SplatFields: Neural Gaussian Splats for Sparse 3D and 4D Reconstruction –Supplementary Material–

Overview. We provide additional details related to training (Sec. A), implementation (Sec. B), and reported results (Sec. C). We refer the reader to the supplementary video for more qualitative results.

A Training

Optimization. Given a set of calibrated multi-view input images and an initial collection of K Gaussian splats $\mathbf{G} = \{\mathcal{G}_k\}_{k=1}^{K}$, differential rasterizer $\mathcal{R}$ based on Gaussian Splatting (Sec. 3) propagates image changes to the scene parameters $\mathbf{G}$. This feedback loop is used to optimize the scene parameters by imposing the photometric loss between the rendered I and the input image I^*:

$$\arg\min_{\mathbf{G}} \mathcal{L}(\mathcal{R}(\mathbf{G}), I^*), \tag{A.1}$$

where the initial collection of splats is initialized either randomly [34], by SfM [67], or by the visual hull [19].

Loss definition. We follow the training scheme used by 3DGS (Sec. 3) and optimize the rendering objective (Eq. A.1) via the Adam optimizer [31]. We also employ the mask loss between rendered and ground-truth masks for the object-level scenes:

$$\mathcal{L} = (1 - \lambda_1)\mathcal{L}_1 + \lambda_1\mathcal{L}_{\text{D-SSIM}} + \lambda_2\mathcal{L}_{\text{MASK}} + \lambda_3\mathcal{L}_{\text{norm}}, \tag{A.2}$$

where $\mathcal{L}_{\text{MASK}}$ is the $\mathcal{L}_1$ loss between the rendered opacity and the ground truth mask akin to [59] and $\mathcal{L}_{\text{norm}}$ is the splat norm regularization term from Sec. 4.

Hyperparameter λ_1 is empirically set to 0.2, λ_2 is set to 0.1 for object-level scenes [47, 80] and to 0 for unbounded scenes, while λ_3 is set to 0.01 for static reconstruction and to 0 for all of the other experiments. We further employ the exponential learning rate decay that starts from 8×10^{-4} until it reaches 1.6×10^{-6} at 40k iterations.

B Implementation Details

B.1 Spatial Autocorrelation

To quantify the spatial similarity of nearby features, we measure local spatial autocorrelation via Moran's I [48] between the features of splats in their local neighborhoods. Specifically, for each splat $\mathcal{G}_k$ and its attribute (color, opacity, and covariance) we query N nearest neighbors $[\mathcal{X}_i]_{i=1}^N$ with associated locations $loc(\mathcal{X}_i) \in \mathbb{R}^3$ and measure Moran's I (Eq. B.3) of its attributes $attr(\mathcal{X}_i) \in \mathbb{R}$:

$$I = \mathop{\mathbb{E}}_{\mathcal{X} \in \mathbf{G}} [I(\mathcal{X})], \tag{B.3}$$

where

$$I(\mathcal{X}) = \frac{N}{\sum_{i=1}^N \sum_{j=1}^N W_{ij}(\mathcal{X})} \frac{\sum_{i=1}^N \sum_{j=1}^N W_{ij}(\mathcal{X}) attr(\mathcal{X}_i) attr(\mathcal{X}_j)}{\sum_{i=1}^N attr(\mathcal{X}_i)^2}, \tag{B.4}$$

$$W_{ij}(\mathcal{X}) = \begin{cases} \|loc(X_i) - loc(X_j)\|_2^{-1} & \text{if } i \neq j \ , \\ 0 & \text{otherwise.} \end{cases} \tag{B.5}$$

For attributes with more than one feature dimension (*e.g.* color and covariance matrix), we average Moran's I across all feature dimensions. In all of the experiments, we set $N = 5$.

Moran's Loss $\mathcal{L}_{Moran}$ (in Tab. 2, C.1, C.2) enhances 3DGS as a straightforward baseline that incorporates the *spatial bias* by enforcing a higher Moran's I score and is implemented as the negative autocorrelation score:

$$\mathcal{L}_{Moran} = \lambda_{Moran} \left(1 - \mathop{\mathbb{E}}_{\mathcal{X} \in \mathbf{G}} [I(\mathcal{X})] \right), \tag{B.6}$$

where λ_{Moran} is empirically set to 0.01.

B.2 SplatFields

In the following, we describe the network architectures.

CNN Generator g_θ consists of three CNN decoders to produce three axis-aligned feature planes $\mathbf{F}$. Each decoder, takes as input a 20×20-resolution noise $\epsilon \in \mathbb{R}^{20\times20\times8}$ with 8 channels to produce the 160×160-resolution feature plane with 16 channels ($\mathbb{R}^{160\times160\times16}$) through up-sampling blocks with residual connections. First, the noise is expanded to 32 channels via an up-sampling CNN layer, which is then processed by a single attention layer and propagated through a ResNet block with two CNN layers to output an intermediate feature ($20 \times 20 \times 32$). This feature is then propagated through four up-sampling blocks until the feature resolution of $160 \times 160 \times 32$ which is then down-scaled to 16 channels via a single CNN layer to form the final tri-plane representation $\mathbf{F} \in \mathbb{R}^{3\times160\times160\times16}$. Each up-sampling block consists of two CNN ResNet blocks (each with two CNN layers) and one up-sampling CNN layer. Then the splat center $\mathbf{p}_k$ is projected onto each axis-aligned feature plane to obtain feature vectors via bi-linear interpolation. These features are then concatenated along the feature dimension and propagated through a tiny 2-layer MLP with 48 neurons to produce the point feature $\mathbf{f}_k \in \mathbb{R}^{48}$.

Deform MLP f_Θ takes as input the splat center $\mathbf{p}_k$ and the feature $\mathbf{f}_k$. The splat location is first positionally encoded [47] with 4 levels and propagated through an 8-layer MLP with 128 neurons that deforms the splat center $\hat{\mathbf{p}}_k$ by predicting its residual akin to [56, 57].

Color Field $f_{\Theta_{\mathbf{c}}}$ takes as input the deformed query point (positionally encoded with 4 levels) along with $\mathbf{f}_k$ and propagates them through a 6-layer MLP with 128 neurons, where the last layer takes as input the viewing direction akin to NeRF [47].

