SplineGS: Robust Motion-Adaptive Spline for Real-Time Dynamic 3D Gaussians from Monocular Video

Supplementary Material

A. Implementation Details

To develop our method, we build on top of the widely used open-source 3DGS codebase [20]. Our SplineGS architecture is trained over 1k iterations in the warm-up stage and 20k iterations in the main training stage. We initialize n^{st} and n^{dy} to 20k, densifying the Gaussians every 100 iterations until 12k. We optimize the number of control points with the proposed MACP every 100 iterations. For depth and 2D tracking estimation, we employ the pre-trained models from UniDepth [38] and CoTracker [19], respectively. The learnable camera extrinsics $[\hat{\boldsymbol{R}}_t|\hat{\boldsymbol{T}}_t]$ are initialized by $[\mathbf{I}|\mathbf{0}]$, while the initial learnable focal length value $\hat{f}$ is set to 500. We maintain the same gradient-based densification [20] for both static $\{G_i^{\text{st}}\}$ and dynamic $\{G_i^{\text{dy}}\}$ 3D Gaussians. We set all λ values of Eq. 14 and 15 to 1.

B. Additional Ablation Study for Motion-Adaptive Control Points Pruning (MACP)

As described in Eq. 8 of the main paper, we compute the error E between $S(t, \mathbf{P})$ and $S(t, \mathbf{P}')$ by projecting the 3D points of each cubic Hermite spline function [2, 8] over time into pixel space of all training cameras. This error is then used to update the new spline function. The 2D error measurement is particularly effective because it directly aligns with the image domain, where pixel-level accuracy is essential for precise spline function updates. To determine the updated spline function, we set the threshold value ϵ of the error E in Eq. 8 to 1. To validate the rationale behind our setup, we conduct an ablation study for novel view synthesis on the NVIDIA dataset [56], examining different MACP settings, including the ablated models without MACP ('w/o MACP ($N_c = 4$)', 'w/o MACP ($N_c = N_f$)' in Table 3-(c)) and with MACP having variations in ϵ values. For the variations in ϵ values, we select 0.2, 1, 2, 3, and 5. Fig. 9 presents the average PSNR values and the average number of control points for dynamic 3D Gaussians after training across all scenes. As shown in Fig. 9, when ϵ is set to an excessively small value ('$\epsilon = 0.2$'), our MAS architecture fails to prune control points effectively, resulting in reduced efficiency. Conversely, when ϵ is too large ('$\epsilon = 5$'), the pruning becomes overly aggressive, resulting in an insufficient number of control points to accurately represent complex motion trajectories. This trade-off underscores the importance of selecting ϵ carefully to achieve a balance between efficiency and representation quality.

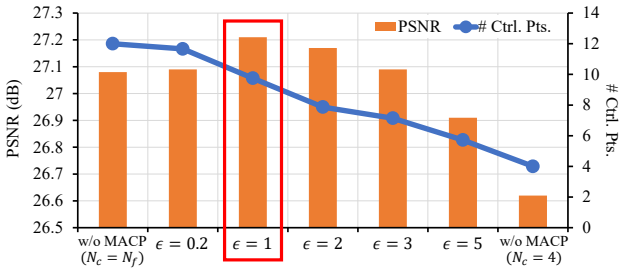


Figure 9. **Ablation study on MACP.** We conduct an ablation study of our Motion-Adaptive Control points Pruning (MACP) method for novel view synthesis on the NVIDIA dataset [56] by adjusting the pruning error threshold ϵ. 'PSNR (dB)' and '# Ctrl. Pts.' denote the average PSNR value and the average number of control points for dynamic 3D Gaussians after training, computed across all scenes, respectively.

C. Memory Footprint Comparison

To further highlight the efficiency of our SplineGS, we compared its memory footprint with other 3DGS-based methods [21, 24, 52, 55], as shown in Table 4. This comparison evaluates the average model storage requirements after optimization on the NVIDIA dataset [56]. The storage requirements of 3DGS-based methods depend on the number of 3D Gaussians, which is determined by their hyperparameters. For consistency, we use the same hyperparameter settings for the 3DGS-based methods [21, 24, 52, 55] as those specified in their original implementations. Ex4DGS [21] requires the largest memory footprint, attributed to its method of explicit keyframe dynamic 3D Gaussian fusion. In contrast, our SplineGS, which achieves state-of-the-art (SOTA) rendering quality as shown in Table 1, utilizes only about one-tenth of the memory footprint required by Ex4DGS [21], thanks to our efficient MAS representation and the MACP method.

