Swift4D: Adaptive divide-and-conquer Gaussian Splatting for compact and efficient reconstruction of dynamic scene

Jiahao Wu, Rui Peng, Zhiyan Wang, Lu Xiao, Luyang Tang, Jinbo Yan, Kaiqiang Xiong, Ronggang Wang*

Peking University

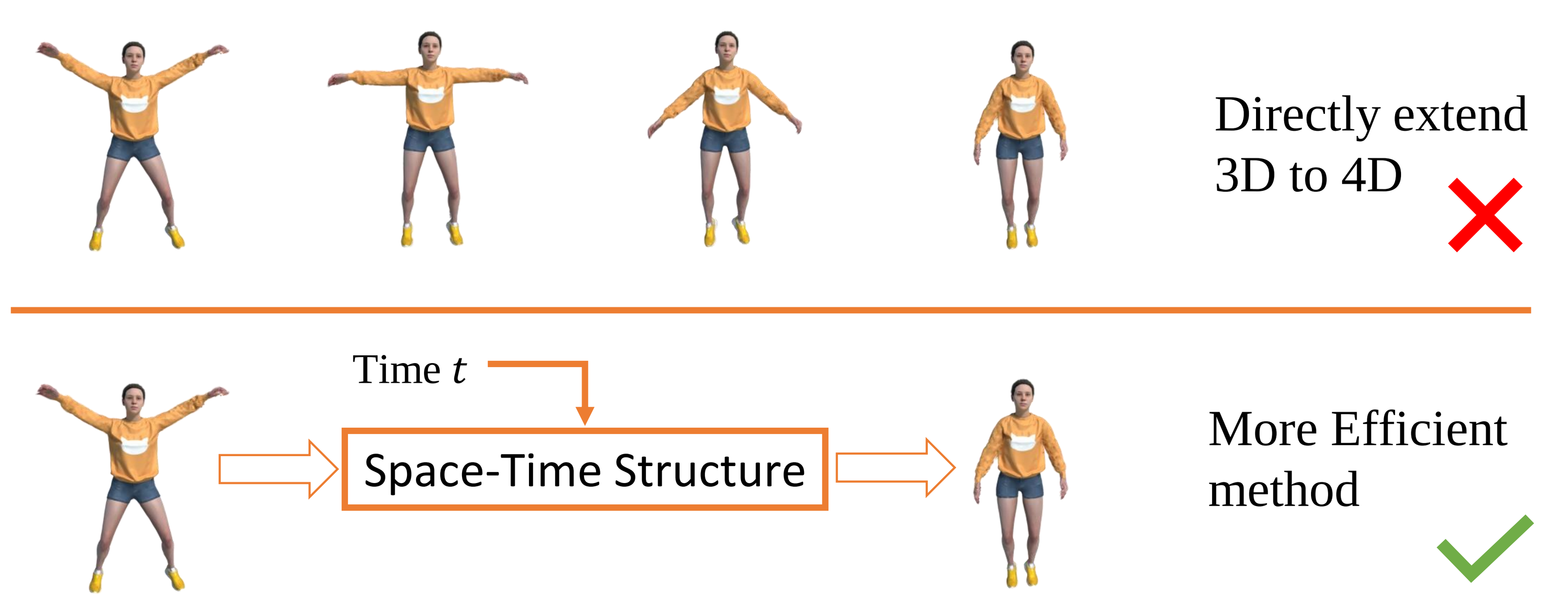
Scan the QR code to get the paper code, weight and datasets

