Gear-NeRF: Free-Viewpoint Rendering and Tracking with Motion-aware Spatio-Temporal Sampling —Supplementary Materials —

Xinhang Liu[1] †* Yu-Wing Tai[2] Chi-Keung Tang[1] †
Pedro Miraldo[3] Suhas Lohit[3] Moitreya Chatterjee[3]
[1]HKUST [2]Dartmouth College [3]Mitsubishi Electric Research Laboratories (MERL)
xliufe@connect.ust.hk, yu-wing.tai@dartmouth.edu, cktang@cse.ust.hk,
miraldo@merl.com, slohit@merl.com, metro.smiles@gmail.com

We begin this supplementary document by reporting per-scene rendering results of Gear-NeRF compared to competing methods, both qualitatively and quantitatively. In Section **B**, we present performance comparisons for the task of tracking in novel views, a new contribution of this work, and compare against baselines adapted for this task. We then present additional ablation studies, discussing the sensitivity of our method to the choice of appropriate hyper-parameters in Section **C**. Besides, we provide a video showing results for dynamic novel view rendering and tracking.

The following summarizes the supplementary materials we present:

1. Per-Scene Rendering Results.
2. Novel-view Tracking Results.
3. Additional Ablation Studies.
4. Discussions on Training and Rendering Efficiency.
5. **A video** showing novel view rendering and tracking in dynamic scenes using Gear-NeRF and other competing methods on different datasets, compiled together in *supplementary_video.mp4*.

A. Per-Scene Rendering Results

In this section, we present a quantitative evaluation of Gear-NeRF and competing techniques for the task of rendering dynamic scenes from novel views, on a per-scene basis for each of the three datasets we conduct experiments on: (i) The Technicolor Lightfield Dataset [8] (ii) The Neural 3D Video Dataset [7], and the (iii) The Google Immersive Dataset [2]. Moreover, to further demonstrate the generalizability of our method vis-á-vis our closest competing baseline, HyperReel [1], we report its performance versus that of our method on some additional sequences for each of these three datasets.

Table **A**, Table **B**, and Table **C** show per-scene quantitative comparison results of our approach against competing methods on the Technicolor dataset [8], the Neural 3D Video dataset [7], and the Google Immersive dataset [2], respectively. The averaged results are presented in Table 1 of the paper and are derived from these per-scene results. We see that in all but a couple of sequences ("Cut Roasted Beef" from the Neural 3D video dataset oe "Theater" from the Technicolor dataset) our proposed approach outperforms all other competing methods, across all the metrics, attesting to the effectiveness of our method. Even

Table A. **Per-scene quantitative comparisons for the task of novel view synthesis for dynamic scenes on the Technicolor dataset [8]. Best** and second best results are highlighted.

Scene	Method	PSNR (↑)	SSIM (↑)	LPIPS (↓)
Train	ST-NeRF [11]	29.16	0.877	0.070
	HyperReel [1]	29.18	0.894	0.054
	MixVoxels [9]	27.34	0.830	0.058
	Ours	**30.55**	**0.957**	**0.049**
Theater	ST-NeRF [11]	31.57	0.866	0.133
	HyperReel [1]	31.69	0.863	0.131
	MixVoxels [9]	27.34	**0.888**	0.134
	Ours	**32.56**	0.887	**0.067**
Painter	ST-NeRF [11]	35.14	0.911	0.102
	HyperReel [1]	35.38	0.916	0.091
	MixVoxels [9]	34.18	0.900	0.076
	Ours	**36.35**	**0.928**	**0.073**
Birthday	ST-NeRF [11]	27.55	0.877	0.097
	HyperReel [1]	27.91	0.873	0.090
	MixVoxels [9]	27.11	0.749	0.142
	Ours	**29.38**	**0.904**	**0.041**

*Work mainly done when XL was an intern at MERL.

†XL and CT are supported in part by the Research Grant Council of the Hong Kong SAR grant no. 16201420.

Figure A. **Qualitative comparisons of competing methods for the task of novel view synthesis of some additional dynamic scenes for the Google Immersive [2] (top row) and the Neural 3D Video [7] (bottom row) datasets.**

