FlipNeRF: Flipped Reflection Rays for Few-shot Novel View Synthesis —Supplementary Material—

Seunghyeon Seo Yeonjin Chang Nojun Kwak
Seoul National Univeristy
{zzzlssh, yjean8315, nojunk}@snu.ac.kr

	Real. Syn. 360° [5]		DTU [3]			LLFF [4]
	4-view	8-view	3-view	6-view	9-view	3-view
Learning Rate	$[1e{-}3, 1e{-}5]$		$[2e{-}3, 2e{-}5]$			
Warm-up Step	512	1024	512	1024		512
Delay Multiplier	$1e{-}2$					
λ_1(for $\mathcal{L}_{\text{NLL}}$)	$[4.0, 1e{-}3]$					
λ_2(for $\mathcal{L}'_{\text{NLL}}$)	$[4e{-}1, 1e{-}4]$	$[4e{-}2, 1e{-}5]$	$[4e{-}1, 1e{-}4]$	$[4e{-}2, 1e{-}5]$	$[4e{-}3, 1e{-}6]$	$[4e{-}3, 1e{-}6]$
λ_3(for $\mathcal{L}_{\text{UE}}$)	$[1e{-}4, 1e{-}1]$	$[1e{-}5, 1e{-}2]$	$[1e{-}4, 1e{-}1]$	$[1e{-}5, 1e{-}2]$	$[1e{-}6, 1e{-}3]$	$[1e{-}6, 1e{-}3]$
λ_4(for $\mathcal{L}'_{\text{UE}}$)	$1e{-}2$	$1e{-}3$	$1e{-}3$	$1e{-}4$	$1e{-}5$	$1e{-}5$
λ_5(for $\mathcal{L}_{\text{BFC}}$)	$1e{-}1$	$1e{-}2$	$1e{-}1$	$1e{-}2$	$1e{-}3$	$1e{-}3$
λ_6(for $\mathcal{L}_{\text{Ori.}}$)	$1e{-}1$	$1e{-}2$	$1e{-}1$	$1e{-}2$	$1e{-}3$	$1e{-}3$

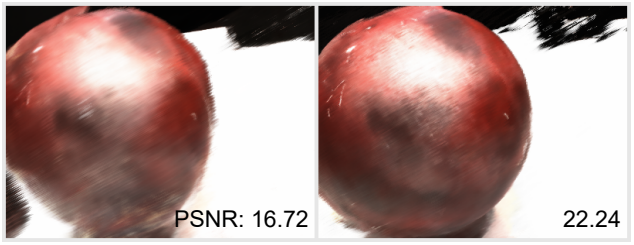
Table A: **Details of hyperparameters and loss balancing terms.** For each dataset, the more training views are provided, the smaller λ's we set to prevent over-regularization, except λ_1 for $\mathcal{L}_{\text{NLL}}$. $[a, b]$ indicates the annealing from a to b.

A. Implementation Details

The detailed hyperparameters and our loss balancing terms by the datasets and the number of training views are provided in Tab. A.

B. Explicit Normalization for the Estimated Surface Normals

As mentioned in Sec. 3.2, we use the weighted sum of blending weights and estimated normal vectors along a ray, *i.e.* $\hat{\mathbf{n}} = \sum_{i=1}^{M} w_i \mathbf{n}_i$, as the surface normals $\hat{\mathbf{n}}$ to derive a flipped reflection direction $\mathbf{d}'$. In this formulation, $\hat{\mathbf{n}}$ is not guaranteed to be a unit vector without an explicit normalization process. However, we empirically found that the normalization rather destabilizes the training and leads to the performance degradation as shown in Fig. A. We conjecture that the inaccurately generated $\mathbf{r}'$ with an explicit normalization provides wrong supervisory signals, especially during the initial training stage when the surface normals are not accurately estimated. Furthermore, as illustrated in Fig. B, $\hat{\mathbf{n}}$ is trained naturally to approximate the unit vector through the training without an explicit normalization process. With our proposed loss terms and regularization techniques for accurate normal estimation, we are able to use $\hat{\mathbf{n}}$ as the surface normal vector without any additional



w/ explicit normalization w/o explicit normalization

Figure A: **Comparison between FlipNeRF using $\hat{\mathbf{n}}$ with and without an explicit normalization.** With explicitly normalized $\hat{\mathbf{n}}$ (left), our FlipNeRF suffers from the training instability and achieves degenerate results. We are able to achieve much superior rendering quality with our proposed surface normals (right), which are trained to approximate unit vectors, thanks to our regularization techniques and masking strategy.

