

ARC-NeRF: Area Ray Casting for Broader Unseen View Coverage in Few-shot Object Rendering

Supplementary Material

Hyperparameter & Balancing Weights	Realistic Synthetic 360° [8]		DTU [5]			Shiny Blender [13]
	4-view	8-view	3-view	6-view	9-view	
LR	[1e-3, 1e-5]		[2e-3, 2e-5]			[1e-3, 1e-5]
Warm-up Iter.	512	1024	512	2048		512
$\eta_{\text{Ori.}}$ (for $\mathcal{L}_{\text{Ori.}}$)	1e-1	1e-2	1e-1	1e-2		1e-1
$\eta_{\text{lum.}}$ (for $l_{\text{lum.}}$)	1e-3	1e-4	1e-3	1e-4	1e-5	1e-3
$\tilde{\eta}_{\text{lum.}}$ (for $\tilde{l}_{\text{lum.}}$)	1e-4	1e-5	1e-4	1e-5	1e-6	1e-4

Table A. **Hyperparameters and balancing weights.** Since our ARC-NeRF is built upon FlipNeRF, we follow the training details for other hyperparameters, which are not mentioned here, as FlipNeRF. $[\alpha, \beta]$ denotes the annealing from α to β .

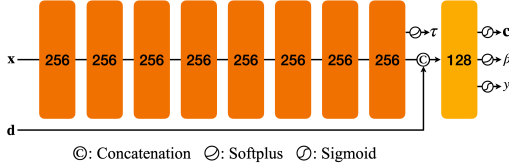


Figure A. **Network architecture of ARC-NeRF.** Our ARC-NeRF estimates the additional output y , *i.e.* the relative luminance for our auxiliary luminance estimation task.

A. Experimental Setting

Implementational details. Our ARC-NeRF is implemented upon FlipNeRF [10], and we follow its overall training scheme. We utilize the scene space annealing strategy during the initial training phase following [9–11]. Furthermore, we adopt the initial warm up and exponential decay for the learning rate. We use the Adam optimizer [7] with gradient clipping set to 0.1 for both each element of the gradient value and the gradient’s norm. Our ARC-NeRF is trained for 500 pixel epochs using a batch size of 4,096 on four NVIDIA RTX 3090 GPUs. Additionally, since our proposed Area Ray encompasses broader areas of unseen views compared to a single ray, we set the masking threshold ψ as 45° , which is smaller than that of FlipNeRF, to avoid over-regularization effect of augmented rays. The related experiment is demonstrated in Tab. C.

Hyperparameters. For additional details on hyperparameters and loss balancing terms based on training views and datasets, kindly refer to Tab. A. Note that our ARC-NeRF follows the same hyperparameters as FlipNeRF for other training losses and schemes which are not specified in Tab. A.

B. Further Details of Method

Architectural details. Our ARC-NeRF leverages the network architecture of mip-NeRF [1], which is commonly used in several few-shot NeRF models [9–11, 17]. Moreover, our ARC-NeRF additionally estimates the relative luminance y . Kindly refer to more details in Fig. A.

Total loss. Our ARC-NeRF is trained to maximize the log-likelihood of the target pixel c_{GT} for both sets of original input rays \mathcal{R} and our proposed Area Rays $\tilde{\mathcal{R}}$, as well as to minimize the mean squared errors (MSE) between the ground-truth and estimated pixel values. Except our proposed $\mathcal{L}_{\text{lum.}}$, we use the same training losses as those of FlipNeRF. Note that we use \mathcal{L}_{MSE} only for \mathcal{R} and exploit a batch of Area Rays instead of flipped reflection rays. Summing up, the total loss over a batch is calculated as follows:

$$\begin{aligned} \mathcal{L}_{\text{Total}} = & \mathcal{L}_{\text{MSE}} + \mathcal{L}_{\text{lum.}} + \eta_{\text{NLL}} \mathcal{L}_{\text{NLL}} + \tilde{\eta}_{\text{NLL}} \tilde{\mathcal{L}}_{\text{NLL}} \\ & + \eta_{\text{UE}} \mathcal{L}_{\text{UE}} + \tilde{\eta}_{\text{UE}} \tilde{\mathcal{L}}_{\text{UE}} + \eta_{\text{BFC}} \mathcal{L}_{\text{BFC}} + \eta_{\text{Ori.}} \mathcal{L}_{\text{Ori.}}, \quad (1) \end{aligned}$$

where $\mathcal{L}_{\text{lum.}} = \eta_{\text{lum.}} l_{\text{lum.}} + \tilde{\eta}_{\text{lum.}} \tilde{l}_{\text{lum.}}$.

