

4C4D: 4 Camera 4D Gaussian Splatting

Supplementary Material

1. Details of Capture System

1.1. Hardware

We build our capture system using four GoPro HERO 12 cameras, a modern consumer-grade portable device that is easily accessible to ordinary users. The entire setup, including the cameras and custom mounting brackets, costs less than 1,500 USD. All four cameras record video with identical settings at a resolution of 1920×1080 and a frame rate of 60 frames per second.

1.2. Capturing Pipeline

We position four cameras facing the region where the dynamic actions occur. The four viewpoints span an effective coverage of approximately 100–120 degrees, providing sufficient overlap to fully observe the scene without causing the severe difficulties associated with non-overlapping views. Due to the inherent temporal misalignment of GoPro devices, we first perform multi-view frame re-alignment to ensure perfect synchronization across all videos.

To obtain accurate camera poses, we use COLMAP [5] to solve the extrinsics before recording videos. Estimating stable poses from only four photos can often be unreliable. Therefore, we capture an additional set of eight images of the same scene to improve multi-view feature matching and enhance pose stability. This step does not involve any extra cameras, where the same four cameras are used to take photos from additional viewpoints. After computing all camera poses, only the four main viewpoints are used for video capture, and only their corresponding poses are utilized in the subsequent 4D Gaussian Splatting training.

1.3. The Dyn4Cam Dataset

We finally collect eight representative action categories to form the Dyn4Cam dataset, including Boxing, Dancing, Chest Fly, Exercising, Jumping Clap, Running, Taichi, and Showing a Toy. These actions span both fast and relatively slow motions and include simple daily movements as well as complex athletic behaviors. This composition ensures good diversity and provides a comprehensive benchmark for dynamic scene reconstruction in sparse views.

2. Implementation Details

We implement 4C4D in PyTorch and adopt the Adam optimizer [1] to optimize the Gaussian primitives. We adopt the MAST3R [3] for initialization, which is more stable in the sparse-camera setting. To ensure stable warm-up, we introduce the Neural Decaying Function after 500 iterations,

which provides more reliable supervision to the neural network and helps stabilize the optimization process.

3. More Experimental Results

3.1. Results on Neural3DV dataset

We report the complete evaluation metrics for each scene in the Neural3DV dataset in Tab. 1 and Tab. 2. Since the 'Flame Salmon' scene is significantly longer than the other five scenes, we restrict the evaluation to its first 300 frames to ensure a consistent experimental setup. Additional visual comparisons are provided in Fig. 1, where all baselines are rendered using the same camera trajectory.

3.2. Results on ENeRF-Outdoor dataset

Tab. 3 presents a detailed breakdown of the per-scene quantitative results on the ENeRF-Outdoor dataset, showcasing the performance of all competing methods across a diverse set of outdoor sequences. To complement these numerical comparisons, additional qualitative visualizations are provided in Fig. 2.

3.3. Results on Dyn4Cam dataset

Additionally, we provide more comprehensive qualitative comparisons with state-of-the-art baselines in Fig. 3 and Fig. 4. These visual examples cover four representative scenes from the Dyn4Cam dataset, namely "Taichi," "Dancing," "Boxing," and "Exercising." These sequences span a wide range of motion patterns and appearance variations. As shown in the figures, our approach consistently produces sharper details, more stable temporal behaviors, and cleaner geometry across different motion types.

4. Video

We provide a video in the supplementary material that includes comparisons between 4C4D and state-of-the-art baselines, along with additional visualizations on our self-captured Dyn4Cam dataset.