Scale $f_{\Theta_{\mathbf{s}}}$ *and Opacity* f_{Θ_α} *Fields* take the same input as the color MLP and are implemented as 5-layer 64-neuron MLPs. The output of the opacity MLP is activated by the sigmoid function.

Rotation Field $f_{\Theta_{\mathbf{O}}}$ is implemented as a 4-layer MLP that takes the same input as the color MLP and predicts a four-dimensional vector that is normalized to produce the quaternion representation.

Flow Field $f_{\Theta_{\mathbf{p}}}$ is utilized only for the 4D reconstruction. It takes as input the deformed splat center $\hat{\mathbf{p}}_k$ (positionally encoded with 4 levels) and the feature vector $\mathbf{f}_k$ and propagates them through an architecture similar to Deform MLP to model the forward flow. In the paper, we consider different types of modeling the flow: DCT [39], SE(3) [52], scaled SE(3) [93], and offsets [56, 57]. See Sec. 5.2 for further details.

All of the MLP fields take time (positionally encoded with 4 levels) as an additional input and are implemented as ResField MLPs [45]. We empirically set the ResFields' rank to 40 for the multi-view dynamic reconstruction on Owlii [80] and to 0 for the monocular reconstruction [82] as the scenes are semi-static.

C Experiment Details

Static reconstruction on Blender (Sec. 5.1). We compare SplatFields with SparseNeRF [75] and with recent 3DGS methods: SuGaR [22], Mip3DGS [91], 3DGS [29], 2DGS [25], and Light3DGS [16] on Blender [47]. Mip3DGS [91], 3DGS [29], and 2DGS [25] are run for 40k iterations like SplatFields, while Light3DGS [16], SuGaR [22], and SparseNeRF [75] are run with their default configurations as they have a particular training scheme. All of the methods are initialized from the randomly sampled points inside the visual hull of the objects and are further supervised with the mask loss implemented as the $\mathcal{L}_1$ distance between the ground truth and the rendered opacity akin to [59].

We further provide extended comparisons of Tab. 2 in Tab. C.1-C.2 for varying number of views ranging from 4 to 12. Consistently with the main paper, SplatFields demonstrates superior metric reconstruction quality over the baseline methods across varying number of input views.

Static reconstruction on DTU (Sec. 5.1). We compare with NeRF (VolRecon [63], ZeroRF [68]) and splatting (3DGS [29], 2DGS [25]) methods on DTU [28] on the task of 3-view reconstruction (Tab. C.3). All of the baselines are run with the default configurations, with the difference that the splatting-based baselines adopt the mask loss for fair comparisons.

Monocular dynamic reconstruction (Sec. 5.2). Our method adopts annealing smooth training [85] and is trained for 30k iterations after being initial-

Table C.1: Sparse static scene reconstruction of Blender [47] scenes. Reported numbers indicate PSNR metric on the novel views ("-" denotes failed runs). Colors denote the 1st, 2nd, and 3rd best-performing model. See Sec. 5.1 for discussion

	mean	Toy	Ficus	Hotdog	Chair	Mic	Ship	Drums	Materials
12 Input Views									
SparseNeRF [75]	-	23.02	18.19	-	26.20	23.26	20.81	19.21	20.80
SparseNeRF *wo.* depth	22.92	24.00	18.84	27.52	27.11	23.35	21.84	19.17	21.50
SuGaR [22]	21.78	23.77	23.08	22.36	25.72	18.72	21.09	19.55	19.94
ScaffoldGS [41]	23.82	23.65	22.78	26.34	25.80	28.28	21.17	20.47	22.06
Mip3DGS [91]	24.86	24.65	25.62	26.53	26.25	28.40	22.52	21.98	22.94
3DGS [29]	25.29	25.14	25.92	27.51	27.10	29.02	22.79	22.10	22.71
Light3DGS [16]	25.39	25.08	27.53	27.10	27.40	28.04	23.02	22.07	22.90
2DGS [25]	25.62	25.50	25.62	29.24	28.52	28.07	23.08	22.19	22.75
3DGS *w.* $\mathcal{L}_{Moran}$	25.44	25.26	26.55	28.96	27.91	27.87	22.33	21.98	22.65
SplatFields3D	25.80	26.98	26.27	29.45	27.42	27.60	23.78	22.55	22.32
10 Input Views									
SparseNeRF [75]	-	22.64	18.27	-	25.30	23.27	20.29	18.61	19.72
SparseNeRF *wo.* depth	22.58	23.89	18.75	27.56	26.42	23.23	21.68	18.20	20.87
SuGaR [22]	21.10	22.78	22.42	23.60	24.25	17.93	20.35	19.11	18.40
ScaffoldGS [41]	22.63	21.98	22.68	24.37	24.15	27.76	20.39	19.64	20.08
Mip3DGS [91]	23.65	23.49	24.97	25.27	24.49	27.69	21.38	21.23	20.66
3DGS [29]	24.11	23.79	25.54	26.16	25.28	28.39	21.87	21.34	20.51
Light3DGS [16]	24.21	23.94	26.95	25.62	25.91	27.45	21.82	21.38	20.60
2DGS [25]	24.42	24.06	25.17	27.92	26.96	27.53	21.83	21.58	20.27
3DGS *w.*$\mathcal{L}_{Moran}$	24.21	23.91	26.09	27.65	25.86	27.07	21.38	21.26	20.46
SplatFields3D	24.94	26.51	25.59	28.29	25.92	27.36	23.12	21.86	20.88
8 Input Views									
SparseNeRF [75]	-	22.33	17.97	-	23.81	23.01	19.85	17.85	20.02
SparseNeRF *wo.* depth	22.20	24.06	18.42	27.09	25.12	23.04	21.23	17.94	20.74
SuGaR [22]	20.62	21.91	22.33	23.01	23.30	18.60	19.59	18.66	17.55
ScaffoldGS [41]	21.53	20.95	21.35	23.77	22.77	26.40	18.88	18.96	19.17
Mip3DGS [91]	22.37	22.05	23.23	24.24	23.57	26.32	19.91	20.10	19.55
3DGS [29]	22.93	22.55	23.69	25.57	24.43	27.37	19.98	20.33	19.49
Light3DGS [16]	22.98	22.67	24.98	24.79	24.40	26.59	20.60	20.41	19.41
2DGS [25]	23.04	22.19	23.63	26.76	25.46	26.24	20.16	20.60	19.25
3DGS *w.* $\mathcal{L}_{Moran}$	23.19	22.79	24.56	26.57	25.14	26.97	19.79	20.41	19.32
SplatFields3D	23.98	24.71	23.97	27.87	25.64	26.49	22.15	21.12	19.85
6 Input Views									
SparseNeRF [75]	-	20.86	18.03	-	22.75	22.40	19.33	16.24	19.54
SparseNeRF *wo.* depth	20.86	22.62	17.63	25.84	22.65	20.72	19.85	17.25	20.30
SuGaR [22]	19.07	19.89	20.61	20.80	21.92	18.26	17.72	16.86	16.53
ScaffoldGS [41]	19.65	18.21	20.72	19.48	22.20	24.31	16.47	17.21	18.62
Mip3DGS [91]	20.04	19.39	21.81	19.70	21.72	24.44	17.02	17.72	18.52
3DGS [29]	20.62	19.80	22.25	21.16	22.75	25.21	17.58	17.77	18.48
Light3DGS [16]	20.76	20.25	23.12	20.66	22.69	24.89	17.83	18.02	18.63
2DGS [25]	20.74	19.38	21.93	23.85	23.26	24.48	16.92	17.91	18.17
3DGS *w.* $\mathcal{L}_{Moran}$	21.03	20.34	23.05	23.92	22.50	24.64	17.20	18.14	18.48
SplatFields3D	22.26	22.41	22.26	26.19	25.03	24.84	19.33	18.97	19.05
4 Input Views									
SparseNeRF [75]	-	20.94	17.48	23.81	21.41	21.52	-	15.37	17.03
SparseNeRF *wo.* depth	17.87	19.31	17.05	23.54	20.26	11.56	17.86	13.64	19.77
SuGaR [22]	16.94	16.96	19.30	19.36	19.07	17.47	15.22	14.73	13.38
ScaffoldGS [41]	16.86	15.40	19.58	17.31	18.40	20.54	14.70	15.27	13.69
Mip3DGS [91]	16.94	16.23	19.60	16.98	18.38	20.56	14.64	14.92	14.21
3DGS [29]	17.37	16.44	19.72	18.65	18.72	20.75	15.43	15.08	14.15
Light3DGS [16]	17.70	16.94	20.35	18.56	18.96	21.53	15.67	15.44	14.19
2DGS [25]	17.58	16.32	19.69	20.67	19.39	21.17	14.45	14.84	14.14
3DGS *w.* $\mathcal{L}_{Moran}$	18.13	17.06	20.53	22.10	18.25	22.06	15.18	15.32	14.53
SplatFields3D	19.16	18.89	20.19	24.31	19.31	21.73	16.83	16.35	15.69