Methods	Memory footprint (MB) ↓	# Gaussian (K)
4DGS (CVPR'24) [52]	50	136
D3DGS (CVPR'24) [55]	92	382
Ex4DGS (NeurIPS'24) [21]	256	436
STGS (CVPR'24) [24]	19	128
SplineGS (Ours)	26	183

Table 4. **Memory footprint comparison results.** 'Memory footprint (MB)' refers to the memory size of each trained model, while '# Gaussian (K)' represents the total number of 3D Gaussians after training.

D. Dynamic 3D Gaussian Trajectory Visualization

Please note that the term *motion tracking* in our main paper (Fig. 6), also referred to as dynamic 3D Gaussian trajectory visualization in 2D space, differs from the term *tracking* used in *2D Tracking* methods such as [19], which aim to find 2D pixel correspondences among *given* video frames. Our SplineGS leverages spline-based motion modeling to directly capture the deformation of each dynamic 3D Gaussian along the temporal axis, enabling the rendering of target novel views. For 2D visualization of the 3D motion of each dynamic 3D Gaussian, which is referred to as *motion tracking* in our main paper, we project its trajectory onto the 2D pixel space of the novel views. We compute a *rasterized* 2D track $\boldsymbol{\mathcal{T}}^G = \{\boldsymbol{\varphi}^G_{t'} | \boldsymbol{\varphi}^G_{t'} \in \mathbb{R}^2\}_{t' \in [t_1, t_2]}$ over the specified time interval $[t_1, t_2]$ as the Gaussians' trajectories visualization shown in Fig. 6 of the main paper. For this motion tracking rasterization, we compute the projected pixel coordinates at time t' for each 3D Gaussian using the camera pose $[\boldsymbol{R}^*|\boldsymbol{T}^*]$ of the target novel view as $\pi_{\hat{\boldsymbol{K}}}(\boldsymbol{R}^* S(t', \mathbf{P}) + \boldsymbol{T}^*)$. Then, we compute $\boldsymbol{\varphi}^G_{t'}$ by replacing the color $\boldsymbol{c}_i$ in Eq. 2 with the projected pixel coordinate as

$$\boldsymbol{\varphi}^G_{t'} = \textstyle\sum_{i \in \mathcal{N}} \pi_{\hat{\boldsymbol{K}}}(\boldsymbol{R}^* S_i(t', \mathbf{P}) + \boldsymbol{T}^*) \alpha_i^{\text{dy}} \prod_{j=1}^{i-1} (1 - \alpha_j^{\text{dy}}), \tag{18}$$

where α_i^{dy} denotes the density of the i^{th} dynamic 3D Gaussian.

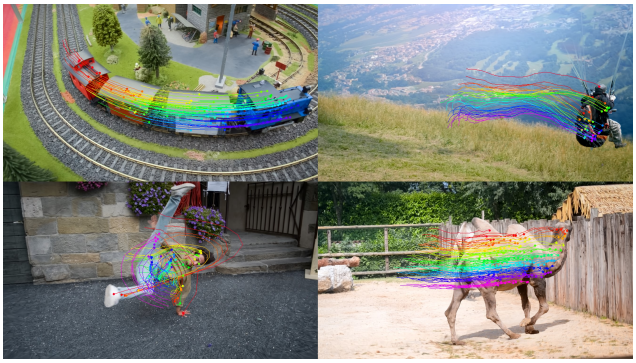
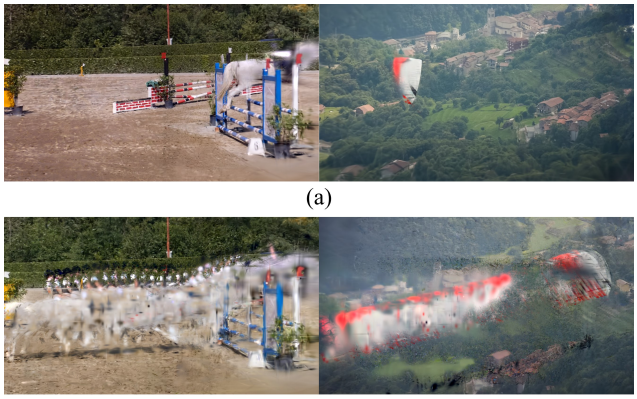


Figure 10. **Visual results of dynamic 3D Gaussian trajectory projected to novel views for our SplineGS.**

