

Supplementary Material

In this supplementary material, we provide additional details on the datasets used in our experiments (Appendix A) and our training cost and more ablation studies (Appendix B). On the results side, we first show more qualitative static and dynamic reconstruction results (Appendix C, Appendix D). We further justify our choice of the bullet-time formulation by showing that our reconstruction is **temporally smooth** (Appendix E), as well as an simple extension to unlock the power for **estimating dynamic deformations** (Appendix F).

A Dataset Details

The static datasets used in our training are as follows: OBJAVERSE [1] is a synthetic object-centric dataset, and we use the 80K-object subset from [2]. MVIMGNET [3] is a real-world object-centric dataset that has 220K objects. RE10K [4] is a real-world scene dataset that has 80K video clips. DL3DV [5] is a real-world scene dataset that has 10K video. We sample DL3DV 10 times more frequently than other datasets to balance the number of training samples. We use a spatial scale of 8 for Objaverse and scale 1 for all other datasets.

The dynamic datasets used in our training are as follows: KUBRICMV is a synthetic multi-view video dataset that has 3K scenes. We rendered this dataset using the Kubric [6] engine. The scene setup follows Movi-E [6] and videos are rendered from all camera poses in the camera trajectory so it produces a multi-view video. POINTODYSSEY [7] is a synthetic monocular dataset with 131 scenes. DYNAMICREPLICA [8] is a synthetic stereo video dataset with 484 training sequences. SPRING [9] is a synthetic stereo video dataset with 37 scenes. PANDA-70M [10] is a real-world monocular video dataset. We use around 40K clips filtered from a random subset. We use scale 6 for Spring and DynamicReplica and 1 for other datasets. More details can be found in Tab. S1

Dataset	Dynamic	Subject	Domain	#Views	#Frames	#Scenes	#Multiplies	#Scale
RE10K [4]		<i>S</i>	Real	-	10M	80K	1	1
MVImgNet [3]		<i>O</i>	Real	-	6.5M	220K	1	1
Objaverse [1]		<i>O</i>	Synthetic	-	4M	80K	1	8
DL3DV [5]		<i>S</i>	Real	-	51M	10K	10	1
PointOdyssey [7]	✓	<i>O+S</i>	Synthetic	1	6K	131	3e3	1
Kubric-MV [6]	✓	<i>O+S</i>	Synthetic	24	70K	3K	2e2	1
DynamicReplica [8]	✓	<i>O+S</i>	Synthetic	2	145K	484	8e2	6
Spring [9]	✓	<i>O+S</i>	Synthetic	2	200K	37	1e4	6
PANDA-70M [10]	✓	<i>O+S</i>	Real	1	19M	40K	10	1

Table S1: **Datasets.** **Dynamic** indicates if the dataset is dynamic or static. **Subject** indicates if the dataset is object-centric (*O*) or scene-centric (*S*). **Domain** indicates if the dataset is captured from the real world or is synthesized. **#Views** denotes the number of synchronized views for a dynamic video. **#Frames** and **#Scenes** are the numbers of image frames and unique scenes in the dataset respectively. **#Multiplies** denotes the number of multiplies we sample the dataset (by scene) in training for balance. **#Scale** is the scale we applied to the dataset so that all datasets have approximately the same metric scale.

B Training Cost Analysis and Effect of Batch Size

The full training of BulletTimer takes ~ 4 days on 32 NVIDIA A100 GPUs. As illustrated in Fig. S1, the training cost is comparable to existing feed-forward 3D reconstruction methods, such as LVSM [18] and LRM [19] (384 GPU-days) or GS-LRM [17] (192 GPU-days). Like these methods, our work also functions as an amortized algorithm: once trained, the inference cost becomes negligible. Taking inference cost also into consideration, per-scene optimization quickly becomes more expensive, with the difference becoming more pronounced with the growing number of scenes.

