Deformable 3D Gaussians for High-Fidelity Monocular Dynamic Scene Reconstruction
– *Supplemental Material* –

Ziyi Yang[1,2] Xinyu Gao[1] Wen Zhou[2] Shaohui Jiao[2] Yuqing Zhang[1] Xiaogang Jin[1†]

[1]Zhejiang University [2]ByteDance Inc.

A. Overview

This supplementary document provides some implementation details and further results that accompany the paper.

- Section B introduces the implementation details of the network architecture in our approach.
- Section C provides additional results, including more visualizations, rendering efficiency, more comparisons, and more ablations.
- Section D discusses the failure cases of our method.

B. Implementation Details

B.1. Network Architecture of the Deformation Field

We learn the deformation field with an MLP network $\mathcal{F}_\theta : (\gamma(\text{sg}(\mathbf{x})), \gamma(t)) \rightarrow (\delta x, \delta r, \delta s)$, which maps from each coordinate of 3D Gaussians and time to their corresponding deviations in position, rotation, and scaling. The weights θ of the MLP are optimized through this mapping. As shown in Fig. 1, our MLP $\mathcal{F}_\theta$ initially processes the input through eight fully connected layers that employ ReLU activations and feature 256-dimensional hidden layers, and outputs a 256-dimensional feature vector. This vector is subsequently passed through three additional fully connected layers (**without activation**) to separately output the offsets over time for position, rotation, and scaling. It should be noted that similar to NeRF, we concatenate the feature vector and the input in the fourth layer. Due to the compact structure of MLP, our additional storage compared to 3D Gaussians is only 2MB.

Our deformation field does not employ any grid/plane-based structures which have been demonstrated to be superior in static scenes because these structures are predicated on a **low-rank tensor assumption** [1]. Dynamic scenes possess a higher rank compared to static scenes, and explicit point-based rendering exacerbates the rank of the scene.

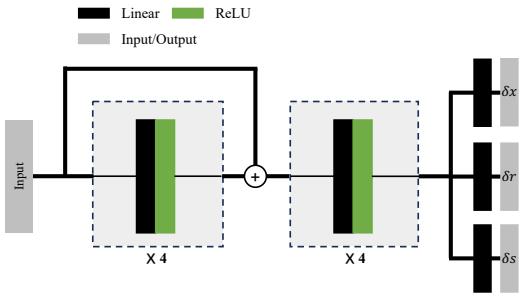

Figure 1. **The architecture of our deformation MLP.**

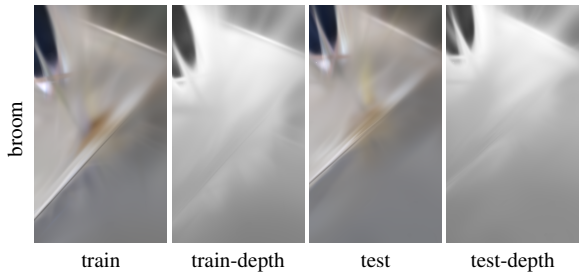

Figure 2. **Failure case on inaccurate pose.** Excessively inaccurate poses can lead to the failure of the convergence on the training set.

B.2. Optimization Loss

During the training of our deformable Gaussians, we deform the 3D Gaussians at each timestep into the canonical space. We then optimize both the deformation network and the 3D Gaussians using a combination of $\mathcal{L}_1$ loss and D-SSIM loss [2]:

$$\mathcal{L} = (1 - \lambda)\mathcal{L}_1 + \lambda\mathcal{L}_{\text{D-SSIM}}, \tag{1}$$

where $\lambda = 0.2$ is used in all our experiments.

