NeRFLiX: High-Quality Neural View Synthesis by Learning a Degradation-Driven Inter-viewpoint MiXer Supplementary Materials

Kun Zhou[1,2*] Wenbo Li[3*] Yi Wang[4]
Tao Hu[3] Nianjuan Jiang[2] Xiaoguang Han[1] Jiangbo Lu[2†]
[1]SSE, CUHK-Shenzhen, [2]SmartMore Corporation [3]CUHK [4]Shanghai AI Laboratory
kunzhou@link.cuhk.edu.cn, {wenboli,taohu}@cse.cuhk.edu.hk
hanxiaoguang@cuhk.edu.cn,{jiangbo.lu,wygamle}@gmail.com

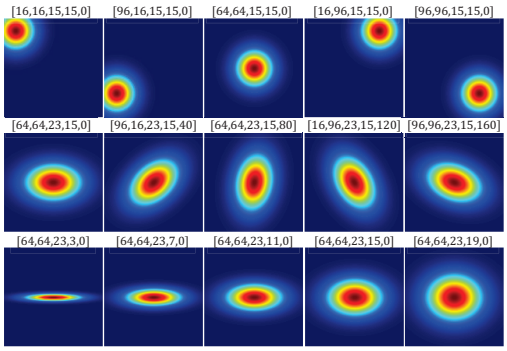


Figure A.1. We give some visualized region-adaptive masks. The parameters refer to the values of $[c_i, c_j; \sigma_i, \sigma_j, A]$ in Eq. (3).

A. NeRF Degradation Simulator

Raw data collection. We collect raw sequences from Vimeo90K [9] and LLFF-T [6]. In total, Vimeo90K contains 64612 7-frame training clips with a 448×256 resolution. Three frames (two reference views and one target view) are selected from a raw sequence of Vimeo90K in a random order. As described in Sec. 5.1, apart from the inherent displacements within the selected views, we add random global offsets to the two reference views, largely enriching the variety of inter-viewpoint changes. On the other hand, we also use the training split of the LLFF dataset, which consists of 8 different forward-facing scenes with 20-62 high-quality input views. Following previous work, we drop the eighth view and use it for evaluation. To construct a training pair from LLFF-T, we randomly select a frame as the target view and then use the proposed view selection algorithm (Sec. 4.3) to choose two reference views that are most overlapped with the target view.

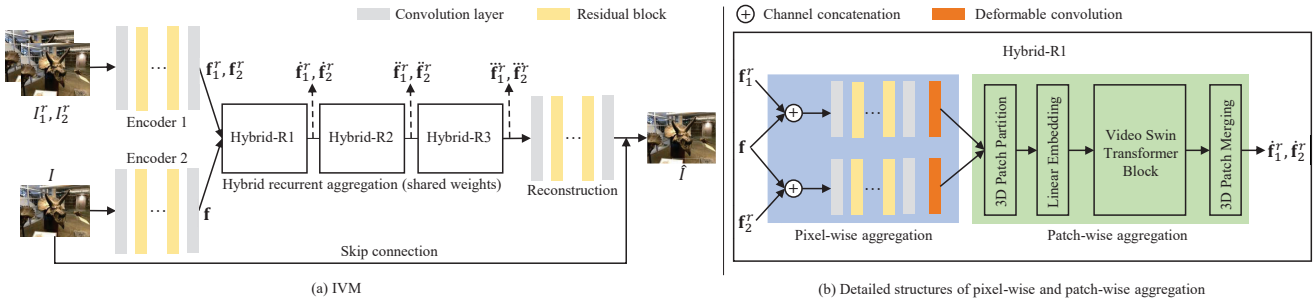
*Equal contribution
†Corresponding author

Settings	10%	50%	100%	PSNR (dB)	SSIM
LLFF-T				26.28	0.837
LLFF-T+	✓			26.71	0.840
LLFF-T+		✓		27.08	0.856
LLFF-T+			✓	**27.39**	**0.867**
TensoRF (Base)	-	-	-	26.70	0.838

Table A.1. Quantitative results of different training data sizes. First, we train an IVM model only using the LLFF-T. Then, we gradually increase the simulated pairs (10%, 50%, 100%) from Vimeo90K [9] to train another three IVM models.