Fig. S2 shows the results of training our model with 1 GPU, 8 GPUs, and 32 GPUs. Although inference fits on a single GPU (see Tab. S5), our training benefits from large batch sizes so we used

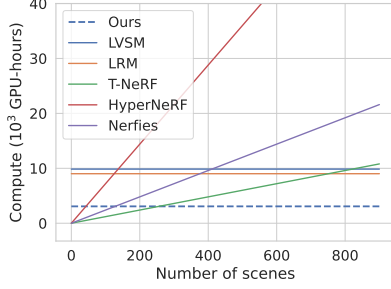


Figure S1: Computation cost.

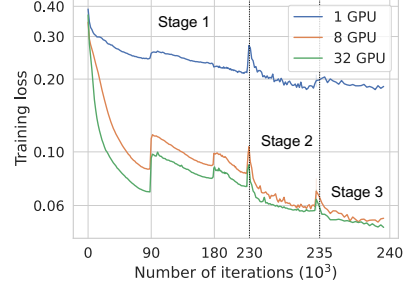


Figure S2: Batch-size ablation.



Figure S3: **Comparison on static scene dataset.** We compare our renderings with the baseline models, trained and tested on the RE10K dataset.

32 32 GPUs (each GPU holds a single batch). The same number of GPUs was also used in both LVSM
 33 and GS-LRM. In line with other fields (LLMs, GenAI), we regard the scalability of our method one
 34 of its key strengths. For ease of reproduction and fine-tuning, we will release our source code and
 35 pretrained checkpoints.

36 C Qualitative Results on Dynamic Scenes

37 We have provided video results on the DyCheck Benchmark [20] and NVIDIA Dynamic Scene
 38 Benchmark [21] in the supplementary material. Additionally, we include novel view synthesis
 39 videos for the DAVIS dataset, DyCheck iPhone dataset, and SORA scenes. We also showcase a
 40 video demonstrating the effects of the NTE module, along with our video results on the Tanks &
 41 Temples static scenes. For access to these video results, please refer to our website by opening the
 42 `index.html` file into a modern browser.

43 D More Results on Static Datasets

44 We provide a comprehensive qualitative comparison of our method against the baselines, MVS-
 45 plat [16] and PixelSplat [15], on the RE10K dataset, as shown in Fig. S3. For each scene, the figure
 46 also displays the input views provided to the networks. Compared to the baselines, our method
 47 produces sharper outputs and more closely aligns with the ground-truth renderings. Note that all
 48 the methods used for the evaluation in this figure are trained exclusively on RE10K. Additionally,
 49 we use two views as context for all methods to ensure fairness in evaluation and to align with the
 50 setup of the baselines. Tab. S2 presents a quantitative evaluation against the baselines under the same
 51 settings. While our static model achieves the best performance among the baselines, our dynamic

Model	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow
PixelNeRF [11]	20.43	0.589	0.550
GPNR [12]	24.11	0.793	0.250
AttnRend [13]	24.78	0.820	0.213
MuRF [14]	26.10	0.858	0.143
PixelSplat [15]	25.89	0.858	0.142
MVSplat [16]	26.39	0.869	0.128
GS-LRM [17]	28.10	0.892	0.114
Ours-Static	26.49	0.886	0.096
Ours-Static†	28.91	0.920	0.070
Ours-BTimer	26.82	0.891	0.089

Table S2: **Quantitative comparison of models performance on RE10K test set.** To be consistent with the baselines, we adopt the 256×256 resolution. Our Bullet Timer has been trained on both static and dynamic scenes, while the other model is only trained on RE10K training set. We highlight the **best**, **second best** and **third best** models. \dagger : 4 input views.



Figure S4: A diverse set of scenes reconstructed using our static model, trained on multiple datasets and capable of generalizing to various scenarios.

52 *BTimer* model, trained for the dynamic task, also demonstrates strong performance on the static task,
53 ranking third on the static benchmark.

54 Our complete static model, trained across all datasets, is capable of reconstructing a highly diverse
55 set of environments. Fig. S4 showcases our model’s reconstructions across a wide variety of scenes,
56 including outdoor forward-facing, outdoor drone shots, outdoor 360-degree views, indoor 360-degree
57 views, and indoor forward-facing scenes, as well as object-centric synthetic scenes. Notably, all these
58 reconstructions are achieved using a shared set of weights, demonstrating that our model, trained
59 across multiple datasets, generalizes effectively to different scenarios.