Method	Sieve PSNR↑	Sieve SSIM↑	Sieve LPIPS↓	Plate PSNR↑	Plate SSIM↑	Plate LPIPS↓	Bell PSNR↑	Bell SSIM↑	Bell LPIPS↓	Press PSNR↑	Press SSIM ↑	Press LPIPS↓
3D-GS	23.16	0.8203	0.2247	16.14	0.6970	0.4093	21.01	0.7885	0.2503	22.89	0.8163	0.2904
TiNeuVox	21.49	0.8265	0.3176	20.58	0.8027	0.3317	23.08	0.8242	0.2568	24.47	0.8613	0.3001
HyperNeRF	25.43	0.8798	0.1645	18.93	0.7709	0.2940	23.06	0.8097	0.2052	26.15	0.8897	0.1959
NeRF-DS	25.78	0.8900	0.1472	20.54	0.8042	0.1996	23.19	0.8212	0.1867	25.72	0.8618	0.2047
Ours (w/o AST)	25.33	0.8620	0.1594	20.32	0.7173	0.3914	25.62	0.8498	0.1540	25.78	0.8613	0.1919
Ours	25.70	0.8715	0.1504	20.48	0.8124	0.2224	25.74	0.8503	0.1537	26.01	0.8646	0.1905
Method	Cup PSNR↑	Cup SSIM↑	Cup LPIPS↓	As PSNR↑	As SSIM↑	As LPIPS↓	Basin PSNR↑	Basin SSIM↑	Basin LPIPS↓	Mean PSNR↑	Mean SSIM ↑	Mean LPIPS↓
3D-GS	21.71	0.8304	0.2548	22.69	0.8017	0.2994	18.42	0.7170	0.3153	20.29	0.7816	0.2920
TiNeuVox	19.71	0.8109	0.3643	21.26	0.8289	0.3967	20.66	0.8145	0.2690	21.61	0.8234	0.2766
HyperNeRF	24.59	0.8770	0.1650	25.58	0.8949	0.1777	20.41	0.8199	0.1911	23.45	0.8488	0.1990
NeRF-DS	24.91	0.8741	0.1737	25.13	0.8778	0.1741	19.96	0.8166	0.1855	23.60	0.8494	0.1816
Ours (w/o AST)	24.80	0.8848	0.1571	26.29	0.8800	0.1830	19.68	0.7869	0.1888	23.97	0.8346	0.2037
Ours	24.86	0.8908	0.1532	26.31	0.8842	0.1783	19.67	0.7934	0.1901	24.11	0.8524	0.1769

Table 1. **Quantitative comparison on NeRF-DS dataset per-scene**. We color each cell as best, second best, and third best. Our method, overall, achieves the best rendering quality and robust convergence in the majority of scenes. It is worth noting that the metrics we used are the same as those in the main text, with LPIPS using the VGG network. Our measurement metrics differ slightly from those used in NeRF-DS and HyperNeRF because their papers use MS-SSIM and LPIPS with the AlexNet.

D-NeRF Dataset			NeRF-DS Dataset			HyperNeRF Dataset		
Scene	**FPS**	**Num (k)**	**Scene**	**FPS**	**Num (k)**	**Scene**	**FPS**	**Num (k)**
Lego	24	300	AS	48	185	Espresso	15	620
Jump	85	90	Basin	29	250	Americano	6	1,300
Bouncing	38	170	Bell	18	400	Cookie	9	1,080
T-Rex	30	220	Cup	35	200	Chicken	10	740
Mutant	40	170	Plate	31	230	Torchocolate	8	1,030
Warrior	172	40	Press	48	185	Lemon	23	420
Standup	93	80	Sieve	35	200	Hand	6	1,750
Hook	45	150				Printer	12	650

Table 2. **Experiments on FPS with respect to the number of 3D Gaussians.** The results of the experiments demonstrate that our method is capable of real-time rendering on a 3090 GPU when the number of point clouds is less than 250k. The excessively high number of 3D Gaussians in HyperNeRF reflects the critical importance of the camera pose accuracy for the convergence of our method.

C. Additional Results

C.1. Per-Scene Results on the NeRF-DS Dataset

In Tab. 1, we provide the results for individual scenes associated with Sec. 4 of the main paper. It can be observed that our method achieved superior metrics in almost every scene compared to those without AST, underscoring the generalizability of AST on real datasets where the pose is not perfectly accurate. Overall, our method outperforms baselines on the NeRF-DS Dataset.