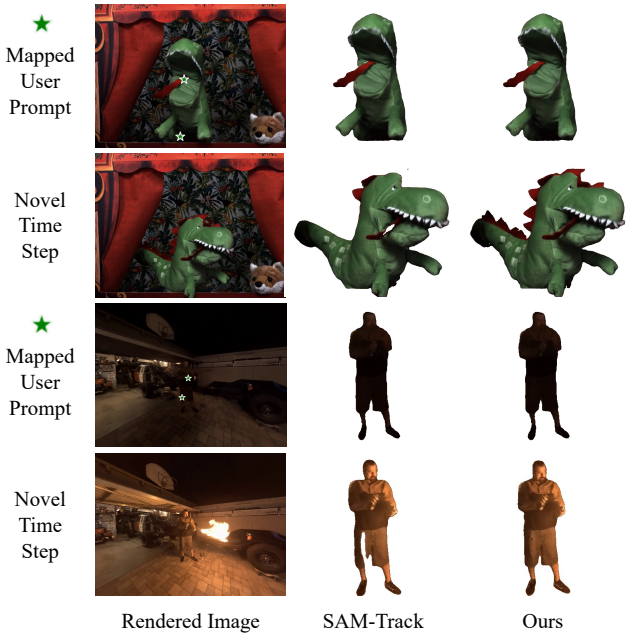

Figure B. **Qualitative comparisons of click-based novel-view tracking of our method versus SAM-Track [5].**

under occasional circumstances when that is not the case, our method still reports performance comparable to HyperReel. Figure A presents qualitative comparisons of rendering results, by our method and HyperReel for some sequences from the Google Immersive [2] and the Neural 3D Video [7] datasets. As is evident from the figure, the frames synthesized by our method look less blurry and better preserves the details (for instance the eye of the lady, the flame, the stem of the glass, or the glasses of the man with the hat) which underscores the effectiveness of our method. More qualitative results can be seen in the attached video.

Non-Lambertian Surfaces: While non-Lambertian surfaces are known to pose challenges for rendering, we observe that they don't undermine the gear selection, perhaps due to object priors from SAM. E.g. the *car* in the scene in Figure C includes reflective surfaces, like windshield yet it is assigned the right gear with its details better reconstructed than competing methods.

Figure C. **Gear selection and rendering of non-Lambertian objects.**

B. Novel-View Tracking Results

Being the first method to achieve free-viewpoint tracking of target objects in the NeRF setting, our approach does not have direct baselines, to the best of our knowledge. Hence, we use the following as baselines for benchmarking: (i) The static scene segmentation approach, SA3D [4] mentioned in Section 5 of the main paper. (ii) We also compare against a monocular video tracking baseline called SAM-Track [5] – a method based on SAM [6] for object tracking in monocular videos. Since SAM-Track only takes a monocular video as input and does not consider the 3D information, we adopted the following procedure to use it as a baseline: Given user-provided click(s) in an input view, we utilize our radiance field representation to map these clicks to a desired target/novel view. SAM-Track can then be used to perform object tracking in the target view using the mapped click(s) as prompts. As the quantitative results in Table D indicate, our method outperforms SAM-Track across all metrics on all datasets for the task of desired novel/target view object tracking. This may be attributed to our method's capability of learning the semantics of the scene by leveraging the 4D SAM embedding field. A rendered SAM feature map is fed into the SAM decoder to obtain the mask of the target ob-

Table B. **Per-scene quantitative comparisons for the task of novel view synthesis for dynamic scenes on the Google Immersive dataset [2]. Best** and second best results are highlighted.

Scene	Method	PSNR (↑)	SSIM (↑)	LPIPS (↓)
Flames	HexPlane [3]	29.31	0.808	0.189
	HyperReel [1]	29.66	0.895	0.129
	MixVoxels [9]	29.01	0.819	0.180
	Ours	**30.87**	**0.903**	**0.120**
Truck	HexPlane [3]	26.89	0.819	0.161
	HyperReel [1]	27.20	0.850	0.153
	MixVoxels [9]	26.59	0.877	0.194
	Ours	**27.46**	**0.892**	**0.136**
Horse	HexPlane [3]	28.45	0.887	0.121
	HyperReel [1]	28.56	0.892	0.114
	MixVoxels [9]	28.13	0.773	0.190
	Ours	**29.05**	**0.895**	**0.110**
Car	HexPlane [3]	24.13	0.719	0.261
	HyperReel [1]	24.58	0.740	0.215
	MixVoxels [9]	24.37	0.724	0.249
	Ours	**25.12**	**0.783**	**0.179**
Welder	HexPlane [3]	25.89	0.778	0.250
	HyperReel [1]	26.07	0.793	0.220
	MixVoxels [9]	24.59	**0.818**	0.277
	Ours	**26.36**	0.810	**0.187**
Exhibit	HexPlane [3]	29.93	0.874	0.159
	HyperReel [1]	31.53	0.907	0.090
	MixVoxels [9]	28.35	0.915	0.148
	Ours	**31.73**	**0.920**	**0.064**
Face Paint 1	HexPlane [3]	28.48	0.841	0.169
	HyperReel [1]	**29.83**	**0.922**	0.093
	MixVoxels [9]	27.84	0.847	0.185
	Ours	29.15	0.901	**0.082**
Face Paint 2	HexPlane [3]	28.58	0.833	0.148
	HyperReel [1]	28.94	0.893	0.106
	MixVoxels [9]	27.50	0.849	0.231
	Ours	**29.24**	**0.903**	**0.076**
Cave	HexPlane [3]	27.35	0.715	0.231
	HyperReel [1]	28.48	0.867	0.184
	MixVoxels [9]	27.93	0.894	0.224
	Ours	**29.68**	**0.880**	**0.144**