60 To further demonstrate the importance of training on multiple datasets for generalization to unseen
61 datasets, we conduct an ablation study on the datasets used to train our static model. Tab. S3 compares
62 the performance of our model when trained individually on a single dataset—RE10K, MVImageNet,
63 DL3DV, or Objaverse—against its performance when trained on all these datasets simultaneously.
64 The evaluation is conducted on a completely unseen dataset, the Tanks & Temples split from the
65 InstantSplat [22] paper. Our model, whether static or dynamic, trained on all datasets significantly
66 outperforms the single-dataset models.

Model	Datasets	PSNR↑	SSIM↑	LPIPS↓
GS-LRM* [17]	RE10K	17.56	0.546	0.310
Ours-Static	Objaverse	7.00	0.363	0.668
	MVImageNet	17.75	0.530	0.343
	DL3DV	17.92	0.566	0.278
	All Static	24.22	0.807	0.093
Ours-Full	+Dynamic	24.13	0.806	0.093

Table S3: **Baseline comparisons on the Tanks & Temples dataset (InstantSplat split).** Test views are 512×512 . LPIPS are computed on 256×256 . We highlight the **best**, **second best** and **third best** models. *: Our reproduced results.

Method	PSNR ↑
w/o 3D Pretrain	17.94
w/ Re10K only 3D Pretrain	21.29
w/o static Co-train	22.79
w/o interpolation supervision	20.54
Full model	24.00

Table S4: Quantitative ablation results on NVIDIA Dynamic Scene Benchmark. Ablation models are trained with 4 context frames.

#Ctx.	Res.	Time	Mem.
4	256^2	0.02s	1.42G
12	256^2	0.15s	2.60G
12	512^2	1.55s	9.68G

Table S5: Inference cost. Model is evaluated on a single NVIDIA A100 GPU.

E Evaluation of Temporal Smoothness

Since our method reconstructs each timestamp individually, it is necessary to understand its temporal smoothness. In this section, we quantitatively evaluate the temporal smoothness of the reconstructed dynamic scenes, with results shown in Tab. S6. We render the reconstructed DyCheck [20] scenes from one of the evaluation fixed cameras and evaluate the rendered video using the *Temporal Flickering* metric in VBench [29]. Concretely, the metric computes the pixel-wise Mean Absolute Error in every two adjacent frames and averages over all pixels and frames:

$$S_{\text{flicker}} = \frac{1}{N} \sum_{i=1}^N \left(\frac{1}{T-1} \sum_{t=1}^{T-1} MAE(f_i^t, f_i^{t+1}) \right), \quad (\text{S.1})$$

where N is the number of videos, T is the number of frames per video, f_i^t is the frame t in video i , and MAE is the Mean Absolute Error between two consecutive frames over all pixel locations. Finally, the metric is normalized into the range of 0 to 1:

$$S_{\text{flicker-norm}} = \frac{255 - S_{\text{flicker}}}{255}. \quad (\text{S.2})$$

The higher the metric, the less flickering will be observed in a video. BTimer achieves the second best on the Temporal Flickering metric, which suggests that our bullet-time prediction, though not explicitly associated across frames, still achieves a better temporal smoothness than other baselines that decode from some temporal representations.

F Visualization of Learned Deformation

While BTimer is primarily intended for novel view synthesis at the bullet timestamp, in order to demonstrate that our model design can be also targeted for building explicit temporal correlations, we train a variant of BTimer that predicts the canonical positions (XYZ) of Gaussians instead of the pixel-aligned depths on the Objaverse4D [30] dataset. The 4 input images are taken from different camera poses and different timestamps. In Fig. S5, we render the reconstructed dynamic object from a fixed viewpoint and find that the model successfully recovers the 3D motion by predicting the positions of the Gaussians at the correctly warped locations. This is further justified by keeping only the partial reconstruction from the 3DGS associated with one of the input images, and we find out that the model learns to warp the Gaussians according to different timesteps.