Hyper-parameter setup. In Eq. (1), the 2D Gaussian noise map n is generated with a zero mean and a standard deviation ranging from 0.01 to 0.05. The isotropic blur kernel g has a size of 5×5. We employ a Gaussian blur kernel to produce blurry contents by randomly selecting kernel sizes (3-7), angles (0-180), and standard deviations (0.2-1.2). Last, in order to obtain a region-adaptive blending map M in Eq. (3), we use random means ($c_i, c_j \in (-16, 144)$), standard deviations ($\sigma_i \in (13, 25), \sigma_j \in (0, 24)$), and orientation angles ($A \in (0, 180)$). Additionally, we visualize some generated masks using different hyper-parameter combinations ($[c_i, c_j; \sigma_i, \sigma_j, A]$) in Fig. A.1.

Training data size. We investigate the influence of training data size. Under the same training and testing setups, we train several models using different training data sizes. As illustrated in Table A.1, we can observe that the final performance is positively correlated with the number of training pairs. Also, we notice the IVM trained with only LLFF-T data or additional few simulated pairs (10% of the Vimeo90K) fails to enhance the TensoRF-rendered results, *i.e.*, there is no obvious improvement compared to TensoRF [2]. This experiment demonstrates the importance of sizable training pairs for training a NeRF restorer.



(a) IVM (b) Detailed structures of pixel-wise and patch-wise aggregation

Figure A.2. The detailed framework architecture of our proposed IVM.

B. Inter-viewpoint Mixer

In Sec. 4.2, we briefly describe the framework architecture of our inter-viewpoint mixer (IVM). Here we provide more details. As illustrated in Fig. A.2(a), there are two convolutional modules ("Encoder 1/2") to extract features of the degraded view I and its two reference views $\{I_1^r, I_2^r\}$, respectively. Then, we develop a hybrid recurrent aggregation module that iteratively performs pixel-wise and patch-wise fusion. At last, a reconstruction module is implemented by a sequence of residual blocks (40 blocks) to output the enhanced view $\hat{I}$. The default channel size is 128.

Feature extraction. Given a rendered view I and its two reference views $I_{1,2}^r$, we aim to utilize the two encoders to extract deep image features $\mathbf{f}$ and $\mathbf{f}_{1,2}^r$, respectively. As detailed in Fig. A.2(a), the two encoders share an identical structure. A convolutional layer is first adopted to convert an RGB frame to a high-dimensional feature. Then we further extract the deep image feature using 5 stacked residual blocks followed by another convolution layer.

Hybrid recurrent aggregation. As depicted in Fig. A.2(a), we employ three hybrid recurrent aggregation blocks (termed "Hybrid-R1(2,3)") to progressively fuse the inter-viewpoint information from the image features ($\mathbf{f}$ and $\mathbf{f}_{\{1,2\}}^r$). Next, we take the first iteration as an example to illustrate our aggregation scheme.

Pixel-wise aggregation. As shown in Fig. A.2(b), we first merge the target view feature $\mathbf{f}$ and one of the reference features $\mathbf{f}_{\{1,2\}}^r$ by channel concatenation. Then we use a convolutional layer to reduce the channel dimension and five residual blocks followed by another convolutional layer to obtain a fused deep feature. Later on, the fused feature and the reference feature are further aggregated through a deformable convolution. And the other reference image follows the same processing pipeline. In this case, we finally obtain two features after the pixel-wise aggregation.

Patch-wise aggregation. We adopt a window-based attention mechanism [5] to accomplish patch-wise aggregation. In detail, the pixel-wisely fused features are first divided into several 3D slices through a 3D patch partition layer. Then, we obtain 3D tokens via a linear embedding operation and aggregate patch-wise information using a video Swin transformer block. Finally, 3D patches are regrouped into a 3D feature map.

In the next iteration, we split the 3D feature map into two "reference" features $\dot{\mathbf{f}}_{\{1,2\}}^r$ and repeat the pixel-wise and patch-wise aggregation. Note that, the weights of pixel-wise and patch-wise modules are shared across all iterations to reduce the model complexity.

Method	IVM-0V	IVM-1V	IVM-2V	IVM-3V
PSNR (dB)	26.87	27.26	27.39	27.44
SSIM	0.846	0.862	0.867	0.869

Table C.2. Quantitative results of different numbers of reference views.