Background

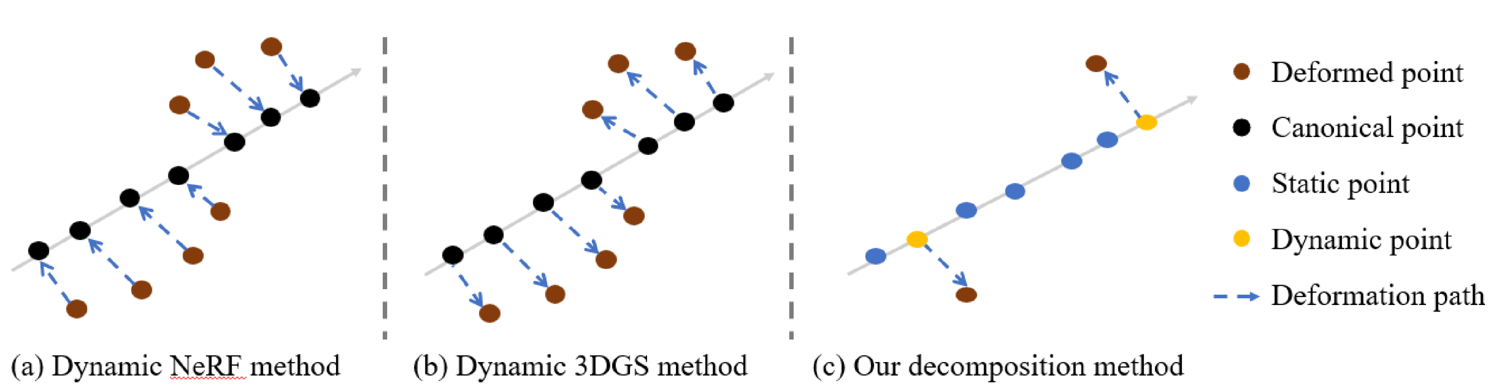
Despite the substantial progress of novel view synthesis (NVS), existing methods, either based on the Neural Radiance Fields (NeRF) or more recently 3D Gaussian Splatting (3DGS), NVS for dynamic scene still a challenge and meaningful task.

Challenges of 3DGS for dynamic scene recon.

Unlike 3D static scene reconstruction, dynamic scene reconstruction requires the introduction of a temporal dimension, making it a 4D reconstruction. However, directly extending 3D to 4D results in significant computational overhead and storage costs. A more efficient approach is to fully leverage the temporal consistency within the scene and incorporate a compact 4D spatiotemporal structure to provide dynamic information, enabling faster and more compact 4D reconstruction.



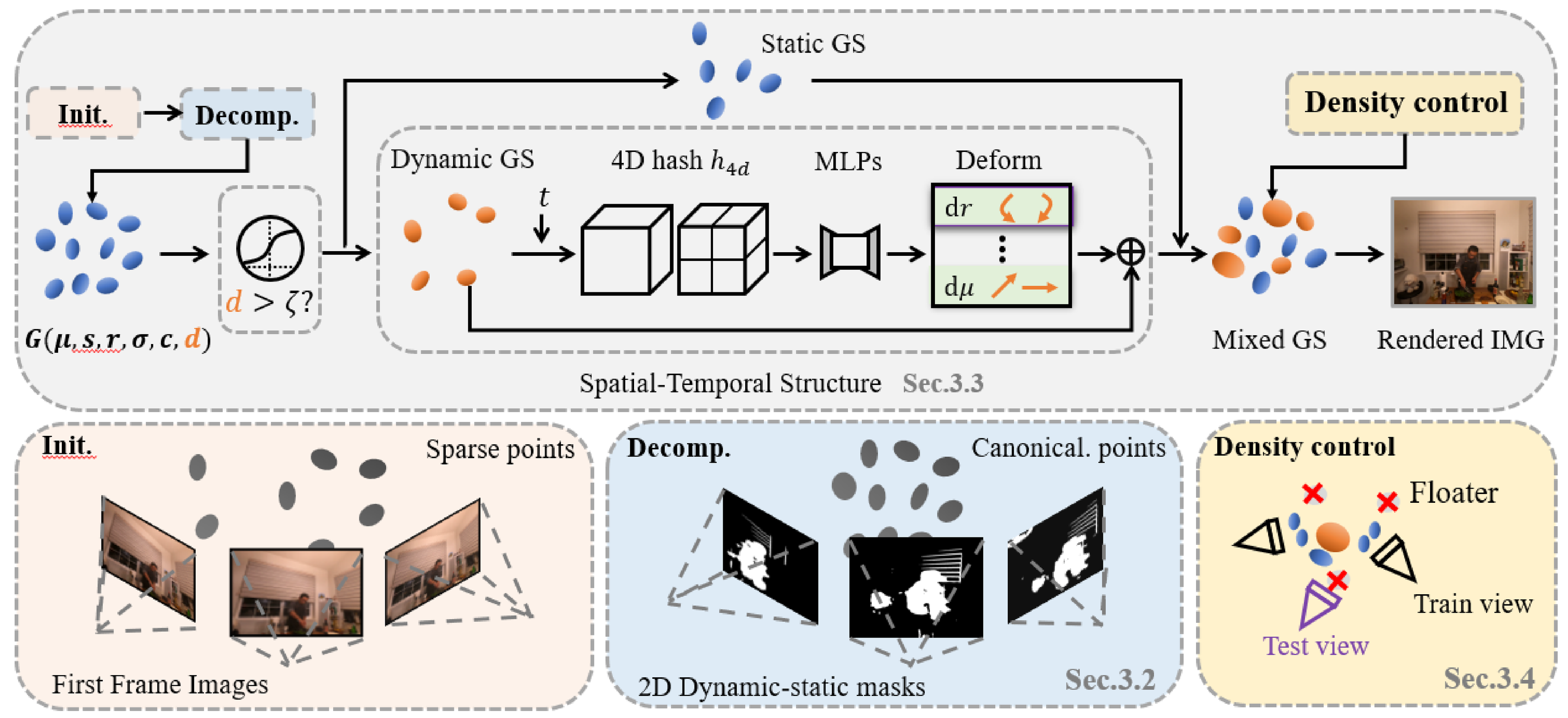
Dynamic scene NVS method compare.