Table C.2: Sparse static scene reconstruction. Synthetic Blender [47] dataset, reported numbers indicate SSIM metric on the novel views. See Sec. 5.1 for discussion

	mean	Toy	Ficus	Hotdog	Chair	Mic	Ship	Drums	Materials
12 Input Views									
SparseNeRF [75]	-	86.07	84.57	-	90.45	92.23	76.28	83.65	85.40
SparseNeRF *wo.* depth	87.54	88.64	85.10	93.96	91.77	92.55	77.69	83.69	86.94
SuGaR [22]	85.60	86.24	89.09	90.29	90.94	85.86	76.81	82.77	82.76
ScaffoldGS [41]	87.47	86.22	90.65	92.06	90.75	96.21	73.02	85.75	85.10
Mip3DGS [91]	89.78	87.81	93.87	93.15	92.46	96.90	76.42	89.67	87.96
3DGS [29]	90.01	88.57	94.11	93.45	93.48	96.98	76.11	89.81	87.55
Light3DGS [16]	90.30	88.51	95.31	93.45	93.52	96.64	76.57	89.98	88.42
2DGS [25]	91.09	90.18	94.20	94.85	94.93	96.77	79.09	90.41	88.27
3DGS *w.* $\mathcal{L}_{Moran}$	90.45	89.31	94.62	94.40	94.07	96.52	76.71	89.97	88.01
SplatFields3D	91.18	91.06	94.36	95.55	92.42	96.17	81.03	90.90	87.98
10 Input Views									
SparseNeRF [75]	-	85.65	84.30	-	89.57	92.27	75.51	82.52	83.82
SparseNeRF *wo.* depth	86.95	88.45	84.65	94.06	91.26	92.39	77.25	82.12	85.42
SuGaR [22]	83.83	84.27	88.55	90.52	88.77	84.39	74.27	80.78	79.10
ScaffoldGS [41]	85.45	83.27	90.14	89.66	88.56	95.77	70.66	83.46	82.10
Mip3DGS [91]	88.00	85.68	93.15	91.76	90.34	96.44	73.98	88.22	84.41
3DGS [29]	88.27	86.30	93.65	92.19	91.23	96.49	73.89	88.48	83.93
Light3DGS [16]	88.76	86.66	94.82	92.41	91.70	96.27	74.37	88.75	85.12
2DGS [25]	89.51	88.42	93.61	93.79	93.26	96.37	76.52	89.38	84.72
3DGS *w.* $\mathcal{L}_{Moran}$	88.94	87.53	94.24	93.32	91.88	95.98	75.10	88.74	84.69
SplatFields3D	90.32	91.34	93.70	95.09	91.14	95.95	80.25	89.85	85.26
8 Input Views									
SparseNeRF [75]	-	84.85	83.98	-	88.39	92.06	74.32	81.44	83.66
SparseNeRF *wo.* depth	86.30	88.27	83.97	93.65	90.23	92.15	76.01	81.38	84.78
SuGaR [22]	82.74	82.04	87.95	89.41	87.60	85.08	72.80	79.84	77.23
ScaffoldGS [41]	83.62	80.54	88.25	89.04	86.23	95.01	68.49	81.49	79.91
Mip3DGS [91]	86.24	82.93	91.03	90.84	89.21	95.77	71.86	86.15	82.14
3DGS [29]	86.63	84.05	91.56	91.63	90.49	95.99	70.96	86.62	81.76
Light3DGS [16]	87.11	84.44	93.01	91.47	90.17	95.76	72.16	86.97	82.92
2DGS [25]	87.72	84.80	91.74	93.13	91.79	95.72	74.06	87.89	82.65
3DGS *w.* $\mathcal{L}_{Moran}$	87.35	85.15	92.57	92.50	91.04	95.88	72.03	87.26	82.36
SplatFields3D	88.94	88.04	91.69	94.68	90.91	95.48	78.76	88.50	83.46
6 Input Views									
SparseNeRF [75]	-	83.25	83.70	-	87.76	91.53	72.96	78.58	83.47
SparseNeRF *wo.* depth	84.42	85.68	82.33	92.51	87.47	90.72	72.50	79.58	84.59
SuGaR [22]	79.85	77.67	85.51	86.44	84.98	84.24	68.93	76.35	74.68
ScaffoldGS [41]	80.34	74.13	87.01	82.79	84.99	93.10	62.49	78.61	79.59
Mip3DGS [91]	83.09	77.68	89.34	86.81	86.46	94.58	66.45	82.08	81.34
3DGS [29]	83.56	78.58	89.79	87.81	87.35	94.81	66.71	82.45	80.94
Light3DGS [16]	84.34	79.64	91.08	88.67	87.49	94.76	67.63	83.09	82.33
2DGS [25]	84.43	78.54	89.71	90.36	88.16	94.59	68.63	83.87	81.61
3DGS *w.* $\mathcal{L}_{Moran}$	84.54	80.20	90.84	90.15	87.76	94.50	67.49	83.47	81.89
SplatFields3D	86.62	84.05	89.56	93.62	89.53	94.50	74.14	85.03	82.55
4 Input Views									
SparseNeRF [75]	-	83.38	82.98	90.95	85.14	90.49	-	76.46	79.48
SparseNeRF *wo.* depth	78.66	78.80	80.86	90.57	82.26	72.23	69.29	71.97	83.28
SuGaR [22]	75.61	72.07	83.08	83.68	80.66	82.48	63.46	70.73	68.69
ScaffoldGS [41]	74.99	68.04	84.68	76.92	77.58	90.24	59.60	71.20	71.68
Mip3DGS [91]	77.67	71.47	85.73	82.68	81.00	91.42	61.17	74.27	73.62
3DGS [29]	78.12	72.17	85.97	83.78	81.49	91.55	62.05	74.80	73.18
Light3DGS [16]	79.38	73.58	86.93	85.98	81.91	92.25	63.27	75.88	75.22
2DGS [25]	79.26	72.84	86.04	86.62	82.54	92.04	63.57	76.25	74.14
3DGS *w.* $\mathcal{L}_{Moran}$	79.65	73.79	87.05	87.79	82.01	92.39	63.31	75.95	74.88
SplatFields3D	82.26	78.23	86.17	92.10	83.85	91.92	70.40	78.67	76.77