C. Additional Results

Number of reference views. By default, we perform inter-viewpoint aggregation using two reference views (termed IVM-2V). We train another three models (IVM-0V, IVM-1V, and IVM-3V) adopting different numbers of reference views. The results are shown in Table C.2. The model without using reference views (IVM-0V) achieves the lowest PSNR and SSIM values compared with other models. Meanwhile, it is observed that the more reference views used, the higher IVM performance, indicating the importance of utilizing high-quality reference views.

View selection. Fig. C.3 exhibits the selected views by our algorithm in different NeRF scenes. We see that the proposed view selection strategy is able to choose the most relevant ones from freely captured views.

Qualitative results. Here, we provide more visual examples to adequately validate the effectiveness of our approach. As shown in Fig. C.4, Fig. C.5, Fig. C.6, Fig. C.7, NeRFLiX consistently improves NeRF-rendered images with clearer details and fewer artifacts for all NeRF models. For example, NeRFLiX successfully recovers recognizable characters, object textures, and more realistic reflectance effects, while effectively eliminating the rendering artifacts.

Figure C.3. Visual comparison between two view selection methods.

Video demo. We also provide a video demo for a clear visual comparison. First, we show some NeRF-rendered views and the restored counterparts of NeRFLiX. Then, we provide two video cases (one is from LLFF and the other is an in-the-wild scene) to compare the rendered views of TensoRF [2] and enhanced results of our NeRFLiX. It is observed that NeRFLiX is capable of producing clearer image details and removing the majority of the NeRF rendering artifacts. Our project page is available at https://redrock303.github.io/nerflix/. Due to restrictions on the upload size of the supplementary materials, we have uploaded this video demo to our project page.