As shown in Fig. 6 of the main paper, D3DGS [55] fails to reconstruct dynamic regions. STGS [24] renders dynamic regions more effectively than D3DGS [55], but it still produces poor visualizations of 3D Gaussian trajectories. In the original STGS [24] paper, they propose the temporal opacity $\sigma_i(t)$ as

$$\sigma_i(t) = \sigma_i^s \exp(-s_i^\tau |t - \mu_i^\tau|^2), \tag{19}$$

where μ_i^τ is the temporal center, s_i^τ is the temporal scaling factor and σ_i^s is the time-independent spatial opacity. To further investigate the motion tracking results of STGS [24], we render novel views for STGS [24] after training by setting the opacity of each 3D Gaussian with (a) its original temporal opacity $\sigma_i(t)$ and (b) the fixed value of time-independent spatial opacity σ_i^s, as shown in Fig. 11.



(a)

(b)

Figure 11. **Visual results of novel view synthesis at a specific time using the same STGS [24] models after optimization with (a) their original time-varying opacity and (b) time-independent spatial opacity, respectively.** Please note that we use their original time-varying opacity during training.

We observe that when the opacity of each 3D Gaussian is set to a time-independent value, the rendered novel view synthesis results show multiple instances of the same moving objects (e.g. a horse or a parachute) appearing simultaneously, as illustrated in Fig. 11-(b). This observation suggests that, to represent a moving object across time, STGS [24] may adjust the opacities of *different* sets of 3D Gaussians through their temporal opacities $\sigma_i(t)$, rather than deforming the spatial 3D positions of a *single* set of 3D Gaussians along the temporal axis. While this approach can produce dynamic rendering results, it may not allow for the direct extraction of 3D Gaussian trajectories along the temporal axis. In contrast, our SplineGS with MAS directly models the motion trajectories of dynamic 3D Gaussians, enabling the extraction of more reasonable 3D trajectories, as shown in Fig. 10.

E. Additional Details for Methodology

Camera Intrinsic. To predict the shared camera intrinsics for our camera parameter estimation, we adopt a pinhole camera model which is widely used in COLMAP-free novel view synthesis methods [30, 32, 37, 50] as

$$\boldsymbol{K} = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \tag{20}$$

where $s = 0$ represents the skewness of the camera, while c_x and c_y denote the coordinates of the principal point in

pixels. Without loss of generality, we assume that $f_x = f_y = f$, indicating equal focal lengths in both directions, and set c_x and c_y to half the width and height of the video frame, respectively.

Other 3D Gaussian Attributes. Other attributes of dynamic 3D Gaussians are represented as follows. For the rotation, we adopt a polynomial function inspired by STGS [24], defined as

$$\boldsymbol{q}_i(t) = \boldsymbol{q}_i^0 + \sum_{k=1}^{n_q} \Delta\boldsymbol{q}_{i,k} t^k, \tag{21}$$

where $\boldsymbol{q}_i^0$ is a time-independent base quaternion of the i^{th} dynamic 3D Gaussian and $\Delta\boldsymbol{q}_{i,k}$ is an offset quaternion of the k^{th}-order term of i^{th} dynamic 3D Gaussian, both of which are learnable parameters. we set $n_q = 1$. This ensures a simple yet effective representation of time-dependent rotations [24]. For the scale, we set it to be time-independent when using the NVIDIA dataset [56]. On the other hand, depending on the scene's characteristics, we can learn the scale's deformation by leveraging the Discrete Cosine Transform (DCT) to capture the continuously varying scale of each dynamic 3D Gaussian, inspired by DynIBaR [26]. The scale function is expressed as

$$\begin{aligned} &\boldsymbol{s}_i(t) = \boldsymbol{s}_i^0 + \Delta\boldsymbol{s}_i(t), \\ &\Delta\boldsymbol{s}_i(t) = \sqrt{2/N_f} \sum_{k=1}^{K} \zeta_{i,k} \cos\left(\tfrac{\pi}{2N_f}(2t+1)k\right), \end{aligned} \tag{22}$$

where $\boldsymbol{s}_i^0$ is a time-independent base scale vector of the i^{th} dynamic 3D Gaussian and $\zeta_{i,k} \in \mathbb{R}^3$ represents the k^{th} coefficient of the i^{th} dynamic 3D Gaussian, both of which are learnable parameters. Here, $K = 10$ controls the number of frequency components used in the DCT, allowing flexible yet compact modeling of temporal scale variations. For the color, following STGS [24], we use the splatted feature rendering to predict the final pixel colors. For static regions, we remove the time-encoded feature while preserving the diffuse and specular features.