Table C.3: Static three-view reconstruction on the DTU dataset [28]. SplatFields demonstrates more accurate reconstructions compared to the NeRF- (VolRecon [63], ZeroRF [68]) and splatting-based (3DGS [29], 2DGS [25]) baselines; the displayed metric is PSNR↑

	mean	Scene ID Number (PSNR↑)														
	PSNR↑	105	106	110	114	118	122	24	37	40	55	63	65	69	83	97
VolRecon	11.42	9.03	15.19	16.45	11.66	17.96	18.29	7.86	6.30	7.86	12.90	6.54	10.06	14.84	7.77	8.64
ZeroRF	19.10	21.36	14.30	20.96	18.86	19.24	23.45	15.78	15.23	16.06	21.02	23.62	18.31	15.05	24.17	19.13
3DGS	19.40	20.07	17.06	17.04	20.56	18.25	20.23	18.76	19.82	18.29	21.03	22.63	20.02	15.86	21.64	19.76
2DGS	20.70	21.25	19.23	19.17	19.90	19.75	22.04	19.71	20.22	19.56	21.95	23.16	22.37	17.64	23.13	21.47
SplatFields	21.07	21.93	19.11	19.77	22.03	21.35	24.49	18.43	19.82	19.67	21.45	23.79	22.64	17.54	23.52	20.52

ized from static 3DGS ran for 3k iterations akin to [85]. We run recent dynamic 3DGS methods [78, 85] with their default configurations, while results for the NeRF-based methods are adopted from the previous work [82, 85].

We further provide additional metrics SSIM and PSNR in Tab. C.4. Note that SSIM and PSNR are less reliable metrics due to noisy camera calibrations.

Multi-view dynamic reconstruction (Sec. 5.2). Dynamic NeRF-based methods (DyNeRF [37], TNeRF [37], DNeRF [57], HyperNeRF [53]) are trained with SDF parametrization as they are better suited for sparse view reconstruction. We use the implementations with ResField MLPs [45] (256 neurons) and train them for 400k iterations, following the training scheme from [45].

For dynamic Gaussian splatting methods (4D-GS [84], Deformable3DGS [85], 4DGaussians [78]), we use their default implementations and adopt additional mask loss with the weight of 0.1. All of these methods, including ours, are trained with a batch size of 5. 4D-GS is trained with default 30k iterations. Deformable3DGS is trained until the full convergence of 200k iterations. 4DGaussians is trained for 30k iterations, we noticed that longer training leads to overfitting and the loss becomes an invalid number.

Additional SSIM metric is reported in Tab. C.5. Akin to the main paper, SplatFields demonstrates consistently better reconstruction quality across all scenes and varying number of input views.

Compute, memory overhead, and inference time. Compared to the original 3DGS, our method requires longer training to converge (∼10 min. for 3DGS vs. ∼70 min. for ours on the Toy scene) and consumes a greater amount of GPU memory (∼5GB for 3DGS vs. about ∼8GB for ours). However, after training, the neural components can be discarded, *leaving the inference speed and memory usage equivalent to that of 3DGS*. In dynamic setups, the training times of our method are comparable to other dynamic 3DGS methods that also employ neural networks. Our neural network architecture comprises ∼1M parameters for the static case. All the run-times reported in the paper are calculated on an NVIDIA RTX 3090.