η ’s and $\tilde{\eta}$ ’s represent the loss balancing weights for the original input rays and additional Area Rays, respectively.

C. Additional Experiments

Viewing direction jittering. For Realistic Synthetic 360° [8] and Shiny Blender [13], which consist of inward-facing synthetic scenes with objects located at the center, we adopt the viewing direction jittering, which is a minor additional strategy slightly improving the performance. We simply add the Gaussian random noise to the input viewing direction \mathbf{d} to improve the robustness for the slight change

	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	Average Err. \downarrow
FlipNeRF [10]	16.47	0.866	0.091	0.095
ARC-NeRF				
w/o view. jitter.	<u>16.66</u>	<u>0.869</u>	<u>0.087</u>	<u>0.093</u>
w/ view. jitter.	16.86	0.873	0.084	0.091

Table B. **Effect of Viewing direction jittering.** On Realistic Synthetic 360° 4-view, we are able to achieve marginal performance improvement while still outperforming FlipNeRF without the jittering strategy.

ψ	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	Average Err. \downarrow
180° (None)	18.15	0.749	0.179	0.120
90°	18.63	0.762	0.163	0.110
75°	<u>19.02</u>	0.764	0.156	<u>0.105</u>
60°	18.94	<u>0.766</u>	<u>0.154</u>	<u>0.105</u>
45°	19.85	0.773	0.146	0.096
30°	18.78	0.765	0.160	0.107
15°	18.65	0.761	0.163	0.111

Table C. **Comparison of masking thresholds.** Our ARC-NeRF excludes a set of Area Rays, whose angle θ between the original input ray is over ψ , *i.e.* the target pixel photo-consistency is relatively low considering the threshold ψ , from a training batch. $\psi = 180^\circ$ (None) uses a whole batch of newly generated Area Rays.

of viewpoints. As shown in Tab. B, we are able to achieve marginal improvement of rendering quality while still outperforming its baseline, FlipNeRF, even without the jittering strategy.

Masking thresholds. Our ARC-NeRF utilizes an additional batch of Area Rays covering a broader area of unseen views, and the high-frequency components of samples along an Area Ray are adaptively regularized via Integrated Positional Encoding (IPE) depending on the angle between the original input direction and the estimated normal vector, *i.e.* the target pixel photo-consistency. As a result, with the same $\psi = 90^\circ$ as FlipNeRF, our ARC-NeRF might suffer from the performance degradation due to over-regularization. As demonstrated in Tab. C, our ARC-NeRF achieves the best result with $\psi = 45^\circ$. The larger ψ becomes than 45° , the worse the performance, as an Area Ray covering too wide area of unseen views leads to over-regularization, which adversely affects the training. On the other hand, a smaller ψ than 45° also leads to poorer performance, as the newly generated Area Rays are excessively filtered, resulting in only a limited number of augmented Area Rays being utilized for training. Note that the masking threshold ψ depends on the characteristics of casting ray rather than being the hyperparameter which needs to be finetuned elaboratively.

	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	Average \downarrow
FreeNeRF [‡] [17]	19.92	0.781	0.125	0.086
FreeNeRF [17]	19.23	0.769	0.149	0.103
ARC-NeRF	<u>19.85</u>	<u>0.773</u>	<u>0.146</u>	<u>0.096</u>



Figure B. **Quantitative and qualitative comparison with FreeNeRF on DTU 3-view.** Although FreeNeRF achieves high-quality of rendering with only a few images, it depends on the white and black prior, which is highly heuristic based on the characteristics of dataset. Our ARC-NeRF achieves comparable performance to FreeNeRF without the heuristic prior for training. [‡] denotes the W&B prior.

Additional results. Fig. B shows the comparison with our ARC-NeRF, FreeNeRF [17], and FreeNeRF without the white and black prior. While FreeNeRF achieves high-quality rendering, it relies on a white and black prior, *i.e.* a highly heuristic approach considering the specific dataset’s characteristics. In contrast, our ARC-NeRF outperforms FreeNeRF without relying on such priors, even showing competitive performance compared to FreeNeRF with the prior.

The quantitative comparisons including the 6/9-view scenarios on DTU and more qualitative results are demonstrated in Tab. D and Fig. C, respectively. Note that we report the results of other methods from their original papers or [9–11], which outperformed the results from the corresponding original paper by modified training curriculum [9]. Our ARC-NeRF achieves competitive performance among the SOTA methods. Furthermore, our supplementary videos show the comparison with other methods on Realistic Synthetic 360° and FreeNeRF on DTU.