F. Limitation

In-the-wild videos often exhibit significant and rapid camera and object movements, resulting in blurry input frames. This blurriness subsequently degrades the quality of the rendered novel views. As shown in Fig. 12, the methods solely designed for dynamic scene reconstruction may overfit to the blurry training frames. A straightforward solution is to employ state-of-the-art 2D deblurring methods to enhance the quality of input frames. Additionally, in future research, we plan to integrate a deblurring approach directly into the reconstruction pipeline. This integration could establish a joint deblurring and rendering optimization framework, addressing low-quality issues and enhancing the final rendered outputs without requiring separate preprocessing.

Training Video Frame SplineGS (Ours) D3DGS [55]

Figure 12. **Limitations of our SplineGS.** When the training video frame contains blurriness, our model cannot effectively reconstruct sharp renderings due to the absence of a deblurring method.

G. Additional Qualitative Results

G.1. Novel View Synthesis on NVIDIA

Figs. 13, 14, and 15 present additional visual comparisons for novel view synthesis on the NVIDIA dataset [56].

G.2. Novel View and Time Synthesis on NVIDIA

Figs. 16, 17, and 18 present additional visual comparisons for novel view and time synthesis on the NVIDIA dataset [56].

G.3. Novel View Synthesis on DAVIS

Figs. 19 and 20 present additional visual comparisons for novel view synthesis on the DAVIS dataset [39].

Figure 13. **Visual comparisons for novel view synthesis on the *Jumping* scene from the NVIDIA dataset.**

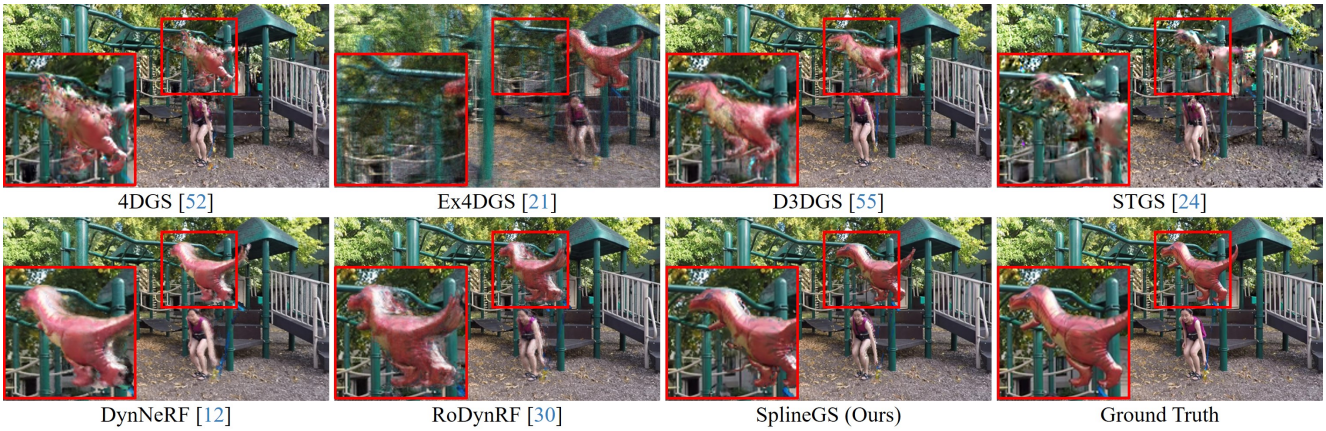


Figure 14. **Visual comparisons for novel view synthesis on the *Playground* scene from the NVIDIA dataset.**

Figure 15. **Visual comparisons for novel view synthesis on the *Truck* scene from the NVIDIA dataset.**

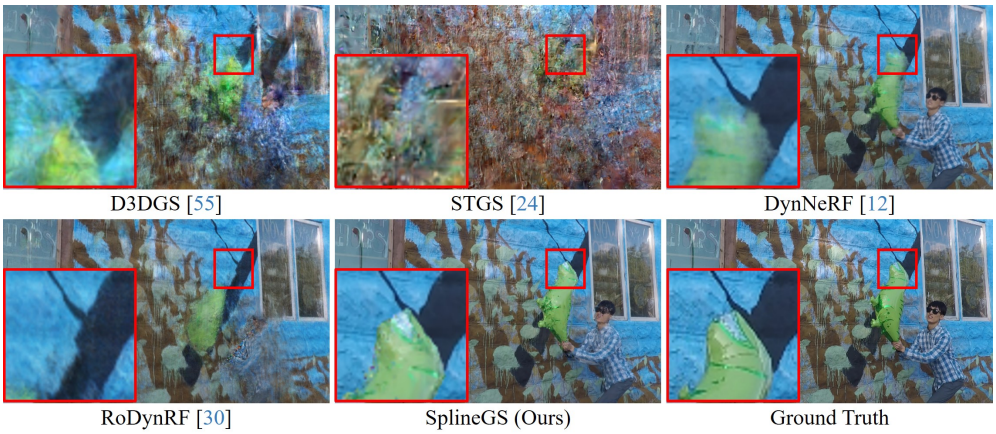


Figure 16. **Visual comparisons for novel view and time synthesis on the *Balloon2* scene from the NVIDIA dataset.**