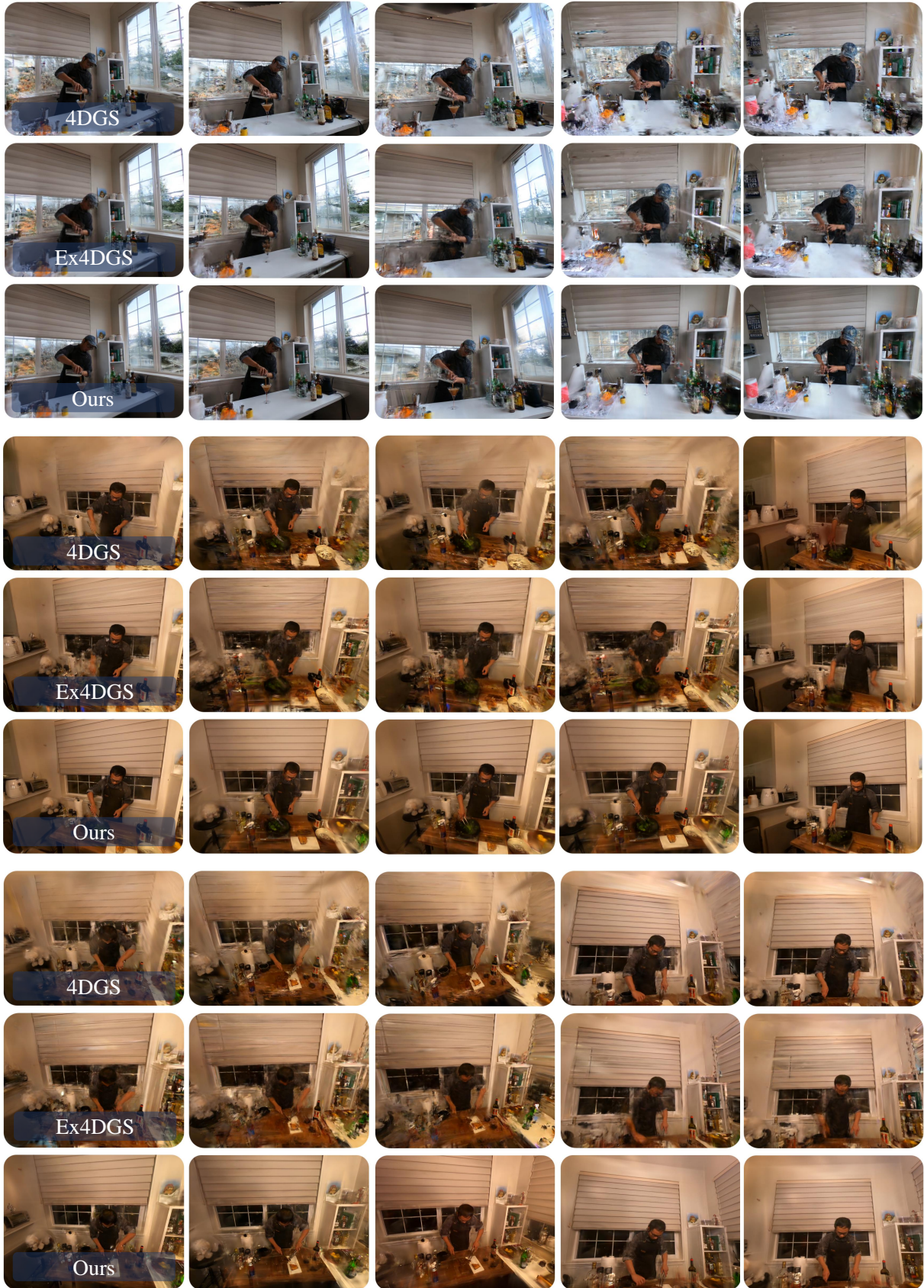


Figure 1. More visual comparisons under Neural3DV dataset.

Method	Coffee Martini				Cook Spinach				Cut Roasted Beef			
	PSNR	DSSIM ₁	DSSIM ₂	LPIPS	PSNR	DSSIM ₁	DSSIM ₂	LPIPS	PSNR	DSSIM ₁	DSSIM ₂	LPIPS
STGS [4]	16.62	0.189	0.129	0.373	18.72	0.134	0.090	0.278	18.65	0.135	0.089	0.286
4DGS [6]	17.50	0.174	0.118	0.289	21.96	0.106	0.067	0.205	21.90	0.102	0.064	0.191
Ex4DGS [2]	17.45	0.179	0.124	0.281	20.61	0.138	0.089	0.224	20.33	0.144	0.092	0.228
Ours	19.57	0.140	0.093	0.203	24.44	0.075	0.046	0.111	23.22	0.083	0.051	0.124

Table 1. Quantitative comparison on Neural3DV dataset (Part 1).

Method	Flame Salmon				Flame Steak				Sear Steak			
	PSNR	DSSIM ₁	DSSIM ₂	LPIPS	PSNR	DSSIM ₁	DSSIM ₂	LPIPS	PSNR	DSSIM ₁	DSSIM ₂	LPIPS
STGS [4]	14.54	0.227	0.162	0.474	18.96	0.131	0.086	0.270	18.72	0.136	0.090	0.271
4DGS [6]	17.81	0.163	0.110	0.283	22.55	0.212	0.146	0.294	21.86	0.102	0.064	0.202
Ex4DGS [2]	15.40	0.193	0.135	0.308	21.39	0.123	0.077	0.200	20.80	0.119	0.078	0.196
Ours	19.15	0.136	0.091	0.205	23.72	0.075	0.046	0.114	23.66	0.078	0.048	0.118

Table 2. Quantitative comparison on Neural3DV dataset (Part 2).

