SpectroMotion: Dynamic 3D Reconstruction of Specular Scenes

Supplementary Material

A. Overview

Supplementary material goes here. This supplementary material presents additional results to complement the main manuscript. In Section B, we detail our coarse-to-fine training strategy, along with the network architectures of the deformable Gaussian MLP and deformable reflection MLP. Subsequently, Section C presents additional results, including comprehensive comparisons, more visualizations, and training and rendering efficiency. Finally, in Section D, we discuss the limitation of our approach and provide visual example of failure case.

B. Implementation Details

We use PyTorch as our framework and 3DGS [2] as our codebase. Our coarse-to-fine training strategy is divided into three sequential stages: static, dynamic, and specular stages.

Static stage. In the static stage, we train the vanilla 3D Gaussian Splatting (3DGS) for 3000 iterations to stabilize the static geometry.

Dynamic stage. After the static stage, we move on to the dynamic stage. During this phase, we introduce a deformable Gaussian MLP to model dynamic objects. First, we optimize both the canonical Gaussians and the deformable Gaussian MLP for 3,000 iterations until the scene reaches a relatively stable state. Then, we introduce the normal loss $\mathcal{L}_{\text{normal}}$, enabling simultaneous optimization of the scene's normal and depth, and perform an additional 3,000 iterations to further refine the geometry. The dynamic stage comprises a total of 6,000 training iterations.

Specular stage. After the dynamic stage concludes, we transition to the specular stage, which involves changing the color representation from complete spherical harmonics to $\mathbf{c_{final}}$. To mitigate potential geometry disruptions due to the initially incomplete $\mathbf{c_{final}}$, we fix the deformable Gaussian MLP and all 3D Gaussian attributes except for zero-order SH, specular tint, and roughness, while temporarily suspending densification. After 6000 iterations, once $\mathbf{c_{final}}$ becomes more complete, we resume optimization of all parameters and reinstate the densification process. Then, after another 3000 iterations, we stop the densification process. Concurrently, during the first 2000 iterations of the specular stage, we optimize only the canonical environment map to learn time-invariant lighting. For the canonical environment map, we use $6 \times 128 \times 128$ learnable parameters. Subsequently, we

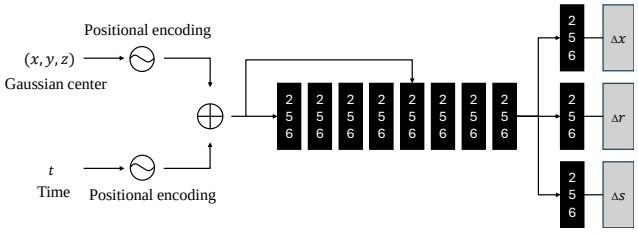

Figure 1. **Architecture of the deformable Gaussian MLP**

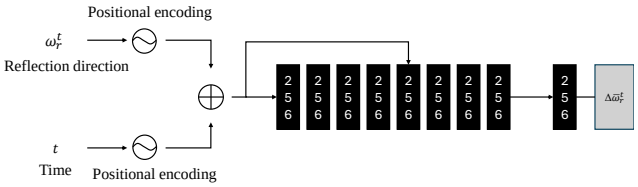

Figure 2. **Architecture of the deformable reflection MLP**

begin optimizing the deformable reflection MLP to capture time-varying lighting effects until the training is complete. The specular stage comprises a total of 31,000 training iterations. For the Peel Banana scene in the HyperNeRF dataset, we do not fix the deformable Gaussian MLP. We resume optimization of all parameters and reinstate the densification process after the first 4000 iterations of the specular stage. Then, after another 2000 iterations, we stop the densification process to prevent excessive growth in the number of 3D Gaussians, which could lead to GPU out-of-memory issues.

For the entire experiment, we train for a total of 40,000 iterations and we use Adam optimizer.

B.1. Network Architecture of the Deformable Gaussian MLP and Deformable reflection MLP

We follow Deformable 3DGS [8] and use deformable Gaussian MLP to predict each coordinate of 3D Gaussians and time to their corresponding deviations in position, rotation, and scaling. As shown in Fig. 1, the MLP initially processes the input through eight fully connected layers that employ ReLU activations, featuring 256-dimensional hidden layers and outputs a 256-dimensional feature vector. This vector is then passed through three additional fully connected layers combined with ReLU activation to separately output the offsets over time for position, rotation, and scaling. Notably, similar to NeRF, the feature vector and the input are concatenated in the fourth layer. For the deformable reflection MLP, we utilize the same network architecture, as shown in Fig. 2.