CNNs *vs.* MLPs on extremely sparse view setups. CNN module enhances the capacity of the SplatFields, which may lead to slight overfitting in extremely sparse scenarios, such as a 4-view setup. However, as additional views are incorporated and the model receives more diverse inputs, the ability of CNNs to

Table C.4: Monocular reconstruction of dynamic sequences from the NeRF-DS dataset [82] with recent state-of-the-art methods. The forward slash in FPS indicates the rendering speed with the inference of neural network *vs.* without when the rendering primitives are extracted and stored for each frame

	Resources		LPIPS↓ ($\times 10^2$)							
	FPS ↑	t ↓	*mean*	Sieve	Plate	Bell	Press	Cup	As	Basin
3D-GS [29]	120+	15 min	29.20	22.47	40.93	25.03	29.04	25.48	29.94	31.53
TiNeuVox [17]	< 1	30 min	27.66	31.76	33.17	25.68	30.01	36.43	39.67	26.90
4DGaussians [78]	120+/50	30 min	21.06	16.39	23.80	21.84	21.68	19.06	22.06	22.57
HyperNeRF [53]	< 1	1 day	19.90	16.45	29.40	20.52	19.59	16.50	17.77	19.11
Deformable3DGS [85]	120+/30	1 h	19.79	15.30	25.04	15.93	29.89	15.38	17.88	19.10
NeRF-DS [82]	< 1	1 day	18.16	14.72	19.96	18.67	20.47	17.37	17.41	18.55
SplatFields4D	120+/30	1 h	17.86	14.72	22.43	16.10	19.26	15.67	17.71	19.11
			PSNR↑							
	FPS ↑	t ↓	*mean*	Sieve	Plate	Bell	Press	Cup	As	Basin
3D-GS [29]	120+	15 min	20.29	23.16	16.14	21.01	22.89	21.71	22.69	18.42
TiNeuVox [17]	< 1	30 min	21.61	21.49	20.58	23.08	24.47	19.71	21.26	20.66
4DGaussians	120+/30	30 min	23.68	26.77	20.51	24.25	25.55	23.69	25.50	19.47
HyperNeRF [53]	< 1	1 day	23.45	25.43	18.93	23.06	26.15	24.59	25.58	20.41
Deformable3DGS [85]	120+/30	1 h	23.54	25.16	19.97	25.02	24.18	24.64	26.26	19.57
NeRF-DS [82]	< 1	1 day	23.60	25.78	20.54	23.19	25.72	24.91	25.13	19.96
SplatFields4D	120+/30	1 h	23.84	25.35	20.36	25.51	25.43	24.29	26.21	19.71
			SSIM↑							
	FPS ↑	t ↓	*mean*	Sieve	Plate	Bell	Press	Cup	As	Basin
3D-GS [29]	120+	15 min	78.16	82.03	69.70	78.85	81.63	83.04	80.17	71.70
TiNeuVox [17]	< 1	30 min	82.34	82.65	80.27	82.42	86.13	81.09	82.89	81.45
4DGaussians	120+/30	30 min	83.22	87.18	79.70	81.14	85.73	86.46	85.73	76.62
HyperNeRF [53]	< 1	1 day	84.88	87.98	77.09	80.97	88.97	87.70	89.49	81.99
Deformable3DGS [85]	120+/30	1 h	84.05	87.58	79.14	84.52	81.22	88.71	88.49	78.69
NeRF-DS [82]	< 1	1 day	84.94	89.00	80.42	82.12	86.18	87.41	87.78	81.66
SplatFields4D	120+/30	1 h	85.17	87.78	80.26	84.74	86.64	88.73	88.59	79.44

capture structural patterns become increasingly beneficial; this is demonstrated by the improved performance in denser view setups. Please also note that the CNN-based SplatFields model is still better than the vanilla 3DGS method in the 4-view setup (Tab. C.1-C.2).

Spatial autocorrelation: sparse *vs.* dense view setup. A simple experiment under different view setups on Toy [47] demonstrates (Tab. C.6) a tendency that overfitting (high ΔPSNR) corresponds to lower autocorrelation, especially for RGB.

Table C.5: Multi-view reconstruction of dynamic sequences from the Owlii dataset [47] under varying number of input views. The reported metric is SSIM↑ averaged across novel views. See Sec. 5.2 for discussion

	10 Input Views				
	mean	Dancer	Exercise	Model	Basketball
4D-GS [84]	95.34	95.31	95.96	94.92	95.16
Deformable3DGS [85]	93.80	94.10	95.09	91.58	94.43
4DGaussians [78]	95.91	95.19	96.47	95.71	96.28
SplatFields4D (30k it)	96.52	96.41	96.72	95.99	96.98
SplatFields4D (40k it)	96.57	96.47	96.76	96.04	97.02
SplatFields4D (100k it)	96.67	96.59	96.83	96.16	97.11
SplatFields4D (200k it)	96.81	96.76	96.92	96.32	97.23
	8 Input Views				
	mean	Dancer	Exercise	Model	Basketball
4D-GS [84]	93.71	94.19	93.94	93.29	93.40
Deformable3DGS [85]	92.37	93.24	93.29	90.33	92.62
4DGaussians [78]	95.00	94.39	95.45	94.92	95.26
SplatFields4D (30k it)	95.99	95.97	96.05	95.44	96.52
SplatFields4D (40k it)	96.04	96.02	96.08	95.49	96.56
SplatFields4D (100k it)	96.15	96.15	96.16	95.62	96.65
SplatFields4D (200k it)	96.28	96.31	96.26	95.78	96.77
	6 Input Views				
	mean	Dancer	Exercise	Model	Basketball
4D-GS [84]	87.23	89.52	87.05	85.16	87.20
Deformable3DGS [85]	90.95	91.73	91.48	89.11	91.48
4DGaussians [78]	93.87	93.58	94.45	93.05	94.40
SplatFields4D (30k it)	95.40	95.62	95.51	94.53	95.95
SplatFields4D (40k it)	95.45	95.67	95.54	94.59	95.99
SplatFields4D (100k it)	95.56	95.81	95.62	94.70	96.09
SplatFields4D (200k it)	95.69	95.99	95.71	94.85	96.21
	4 Input Views				
	mean	Dancer	Exercise	Model	Basketball
4D-GS [84]	78.94	80.81	78.72	78.09	78.15
Deformable3DGS [85]	87.10	89.04	87.67	85.05	86.63
4DGaussians [78]	89.50	90.34	90.67	87.47	89.51
SplatFields4D (30k it)	91.46	92.61	91.99	88.98	92.28
SplatFields4D (40k it)	91.49	92.65	92.01	88.99	92.31
SplatFields4D (100k it)	91.54	92.76	92.04	89.01	92.36
SplatFields4D (200k it)	91.60	92.89	92.07	89.02	92.41