Method	Actor1_4				Actor2_3				Actor5_6			
	PSNR	DSSIM ₁	DSSIM ₂	LPIPS	PSNR	DSSIM ₁	DSSIM ₂	LPIPS	PSNR	DSSIM ₁	DSSIM ₂	LPIPS
STGS [4]	15.50	0.314	0.186	0.690	15.55	0.313	0.184	0.686	15.44	0.316	0.187	0.682
4DGS [6]	23.24	0.194	0.119	0.154	23.64	0.186	0.114	0.154	23.71	0.145	0.087	0.139
Ex4DGS [2]	20.98	0.229	0.142	0.269	22.02	0.236	0.145	0.283	22.68	0.208	0.125	0.235
Ours	24.07	0.182	0.109	0.129	24.39	0.175	0.104	0.124	24.50	0.130	0.077	0.111

Table 3. Quantitative comparison on ENeRF-Outdoor dataset.

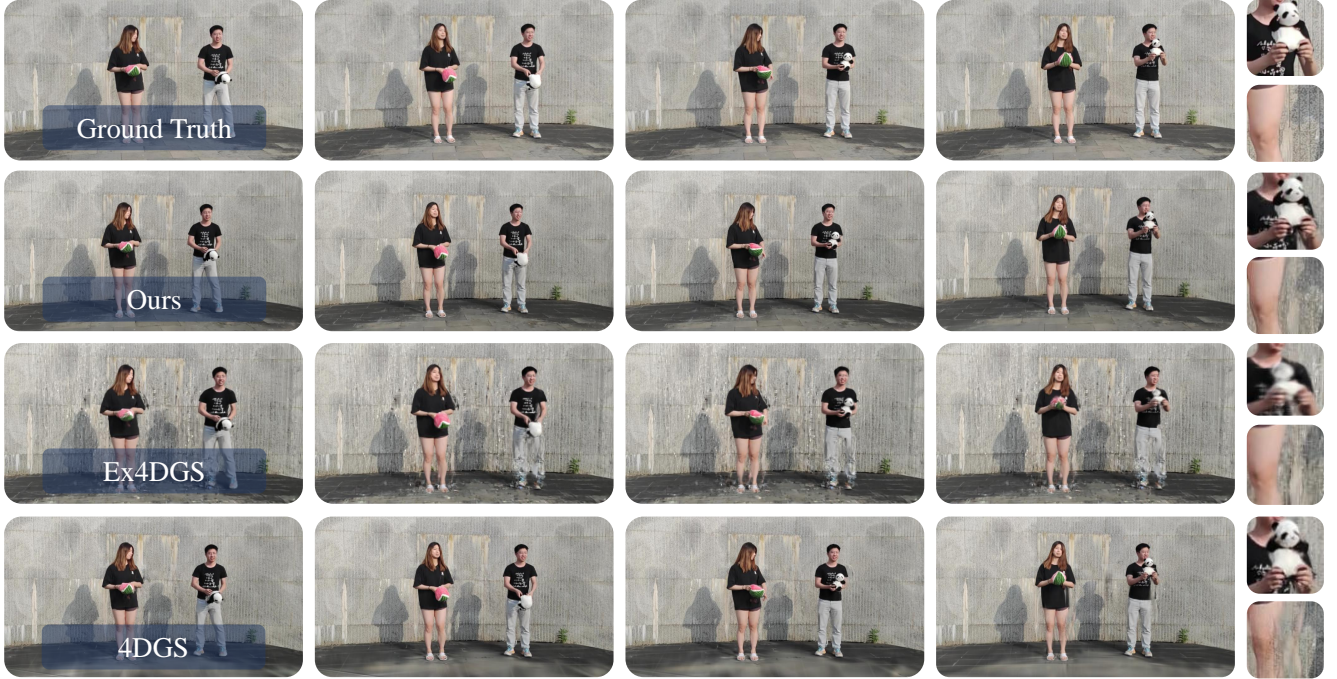


Figure 2. More visual comparisons under ENeRF-Outdoor dataset.