D. Limitations and Future Work

Our proposed ARC-NeRF excels at capturing finer textures and details of object surfaces but shows limitations when addressing complex backgrounds, such as unbounded scenes composed of widely varying depths. This is due to potential obstacles between cast rays, which hinder pixel photo-consistency. Developing a new ray parameterization that can cover broad unseen view areas across significantly varied depths while effectively dealing with the obstacles would be a meaningful extension toward few-shot view synthesis for unbounded scenes as a future work.

	Method	PSNR \uparrow			SSIM \uparrow			LPIPS \downarrow			Avg. Err. \downarrow		
		3-view	6-view	9-view	3-view	6-view	9-view	3-view	6-view	9-view	3-view	6-view	9-view
Mip-NeRF [1]	-	8.68	16.54	23.58	0.571	0.741	0.879	0.353	0.198	0.092	0.323	0.148	0.056
3DGS [6]		14.18	-	-	0.628	-	-	0.301	-	-	0.191	-	-
PixelNeRF [18]	Pre-training	16.82	19.11	20.40	0.695	0.745	0.768	0.270	0.232	0.220	0.147	0.115	0.100
PixelNeRF [†] [18]		18.95	20.56	21.83	0.710	0.753	0.781	0.269	0.223	0.203	0.125	0.104	0.090
SRF [3]		15.32	17.54	18.35	0.671	0.730	0.752	0.304	0.250	0.232	0.171	0.132	0.120
SRF [†] [3]		15.68	18.87	20.75	0.698	0.757	0.785	0.281	0.225	0.205	0.162	0.114	0.093
MVSNeRF [2]		18.63	20.70	22.40	0.769	0.823	0.853	0.197	0.156	0.135	0.113	0.088	0.068
MVSNeRF [†] [2]		18.54	20.49	22.22	0.769	0.822	0.853	0.197	0.155	0.135	0.113	0.089	0.069
DietNeRF [4]	Regularization	11.85	20.63	23.83	0.633	0.778	0.823	0.314	0.201	0.173	0.243	0.101	0.068
RegNeRF [9]		18.89	22.20	24.93	0.745	0.841	0.884	0.190	0.117	0.089	0.112	0.071	0.047
MixNeRF [11]		18.95	22.30	25.03	0.744	0.835	0.879	0.203	0.102	0.065	0.113	0.066	0.042
SimpleNeRF [12]		16.25	20.60	22.75	0.751	0.828	0.856	0.249	0.190	0.176	0.143	0.088	0.071
DiffusioNeRF [15]		16.20	20.34	25.18	0.698	0.818	0.883	0.160	0.093	0.046	0.128	0.072	0.036
SparseNeRF [14]		19.55	-	-	0.769	-	-	0.201	-	-	0.102	-	-
FreeNeRF [‡] [17]		19.92	23.25	25.60	0.781	0.838	0.877	0.125	0.085	0.057	0.086	0.058	0.038
FreeNeRF [17]		19.23	22.77	25.59	0.769	0.835	0.877	0.149	0.088	0.057	0.103	0.063	0.039
FlipNeRF [10]		19.55	22.45	25.12	0.767	0.839	0.882	0.180	0.098	0.062	0.101	0.064	0.041
SparseGS [16]		18.89	-	-	0.702	-	-	0.229	-	-	0.117	-	-
ARC-NeRF		19.85	22.73	25.14	0.773	0.842	0.886	0.146	0.084	0.057	0.096	0.060	0.040

Table D. **Additional quantitative comparison on DTU.** Our ARC-NeRF shows competitive performance by outperforming other methods on most metrics, even without relying on any dataset-specific priors. [†] and [‡] indicate fine-tuning and W&B prior, respectively.