Table 1. **Quantitative comparison on the NeRF-DS [6] dataset with our labeled dynamic specular masks.** We report PSNR, SSIM, and LPIPS (VGG) of previous methods on dynamic specular objects using the dynamic specular objects mask generated by Track Anything [7]. The best, the second best, and third best results are denoted by red, orange, yellow.

	As			Basin			Bell			Cup		
Method	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Deformable 3DGS [8]	24.14	0.7432	0.2957	17.45	0.5530	0.3138	19.42	0.5516	0.2940	20.10	0.5446	0.3312
4DGS [5]	22.70	0.6993	0.3517	16.61	0.4797	0.4084	14.64	0.2596	0.4467	18.90	0.4132	0.4032
GaussianShader [1]	19.27	0.5652	0.5232	15.71	0.4163	0.5941	12.10	0.1676	0.6764	14.90	0.3634	0.6146
GS-IR [3]	19.32	0.5857	0.4782	15.21	0.4009	0.5644	12.09	0.1757	0.6722	14.80	0.3445	0.6046
NeRF-DS [6]	23.67	0.7478	0.3635	17.98	0.5537	0.4211	14.73	0.2439	0.5931	19.95	0.5079	0.3494
HyperNeRF [4]	17.37	0.6934	0.3834	18.75	0.5671	0.4125	13.93	0.2292	0.6051	15.07	0.4860	0.4183
Ours	24.51	0.7534	0.2896	17.71	0.5675	0.3048	19.60	0.5680	0.2862	20.13	0.5384	0.3368

	Plate			Press			Sieve			**Mean**		
Method	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Deformable 3DGS [8]	16.12	0.5192	0.3544	19.64	0.6384	0.3268	20.74	0.5283	0.3109	19.66	0.5826	0.3181
4DGS [5]	13.93	0.4095	0.4229	20.17	0.5434	0.4339	19.70	0.4498	0.3879	18.09	0.4649	0.4078
GaussianShader [1]	9.87	0.2992	0.6812	16.84	0.4408	0.6093	16.19	0.3241	0.5862	14.98	0.3681	0.6121
GS-IR [3]	11.09	0.3254	0.6270	16.43	0.4083	0.5776	16.42	0.3339	0.5749	15.05	0.3678	0.5856
NeRF-DS [6]	14.80	0.4518	0.3987	19.77	0.5835	0.5035	20.28	0.5173	0.4067	18.74	0.5151	0.4337
HyperNeRF [4]	16.03	0.4629	0.3775	14.10	0.5365	0.5023	18.39	0.5296	0.3949	16.23	0.5007	0.4420
Ours	16.53	0.5369	0.3041	21.70	0.6630	0.3252	20.36	0.5089	0.3190	20.08	0.5909	0.3094