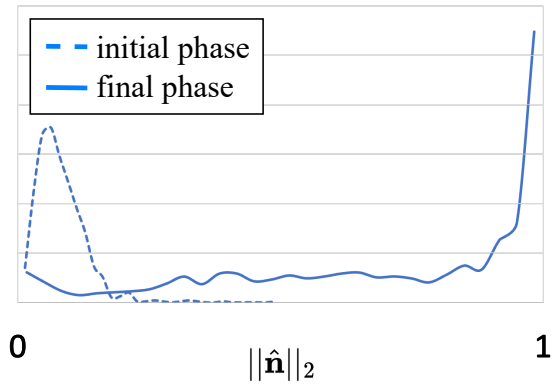
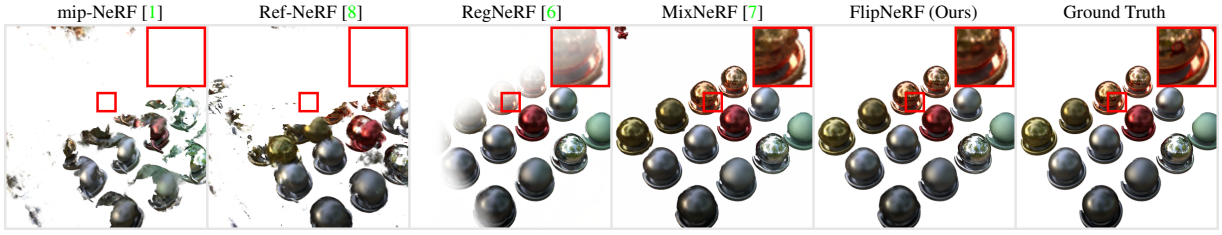


Figure B: **Distribution of $\|\hat{\mathbf{n}}\|_2$ without an explicit normalization process.** At an initial training stage, the estimated surface normals $\hat{\mathbf{n}}$ are not unit vectors as most of $\|\hat{\mathbf{n}}\|_2$ are far from 1. However, our proposed $\hat{\mathbf{n}}$ is trained to be a unit vector naturally through the training without an explicit normalization process, and most of $\|\hat{\mathbf{n}}\|_2$ are concentrated close to 1 after the training, which indicates that $\hat{\mathbf{n}}$ is successfully approximated to a unit vector.

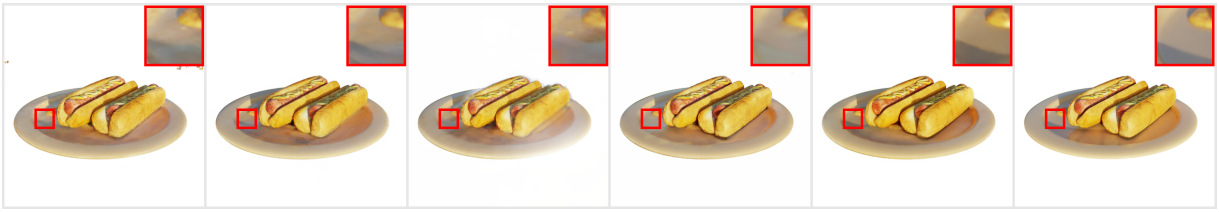
normalization process.

C. Additional Qualitative Results

The additional qualitative comparisons are provided in Fig. C, Fig. D, and Fig. E. Furthermore, we provide more qualitative results of our FlipNeRF in Fig. F, Fig. G, and Fig. H. Our FlipNeRF shows superior rendering quality compared to other baselines.