ject at every time step. In contrast, SAM-Track uses SAM to acquire the object mask only for the first frame and employs a mask tracker [10] to obtain masks for subsequent time steps. This is also demonstrated in Figure **B** where our approach better renders the scene without introducing artifacts as opposed to SAM-Track. More qualitative results can be seen in the attached video.

Table **E** reveals that our approach better segments the target object, given a rendered frame, as compared to SA3D [4]. We attribute this gain to the fact that our method unlike SA3D reasons about the temporal dynamics of the scene and can thus better assess/predict the location of the target object.

Table C. **Per-scene quantitative comparisons for the task of novel view synthesis for dynamic scenes on the Neural 3D Video [7]. Best** and second best results are highlighted.

Scene	Method	PSNR (↑)	SSIM (↑)	LPIPS (↓)
Cut Roasted Beef	ST-NeRF [11]	**32.97**	0.950	0.047
	HyperReel [1]	32.63	0.942	**0.049**
	MixVoxels [9]	32.34	**0.962**	0.138
	Ours	32.74	0.944	0.057
Coffee Martini	ST-NeRF [11]	29.18	0.904	0.102
	HyperReel [1]	28.43	0.896	0.090
	MixVoxels [9]	28.08	0.901	0.079
	Ours	**29.71**	**0.918**	**0.070**
Flame Steak	ST-NeRF [11]	31.75	0.903	0.061
	HyperReel [1]	32.49	0.946	0.051
	MixVoxels [9]	31.54	0.946	0.133
	Ours	**33.20**	**0.952**	**0.045**
Cook Spinach	ST-NeRF [11]	32.84	0.942	0.049
	HyperReel [1]	32.56	0.940	0.056
	MixVoxels [9]	31.71	**0.960**	0.144
	Ours	**33.18**	0.946	**0.046**
Flame Salmon	ST-NeRF [11]	27.74	0.781	0.132
	HyperReel [1]	28.03	0.891	0.100
	MixVoxels [9]	28.88	**0.930**	0.212
	Ours	**29.66**	0.912	**0.073**
Sear Steak	ST-NeRF [11]	31.72	0.862	0.094
	HyperReel [1]	**32.58**	0.951	**0.046**
	MixVoxels [9]	31.60	**0.967**	0.128
	Ours	32.31	0.942	0.054

Table D. **Quantitative comparisons for fixed novel view tracking versus SAM-Track [5].**

Dataset	Method	mIoU	Accuracy
Technicolor [8]	SAM-Track [5]	95.6	96.1
	Ours	**96.0**	**96.9**
Neural 3D Video [7]	SAM-Track [5]	94.1	94.5
	Ours	**95.1**	**95.5**
Google Immersive [2]	SAM-Track [5]	93.4	94.0
	Ours	**95.7**	**96.3**

Table E. **Quantitative comparisons for free-viewpoint tracking:** t-mIoU and t-Acc are metrics used for evaluating novel view masks at novel time steps, not applicable to SA3D. Reported metrics are averages over all scenes for each dataset.

Dataset	Method	mIoU (↑)	Acc. (↑)	t-mIoU (↑)	t-Acc. (↑)
Technicolor [8]	SA3D [4]	96.4	97.1	N/A	N/A
	Ours	**97.4**	**97.6**	**92.1**	**93.3**
Google Immersive [2]	SA3D [4]	94.1	94.8	N/A	N/A
	Ours	**94.3**	**95.0**	**91.5**	**92.8**
Neural 3D Video [7]	SA3D [4]	93.1	94.0	N/A	N/A
	Ours	**93.4**	**94.3**	**90.6**	**92.3**

Table F. **Ablation study on the top-k selection in gear assignment. Best** and second best results are highlighted.

Method	PSNR (↑)	SSIM (↑)	LPIPS (↓)
Ours ($k = 1$)	27.10	0.879	0.139
Ours ($k = 2$)	27.43	0.890	0.145
Ours ($k = 3$)	**27.49**	**0.892**	**0.136**
Ours ($k = 4$)	26.14	0.777	0.158
Ours ($k = 5$)	26.39	0.790	0.161

Table G. **Ablation Study on the point splitting strategy in motion-aware spatial sampling. Best** and second best results are highlighted.