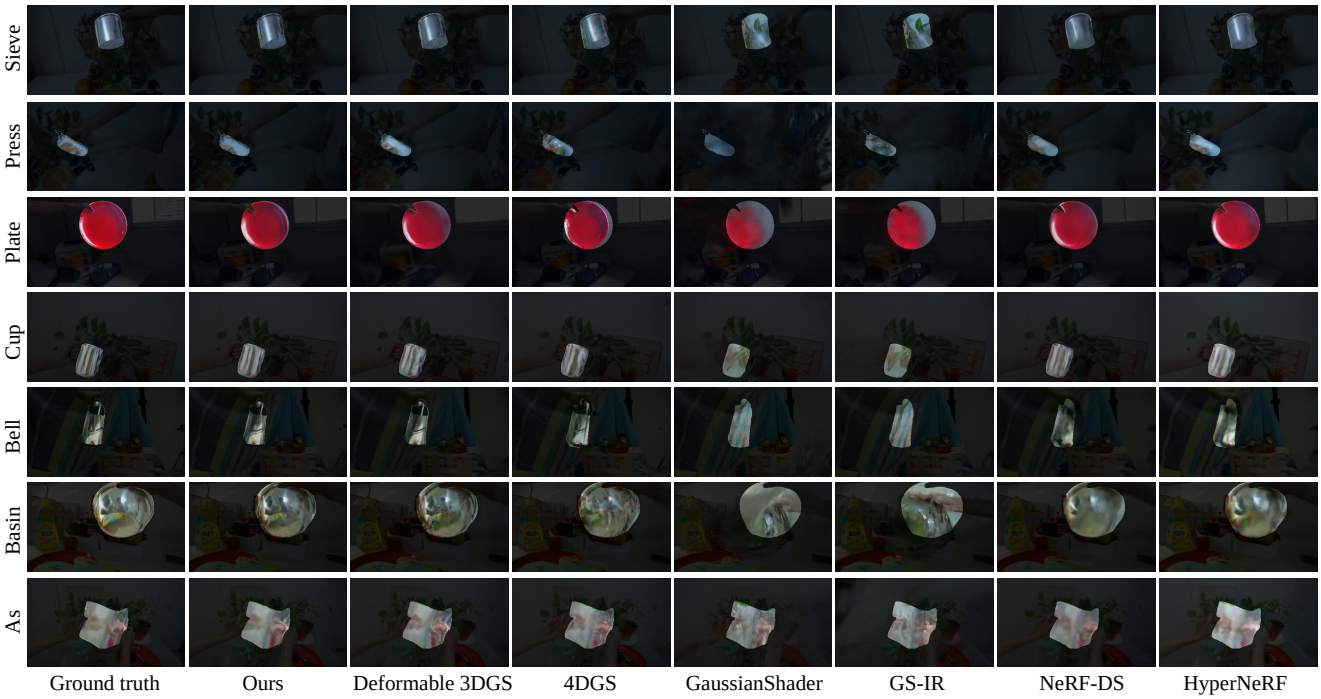

Figure 3. **Qualitative comparison on NeRF-DS [6] dataset with labeled dynamic specular masks.**

C. Additional Results

C.1. Dynamic specular object of NeRF-DS dataset.

Since each scene in the NeRF-DS dataset [6] contains not only dynamic specular objects but also static background objects, we use Track Anything [7] to obtain masks for the