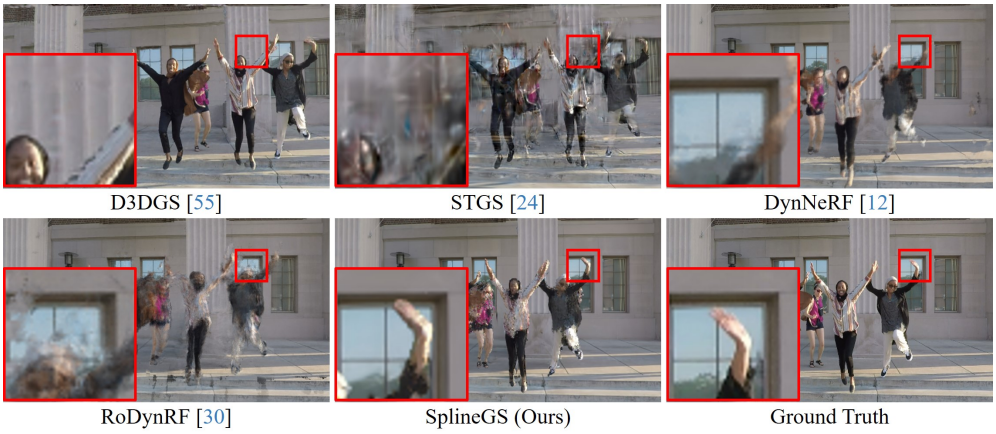


Figure 17. **Visual comparisons for novel view and time synthesis on the *Jumping* scene from the NVIDIA dataset.**

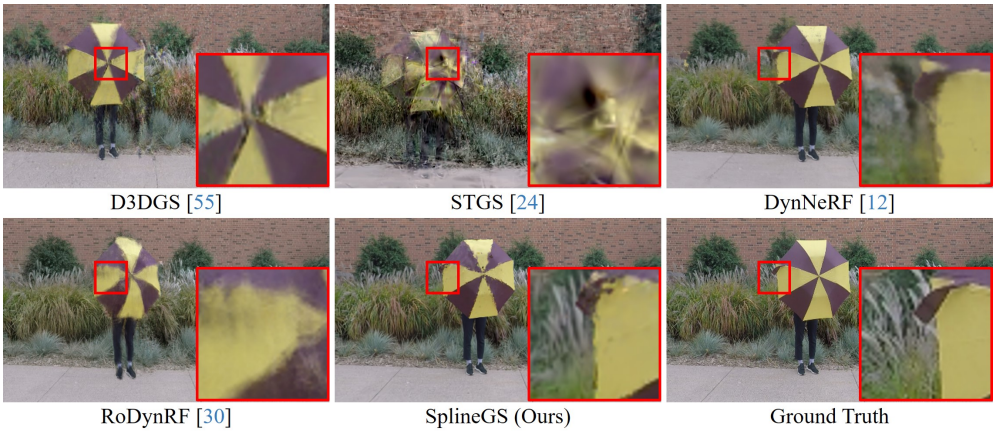


Figure 18. **Visual comparisons for novel view and time synthesis on the *Umbrella* scene from the NVIDIA dataset.**