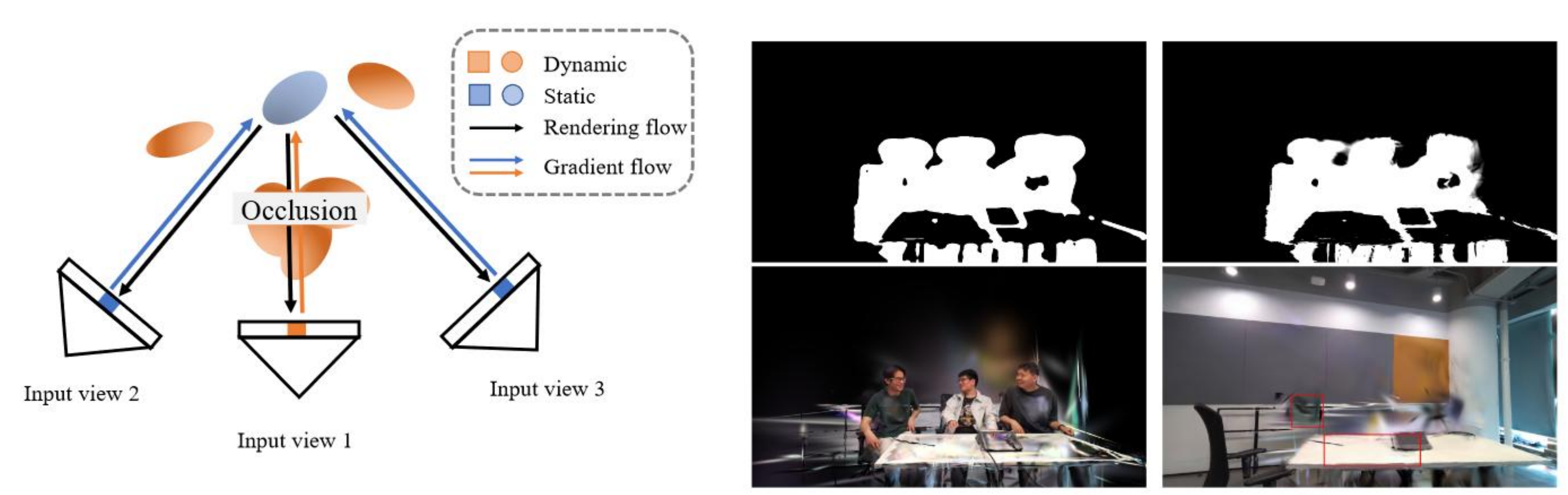
(a) D-NeRF proposes mapping deformation field points to canonical space, a widely adopted practice in NeRF-based methods;
(b) 4DGaussian and DeformableGS propose mapping canonical space points to the deformation field;
(c) We propose dividing the points in canonical space into dynamic and static, and then mapping only the dynamic points to the deformation space.