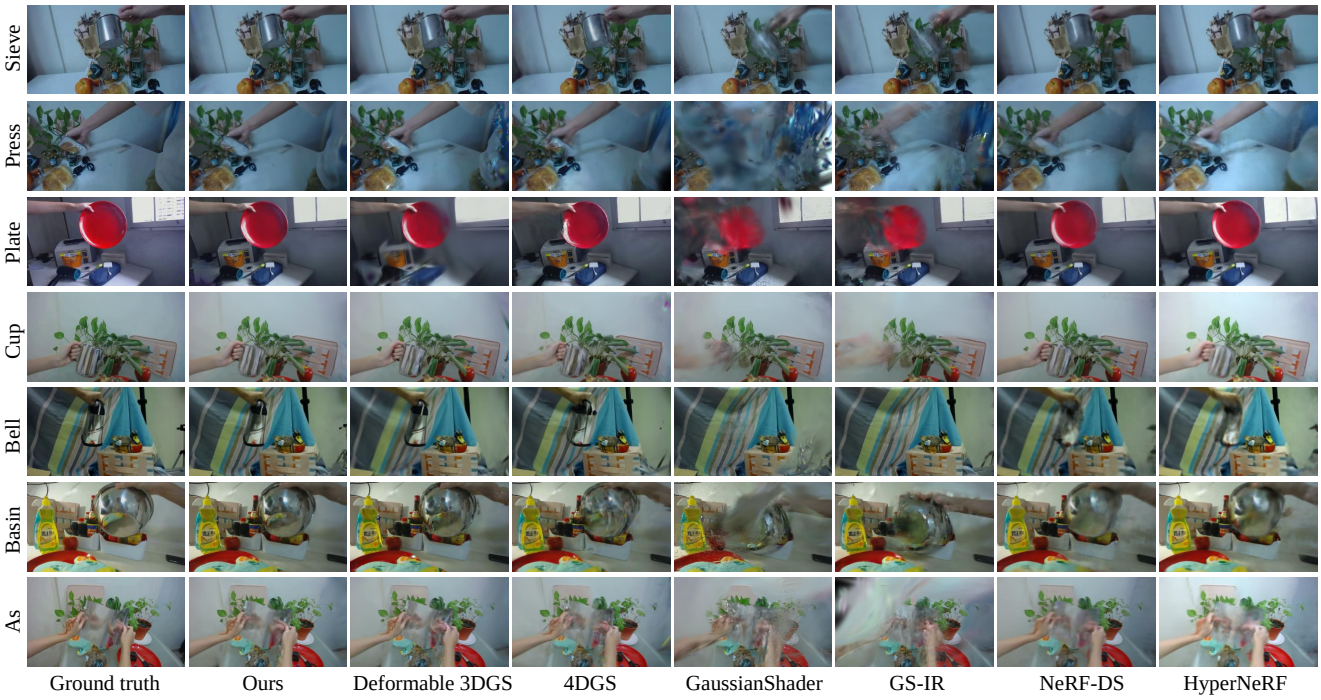

Figure 4. **Qualitative comparison on the NeRF-DS [6] dataset.**

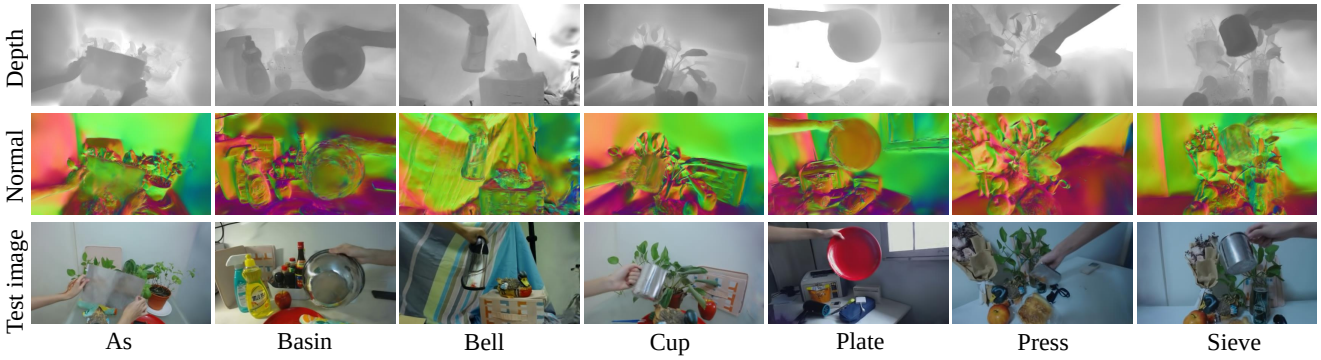

Figure 5. **Visualized our rendered test images, normal maps, and depth maps.**

dynamic specular objects. This allows us to evaluate only the dynamic specular objects. As shown in Tab. 1 and Fig. 3, our method outperforms baselines when evaluating the dynamic specular objects in these monocular sequences.

C.2. Per-Scene results on the NeRF-DS Dataset

In Fig. 4, we present qualitative results for each scene in the NeRF-DS dataset [6]. The visualizations demonstrate that our method achieves superior rendering quality compared to other approaches. We also provide rendered test images and their corresponding normal maps and depth maps for each scene in the NeRF-DS dataset in Fig. 5.

C.3. Deformation magnitudes and color decomposition

Unlike NeRF-DS [6], our approach does not require mask supervision to clearly distinguish between static and dynamic objects, as illustrated in Fig. 6. Additionally, Fig. 7 illustrates our method's decomposition results. As shown, our approach consistently achieves a realistic separation of specular and diffuse components across different scenes in the NeRF-DS dataset [6].

C.4. Training and rendering efficiency

In Tab. 2, we present the training time, FPS, and number of Gaussians from our experiments on each scene in the NeRF-DS dataset [6]. The results show that for scenes with

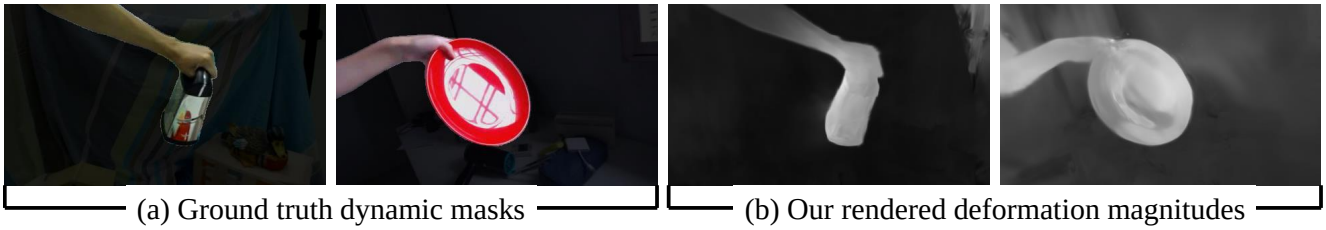

Figure 6. **Visualized our deformation magnitudes.** (a) The left side shows the ground truth of the dynamic object, while (b) on the right side, we render the magnitude of the output of the position residual by our deformable Gaussian MLP. The brighter areas indicate greater movement of the 3D Gaussians. The figure shows that even without mask supervision, our method can still effectively distinguish which objects are dynamic.