Model	apple	block	windmill	space	spin	teddy	wheel	Average
Ground Truth	0.9878	0.9767	0.9940	0.9939	0.9829	0.9759	0.9650	0.9823
TiNeuVox [23]	0.9807	0.9814	0.9879	0.9949	0.9832	0.9782	0.9695	0.9823
T-NeRF [24]	0.9831	0.9791	0.9866	0.9907	0.9828	0.9730	0.9624	0.9797
Nerfies [25]	0.9817	0.9791	0.9868	0.9918	0.9809	0.9720	0.9609	0.9790
HyperNeRF [26]	0.9825	0.9784	0.9865	0.9914	0.9821	0.9720	0.9584	0.9787
PGDVS [27]	0.9719	0.9738	0.9956	0.9903	0.9816	0.9649	0.9517	0.9757
BTimer (Ours)	0.9835	0.9760	0.9884	0.9881	0.9789	0.9746	0.9745	0.9806

Table S6: **Temporal Flickering [28] evaluation on the DyCheck [20] dataset.** There are 7 scenes and we report their average. We highlight the **best**, **second best** and **third best**.

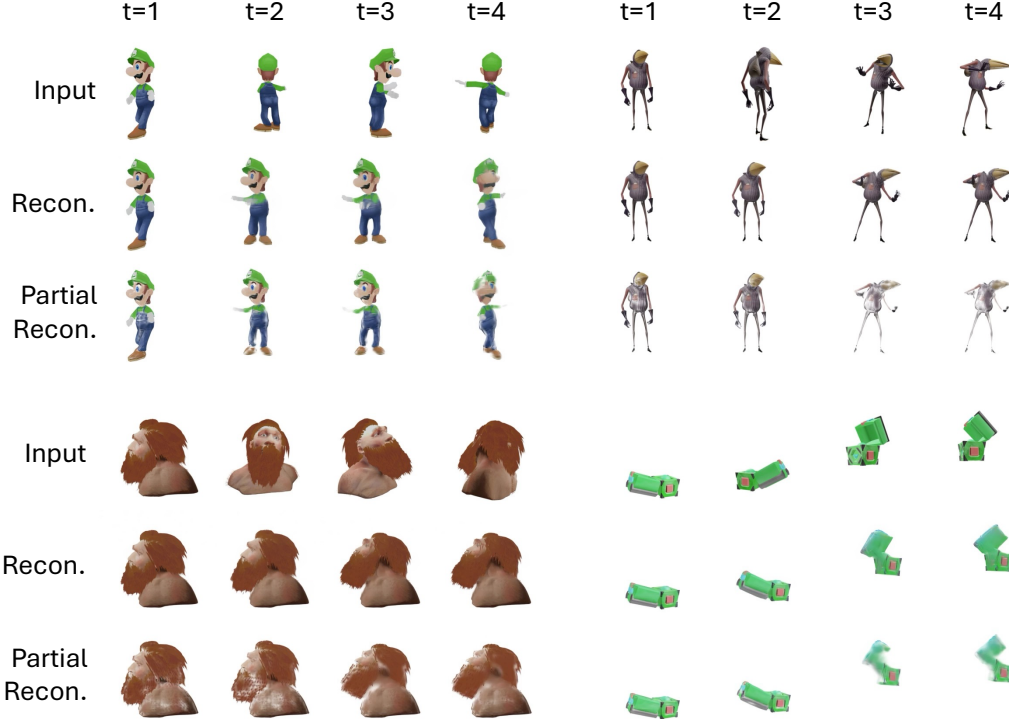


Figure S5: **Learned deformation visualization.** In each example, the first row shows the 4 input images captured from different viewpoints and timestamps, the second row is our reconstruction rendered from a fixed viewpoint, and the third row keeps only Gaussians associated from one of the input images.

91 G Visualization of Learned Scene Flow

92 Although BTimer is not trained with scene flow supervision, we show that our model effectively
 93 model scene flows under the hood in the process of learning dynamic reconstruction. We treat the
 94 Gaussian associated with each pixel in the input images as a point and treat its trajectory over time as
 95 a scene flow. The visualization in Fig. S6 suggests that the learned scene flows closely represent the
 96 actual object motion.