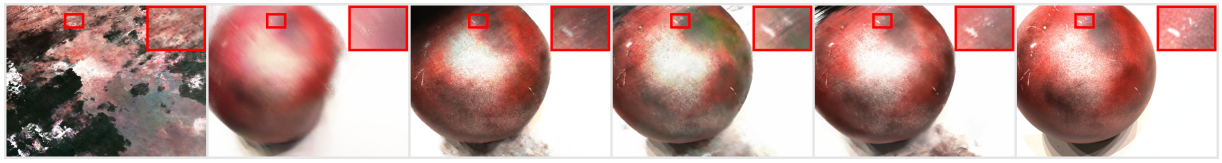
(a) 4-view



(b) 8-view

Figure C: **Additional qualitative comparisons on Realistic Synthetic 360°.**

(a) 3-view



(b) 6-view

(c) 9-view

Figure D: **Additional qualitative comparisons on DTU.**

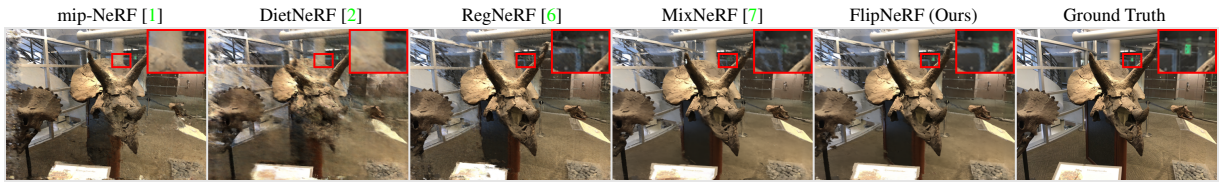
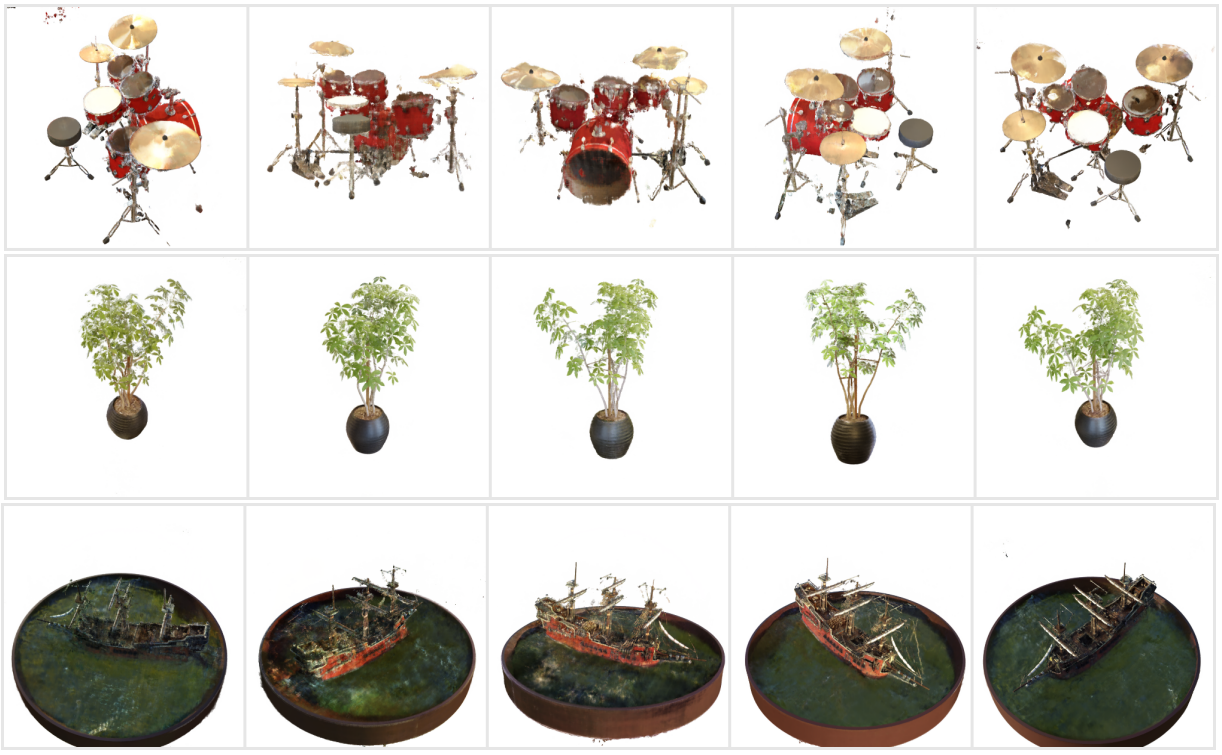