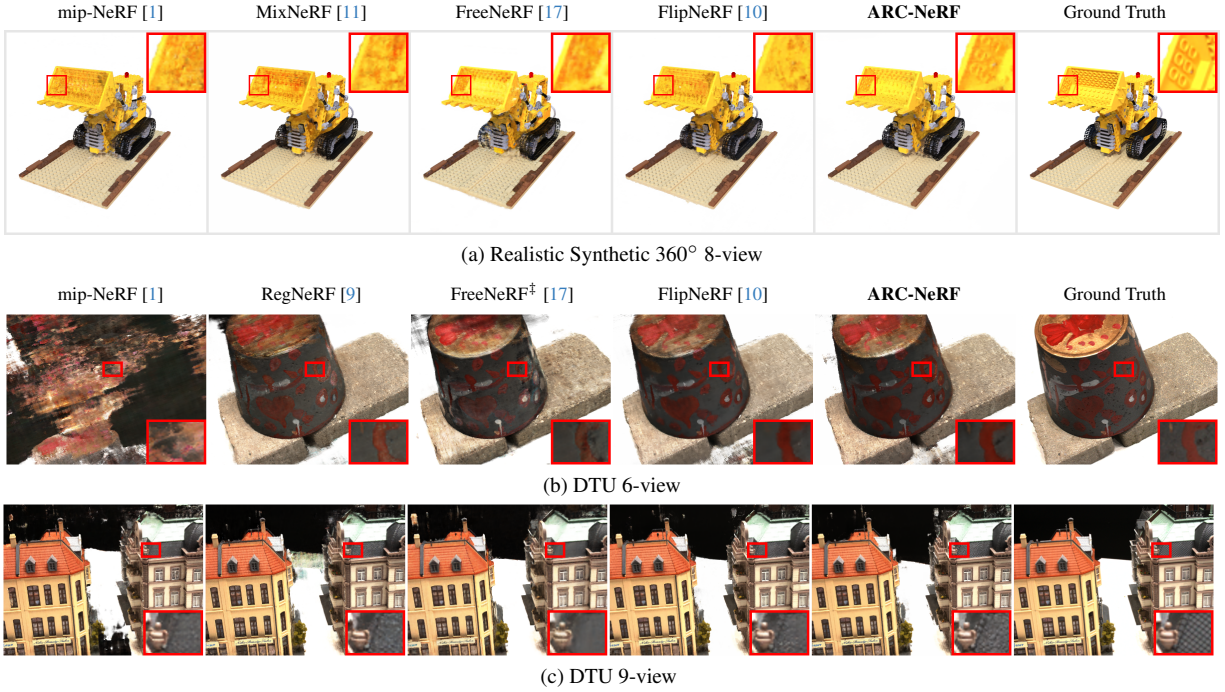


Figure C. **Additional qualitative comparisons.** Our ARC-NeRF achieves high-quality renderings with fine details and clearer texture.

References

- [1] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2021. 1, 3
- [2] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14124–14133, 2021. 3
- [3] Julian Chibane, Aayush Bansal, Verica Lazova, and Gerard Pons-Moll. Stereo radiance fields (srf): Learning view synthesis for sparse views of novel scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern*

- Recognition*, pages 7911–7920, 2021. 3
- [4] Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting nerf on a diet: Semantically consistent few-shot view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5885–5894, 2021. 3
 - [5] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engin Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 406–413, 2014. 1
 - [6] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 3
 - [7] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 1
 - [8] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 1
 - [9] Michael Niemeyer, Jonathan T Barron, Ben Mildenhall, Mehdi SM Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5480–5490, 2022. 1, 2, 3
 - [10] Seunghyeon Seo, Yeonjin Chang, and Nojun Kwak. Flipnerf: Flipped reflection rays for few-shot novel view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22883–22893, 2023. 1, 2, 3
 - [11] Seunghyeon Seo, Donghoon Han, Yeonjin Chang, and Nojun Kwak. Mixnerf: Modeling a ray with mixture density for novel view synthesis from sparse inputs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20659–20668, 2023. 1, 2, 3
 - [12] Nagabhushan Somraj, Adithyan Karanayil, and Rajiv Soundararajan. Simplenerf: Regularizing sparse input neural radiance fields with simpler solutions. In *SIGGRAPH Asia 2023 Conference Papers*, pages 1–11, 2023. 3
 - [13] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T Barron, and Pratul P Srinivasan. Ref-nerf: Structured view-dependent appearance for neural radiance fields. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5481–5490. IEEE, 2022. 1
 - [14] Guangcong Wang, Zhaoxi Chen, Chen Change Loy, and Ziwei Liu. Sparsenerf: Distilling depth ranking for few-shot novel view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9065–9076, 2023. 3
 - [15] Jamie Wynn and Daniyar Turmukhambetov. Diffusionerf: Regularizing neural radiance fields with denoising diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4180–4189, 2023. 3
 - [16] Haolin Xiong, Sairisheek Muttukuru, Rishi Upadhyay, Pradyumna Chari, and Achuta Kadambi. Sparsegs: Real-time 360 {deg} sparse view synthesis using gaussian splatting. *arXiv preprint arXiv:2312.00206*, 2023. 3
 - [17] Jiawei Yang, Marco Pavone, and Yue Wang. Freenerf: Improving few-shot neural rendering with free frequency regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8254–8263, 2023. 1, 2, 3
 - [18] 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. 3