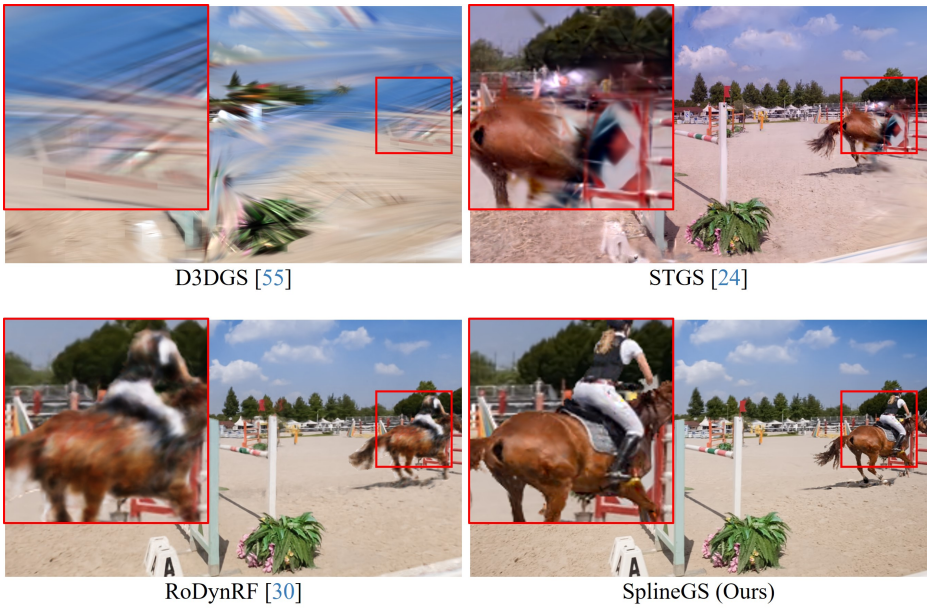


Figure 19. **Visual comparisons for novel view synthesis on the *Horsejump-high* scene from the DAVIS dataset.**

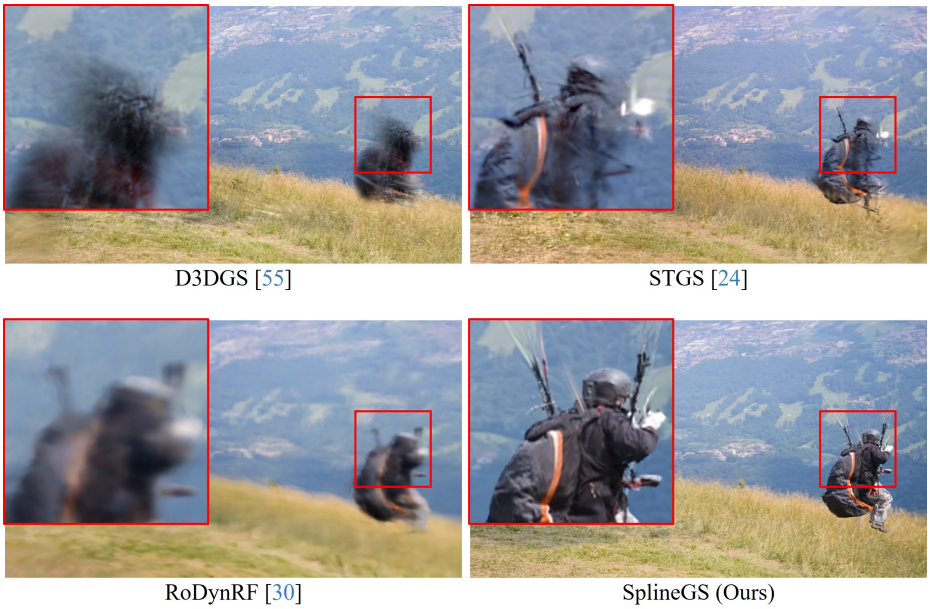


Figure 20. **Visual comparisons for novel view synthesis on the *Paragliding-launch* scene from the DAVIS dataset.**

References

[1] Fast dynamic radiance fields with time-aware neural voxels. In *SIGGRAPH Asia 2022 Conference Papers*, 2022. 2

[2] J Harold Ahlberg, Edwin Norman Nilson, and Joseph Leonard Walsh. *The Theory of Splines and Their Applications: Mathematics in Science and Engineering: A Series of Monographs and Textbooks, Vol. 38*. Elsevier, 2016. 2, 3, 4, 1

[3] ShahRukh Athar, Zexiang Xu, Kalyan Sunkavalli, Eli Shechtman, and Zhixin Shu. Rignerf: Fully controllable neural 3d portraits. In *CVPR*, 2022. 2

[4] 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 *CVPR*, 2023. 2

[5] Jeongmin Bae, Seoha Kim, Youngsik Yun, Hahyun Lee, Gun Bang, and Youngjung Uh. Per-gaussian embedding-based deformation for deformable 3d gaussian splatting. In *European Conference on Computer Vision*, pages 321–335. Springer, 2024. 6

[6] Ang Cao and Justin Johnson. Hexplane: A fast representation for dynamic scenes. In *CVPR*, 2023. 2

[7] Mengyu Chu, You Xie, Jonas Mayer, Laura Leal-Taixé, and Nils Thuerey. Learning temporal coherence via self-supervision for gan-based video generation. *ACM Transactions on Graphics (TOG)*, 2020. 7

