HAC: Hash-grid Assisted Context for 3D Gaussian Splatting Compression

Supplementary Material

Yihang Chen[1,2], Qianyi Wu[2], Weiyao Lin[1⋆], Mehrtash Harandi[2], and Jianfei Cai[2]

[1] Shanghai Jiao Tong University
[2] Monash University
{yhchen.ee, wylin}@sjtu.edu.cn,
{qianyi.wu, mehrtash.harandi, jianfei.cai}@monash.edu

Abstract. This is the supplementary material for our paper. Herein, we offer more details of implementations, an extra experiment, quantitative per-scene results across all datasets, and a comprehensive notation table.

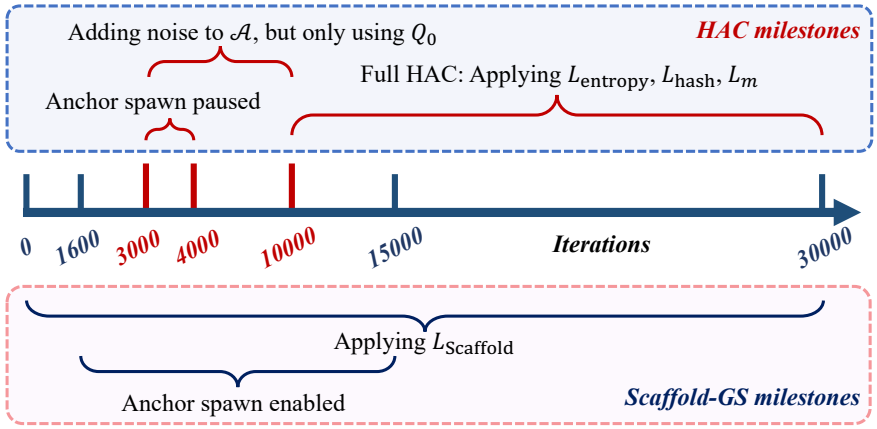

Fig. 1: Detailed training process of our HAC model. We use the blue box to indicate the training process related to our model, while using the pink box for Scaffold-GS [5].

1 More Implementation Details

1.1 Training Process

We provide a detailed overview of the training process for our HAC framework, as illustrated in Fig. 1.

⋆ Corresponding author

Table 1: Bit allocation among anchor's attributes. We set $\lambda_e = 4e - 3$. When calculating per-param size, we only consider *valid* anchors that are not pruned.

Dataset	Total size (MB)			Per-param size (bit)		
	$\boldsymbol{f}^a$	$\boldsymbol{l}$	$\boldsymbol{o}$	$\boldsymbol{f}^a$	$\boldsymbol{l}$	$\boldsymbol{o}$
Bungee-NeRF [7]	8.76	2.53	3.62	3.03	7.27	4.56
Synthetic-NeRF [6]	0.31	0.09	0.12	1.33	3.58	3.76

During the initial 3000 iterations, no additional techniques are applied to impact the original training process of Scaffold-GS [5], ensuring a stable start of the anchor attribute training and anchor spawning.
From iteration 3000 to 10000, we introduce "adding noise" operations to anchor attributes $\mathcal{A}$, which allows the model to adapt to the quantization process. Note that, in this stage, we only apply Q_0 for quantization without using $\boldsymbol{r}$ for refinement, therefore, we do not need the hash grid. Specifically, we pause the anchor spawning process between iterations 3000 and 4000 for a transitional period, as the sudden introduction of quantization may introduce instability to the spawning process. Once the parameters are fitted to the quantization after iteration 4000, we re-enable the spawning process. Note that we do not incorporate the hash grid in this stage (*i.e.*, before iteration 10000) because we want to provide a transition for the anchor attributes and the spawning process to fit the quantization operation, enabling a more stable training process when the hash grid is incorporated in the further iterations.
After iteration 10000, assuming the 3D model is adequately fitted to the quantization, we fully integrate our HAC framework to jointly train the binary hash grid. Notably, the bound of the hash grid is determined using the maximum and minimum anchor locations at the 10000th iteration, which are then utilized to normalize anchor locations for interpolation in the hash grid. This comprehensive pipeline ensures a stable training process to reduce the model size via entropy constraints while maintaining a high-quality fidelity.