Figure E: **Qualitative comparisons on LLFF 3-view.**

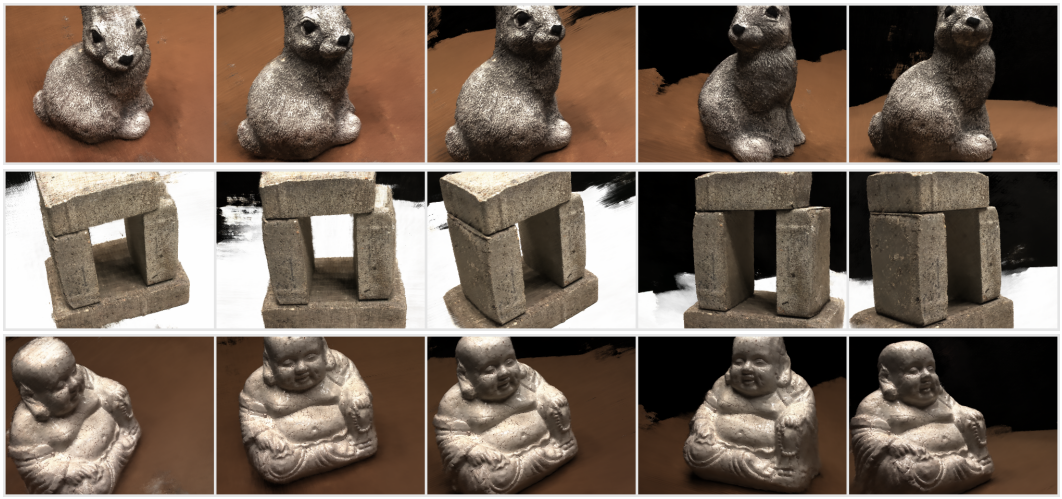
(a) 4-view



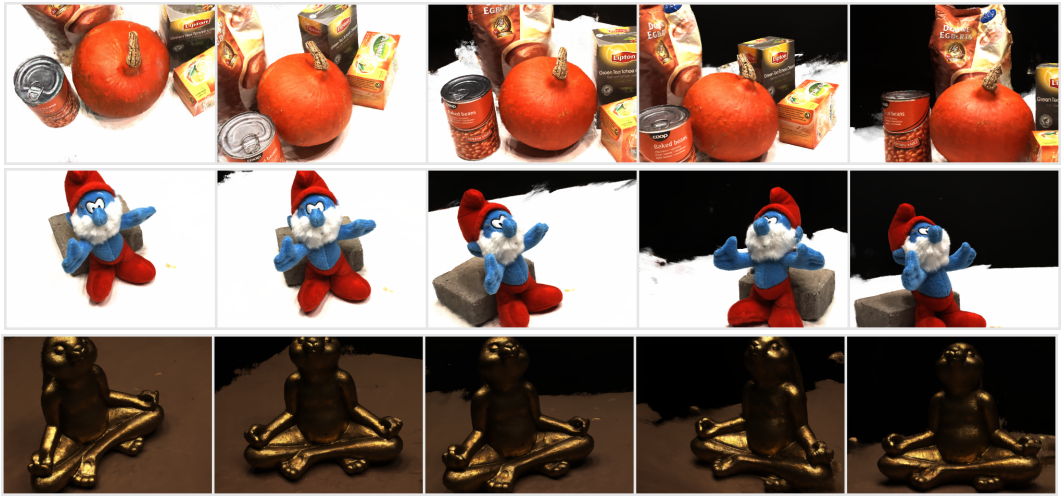
(b) 8-view

Figure F: **Additional qualitative results of our FlipNeRF on Realistic Synthetic 360°.**

(a) 3-view



(b) 6-view



(c) 9-view

Figure G: **Additional qualitative results of our FlipNeRF on DTU.**

Figure H: **Additional qualitative results of our FlipNeRF on LLFF 3-view.**

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. 3

[2] 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

[3] 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

[4] Ben Mildenhall, Pratul P Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)*, 38(4):1–14, 2019. 1

[5] 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

[6] 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. 3

[7] 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. 3

[8] 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. 3

[9] 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