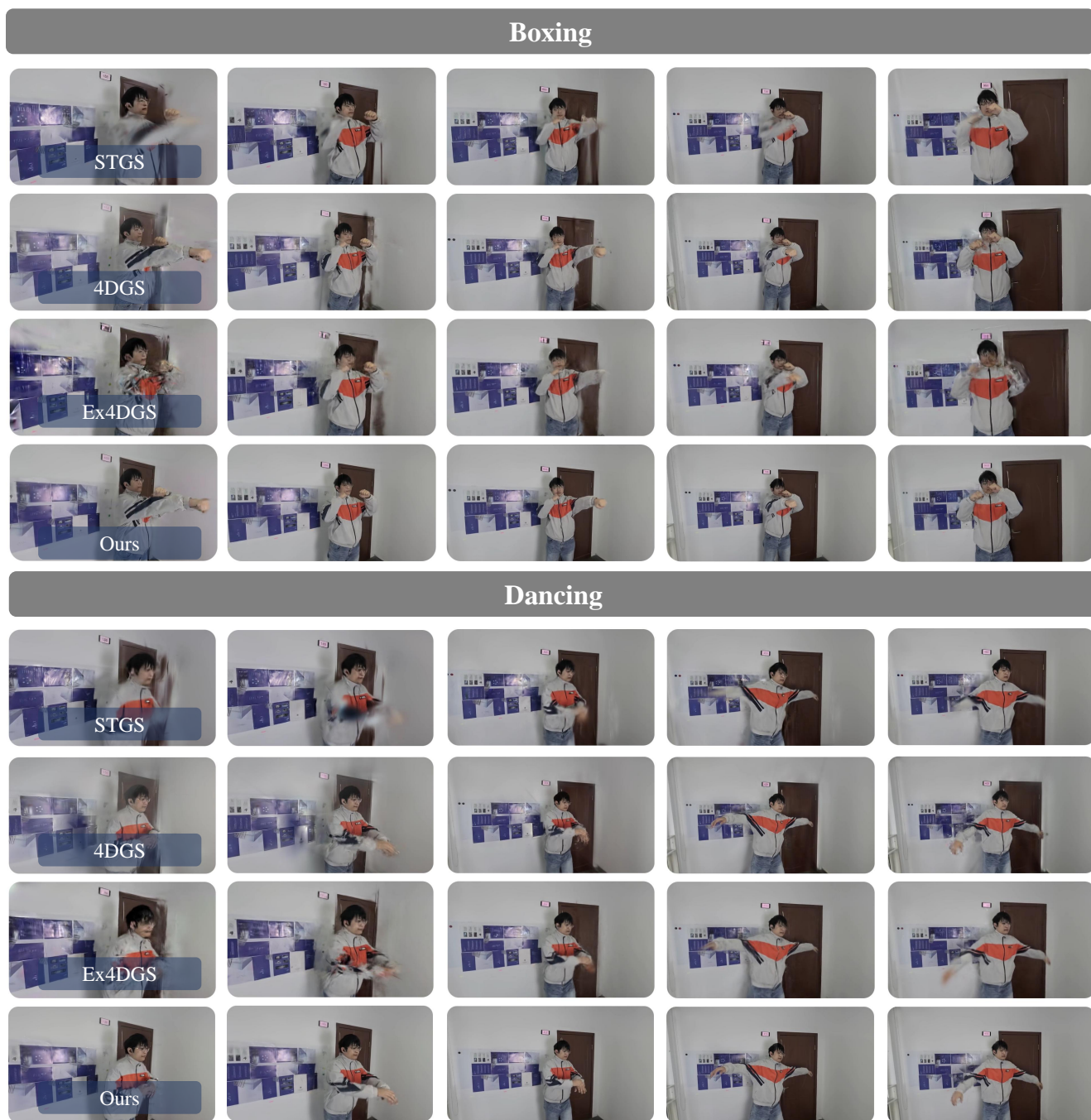


Figure 3. More visual comparisons under Dyn4Cam dataset (Part 1).