1.2 Sampling Strategy

During training, employing all anchors for entropy training in each iteration could result in prolonged training time and potential out-of-memory (OOM) issues. Therefore, we adopt a sampling strategy that, in each iteration, we only randomly sample and entropy train 5% of anchors from those used for rendering. This approach ensures faster training speeds while still preserving satisfactory RD performance.

2 Additional Experiments

We investigate bit allocation among the anchor's three attributes, as depicted in Tab. 1. When viewing from the total size, feature $\boldsymbol{f}^a$ contributes the most

due to its highest dimensionality. However, as it needs to be inputted into MLPs to extract Gaussian attributes, it exhibits the most significant dimensional redundancy, resulting in the smallest per-parameter bit. Conversely, this is not the case for scaling $\boldsymbol{l}$ and offsets $\boldsymbol{o}$, which are directly used for rendering, making much fewer dimensional redundancies. Additionally, as $\boldsymbol{l}$ and $\boldsymbol{o}$ are always of higher decimal precision, their value distributions are more difficult to accurately predict, resulting higher per-parameter bit consumption.

3 Quantitative Results of Each Scene

3.1 Detailed Results of Our HAC Framework

Detailed per-scene results of Synthetic-NeRF dataset [6] are given in Tab. 2.
Detailed per-scene results of Mip-NeRF360 dataset [1] are given in Tab. 3.
Detailed per-scene results of Tank&Temples dataset [4] are given in Tab. 4.
Detailed per-scene results of DeepBlending dataset [2] are given in Tab. 5.
Detailed per-scene results of BungeeNeRF dataset [7] are given in Tab. 6.

3.2 Detailed Results of the Base Models

We also give detailed per-scene results of all datasets of our two base models 3DGS [3] and Scaffold-GS [5] in Tab. 7 and Tab. 8, respectively.

4 Notation Table

Please refer to Tab. 9 for detailed notation explanations.

Table 2: Per-scene results of Synthetic-NeRF dataset [6] of our approach.

λ_e	Scenes	PSNR↑	SSIM↑	LPIPS↓	SIZE↓
0.004	chair	34.02	0.981	0.018	0.82
	drums	26.20	0.950	0.044	1.23
	ficus	34.27	0.983	0.016	0.71
	hotdog	36.44	0.979	0.033	0.51
	lego	34.25	0.976	0.027	0.97
	materials	30.20	0.959	0.045	1.07
	mic	35.39	0.989	0.011	0.55
	ship	31.24	0.902	0.124	1.82
	AVG	**32.75**	**0.965**	**0.040**	**0.96**
0.003	chair	34.33	0.982	0.017	0.89
	drums	26.26	0.951	0.043	1.42
	ficus	34.57	0.984	0.015	0.82
	hotdog	36.70	0.980	0.031	0.54
	lego	34.65	0.977	0.024	1.07
	materials	30.29	0.960	0.043	1.20
	mic	35.62	0.990	0.010	0.62
	ship	31.32	0.903	0.121	1.86
	AVG	**32.97**	**0.966**	**0.038**	**1.05**
0.002	chair	34.73	0.984	0.016	1.03
	drums	26.32	0.952	0.043	1.45
	ficus	34.90	0.985	0.014	0.94
	hotdog	37.11	0.981	0.029	0.64
	lego	35.04	0.979	0.022	1.25
	materials	30.53	0.961	0.041	1.45
	mic	35.92	0.990	0.010	0.67
	ship	31.38	0.903	0.119	1.99
	AVG	**33.24**	**0.967**	**0.037**	**1.18**
0.001	chair	35.21	0.985	0.014	1.32
	drums	26.38	0.952	0.041	1.95
	ficus	35.37	0.986	0.013	1.20
	hotdog	37.47	0.983	0.026	0.79
	lego	35.51	0.981	0.019	1.61
	materials	30.58	0.961	0.040	1.62
	mic	36.25	0.991	0.009	0.81
	ship	31.48	0.904	0.116	2.50
	AVG	**33.53**	**0.968**	**0.035**	**1.47**
0.0005	chair	35.49	0.986	0.013	1.67
	drums	26.45	0.952	0.041	2.32
	ficus	35.30	0.986	0.013	1.53
	hotdog	37.87	0.984	0.024	0.97
	lego	35.67	0.981	0.019	1.90
	materials	30.70	0.962	0.039	2.07
	mic	36.71	0.992	0.008	1.01
	ship	31.52	0.904	0.115	3.39
	AVG	**33.71**	**0.968**	**0.034**	**1.86**

Table 3: Per-scene results of Mip-NeRF360 dataset [1] of our approach.