C.2. Results on the HyperNeRF Dataset

We visualize the results of the HyperNeRF dataset in Fig. 3. Notably, metrics designed to assess image rendering quality, such as PSNR, tend to penalize minor offsets more heavily than blurring. Therefore, for datasets with less accurate camera poses, like HyperNeRF, our method's quantitative metrics might not consistently outperform those of methods yielding blurred outputs when faced with imprecise camera poses. Despite this, our rendered images often exhibit fewer artifacts and greater clarity. This phenomenon aligns with observations reported in Nerfies [4] and HyperNeRF [5].

C.3. Results on Rendering Efficiency

In our research, we present comprehensive Frames Per Second (FPS) testing results in Tab. 2. Tests were conducted on one NVIDIA RTX 3090. It is observed that when the number of point clouds remains below ∼250k, our method can achieve real-time rendering at rates greater than 30 FPS. A point of note is that the point cloud count reconstructed from the HyperNeRF dataset significantly exceeds that of other datasets, reaching a level of 1,000k. This excessive count is attributed to the highly inaccurate camera poses within the HyperNeRF dataset. In contrast, the NeRF-

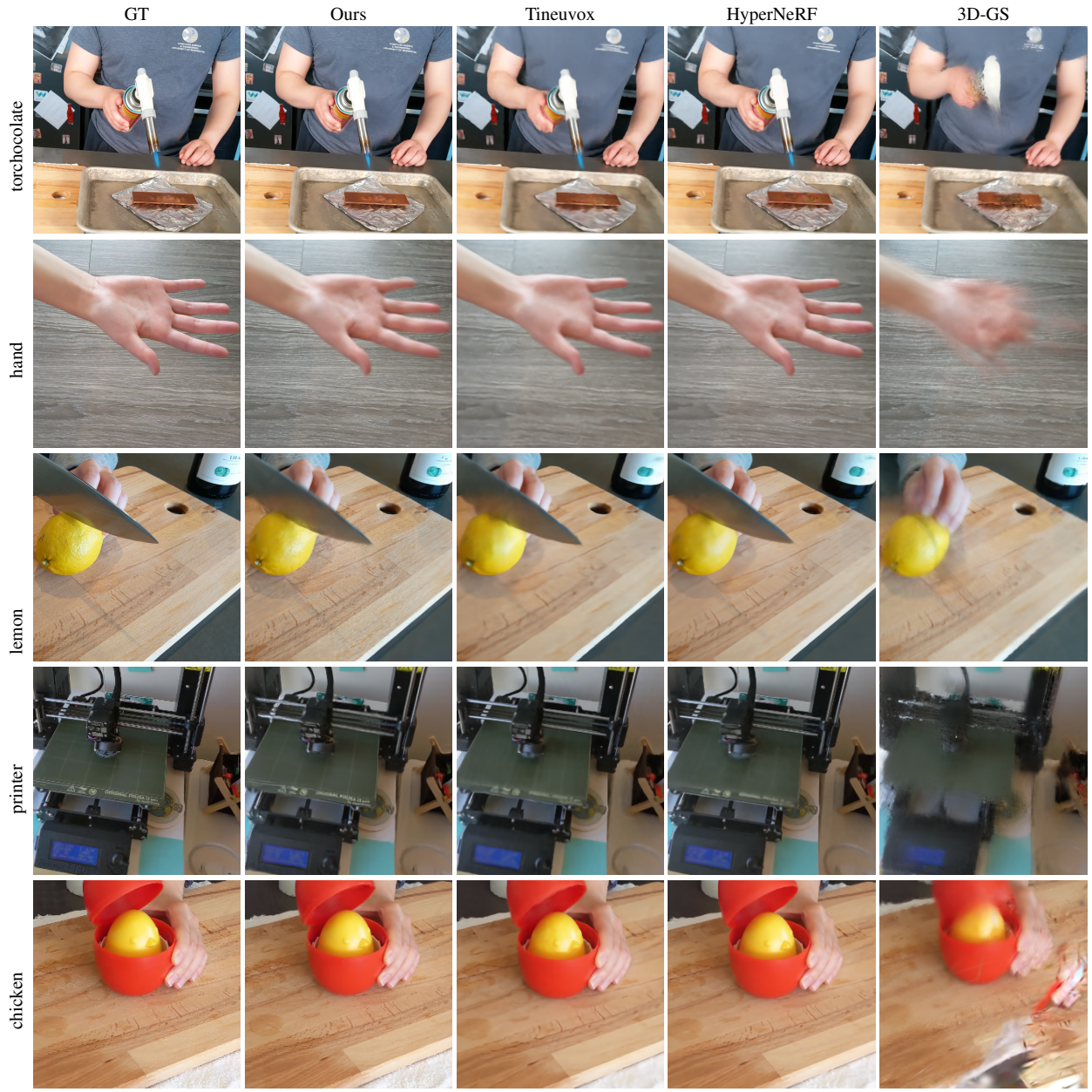

Figure 3. **Qualitative comparisons of baselines and our method on HyperNeRF dataset.** The first three rows present the results of the time interpolation task, while the last two rows depict the outcomes of the novel viewpoint synthesis task. Experimental results indicate that our method can achieve superior rendering quality on real datasets where the pose is not absolutely precise.

	lego	jump	bouncing	trex	mutant	warrior	standup	hook	mean
ours	**33.07**	**37.72**	41.01	38.10	42.63	41.54	44.62	37.42	39.51
ours-SE(3)	32.91	37.60	**41.05**	**38.29**	**42.83**	**41.73**	**44.68**	**37.60**	**39.58**

Table 3. **Comparison with SE(3) deformation field on the D-NeRF dataset.**