Method overview

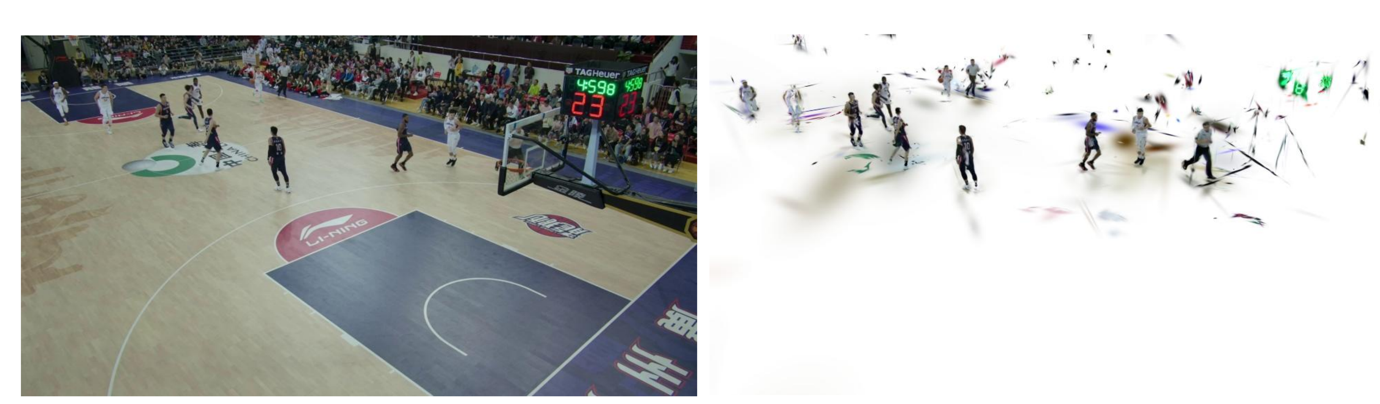
In this paper we propose **Swift4D**, a divide-and-conquer 3D Gaussian Splatting method that can handle static and dynamic primitives separately, achieving a good trade-off between rendering quality and efficiency, motivated by the fact that most of the scene is the static primitive and does not require additional dynamic properties. Concretely, we focus on modeling dynamic transformations only for the dynamic primitives which benefits both efficiency and quality. This divide-and-conquer method facilitates efficient training and reduces storage redundancy.