λ_e	Scenes	PSNR↑	SSIM↑	LPIPS↓	SIZE↓
0.004	bicycle	25.05	0.742	0.264	27.54
	garden	27.28	0.842	0.151	22.69
	stump	26.58	0.762	0.269	18.11
	room	31.55	0.921	0.208	5.53
	counter	29.35	0.911	0.195	7.26
	kitchen	31.16	0.923	0.131	8.05
	bonsai	32.28	0.942	0.189	8.56
	flower	21.26	0.572	0.381	19.59
	treehill	23.30	0.645	0.356	20.04
	AVG	**27.53**	**0.807**	**0.238**	**15.26**
0.003	bicycle	25.05	0.742	0.261	30.02
	garden	27.36	0.844	0.148	24.62
	stump	26.64	0.763	0.265	19.85
	room	31.71	0.922	0.206	5.72
	counter	29.54	0.913	0.191	7.93
	kitchen	31.22	0.925	0.128	8.84
	bonsai	32.50	0.944	0.186	9.40
	flower	21.26	0.571	0.383	20.67
	treehill	23.26	0.645	0.356	22.08
	AVG	**27.62**	**0.808**	**0.236**	**16.57**
0.002	bicycle	25.10	0.742	0.262	33.14
	garden	27.43	0.847	0.143	27.52
	stump	26.59	0.761	0.268	21.75
	room	31.87	0.925	0.201	6.47
	counter	29.65	0.915	0.189	8.88
	kitchen	31.46	0.928	0.125	10.05
	bonsai	32.70	0.945	0.184	10.51
	flower	21.32	0.576	0.377	23.73
	treehill	23.34	0.647	0.350	24.83
	AVG	**27.72**	**0.809**	**0.233**	**18.54**
0.001	bicycle	25.11	0.742	0.259	39.15
	garden	27.46	0.849	0.139	32.17
	stump	26.59	0.763	0.264	25.26
	room	31.90	0.926	0.198	7.85
	counter	29.74	0.918	0.184	10.44
	kitchen	31.63	0.930	0.122	12.07
	bonsai	32.97	0.948	0.180	12.72
	flower	21.27	0.575	0.377	27.55
	treehill	23.26	0.648	0.345	29.65
	AVG	**27.77**	**0.811**	**0.230**	**21.87**
0.0005	bicycle	25.05	0.742	0.258	44.01
	garden	27.50	0.850	0.139	36.27
	stump	26.57	0.762	0.264	28.93
	room	32.19	0.929	0.194	9.16
	counter	29.75	0.918	0.185	12.22
	kitchen	31.81	0.931	0.120	13.96
	bonsai	33.16	0.949	0.178	14.90
	flower	21.28	0.575	0.376	31.24
	treehill	23.22	0.646	0.346	34.42
	AVG	**27.83**	**0.811**	**0.229**	**25.01**

Table 4: Per-scene results of Tank&Temples dataset [4] of our approach.

λ_e	Scenes	PSNR↑	SSIM↑	LPIPS↓	SIZE↓
0.004	truck	25.88	0.878	0.158	9.26
	train	22.19	0.815	0.216	6.94
	AVG	**24.04**	**0.846**	**0.187**	**8.10**
0.003	truck	25.99	0.880	0.153	9.80
	train	22.49	0.817	0.213	7.59
	AVG	**24.24**	**0.849**	**0.183**	**8.70**
0.002	truck	25.99	0.881	0.153	11.15
	train	22.66	0.819	0.210	8.64
	AVG	**24.33**	**0.850**	**0.181**	**9.90**
0.001	truck	26.02	0.883	0.147	12.42
	train	22.78	0.823	0.207	10.07
	AVG	**24.40**	**0.853**	**0.177**	**11.24**
0.0005	truck	26.00	0.883	0.146	15.12
	train	22.49	0.823	0.206	11.19
	AVG	**24.25**	**0.853**	**0.176**	**13.16**

Table 5: Per-scene results of DeepBlending dataset [2] of our approach.