	as	basin	bell	cup	plate	press	sieve	mean
ours	26.31	**19.67**	**25.74**	**24.86**	**20.48**	**26.01**	**25.70**	**24.11**
ours-SE(3)	**26.37**	19.64	25.43	24.83	20.28	25.63	25.46	23.95

Table 4. **Comparison with SE(3) deformation field on the NeRF-DS dataset.**

Figure 4. **Failure case on few training viewpoints.** The first three columns represent the original dataset configurations. The term **swap** indicates the exchange of training and test sets, thereby ensuring that the model's inputs possess a sufficiently diverse array of viewpoints

	Hell Warrior			Mutant			Hook		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
w/o δs	41.55	0.9878	0.0223	42.15	0.9949	0.0053	37.01	0.9859	0.0153
w/o δr	41.17	0.9866	0.0256	42.51	0.9950	0.0054	36.82	0.9852	0.0167
w r&s	40.39	0.9833	0.0323	41.30	0.9934	0.0075	36.15	0.9818	0.0214
ours	41.54	0.9873	0.0234	42.63	0.9951	0.0052	37.42	0.9867	0.0144
	Bouncing Balls			Lego			T-Rex		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
w/o δs	40.82	0.9952	0.0095	31.30	0.9705	0.0260	37.39	0.9928	0.0105
w/o δr	41.11	0.9953	0.0092	32.87	0.9783	0.0192	37.99	0.9931	0.0101
w r&s	39.89	0.9945	0.0117	33.71	0.9798	0.0181	37.06	0.9923	0.0113
ours	41.01	0.9953	0.0093	33.07	0.9794	0.0183	38.10	0.9933	0.0098
	Stand Up			Jumping Jacks			Mean		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
w/o δs	44.05	0.9946	0.0074	37.49	0.9895	0.0129	38.97	0.9889	0.0137
w/o δr	44.18	0.9946	0.0075	37.48	0.9893	0.0138	39.27	0.9897	0.0134
w r&s	42.88	0.9932	0.0097	37.00	0.9878	0.0164	38.55	0.9883	0.0160
ours	44.62	0.9951	0.0063	37.72	0.9897	0.0126	39.51	0.9902	0.0124

Table 5. **Ablations on network architecture.** We color each cell as best and second best. δr and δs denote the output components of the MLP model. The term **w r&s** signifies that the model inputs include not only time and the position of the 3D Gaussians but also the 3D Gaussians' rotation and scaling. The experimental outcomes affirm that our network architecture is the most advantageous.