Dynamic and static decomposition

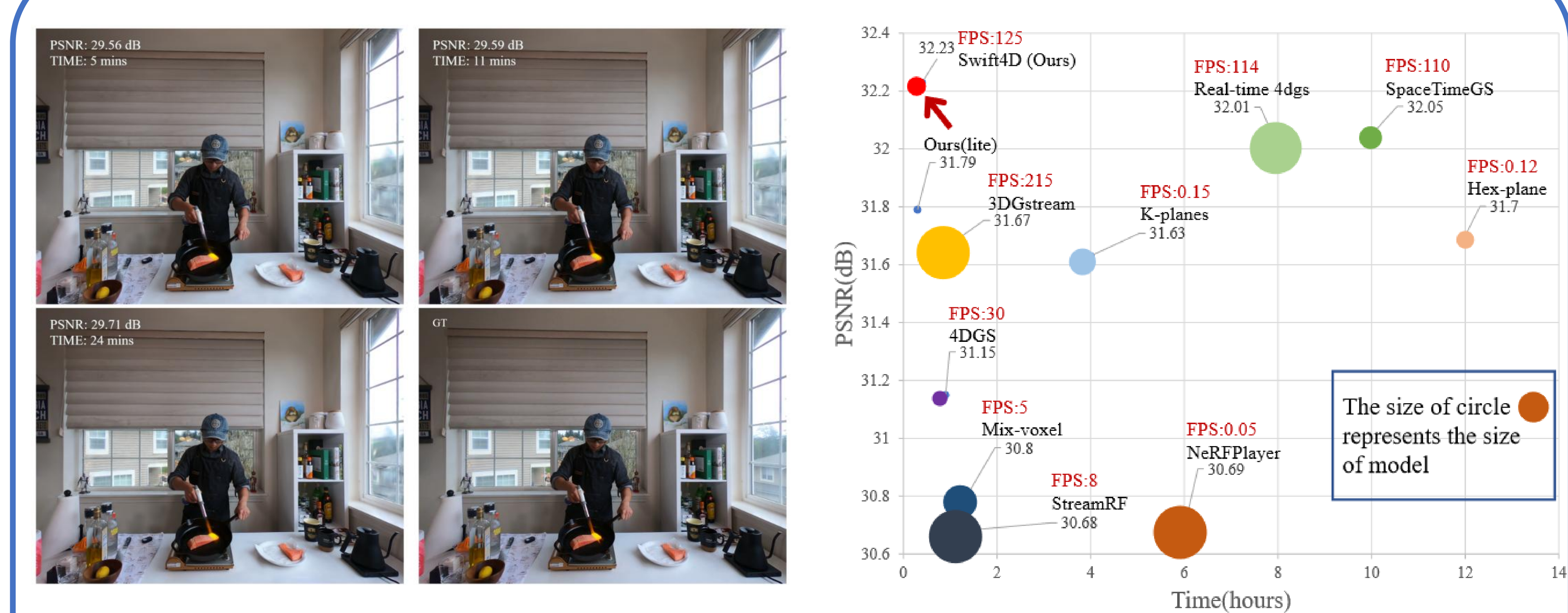


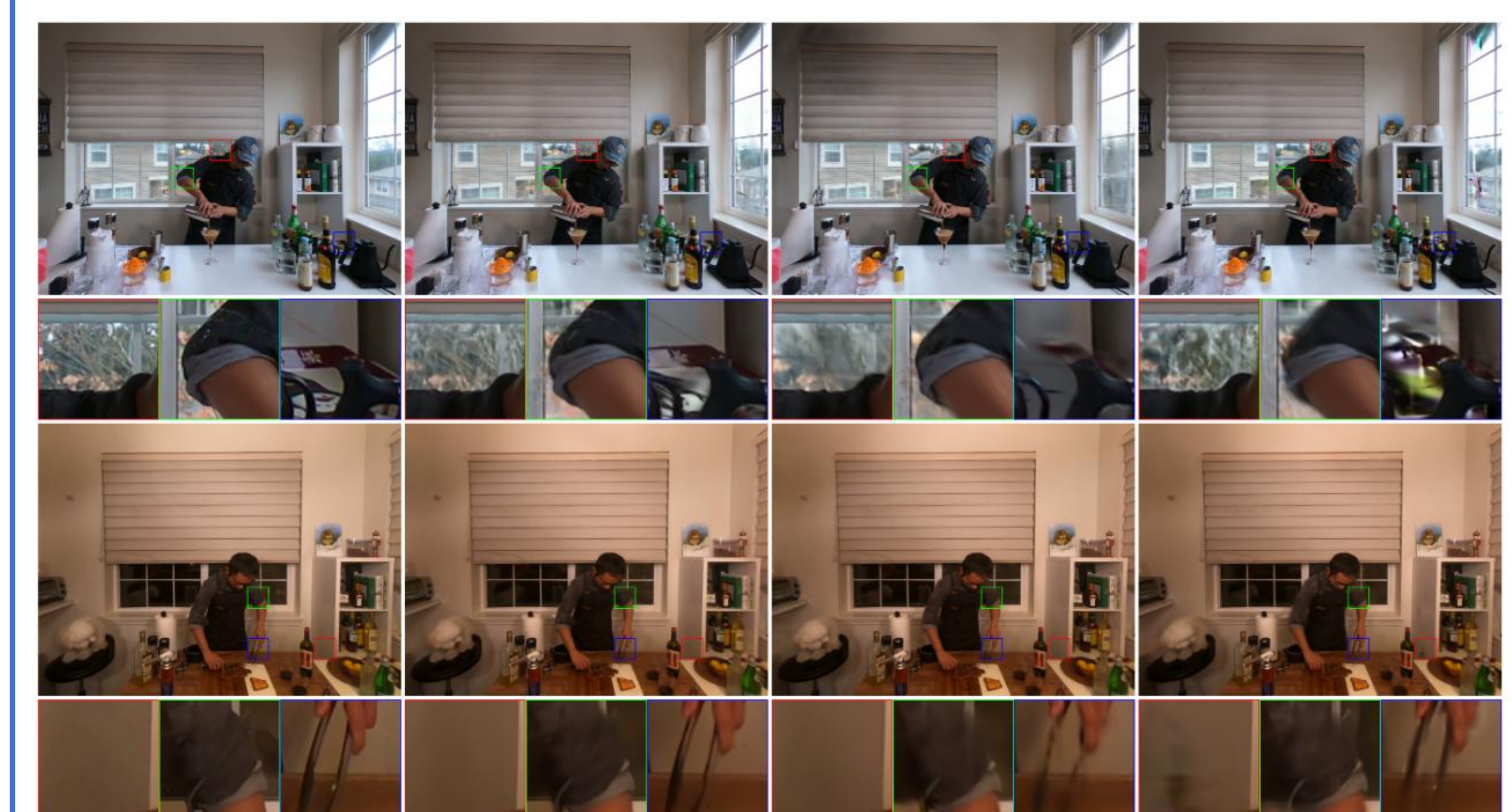
- Calculate the dynamic and static masks.
- Render the parameter d to obtain the dynamic value images.
- Calculate the loss between the dynamic and static mask and dynamic value images to optimize the parameter d