λ_e	Scenes	PSNR↑	SSIM↑	LPIPS↓	SIZE↓
0.004	playroom	30.44	0.902	0.272	3.15
	drjohnson	29.53	0.903	0.265	5.55
	AVG	**29.98**	**0.902**	**0.269**	**4.35**
0.003	playroom	30.61	0.903	0.269	3.66
	drjohnson	29.67	0.904	0.261	5.71
	AVG	**30.14**	**0.903**	**0.265**	**4.69**
0.002	playroom	30.66	0.905	0.265	4.12
	drjohnson	29.69	0.905	0.258	6.51
	AVG	**30.17**	**0.905**	**0.262**	**5.32**
0.001	playroom	30.84	0.906	0.262	5.03
	drjohnson	29.85	0.906	0.255	7.67
	AVG	**30.34**	**0.906**	**0.258**	**6.35**
0.0005	playroom	30.66	0.906	0.259	6.08
	drjohnson	29.76	0.906	0.255	9.09
	AVG	**30.21**	**0.906**	**0.257**	**7.58**

Table 6: Per-scene results of BungeeNeRF dataset [7] of our approach.

λ_e	Scenes	PSNR↑	SSIM↑	LPIPS↓	SIZE↓
0.004	amsterdam	26.80	0.865	0.224	22.49
	bilbao	27.65	0.864	0.231	17.14
	hollywood	24.25	0.748	0.347	16.55
	pompidou	25.16	0.829	0.266	20.40
	quebec	29.33	0.918	0.192	15.06
	rome	25.68	0.845	0.243	19.30
	AVG	**26.48**	**0.845**	**0.250**	**18.49**
0.003	amsterdam	26.95	0.873	0.214	24.41
	bilbao	27.82	0.872	0.218	18.76
	hollywood	24.27	0.753	0.342	17.87
	pompidou	25.34	0.837	0.255	22.49
	quebec	29.67	0.924	0.185	16.15
	rome	25.98	0.855	0.231	20.83
	AVG	**26.67**	**0.852**	**0.241**	**20.08**
0.002	amsterdam	27.13	0.880	0.202	27.14
	bilbao	28.02	0.880	0.205	20.91
	hollywood	24.43	0.763	0.330	20.09
	pompidou	25.27	0.842	0.249	24.85
	quebec	29.98	0.929	0.175	17.90
	rome	26.28	0.866	0.219	23.07
	AVG	**26.85**	**0.860**	**0.230**	**22.33**
0.001	amsterdam	27.25	0.886	0.190	31.84
	bilbao	27.98	0.886	0.190	24.38
	hollywood	24.59	0.772	0.319	23.41
	pompidou	25.58	0.851	0.236	29.19
	quebec	30.30	0.934	0.163	21.23
	rome	26.61	0.876	0.203	26.91
	AVG	**27.05**	**0.868**	**0.217**	**26.16**
0.0005	amsterdam	27.24	0.891	0.180	36.31
	bilbao	28.09	0.891	0.181	27.72
	hollywood	24.60	0.778	0.313	26.00
	pompidou	25.60	0.853	0.231	33.55
	quebec	30.19	0.936	0.155	24.56
	rome	26.73	0.881	0.194	30.21
	AVG	**27.08**	**0.872**	**0.209**	**29.72**

Table 7: Per-scene results of all evaluated datasets of 3DGS [3].