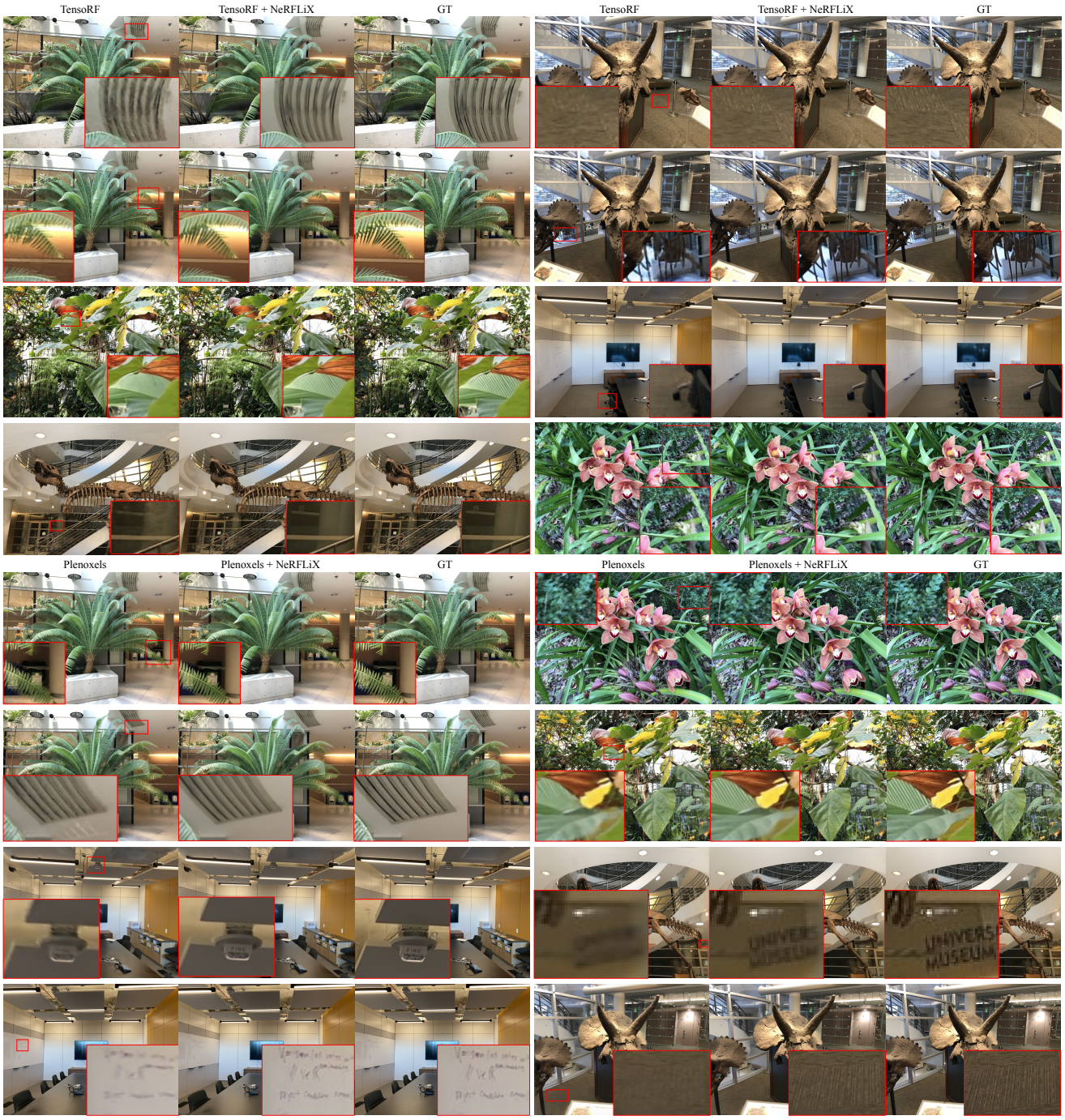


Figure C.4. Qualitative evaluation of the improvement over two SOTA NeRF models (TensoRF [2] and Plenoxels [3]) on LLFF [6] under LLFF-P1.