97 References

- 98 [1] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt,
 99 Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe
 100 of annotated 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and*
 101 *Pattern Recognition*, pages 13142–13153, 2023.

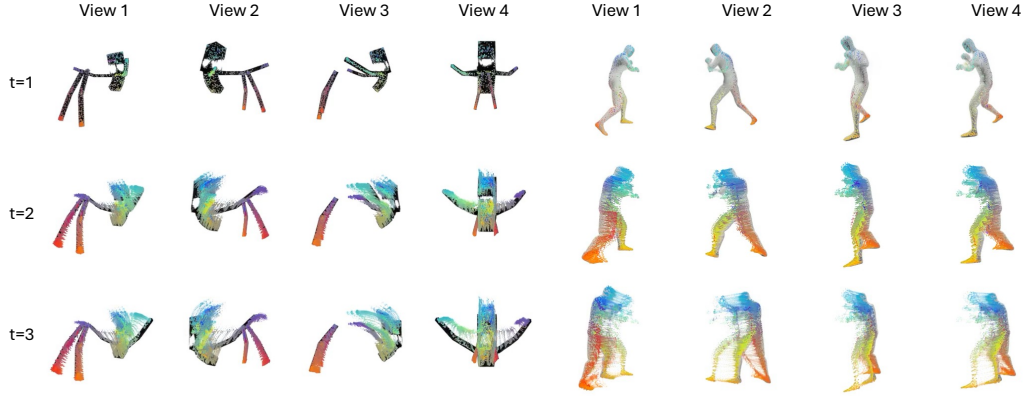


Figure S6: **Learned scene flow visualization.** We color the Gaussians by the pixel positions they associate with. As the Gaussians move, their trajectories are considered as scene flows. Our model learns meaningful scene flows that closely represent the object motion without any scene flow supervision.

- 102 [2] Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu.
103 Lgm: Large multi-view gaussian model for high-resolution 3d content creation. In *European*
104 *Conference on Computer Vision*, pages 1–18. Springer, 2025.
- 105 [3] Xianggang Yu, Mutian Xu, Yidan Zhang, Haolin Liu, Chongjie Ye, Yushuang Wu, Zizheng
106 Yan, Chenming Zhu, Zhangyang Xiong, Tianyou Liang, et al. Mvimgnet: A large-scale dataset
107 of multi-view images. In *Proceedings of the IEEE/CVF conference on computer vision and*
108 *pattern recognition*, pages 9150–9161, 2023.
- 109 [4] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnifi-
110 cation: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*,
111 2018.
- 112 [5] Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo,
113 Zixun Yu, Yawen Lu, et al. D13dv-10k: A large-scale scene dataset for deep learning-based
114 3d vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern*
115 *Recognition*, pages 22160–22169, 2024.
- 116 [6] Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J
117 Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, et al. Kubric: A scalable
118 dataset generator. In *Proceedings of the IEEE/CVF conference on computer vision and pattern*
119 *recognition*, pages 3749–3761, 2022.
- 120 [7] Yang Zheng, Adam W. Harley, Bokui Shen, Gordon Wetzstein, and Leonidas J. Guibas.
121 Pointodyssey: A large-scale synthetic dataset for long-term point tracking. In *ICCV*, 2023.
- 122 [8] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and
123 Christian Rupprecht. Dynamicstereo: Consistent dynamic depth from stereo videos. *CVPR*,
124 2023.
- 125 [9] Lukas Mehl, Jenny Schmalfluss, Azin Jahedi, Yaroslava Nalivayko, and Andrés Bruhn. Spring:
126 A high-resolution high-detail dataset and benchmark for scene flow, optical flow and stereo. In
127 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages
128 4981–4991, 2023.
- 129 [10] Tsai-Shien Chen, Aliaksandr Siarohin, Willi Menapace, Ekaterina Deyneka, Hsiang-wei Chao,
130 Byung Eun Jeon, Yuwei Fang, Hsin-Ying Lee, Jian Ren, Ming-Hsuan Yang, et al. Panda-70m:
131 Captioning 70m videos with multiple cross-modality teachers. In *Proceedings of the IEEE/CVF*
132 *Conference on Computer Vision and Pattern Recognition*, pages 13320–13331, 2024.