DS dataset, while also being derived from the real world, exhibits more accurate poses, resulting in a reconstructed point cloud count within a reasonable range. This issue of an overabundant point cloud count occurs not only in scenes with inaccurate poses but also in those with sparse viewpoints, as evidenced in scenes like the D-NeRF's Lego scene, which was trained on merely 50 images.

C.4. More Ablations

Network architecture. We present ablation experiments on the architecture of our purely implicit network, as shown in Tab. 5. The results of these experiments suggest that the structure within our pipeline is optimal. Notably, we did not adopt Grid/Plane-based structures because dynamic scenes do not conform to the **low-rank assumption**. Furthermore,

	lego	jump	bouncing	trex	mutant	warrior	standup	hook	mean
ours	**33.07**	**37.72**	41.01	38.10	**42.63**	**41.54**	**44.62**	**37.42**	**39.51**
ours-white	32.03	36.87	**43.52**	**38.57**	42.11	32.75	42.40	36.60	38.10
ours-best	33.07	37.72	43.52	38.57	42.63	41.54	44.62	37.42	39.89

Table 6. **Comparison with different background colors on the D-NeRF dataset.**We explored the impact of different background colors on rendering metrics using the D-NeRF dataset. The experimental results showed that overall, a black background yielded higher metrics, while bouncing and trex scenes performed better with a white background, and the warrior scene had higher metrics with a black background. To ensure **experimental consistency**, we uniformly used a black background in the main text. If one wishes to pursue the best metrics for a specific scene, one can refer to this table to adjust the background color.

the explicit point-based rendering of 3D-GS exacerbates the rank of dynamic scenes. Our early experimental validations have corroborated this assertion.

C.5. Background color

In the research of Neural Rendering, it's common to use a black or white background for rendering scenes without a background. In our experiments, we found that the background color has an impact on certain scenes in the D-NeRF dataset. The experimental results are shown in Tab. 6. Overall, a black background yields better rendering results. For the sake of consistency in our experiments, we uniformly used a black background in our main text experiments. However, for the bouncing and trex scenes, using a white background can produce better results.

C.6. Deformation using SE(3) Field

Drawing inspiration from Nerfies [4], we applied a 6-DOF SE(3) field that accounts for rotation to the transformation of 3D Gaussian positions. The experimental results, presented in Tab. 3 and Tab. 4, indicate that this constraint offers a minor improvement on the D-NeRF dataset. However, it appears to diminish the quality on the more complex real-world NeRF-DS dataset. Moreover, the additional computational overhead introduced by the SE(3) Field approximately increases 50 % of the training time and results in about a 20% decrease in FPS during rendering. Consequently, we opted to utilize a direct addition without imposing SE(3) constraints on the transformation of position.

D. Failure Cases

Inaccurate poses. In our research, we find that inaccurate poses can lead to the failure of the convergence of deformable-gs, as illustrated in Fig. 2. For implicit representations, their inherent smoothness can maintain robustness in the face of minor deviations in pose. However, for the explicit point-based rendering, such inaccuracies are particularly detrimental, resulting in inconsistencies in the scene at different moments.

Few training viewpoints. In our study, a notable scarcity of training views presents a dual challenge: both few-shot learning and a limited number of viewpoints. Either aspect can lead to overfitting in deformable-gs and even in 3D-GS on the training set. As demonstrated in Fig. 4, significant overfitting is evident in the DeVRF [3] dataset. The training set for this scene contains 100 images, but the viewpoints for training are limited to only four. However, by swapping the training and test sets, where the test set contained an equal number of 100 images and viewpoints, we obtained markedly better results.

References

[1] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *European Conference on Computer Vision (ECCV)*, pages 333–350, 2022. 1

[2] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):1–14, 2023. 1

[3] Jia-Wei Liu, Yan-Pei Cao, Weijia Mao, Wenqiao Zhang, David Junhao Zhang, Jussi Keppo, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Devrf: Fast deformable voxel radiance fields for dynamic scenes. *Advances in Neural Information Processing Systems*, 35:36762–36775, 2022. 5

[4] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5865–5874, 2021. 2, 5

[5] 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*, 40(6):1–12, 2021. 2