Datasets	Scenes	PSNR↑	SSIM↑	LPIPS↓	SIZE↓
Synthetic-NeRF	chair	35.65	0.988	0.010	115.77
	drums	26.28	0.955	0.037	92.87
	ficus	35.48	0.987	0.012	64.53
	hotdog	38.05	0.985	0.020	43.37
	lego	35.98	0.982	0.017	80.53
	materials	30.48	0.960	0.037	38.50
	mic	36.76	0.993	0.006	48.31
	ship	31.73	0.907	0.107	63.82
	AVG	**33.80**	**0.970**	**0.031**	**68.46**
Mip-NeRF360	bicycle	25.11	0.746	0.245	1336.45
	garden	27.30	0.856	0.122	1327.99
	stump	26.66	0.770	0.242	1070.92
	room	31.74	0.926	0.197	353.10
	counter	29.07	0.914	0.184	277.39
	kitchen	31.47	0.931	0.117	414.33
	bonsai	32.12	0.946	0.181	295.33
	flower	21.36	0.588	0.360	814.24
	treehill	22.62	0.636	0.347	812.63
	AVG	**27.49**	**0.813**	**0.222**	**744.71**
Tank&Temples	truck	25.38	0.877	0.148	606.99
	train	22.00	0.811	0.208	254.91
	AVG	**23.69**	**0.844**	**0.178**	**430.95**
DeepBlending	playroom	29.83	0.900	0.247	551.93
	drjohnson	29.02	0.898	0.247	775.91
	AVG	**29.42**	**0.899**	**0.247**	**663.92**
BungeeNeRF	amsterdam	26.03	0.874	0.170	1458.14
	bilbao	26.35	0.864	0.191	1350.37
	hollywood	23.44	0.767	0.241	1601.76
	pompidou	21.20	0.772	0.266	2169.21
	quebec	28.83	0.923	0.156	1468.76
	rome	23.34	0.848	0.206	1649.12
	AVG	**24.87**	**0.841**	**0.205**	**1616.23**

Table 8: Per-scene results of all evaluated datasets of Scaffold-GS [5].

Datasets	Scenes	PSNR↑	SSIM↑	LPIPS↓	SIZE↓
Synthetic-NeRF	chair	34.96	0.985	0.013	15.50
	drums	26.36	0.949	0.045	26.93
	ficus	34.66	0.984	0.015	16.46
	hotdog	37.82	0.984	0.022	11.31
	lego	35.48	0.981	0.018	19.84
	materials	30.37	0.958	0.043	23.12
	mic	36.37	0.991	0.008	14.83
	ship	31.27	0.896	0.119	26.90
	AVG	**33.41**	**0.966**	**0.035**	**19.36**
Mip-NeRF360	bicycle	24.50	0.705	0.306	248.00
	garden	27.17	0.842	0.146	271.00
	stump	26.27	0.784	0.284	493.00
	room	31.93	0.925	0.202	133.00
	counter	29.34	0.914	0.191	194.00
	kitchen	31.30	0.928	0.126	173.00
	bonsai	32.70	0.946	0.185	258.00
	flower	21.14	0.566	0.417	253.00
	treehill	23.19	0.642	0.410	262.00
	AVG	**27.50**	**0.806**	**0.252**	**253.89**
Tank&Temples	truck	25.77	0.883	0.147	107.00
	train	22.15	0.822	0.206	66.00
	AVG	**23.96**	**0.853**	**0.177**	**86.50**
DeepBlending	playroom	30.62	0.904	0.258	63.00
	drjohnson	29.80	0.907	0.250	69.00
	AVG	**30.21**	**0.906**	**0.254**	**66.00**
BungeeNeRF	amsterdam	27.16	0.898	0.188	223.00
	bilbao	26.60	0.857	0.257	178.00
	hollywood	24.49	0.787	0.318	155.00
	pompidou	24.94	0.839	0.271	209.00
	quebec	30.28	0.936	0.190	159.00
	rome	26.23	0.873	0.225	174.00
	AVG	**26.62**	**0.865**	**0.241**	**183.00**

Table 9: Notation Table. With slight abuse of notation, we use L_{3d} and L_{2d} to represent the number of levels of the 3D and 2D part of the hash grid, respectively.