Table C.6: Moran's I on Toy [47] for a varying number of views

#Views	5	25	50	75	100
Train PSNR	51.85	45.10	41.11	40.23	40.03
Test PSNR ↑	18.07	30.12	33.75	35.02	**35.34**
ΔPSNR ↓	33.78	14.98	7.36	5.21	**4.69**
Moran RGB ↑	0.467	0.588	0.634	0.655	**0.661**
Moran Opacity ↑	0.710	0.746	0.744	**0.742**	0.736
Moran Covariance ↑	0.426	0.414	0.452	0.465	**0.476**

References

1. Alexa, M., Gross, M., Pauly, M., Pfister, H., Stamminger, M., Zwicker, M.: Point-based computer graphics. In: SIGGRAPH notes (2004) 4
2. Aliev, K.A., Sevastopolsky, A., Kolos, M., Ulyanov, D., Lempitsky, V.: Neural point-based graphics. In: ECCV (2020) 4
3. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In: CVPR (2022) 4
4. Botsch, M., Hornung, A., Zwicker, M., Kobbelt, L.: High-quality surface splatting on today's gpus. In: Proceedings Eurographics/IEEE VGTC Symposium Point-Based Graphics, 2005. pp. 17–141. IEEE (2005) 2, 4
5. Cai, H., Feng, W., Feng, X., Wang, Y., Zhang, J.: Neural surface reconstruction of dynamic scenes with monocular rgb-d camera. In: NeurIPS (2022) 4
6. Cao, A., Johnson, J.: Hexplane: A fast representation for dynamic scenes. CVPR (2023) 14
7. Chan, E.R., Lin, C.Z., Chan, M.A., Nagano, K., Pan, B., De Mello, S., Gallo, O., Guibas, L.J., Tremblay, J., Khamis, S., et al.: Efficient geometry-aware 3d generative adversarial networks. In: CVPR (2022) 3, 4, 5, 8
8. Chen, A., Xu, Z., Geiger, A., Yu, J., Su, H.: Tensorf: Tensorial radiance fields. In: ECCV (2022) 2, 4
9. Chen, G., Wang, W.: A survey on 3d gaussian splatting. arXiv preprint arXiv:2401.03890 (2024) 2, 4
10. Chen, H., Gu, J., Chen, A., Tian, W., Tu, Z., Liu, L., Su, H.: Single-stage diffusion nerf: A unified approach to 3d generation and reconstruction. In: ICCV (2023) 4, 14
11. Das, D., Wewer, C., Yunus, R., Ilg, E., Lenssen, J.E.: Neural parametric gaussians for monocular non-rigid object reconstruction. In: CVPR (2024) 5
12. Deng, C., Jiang, C., Qi, C.R., Yan, X., Zhou, Y., Guibas, L., Anguelov, D., et al.: Nerdi: Single-view nerf synthesis with language-guided diffusion as general image priors. In: CVPR (2023) 4
13. Deng, K., Liu, A., Zhu, J.Y., Ramanan, D.: Depth-supervised nerf: Fewer views and faster training for free. In: CVPR (2022) 4
14. Drebin, R.A., Carpenter, L., Hanrahan, P.: Volume rendering. SIGGRAPH (1988) 4
15. Duckworth, D., Hedman, P., Reiser, C., Zhizhin, P., Thibert, J.F., Lučić, M., Szeliski, R., Barron, J.T.: Smerf: Streamable memory efficient radiance fields for real-time large-scene exploration. arXiv preprint arXiv:2312.07541 (2023) 4
16. Fan, Z., Wang, K., Wen, K., Zhu, Z., Xu, D., Wang, Z.: Lightgaussian: Unbounded 3d gaussian compression with 15x reduction and 200+ fps. arXiv preprint arXiv:2311.17245 (2023) 3, 4, 9, 10, 5
17. Fang, J., Yi, T., Wang, X., Xie, L., Zhang, X., Liu, W., Nießner, M., Tian, Q.: Fast dynamic radiance fields with time-aware neural voxels. In: SIGGRAPH Asia (2022) 11, 12, 7
18. Fei, B., Xu, J., Zhang, R., Zhou, Q., Yang, W., He, Y.: 3d gaussian as a new vision era: A survey. arXiv preprint arXiv:2402.07181 (2024) 2, 4
19. Franco, J.S., Boyer, E.: Exact polyhedral visual hulls. In: BMVC (2003) 1
20. Gao, K., Gao, Y., He, H., Lu, D., Xu, L., Li, J.: Nerf: Neural radiance field in 3d vision, a comprehensive review. arXiv preprint arXiv:2210.00379 (2022) 2
21. Grossman, J.P., Dally, W.J.: Point sample rendering. In: Eurographics Workshop (1998) 4

