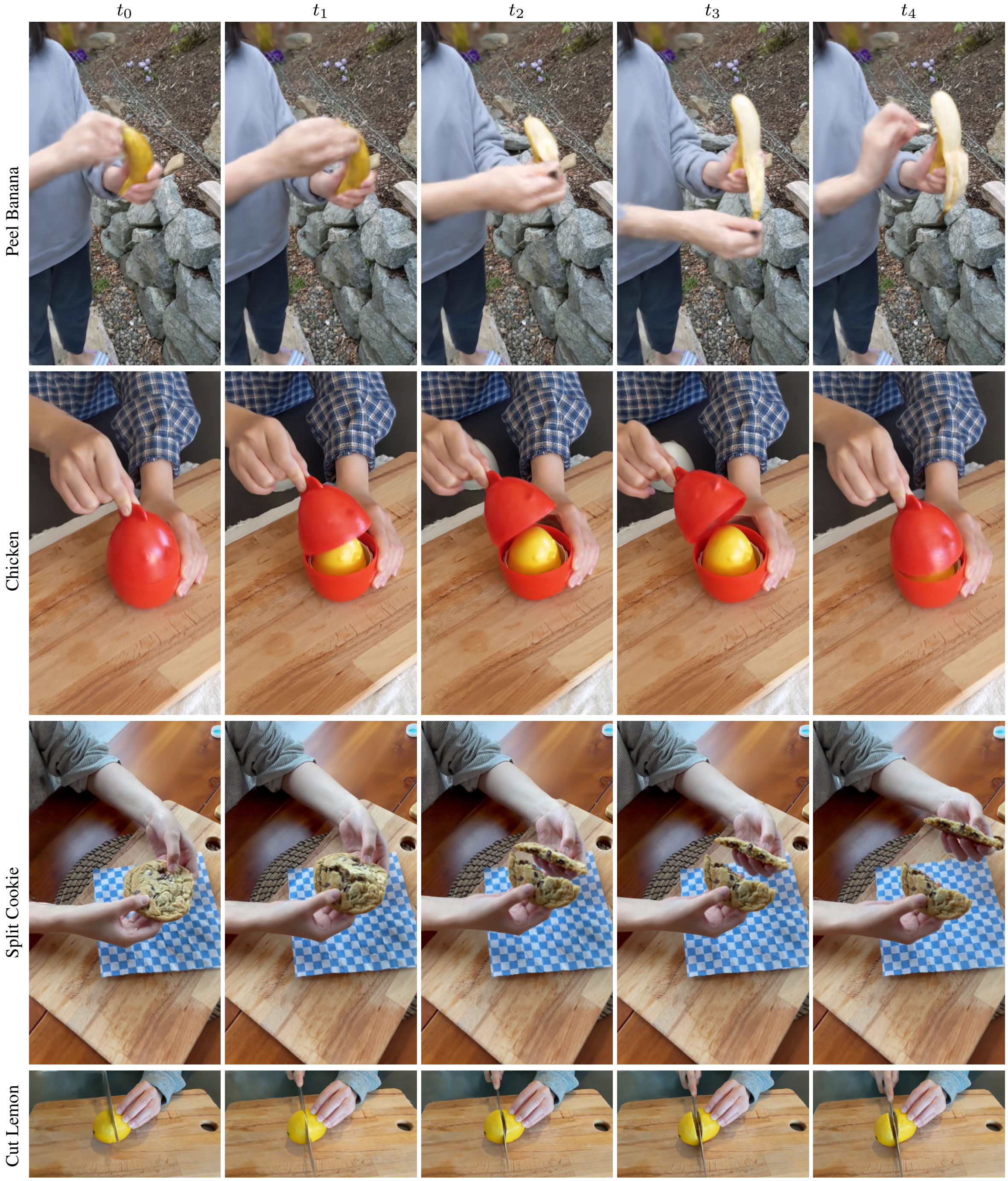

Figure 9. **Temporal Interpolation Capability on HyperNeRF dataset.** We show the temporal interpolation capabilities of our method. Specifically, we showcase our ability to perform time interpolation by maintaining a fixed camera viewpoint while observing the temporal changes in scene content.

ence on Computer Vision (ICCV), 2023. 4, 5

[7] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (TOG)*, 2023. 5

[8] Lingzhi Li, Zhen Shen, Zhongshu Wang, Li Shen, and Ping Tan. Streaming radiance fields for 3D video synthesis. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 2

[9] Tianye Li, Mira Slavcheva, Michael Zollhoefer, Simon

Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, Richard Newcombe, et al. Neural 3D video synthesis from multi-view video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 3

[10] 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 *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 1

[11] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M Seitz. HyperNeRF: A higher-dimensional representation for topologically varying neural radiance fields. *ACM Transactions on Graphics (TOG)*, 2021. 2

[12] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-NeRF: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[13] Richard Shaw, Jifei Song, Arthur Moreau, Michal Nazarczuk, Sibi Catley-Chandar, Helisa Dhamo, and Eduardo Perez-Pellitero. SWAGS: Sampling windows adaptively for dynamic 3D Gaussian splatting. *arXiv preprint arXiv:2312.13308*, 2023. 2

[14] Liangchen Song, Anpei Chen, Zhong Li, Zhang Chen, Lele Chen, Junsong Yuan, Yi Xu, and Andreas Geiger. NeRFPlayer: A streamable dynamic scene representation with decomposed neural radiance fields. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 2023. 2

[15] Feng Wang, Sinan Tan, Xinghang Li, Zeyue Tian, Yafei Song, and Huaping Liu. Mixed neural voxels for fast multi-view video synthesis. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2023. 2

[16] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4D Gaussian splatting for real-time dynamic scene rendering. *arXiv preprint arXiv:2310.08528*, 2023. 4