Notation	Shape	Definition
$\boldsymbol{x}$	$\mathbb{R}^3$	A random 3D point
$\boldsymbol{\mu}$	$\mathbb{R}^3$	Location of Gaussians in 3DGS [3]
$\boldsymbol{\Sigma}$	$\mathbb{R}^{3\times3}$	Covariance matrix of Gaussians
$\boldsymbol{S}$	$\mathbb{R}^{3\times3}$	Scale matrix of Gaussians
$\boldsymbol{R}$	$\mathbb{R}^{3\times3}$	Rotation matrix of Gaussians
α	$\mathbb{R}^1$	Opacity of Gaussians after 2D projection
$\boldsymbol{c}$	$\mathbb{R}^3$	View-dependent color of Gaussians
I		Number of Gaussians contributed to the rendering
$\boldsymbol{C}$	$\mathbb{R}^3$	The obtained pixel value after rendering
$\boldsymbol{x}^a$	$\mathbb{R}^3$	Anchor location
$\boldsymbol{f}^a$	$\mathbb{R}^{D^a}$	Feature of the anchor
$\boldsymbol{l}$	$\mathbb{R}^6$	Scaling of the anchor
$\boldsymbol{o}$	$\mathbb{R}^{3K}$	Offsets of the anchor
$\mathcal{A}$		The set of anchor's attributes including $\{\boldsymbol{f}^a, \boldsymbol{l}, \boldsymbol{o}\}$
D^a		Dimension of $\boldsymbol{f}^a$
K		Number of offsets per anchor
$\mathcal{H}$		A 3D-2D mixed binary hash grid
T		Table size of the hash grid at each level
L		Number of levels of the hash grid
D^h		Dimension of the vectors of the hash grid
$\boldsymbol{\theta}$	$\mathbb{R}^{D^h}$	A vector of the hash grid
$\boldsymbol{f}^h$	$\mathbb{R}^{D^h(L_{3d}+3L_{2d})}$	Feature obtained by interpolation of $\boldsymbol{x}^a$ in $\mathcal{H}$
$\boldsymbol{f}$	$\mathbb{R}^D$	Any of anchor's attribute vectors $\in \{\boldsymbol{f}^a, \boldsymbol{l}, \boldsymbol{o}\}$
$\hat{\boldsymbol{f}}$	$\mathbb{R}^D$	Quantized version of $\boldsymbol{f}$
D		Dimension of $\boldsymbol{f}$, which $\in \{D^a, 6, 3K\}$
$\boldsymbol{q}$	$\mathbb{R}^1$	Quantization step of $\boldsymbol{f}$
$\boldsymbol{r}$	$\mathbb{R}^1$	Quantization step refinement term
$\boldsymbol{\mu}$	$\mathbb{R}^D$	Estimated mean value for distribution modeling
$\boldsymbol{\sigma}$	$\mathbb{R}^D$	Estimated standard deviation for distribution modeling
Q_0		Base quantization step, which varies for $\boldsymbol{f}^a, \boldsymbol{l}, \boldsymbol{o}$.
N		Total number of anchors
h_f		Occurrence frequency of "+1" in $\mathcal{H}$
M_+		Total number of "+1" $\mathcal{H}$
M_-		Total number of "-1" $\mathcal{H}$
L_{Scaffold}		The loss item used in Scaffold-GS [5]
L_{entropy}		Entropy loss for measuring bits of $\mathcal{A}$
L_{hash}		Entropy loss for measuring bits of $\mathcal{H}$
L_m		Masking loss
$Loss$		The total loss
λ_e		Tradeoff parameter to achieve variable birate
λ_m		Tradeoff parameter to balance masking ratio
MLP_q		The MLP to deduce r from $\boldsymbol{f}^h$
MLP_c		The MLP to deduce μ and σ from $\boldsymbol{f}^h$
ϕ		Probability density function of Gaussian distribution
Φ		Cumulative distribution function of Gaussian distribution

References

1. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5470–5479 (2022)
2. Hedman, P., Philip, J., Price, T., Frahm, J.M., Drettakis, G., Brostow, G.: Deep blending for free-viewpoint image-based rendering. ACM Transactions on Graphics (ToG) **37**(6), 1–15 (2018)
3. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics **42**(4) (2023)
4. Knapitsch, A., Park, J., Zhou, Q.Y., Koltun, V.: Tanks and temples: Benchmarking large-scale scene reconstruction. ACM Transactions on Graphics (ToG) **36**(4), 1–13 (2017)
5. Lu, T., Yu, M., Xu, L., Xiangli, Y., Wang, L., Lin, D., Dai, B.: Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2024)
6. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM **65**(1), 99–106 (2021)
7. Xiangli, Y., Xu, L., Pan, X., Zhao, N., Rao, A., Theobalt, C., Dai, B., Lin, D.: Bungeenerf: Progressive neural radiance field for extreme multi-scale scene rendering. In: European conference on computer vision. pp. 106–122. Springer (2022)