Our method can also be applied to extend 2D segmentation to 3D segmentation. Finally, welcome to use our dataset testing your methods.

Experiments

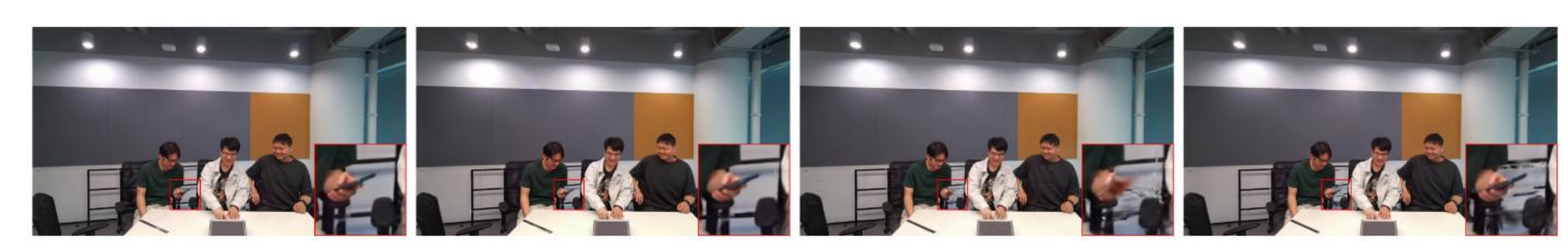




(a) GT (b) Ours (c) RTGS (d) STGS

Table 1: **Quality comparison on the N3DV dataset.** The best and the second best results are denoted by red and blue. [1] online method.

Method	PSNR ↑	DSSIM ↓	LPIPS ↓	Time ↓	Size(MB) ↓	FPS ↑
DyNeRF Li et al. (2022b)	29.58	0.020	0.099	1300.0 hours	30	0.02
NeRFPlayer Song et al. (2023)	30.69	-	0.111	6.0 hours	5100	0.05
HexPlane Cao & Johnson (2023)	31.70	0.014	0.075	12.0 hours	240	0.21
K-Planes Fridovich-Keil et al. (2023)	31.63	0.018	-	5.0 hours	300	0.15
4DGS Wu et al. (2024)	31.02	0.030	0.150	50 mins	90	30
3DGStream [1] Sun et al. (2024)	31.67	-	-	60 mins	2340	215
SpaceTimeGS Li et al. (2024)	32.05	0.014	0.044	10.0 hours	200	110
Real-Time4DGS Yang et al. (2023)	32.01	0.014	0.055	9.0 hours	> 1000	114
Swift4DLite(Ours)	31.79	0.017	0.072	20 mins	30	128
Swift4D(Ours)	32.23	0.014	0.043	25 mins	120	125



(a) GT (b) Ours (c) 3DGStream (d) 3DGS