Method	PSNR (↑)	SSIM (↑)	LPIPS (↓)
Ours ($2^{p(\mathbf{x},t)-1}$)	27.49	0.892	0.136
Ours ($3^{p(\mathbf{x},t)-1}$)	**27.98**	**0.914**	**0.125**
Ours ($2p(\mathbf{x},t)-1$)	26.46	0.815	0.140

C. Additional Ablation Studies

In this section, we present some additional ablation results on the hyper-parameters of our model.

Top-k in Gear Assignment Updates: For gear assignment updates, we employ a patch-based approach to identify regions with the top-k highest or lowest average rendering loss to obtain positive or negative prompts for subsequent steps. We perform an ablation study on the *Truck* scene of the Google Immersive dataset [2]. Table **F** reveals that both excessively high or low values of k do not yield optimal performance. We note that a selection of $k = 4$ or 5 leads to gear upshifts for inappropriate regions, weakening the efficacy of our motion-aware spatio-temporal sampling strategy. In our experiments, we uniformly applied $k = 3$ across all scenes, which yielded satisfactory results.

Sampling Point Splitting: In our motion-aware spatial sampling, we adopt a 3D sampling point-splitting strategy. Specifically, we split each sampled 3D point into $2^{p(\mathbf{x},t)-1}$ points. We conduct an ablation study on the number of points a sampling point is split into. To elaborate, in addition to splitting one point into $2^{p(\mathbf{x},t)-1}$ points, we explore variants, including splitting into $3^{p(\mathbf{x},t)-1}$ points and $2p(\mathbf{x},t) - 1$ points, on the *Truck* scene of the Google Immersive dataset [2]. As shown in Table **G**, the additional sampling points generated by the $2p(\mathbf{x},t) - 1$ strategy are insufficient, resulting in a decrease in rendering quality. In contrast, $3^{p(\mathbf{x},t)-1}$ achieves better quality than $2^{p(\mathbf{x},t)-1}$. However, an excessive number of sampling points leads to a reduction in training speed, while providing a marginal performance boost, which is why we stick with the strategy of splitting into $2^{p(\mathbf{x},t)-1}$ points.

References

[1] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1, 3

[2] Michael Broxton, John Flynn, Ryan Overbeck, Daniel Erickson, Peter Hedman, Matthew Duvall, Jason Dourgarian, Jay Busch, Matt Whalen, and Paul Debevec. Immersive light field video with a layered mesh representation. *ACM Transactions on Graphics (TOG)*, 39(4):86–1, 2020. 1, 2, 3, 4

[3] Ang Cao and Justin Johnson. Hexplane: A fast representation for dynamic scenes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 130–141, 2023. 3

[4] Jiazhong Cen, Zanwei Zhou, Jiemin Fang, Chen Yang, Wei Shen, Lingxi Xie, Dongsheng Jiang, Xiaopeng Zhang, and Qi Tian. Segment anything in 3d with nerfs, 2023. 2, 3

[5] Yangming Cheng, Liulei Li, Yuanyou Xu, Xiaodi Li, Zongxin Yang, Wenguan Wang, and Yi Yang. Segment and track anything. *arXiv preprint arXiv:2305.06558*, 2023. 2, 3

[6] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 2

[7] Tianye Li, Mira Slavcheva, Michael Zollhoefer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, Richard Newcombe, et al. Neural 3d video synthesis from multi-view video. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5521–5531, 2022. 1, 2, 3

[8] Neus Sabater, Guillaume Boisson, Benoit Vandame, Paul Kerbiriou, Frederic Babon, Matthieu Hog, Remy Gendrot, Tristan Langlois, Olivier Bureller, Arno Schubert, et al. Dataset and pipeline for multi-view light-field video. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 30–40, 2017. 1, 3

[9] Feng Wang, Sinan Tan, Xinghang Li, Zeyue Tian, and Huaping Liu. Mixed neural voxels for fast multi-view video synthesis. *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 1, 3

[10] Zongxin Yang and Yi Yang. Decoupling features in hierarchical propagation for video object segmentation, 2022. 3

[11] Jiakai Zhang, Xinhang Liu, Xinyi Ye, Fuqiang Zhao, Yanshun Zhang, Minye Wu, Yingliang Zhang, Lan Xu, and Jingyi Yu. Editable free-viewpoint video using a layered neural representation. *ACM Transactions on Graphics (TOG)*, 40(4):1–18, 2021. 1, 3