- [11] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4578–4587, 2021.
- [12] Mohammed Suhail, Carlos Esteves, Leonid Sigal, and Ameesh Makadia. Generalizable patch-based neural rendering. In *European Conference on Computer Vision*, pages 156–174. Springer, 2022.
- [13] Yilun Du, Cameron Smith, Ayush Tewari, and Vincent Sitzmann. Learning to render novel views from wide-baseline stereo pairs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4970–4980, 2023.
- [14] Haofei Xu, Anpei Chen, Yuedong Chen, Christos Sakaridis, Yulun Zhang, Marc Pollefeys, Andreas Geiger, and Fisher Yu. Murf: Multi-baseline radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20041–20050, 2024.
- [15] David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19457–19467, 2024.
- [16] Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. In *European Conference on Computer Vision*, pages 370–386. Springer, 2025.
- [17] Kai Zhang, Sai Bi, Hao Tan, Yuanbo Xiangli, Nanxuan Zhao, Kalyan Sunkavalli, and Zexiang Xu. Gs-lrm: Large reconstruction model for 3d gaussian splatting. In *European Conference on Computer Vision*, pages 1–19. Springer, 2025.
- [18] Haian Jin, Hanwen Jiang, Hao Tan, Kai Zhang, Sai Bi, Tianyuan Zhang, Fujun Luan, Noah Snavely, and Zexiang Xu. Lvsm: A large view synthesis model with minimal 3d inductive bias. *arXiv preprint arXiv:2410.17242*, 2024.
- [19] Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. Lrm: Large reconstruction model for single image to 3d. *arXiv preprint arXiv:2311.04400*, 2023.
- [20] Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang. Dynamic view synthesis from dynamic monocular video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5712–5721, 2021.
- [21] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6498–6508, 2021.
- [22] Zhiwen Fan, Wenyan Cong, Kairun Wen, Kevin Wang, Jian Zhang, Xinghao Ding, Danfei Xu, Boris Ivanovic, Marco Pavone, Georgios Pavlakos, et al. Instantsplat: Unbounded sparse-view pose-free gaussian splatting in 40 seconds. *arXiv preprint arXiv:2403.20309*, 2, 2024.
- [23] Jiemin Fang, Taoran Yi, Xinggang Wang, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Matthias Nießner, and Qi Tian. Fast dynamic radiance fields with time-aware neural voxels. In *SIG-GRAPH Asia 2022 Conference Papers*, pages 1–9, 2022.
- [24] Hang Gao, Ruilong Li, Shubham Tulsiani, Bryan Russell, and Angjoo Kanazawa. Monocular dynamic view synthesis: A reality check. *Advances in Neural Information Processing Systems*, 35:33768–33780, 2022.
- [25] 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.
- [26] 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. *arXiv preprint arXiv:2106.13228*, 2021.

- 183 [27] Xiaoming Zhao, R Alex Colburn, Fangchang Ma, Miguel Ángel Bautista, Joshua M Susskind,
184 and Alex Schwing. Pseudo-generalized dynamic view synthesis from a video. In *The Twelfth*
185 *International Conference on Learning Representations*, 2024.
- 186 [28] Yi-Hua Huang, Yang-Tian Sun, Ziyi Yang, Xiaoyang Lyu, Yan-Pei Cao, and Xiaojuan Qi.
187 Sc-gs: Sparse-controlled gaussian splatting for editable dynamic scenes. In *Proceedings of the*
188 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4220–4230, 2024.
- 189 [29] Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang,
190 Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark
191 suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer*
192 *Vision and Pattern Recognition*, pages 21807–21818, 2024.
- 193 [30] Jiawei Ren, Kevin Xie, Ashkan Mirzaei, Hanxue Liang, Xiaohui Zeng, Karsten Kreis, Ziwei Liu,
194 Antonio Torralba, Sanja Fidler, Seung Wook Kim, et al. L4gm: Large 4d gaussian reconstruction
195 model. *arXiv preprint arXiv:2406.10324*, 2024.