22. Guédon, A., Lepetit, V.: Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. In: CVPR (2024) 3, 4, 9, 10, 5
23. Hu, L., Zhang, H., Zhang, Y., Zhou, B., Liu, B., Zhang, S., Nie, L.: Gaussianavatar: Towards realistic human avatar modeling from a single video via animatable 3d gaussians. In: CVPR (2024) 5
24. Hu, S., Hu, T., Liu, Z.: Gauhuman: Articulated gaussian splatting from monocular human videos. In: CVPR (2024) 5
25. Huang, B., Yu, Z., Chen, A., Geiger, A., Gao, S.: 2d gaussian splatting for geometrically accurate radiance fields. In: SIGGRAPH (2024) 10, 11, 12, 3, 4, 5, 6
26. Huang, Y.H., Sun, Y.T., Yang, Z., Lyu, X., Cao, Y.P., Qi, X.: Sc-gs: Sparse-controlled gaussian splatting for editable dynamic scenes. In: CVPR (2024) 5
27. Jain, A., Tancik, M., Abbeel, P.: Putting nerf on a diet: Semantically consistent few-shot view synthesis. In: ICCV (2021) 4, 14
28. Jensen, R., Dahl, A., Vogiatzis, G., Tola, E., Aanæs, H.: Large scale multi-view stereopsis evaluation. In: CVPR (2014) 9, 12, 3, 6
29. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ToG (2023) 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12
30. Kilian, M., Mitra, N.J., Pottmann, H.: Geometric modeling in shape space. In: SIGGRAPH (2007) 4
31. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: ICLR (2015) 6, 1
32. Kocabas, M., Chang, R., Gabriel, J., Tuzel, O., Ranjan, A.: Hugs: Human gaussian splats. In: CVPR (2024) 5
33. Kopanas, G., Philip, J., Leimkühler, T., Drettakis, G.: Point-based neural rendering with per-view optimization. Computer Graphics Forum (2021) 4
34. Lassner, C., Zollhofer, M.: Pulsar: Efficient sphere-based neural rendering. In: CVPR (2021) 4, 1
35. Lei, J., Wang, Y., Pavlakos, G., Liu, L., Daniilidis, K.: Gart: Gaussian articulated template models. In: CVPR (2024) 5
36. Li, R., Gao, H., Tancik, M., Kanazawa, A.: Nerfacc: Efficient sampling accelerates nerfs. In: ICCV (2023) 4
37. Li, T., Slavcheva, M., Zollhoefer, M., Green, S., Lassner, C., Kim, C., Schmidt, T., Lovegrove, S., Goesele, M., Newcombe, R., et al.: Neural 3d video synthesis from multi-view video. In: CVPR (2022) 4, 13, 14, 6
38. Li, Z., Chen, Z., Li, Z., Xu, Y.: Spacetime gaussian feature splatting for real-time dynamic view synthesis. In: CVPR (2024) 5
39. Li, Z., Wang, Q., Cole, F., Tucker, R., Snavely, N.: Dynibar: Neural dynamic image-based rendering. In: CVPR (2023) 14, 3
40. Lin, Y., Dai, Z., Zhu, S., Yao, Y.: Gaussian-flow: 4d reconstruction with dynamic 3d gaussian particle. In: CVPR (2024) 5
41. Lu, T., Yu, M., Xu, L., Xiangli, Y., Wang, L., Lin, D., Dai, B.: Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. In: CVPR (2024) 10, 4, 5
42. Lu, Z., Guo, X., Hui, L., Chen, T., Yang, M., Tang, X., Zhu, F., Dai, Y.: 3d geometry-aware deformable gaussian splatting for dynamic view synthesis. In: CVPR (2024) 5
43. Luiten, J., Kopanas, G., Leibe, B., Ramanan, D.: Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. In: 3DV (2024) 4, 8

44. Mihajlovic, M., Bansal, A., Zollhoefer, M., Tang, S., Saito, S.: Keypointnerf: Generalizing image-based volumetric avatars using relative spatial encoding of keypoints. In: ECCV (2022) 4
45. Mihajlovic, M., Prokudin, S., Pollefeys, M., Tang, S.: ResFields: Residual neural fields for spatiotemporal signals. In: ICLR (2024) 3, 5, 7, 8, 12, 13, 14, 6
46. Mihajlovic, M., Weder, S., Pollefeys, M., Oswald, M.R.: Deepsurfels: Learning online appearance fusion. In: CVPR (2021) 4
47. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: ECCV (2020) 1, 2, 3, 4, 7, 9, 10, 13, 5, 8
48. Moran, P.A.: Notes on continuous stochastic phenomena. Biometrika (1950) 2, 6, 9, 1
49. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ToG (2022) 2, 4
50. Niemeyer, M., Barron, J.T., Mildenhall, B., Sajjadi, M.S., Geiger, A., Radwan, N.: Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In: CVPR (2022) 2, 4
51. Pang, H., Zhu, H., Kortylewski, A., Theobalt, C., Habermann, M.: Ash: Animatable gaussian splats for efficient and photoreal human rendering. In: CVPR (2024) 5
52. Park, K., Sinha, U., Barron, J.T., Bouaziz, S., Goldman, D.B., Seitz, S.M., Martin-Brualla, R.: Nerfies: Deformable neural radiance fields. In: ICCV (2021) 4, 13, 14, 3
53. Park, K., Sinha, U., Hedman, P., Barron, J.T., Bouaziz, S., Goldman, D.B., Martin-Brualla, R., Seitz, S.M.: Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. ToG (2021) 4, 11, 12, 13, 14, 6, 7
54. Peng, S., Yan, Y., Shuai, Q., Bao, H., Zhou, X.: Representing volumetric videos as dynamic mlp maps. In: CVPR (2023) 2
55. Peng, S., Niemeyer, M., Mescheder, L., Pollefeys, M., Geiger, A.: Convolutional occupancy networks. In: ECCV (2020) 3, 8
56. Prokudin, S., Ma, Q., Raafat, M., Valentin, J., Tang, S.: Dynamic point fields. In: ICCV (2023) 4, 3
57. Pumarola, A., Corona, E., Pons-Moll, G., Moreno-Noguer, F.: D-nerf: Neural radiance fields for dynamic scenes. In: CVPR (2021) 4, 12, 13, 14, 3, 6
58. Qian, S., Kirschstein, T., Schoneveld, L., Davoli, D., Giebenhain, S., Nießner, M.: Gaussianavatars: Photorealistic head avatars with rigged 3d gaussians. In: CVPR (2024) 5
59. Qian, Z., Wang, S., Mihajlovic, M., Geiger, A., Tang, S.: 3dgs-avatar: Animatable avatars via deformable 3d gaussian splatting. In: CVPR (2024) 5, 1, 3
60. Rahaman, N., Baratin, A., Arpit, D., Draxler, F., Lin, M., Hamprecht, F., Bengio, Y., Courville, A.: On the spectral bias of neural networks. In: ICML (2019) 2, 7
61. Reiser, C., Peng, S., Liao, Y., Geiger, A.: Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In: ICCV (2021) 4
62. Reiser, C., Szeliski, R., Verbin, D., Srinivasan, P., Mildenhall, B., Geiger, A., Barron, J., Hedman, P.: Merf: Memory-efficient radiance fields for real-time view synthesis in unbounded scenes. ToG (2023) 4
63. Ren, Y., Wang, F., Zhang, T., Pollefeys, M., Süsstrunk, S.: Volrecon: Volume rendering of signed ray distance functions for generalizable multi-view reconstruction. In: CVPR (2023) 3, 6
64. Rückert, D., Franke, L., Stamminger, M.: Adop: Approximate differentiable one-pixel point rendering. ToG (2022) 4