[8] C De Boor. A practical guide to splines. *Springer-Verlag google schola*, 1978. 2, 3, 4, 1

[9] Gerald Farin. *Curves and surfaces for CAGD: a practical guide*. Elsevier, 2001. 2, 7, 8

[10] Sara Fridovich-Keil, Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa. K-planes: Explicit radiance fields in space, time, and appearance. In *CVPR*, 2023. 2

[11] Yang Fu, Sifei Liu, Amey Kulkarni, Jan Kautz, Alexei A. Efros, and Xiaolong Wang. Colmap-free 3d gaussian splatting. In *CVPR*, 2024. 3

[12] Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang. Dynamic view synthesis from dynamic monocular video. In *ICCV*, 2021. 2, 6, 7

[13] Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012. 5

[14] Yi-Hua Huang, Yang-Tian Sun, Ziyi Yang, Xiaoyang Lyu, Yan-Pei Cao, and Xiaojuan Qi. Sc-gs: Sparse-controlled gaussian splatting for editable dynamic scenes. *CVPR*, 2024. 2, 6

[15] Woobin Im, Geonho Cha, Sebin Lee, Jumin Lee, Juhyeong Seon, Dongyoon Wee, and Sung-Eui Yoon. Regularizing dynamic radiance fields with kinematic fields. In *European Conference on Computer Vision*, pages 312–328. Springer, 2024. 6

[16] Wei Jiang, Kwang Moo Yi, Golnoosh Samei, Oncel Tuzel, and Anurag Ranjan. Neuman: Neural human radiance field from a single video. In *ECCV*, 2022. 2

[17] Yifan Jiang, Peter Hedman, Ben Mildenhall, Dejia Xu, Jonathan T Barron, Zhangyang Wang, and Tianfan Xue. Alignerf: High-fidelity neural radiance fields via alignment-aware training. In *CVPR*, 2023. 3

[18] Moritz Kappel, Florian Hahlbohm, Timon Scholz, Susana Castillo, Christian Theobalt, Martin Eisemann, Vladislav Golyanik, and Marcus Magnor. D-npc: Dynamic neural point clouds for non-rigid view synthesis from monocular video. *arXiv preprint arXiv:2406.10078*, 2024. 6

[19] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker: It is better to track together. *arXiv*, 2023. 4, 1, 2

[20] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 2023. 2, 3, 4, 7, 1

[21] Junoh Lee, Chang-Yeon Won, Hyunjun Jung, Inhwan Bae, and Hae-Gon Jeon. Fully explicit dynamic gaussian splatting. In *NeurIPS*, 2024. 3, 6, 1

[22] Yao-Chih Lee, Zhoutong Zhang, Kevin Blackburn-Matzen, Simon Niklaus, Jianming Zhang, Jia-Bin Huang, and Feng Liu. Fast view synthesis of casual videos with soup-of-planes. In *ECCV*, 2024. 3, 6

[23] Jiahui Lei, Yijia Weng, Adam Harley, Leonidas Guibas, and Kostas Daniilidis. Mosca: Dynamic gaussian fusion from casual videos via 4d motion scaffolds. *arXiv*, 2024. 3, 6

[24] Zhan Li, Zhang Chen, Zhong Li, and Yi Xu. Spacetime gaussian feature splatting for real-time dynamic view synthesis. In *CVPR*. 1, 2, 3, 6, 7, 8

[25] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *CVPR*, 2021. 2, 7

[26] Zhengqi Li, Qianqian Wang, Forrester Cole, Richard Tucker, and Noah Snavely. Dynibar: Neural dynamic image-based rendering. In *CVPR*, 2023. 2, 3

[27] Yiqing Liang, Numair Khan, Zhengqin Li, Thu Nguyen-Phuoc, Douglas Lanman, James Tompkin, and Lei Xiao. Gaufre: Gaussian deformation fields for real-time dynamic novel view synthesis. *arXiv*, 2023. 2, 3

[28] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. In *ICCV*, 2021. 3

[29] Philipp Lindenberger, Paul-Edouard Sarlin, Viktor Larsson, and Marc Pollefeys. Pixel-perfect structure-from-motion with featuremetric refinement. In *ICCV*, 2021. 3

[30] Yu-Lun Liu, Chen Gao, Andreas Meuleman, Hung-Yu Tseng, Ayush Saraf, Changil Kim, Yung-Yu Chuang, Johannes Kopf, and Jia-Bin Huang. Robust dynamic radiance fields. In *CVPR*, 2023. 2, 3, 5, 6, 7