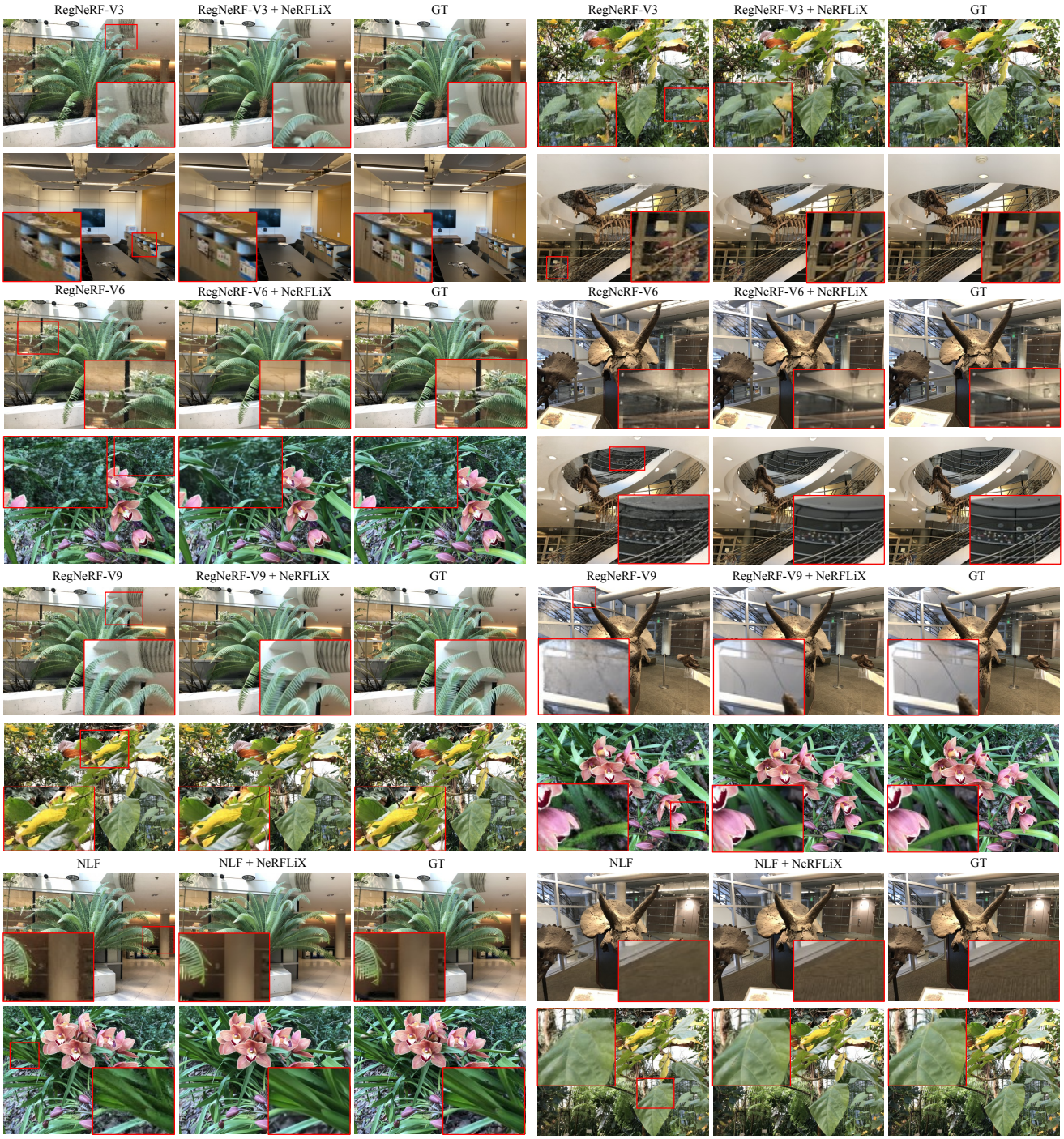
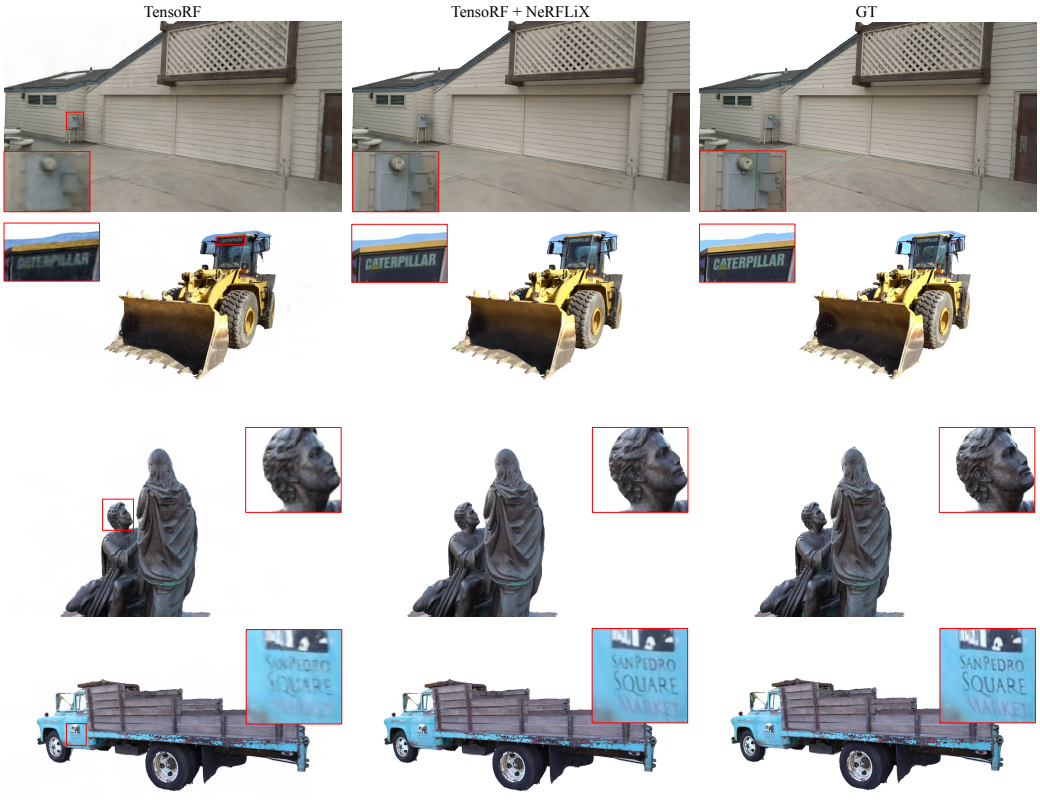
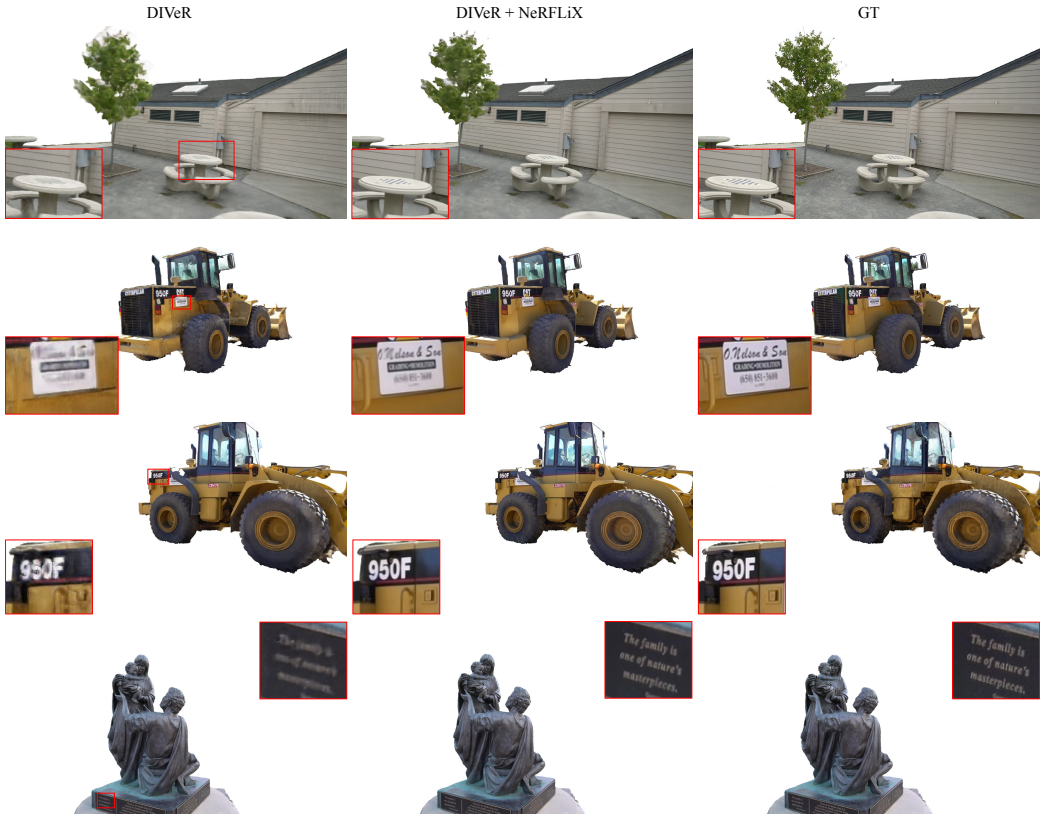