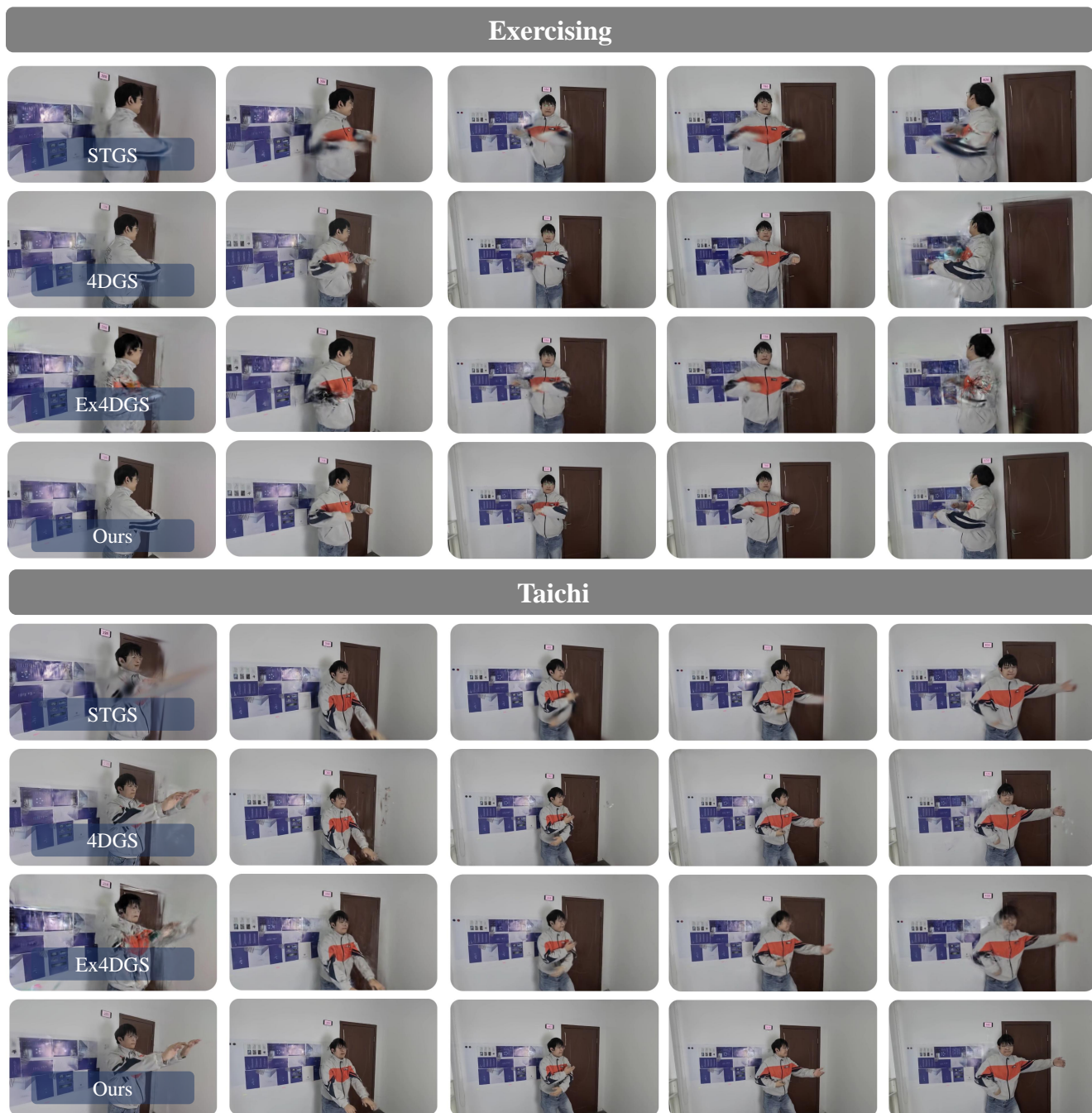


Figure 4. More visual comparisons under Dyn4Cam dataset (Part 2).

References

- [1] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. [1](#)
- [2] Junoh Lee, ChangYeon Won, Hyunjun Jung, Inhwon Bae, and Hae-Gon Jeon. Fully explicit dynamic gaussian splatting. *Advances in Neural Information Processing Systems*, 37:5384–5409, 2024. [3](#)
- [3] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *European conference on computer vision*, pages 71–91. Springer, 2024. [1](#)
- [4] Zhan Li, Zhang Chen, Zhong Li, and Yi Xu. Spacetime gaussian feature splatting for real-time dynamic view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8508–8520, 2024. [3](#)
- [5] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016. [1](#)
- [6] Zeyu Yang, Hongye Yang, Zijie Pan, and Li Zhang. Real-time photorealistic dynamic scene representation and rendering with 4d gaussian splatting. *arXiv preprint arXiv:2310.10642*, 2023. [3](#)