[31] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. In *3DV*, 2024. 2

[32] Quan Meng, Anpei Chen, Haimin Luo, Minye Wu, Hao Su, Lan Xu, Xuming He, and Jingyi Yu. Gnerf: Gan-based neural radiance field without posed camera. In *ICCV*, 2021. 3, 2

[33] Xingyu Miao, Yang Bai, Haoran Duan, Fan Wan, Yawen Huang, Yang Long, and Yefeng Zheng. Ctnerf: Cross-time transformer for dynamic neural radiance field from monocular video. *Pattern Recognition*, 156:110729, 2024. 6

[34] 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 *ECCV*, 2020. 2, 5

[35] 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 *ICCV*, 2021. 2

[36] 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 Trans. Graph.*, 2021. 2

[37] Keunhong Park, Philipp Henzler, Ben Mildenhall, Jonathan T. Barron, and Ricardo Martin-Brualla. Camp: Camera preconditioning for neural radiance fields. *ACM Trans. Graph.*, 2023. 3, 2

[38] Luigi Piccinelli, Yung-Hsu Yang, Christos Sakaridis, Mattia Segu, Siyuan Li, Luc Van Gool, and Fisher Yu. Unidepth: Universal monocular metric depth estimation. In *CVPR*, 2024. 4, 5, 1

[39] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv*, 2018. 1, 6, 7, 3

[40] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *CVPR*, 2021. 2

[41] Vincent Raoult, Sarah Reid-Anderson, Andreas Ferri, and Jane E Williamson. How reliable is structure from motion (sfm) over time and between observers? a case study using coral reef bommies. *Remote Sensing*, 2017. 3

[42] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 1, 2, 3, 5, 6, 7

[43] Ruizhi Shao, Zerong Zheng, Hanzhang Tu, Boning Liu, Hongwen Zhang, and Yebin Liu. Tensor4d: Efficient neural 4d decomposition for high-fidelity dynamic reconstruction and rendering. In *CVPR*, 2023. 2

[44] 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*, 2023. 2

[45] Carole H. Sudre, Wenqi Li, Tom Vercauteren, Sebastien Ourselin, and M. Jorge Cardoso. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, 2017. 5

[46] Fengrui Tian, Shaoyi Du, and Yueqi Duan. MonoNeRF: Learning a generalizable dynamic radiance field from monocular videos. In *ICCV*, 2023. 6

[47] Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Christoph Lassner, and Christian Theobalt. Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video. In *ICCV*, 2021. 2

[48] Diwen Wan, Ruijie Lu, and Gang Zeng. Superpoint gaussian splatting for real-time high-fidelity dynamic scene reconstruction. *arXiv preprint arXiv:2406.03697*, 2024. 6

[49] Qianqian Wang, Vickie Ye, Hang Gao, Jake Austin, Zhengqi Li, and Angjoo Kanazawa. Shape of motion: 4d reconstruction from a single video. *arXiv preprint arXiv:2407.13764*, 2024. 6

[50] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. Nerf-: Neural radiance fields without known camera parameters. *CoRR*, 2021. 3, 2

[51] Chung-Yi Weng, Brian Curless, Pratul P. Srinivasan, Jonathan T. Barron, and Ira Kemelmacher-Shlizerman. HumanNeRF: Free-viewpoint rendering of moving people from monocular video. In *CVPR*, 2022. 2

[52] 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. In *CVPR*. 2, 6, 7, 8, 1

[53] Gengshan Yang, Minh Vo, Neverova Natalia, Deva Ramanan, Vedaldi Andrea, and Joo Hanbyul. Banmo: Building animatable 3d neural models from many casual videos. In *CVPR*, 2022. 2

[54] Jinyu Yang, Mingqi Gao, Zhe Li, Shang Gao, Fangjing Wang, and Feng Zheng. Track anything: Segment anything meets videos. *arXiv*, 2023. 5

[55] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In *CVPR*, 2024. 1, 2, 6, 7, 8

[56] Jae Shin Yoon, Kihwan Kim, Orazio Gallo, Hyun Soo Park, and Jan Kautz. Novel view synthesis of dynamic scenes with globally coherent depths from a monocular camera. In *CVPR*, 2020. 1, 6, 7, 3

[57] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 6

[58] Kaichen Zhou, Jia-Xing Zhong, Sangyun Shin, Kai Lu, Yiyuan Yang, Andrew Markham, and Niki Trigoni. Dynpoint: Dynamic neural point for view synthesis. *Advances in Neural Information Processing Systems*, 36:69532–69545, 2023. 6