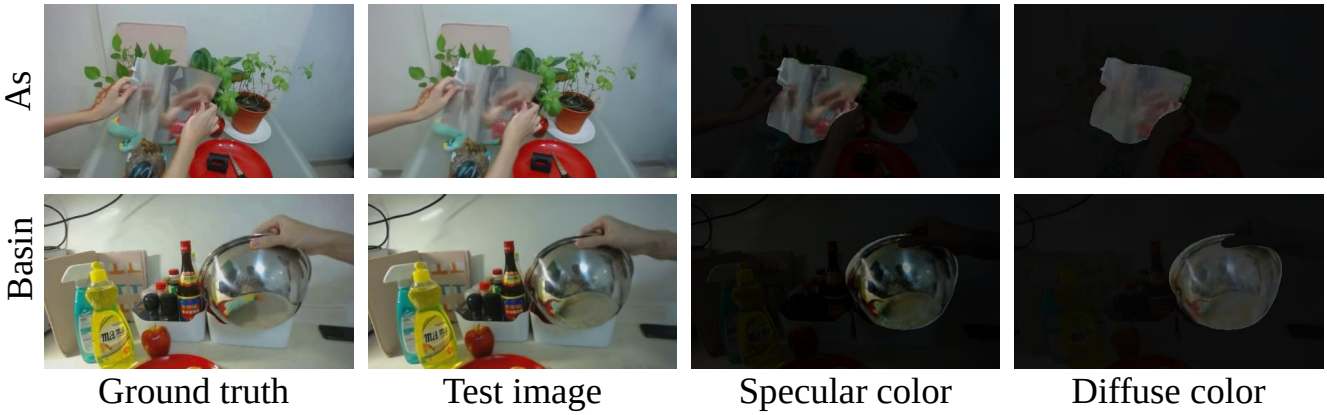

Figure 7. **Visualized our specular and diffuse color.** Specular regions are emphasized while non-specular areas are dimmed to highlight the results of specular region color decomposition.

Table 2. **Training and rendering efficiency on NeRF-DS [6] dataset**

Scene	Training time (hr)	FPS	Number of Gaussians (k)
as	1.2	31	170
basin	1.3	25	218
bell	1.9	18	335
cup	1.7	30	177
plate	1.1	27	187
press	1.0	31	172
sieve	1.1	29	178

fewer than 178k Gaussians, our method achieves real-time rendering greater than or equal to 30 FPS. The experiments are conducted on an NVIDIA RTX 4090 GPU.

C.5. In-the-Wild / Non-Staged Scenes.

We ran our method on the Waymo dataset, which features reflective vehicles and complex motion. As shown in Fig 8 below, our approach outperforms Deformable 3DGS, particularly in handling specular surfaces. These results underscore the robustness of our method in in-the-wild settings.

D. Limitation

In some dramatic scenes, relying solely on the deformable Gaussian MLP and coarse-to-fine training strategy is insufficient, such as when an arm or body enters or exits the scene, leading to many floaters occurring. We provide visual results in Fig. 9.

References

[1] Yingwenqi Jiang, Jiadong Tu, Yuan Liu, Xifeng Gao, Xiaoxiao Long, Wenping Wang, and Yuexin Ma. Gaussianshader: 3d gaussian splatting with shading functions for reflective surfaces. *arXiv preprint arXiv:2311.17977*, 2023. 2

[2] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering, 2023. 1

[3] Zhihao Liang, Qi Zhang, Ying Feng, Ying Shan, and Kui Jia. Gs-ir: 3d gaussian splatting for inverse rendering. *arXiv preprint arXiv:2311.16473*, 2023. 2

[4] 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, 2021. 2

Figure 8. **Comparison of our method and Deformable 3DGS on reflective cars.**

Figure 9. **Failure cases of modeling dramatic scene changes.** There are dramatic scenes where an arm or body enters or exits the scene, leading to many floaters occurring.

[5] 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. 2

[6] Zhiwen Yan, Chen Li, and Gim Hee Lee. Nerf-ds: Neural radiance fields for dynamic specular objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8285–8295, 2023. 2, 3, 4

[7] Jinyu Yang, Mingqi Gao, Zhe Li, Shang Gao, Fangjing Wang, and Feng Zheng. Track anything: Segment anything meets videos. *arXiv preprint arXiv:2304.11968*, 2023. 2

[8] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. *arXiv preprint arXiv:2309.13101*, 2023. 1, 2