65. Saito, S., Schwartz, G., Simon, T., Li, J., Nam, G.: Relightable gaussian codec avatars. In: CVPR (2024) 5
66. Sara Fridovich-Keil and Alex Yu, Tancik, M., Chen, Q., Recht, B., Kanazawa, A.: Plenoxels: Radiance fields without neural networks. In: CVPR (2022) 4
67. Schönberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: CVPR (2016) 7, 1
68. Shi, R., Wei, X., Wang, C., Su, H.: Zerorf: Fast sparse view 360deg reconstruction with zero pretraining. In: CVPR (2024) 3, 7, 8, 11, 12, 6
69. Singh, M., Fuenmayor, E., Hinchy, E.P., Qiao, Y., Murray, N., Devine, D.: Digital twin: Origin to future. Applied System Innovation (2021) 1
70. Tancik, M., Casser, V., Yan, X., Pradhan, S., Mildenhall, B., Srinivasan, P.P., Barron, J.T., Kretzschmar, H.: Block-nerf: Scalable large scene neural view synthesis. In: CVPR (2022) 4
71. Tewari, A., Thies, J., Mildenhall, B., Srinivasan, P., Tretschk, E., Yifan, W., Lassner, C., Sitzmann, V., Martin-Brualla, R., Lombardi, S., et al.: Advances in neural rendering. In: Computer Graphics Forum (2022) 3, 4, 7
72. Tosi, F., Zhang, Y., Gong, Z., Sandström, E., Mattoccia, S., Oswald, M.R., Poggi, M.: How nerfs and 3d gaussian splatting are reshaping slam: a survey. arXiv preprint arXiv:2402.13255 (2024) 2
73. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Deep image prior. In: CVPR (2018) 7, 8
74. Wang, C., Eckart, B., Lucey, S., Gallo, O.: Neural trajectory fields for dynamic novel view synthesis. arXiv preprint arXiv:2105.05994 (2021) 4, 13
75. Wang, G., Chen, Z., Loy, C.C., Liu, Z.: Sparsenerf: Distilling depth ranking for few-shot novel view synthesis. In: ICCV (2023) 9, 10, 3, 4, 5
76. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing (2004) 6
77. Wiles, O., Gkioxari, G., Szeliski, R., Johnson, J.: Synsin: End-to-end view synthesis from a single image. In: CVPR (2020) 4
78. Wu, G., Yi, T., Fang, J., Xie, L., Zhang, X., Wei, W., Liu, W., Tian, Q., Xinggang, W.: 4d gaussian splatting for real-time dynamic scene rendering. In: CVPR (2024) 3, 5, 11, 12, 13, 14, 6, 7, 8
79. Xie, Y., Takikawa, T., Saito, S., Litany, O., Yan, S., Khan, N., Tombari, F., Tompkin, J., Sitzmann, V., Sridhar, S.: Neural fields in visual computing and beyond. In: Computer Graphics Forum. Wiley Online Library (2022) 1, 3, 4
80. Xu, Y., Lu, Y., Wen, Z.: Owlii dynamic human mesh sequence dataset. In: ISO/IEC JTC1/SC29/WG11 m41658, 120th MPEG Meeting (2017) 8, 9, 12, 14, 1, 3
81. Xu, Y., Chen, B., Li, Z., Zhang, H., Wang, L., Zheng, Z., Liu, Y.: Gaussian head avatar: Ultra high-fidelity head avatar via dynamic gaussians. In: CVPR (2024) 5
82. Yan, Z., Li, C., Lee, G.H.: Nerf-ds: Neural radiance fields for dynamic specular objects. In: CVPR (2023) 11, 12, 3, 6, 7
83. Yang, C., Li, S., Fang, J., Liang, R., Xie, L., Zhang, X., Shen, W., Tian, Q.: Gaussianobject: Just taking four images to get a high-quality 3d object with gaussian splatting. arXiv preprint arXiv:2402.10259 (2024) 14
84. Yang, Z., Yang, H., Pan, Z., Zhu, X., Zhang, L.: Real-time photorealistic dynamic scene representation and rendering with 4d gaussian splatting. In: ICLR (2024) 3, 5, 13, 14, 6, 8
85. Yang, Z., Gao, X., Zhou, W., Jiao, S., Zhang, Y., Jin, X.: Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In: CVPR (2024) 3, 5, 11, 12, 13, 14, 6, 7, 8

86. Yariv, L., Hedman, P., Reiser, C., Verbin, D., Srinivasan, P.P., Szeliski, R., Barron, J.T., Mildenhall, B.: Bakedsdf: Meshing neural sdfs for real-time view synthesis. In: SIGGRAPH (2023) 4
87. Yifan, W., Serena, F., Wu, S., Öztireli, C., Sorkine-Hornung, O.: Differentiable surface splatting for point-based geometry processing. ToG (2019) 4
88. Yu, A., Li, R., Tancik, M., Li, H., Ng, R., Kanazawa, A.: PlenOctrees for real-time rendering of neural radiance fields. In: ICCV (2021) 4, 5
89. Yu, A., Ye, V., Tancik, M., Kanazawa, A.: pixelnerf: Neural radiance fields from one or few images. In: CVPR (2021) 2, 4, 14
90. Yu, H., Julin, J., Milacski, Z.A., Niinuma, K., Jeni, L.A.: Cogs: Controllable gaussian splatting. In: CVPR (2024) 5
91. Yu, Z., Chen, A., Huang, B., Sattler, T., Geiger, A.: Mip-splatting: Alias-free 3d gaussian splatting. In: CVPR (2024) 3, 4, 9, 10, 5
92. Zhang, J., Yang, G., Tulsiani, S., Ramanan, D.: Ners: Neural reflectance surfaces for sparse-view 3d reconstruction in the wild. NeurIPS (2021) 4
93. Zhang, Y., Prokudin, S., Mihajlovic, M., Ma, Q., Tang, S.: Degrees of freedom matter: Inferring dynamics from point trajectories. In: CVPR (2024) 13, 14, 3
94. Zwicker, M., Pfister, H., Van Baar, J., Gross, M.: Ewa volume splatting. In: VIS (2001) 2, 4, 6
95. Zwicker, M., Pfister, H., Van Baar, J., Gross, M.: Surface splatting. In: PACMCGIT (2001) 2, 4