Figure C.5. Qualitative evaluation of the improvement over two SOTA NeRF models (RegNeRF [7] and NLF [1]) on LLFF [6] under LLFF-P2. RegNeRF-V3(6,9) takes 3(6,9) input views for training.



(a) Qualitative evaluation of the improvement over TensoRF [2] on Tanks and Temples [4].



(b) Qualitative evaluation of the improvement over DIVeR [8] on Tanks and Temples [4].

Figure C.6. Qualitative evaluation of the improvement over two SOTA NeRF models on Tanks and Temples [4].

Figure C.7. Qualitative evaluation of the improvement over two SOTA NeRF models (Plenoxels [3] and TensoRF [2]) on noisy LLFF Synthetic.

References

[1] Benjamin Attal, Jia-Bin Huang, Michael Zollhöfer, Johannes Kopf, and Changil Kim. Learning neural light fields with ray-space embedding networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 5

[2] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *European Conference on Computer Vision (ECCV)*, 2022. 1, 3, 4, 6, 7

[3] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5501–5510, 2022. 4, 7

[4] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36(4):1–13, 2017. 6

[5] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3202–3211, 2022. 2

[6] 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, 4, 5

[7] Michael Niemeyer, Jonathan T. Barron, Ben Mildenhall, Mehdi S. M. Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022. 5

[8] Liwen Wu, Jae Yong Lee, Anand Bhattad, Yu-Xiong Wang, and David Forsyth. Diver: Real-time and accurate neural radiance fields with deterministic integration for volume rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16200–16209, 2022. 6

[9] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *IJCV*, 127(8):1106–1125, 2019. 1