A. Additional Ablation Analysis

Besides the ablation study shown in Sec. 4.2, we provide more analyses of choices when we design the network architecture. The baseline network has the backbone with depth as 60 layers and 3 Residual Blocks (RB) in each Super-Resolution (SR) module. We use the data from `Chair` in the synthetic 360° dataset to train all models with 300K iterations on Nvidia V100 GPU with batch size as 8. The latency is profiled using iPhone 13 (iOS 16) with CoreMLTools [11]. For the models that are too big to fit into the Nvidia V100 GPU, we only benchmark the latency instead of training the models. The conducted comparisons are introduced as follows:

- **Number of RB per SR module.** We verify the number of RB in each SR module. From Tab. 6, using 3 RB in each SR model, *i.e.*, 3RB per SR, achieves a better trade-off between network performance, *e.g.*, PSNR, and latency.
- **Number of output channels in SR.** We change the number of output channels of the RB in SR. We choose the design of 64 output channels in the first two SR modules and 16 channels in the last SR module, *i.e.*, C64-64-16 in Tab. 6, because it has a real-time inference speed on the mobile device while maintaining a good performance.
- **Width of backbone.** We adopt different width, *i.e.,* the number of channels in the convolution layers, for the backbone. The network with the width as 256, *i.e.,* W256 in Tab. 6, gives us the better trade-off between latency and network performance.
- **Depth of backbone.** Lastly, we show how the depth, *i.e.,* the number of layers, in the backbone affects the performance. We chose the depth as 60, *i.e.,* D60 in Tab. 6, due to the better PSNR and satisfied latency.

B. Per-Scene Quantitative Results

Here we provide detailed per-scene comparison results (PSNR, SSIM, and LPIPS) on the synthetic 360° (Tab. 7, Tab. 8, and Tab. 9) and forward-facing (Tab. 10, Tab. 11, and Tab. 12) datasets. Please note that as we implement our framework following NeRF-Pytorch [44] and R2L [41], we use the same evaluating pipeline for measuring the quantitative metrics. We also compare our results with the models trained by NeRF-Pytorch [44]. As can be seen from the comparisons, our approach (MobileR2L) achieves comparable or even better results than NeRF [33] and NeRF-Pytorch [44].

C. More Visual Comparisons

In Fig. 8, we present more visual comparison results on the synthetic 360° and forward-facing datasets. We note, (1) MobileR2L can produce the *correct* texture details that NeRF cannot, *e.g.*, on the scene `Mic`, NeRF almost loses the grid texture of the Mic while our MobileR2L manages to render it out; similarly, on the scene `Horn`, there is (at least) one button on the glass wall missed by NeRF while our MobileR2L does not. (2) When both MobileR2L and NeRF can render out the details, MobileR2L typically generates *clearer, sharper, and less noisy* results: on the scene `T-Rex`, it is obvious that our MobileR2L renders much less noisy railing; similar phenomenon can also be observed on the scene `Hotdog`, `Material`, and `Drum`.

D. Power Usage

We profile the power usage of our proposed method on MacBook Pro (chip: Apple M1 Pro, OS: Ventura, V13.1) with the public tool[2], due to no publicly available tools for benchmarking the power usage on iPhones. Tab. 13 shows the power usage (in W) when running MobileNeRF (on GPU) and our work (on the neural engine) for the Synthetic 360° and Forward-facing datasets. Our work consumes less power than MobileNeRF, especially for real-world scenes (7.7× less on Forward-facing). MobileNeRF requires to load complicated textures while ours does not. We also profile the CPU and RAM usage for MobileNeRF and our work, which are similar (RAM: 1GB, CPU: 2W).

Table 6. **Ablation analysis on network architectures.** We report the number of parameters (#Params), PSNR, SSIM, LPIPS, and Latency (ms, on iPhone 13) for each design choice.

	#Params	PSNR↑	SSIM↑	LPIPS↓	Latency↓
1RB per SR	3.9M	31.58	0.9973	0.0503	22.38
2RB per SR	3.9M	31.63	0.9973	0.0484	26.21
3RB per SR	3.9M	31.73	0.9973	0.0368	30.42
4RB per SR	4.0M	31.59	0.9972	0.0508	33.80
C16-16-16	3.8M	31.01	0.9969	0.0644	22.77
C32-32-32	3.9M	31.39	0.9972	0.0525	35.77
C64-64-16	3.9 M	31.63	0.9973	0.0484	26.21
C64-64-64	3.9M	-	-	-	59.31
W64	0.3M	28.83	0.9951	0.0896	13.22
W128	1.0M	30.23	0.9963	0.0699	17.91
W256	3.9M	31.63	0.9973	0.0484	26.21
W384	8.7M	32.28	0.9977	0.0359	39.08
W512	15.4M	-	-	-	53.47
D30	1.9M	30.86	0.9965	0.0609	18.72
D60	3.9M	31.63	0.9973	0.0484	26.21
D80	5.2M	31.60	0.9972	0.0499	31.49
D100	6.6M	-	-	-	36.55

[2]https://github.com/tlkh/asitop

Table 7. Per-scene PSNR↑ comparison on the Synthetic 360° dataset between NeRF [33], NeRF-Pytorch [44], and our approach.

Method	Chair	Drums	Ficus	Hotdog	Lego	Materials	Mic	Ship	Average
NeRF [33]	33.00	25.01	30.13	36.18	32.54	29.62	32.91	28.65	31.01
NeRF-Pytorch [44]	33.31	25.14	30.28	36.52	31.80	29.25	32.50	28.54	30.92
MobileR2L (Ours)	33.66	25.05	29.80	36.84	32.18	30.54	34.37	28.75	31.34

Table 8. Per-scene SSIM↑ comparison on the Synthetic 360° dataset between NeRF [33], NeRF-Pytorch [44], and our approach.

Method	Chair	Drums	Ficus	Hotdog	Lego	Materials	Mic	Ship	Average
NeRF [33]	0.967	0.925	0.964	0.974	0.961	0.949	0.980	0.856	0.947
NeRF-Pytorch [44]	0.998	0.985	0.996	0.998	0.991	0.989	0.996	0.980	0.991
MobileR2L (Ours)	0.998	0.986	0.996	0.998	0.992	0.992	0.997	0.982	0.993

Table 9. Per-scene LPIPS↓ comparison on the Synthetic 360° dataset between NeRF [33], NeRF-Pytorch [44], and our approach.

Method	Chair	Drums	Ficus	Hotdog	Lego	Materials	Mic	Ship	Average
NeRF [33]	0.046	0.091	0.044	0.121	0.050	0.063	0.028	0.206	0.081
NeRF-Pytorch [44]	0.025	0.066	0.023	0.022	0.029	0.035	0.021	0.144	0.045
MobileR2L (Ours)	0.027	0.083	0.025	0.026	0.043	0.029	0.012	0.162	0.051

Table 10. Per-scene PSNR↑ comparison on the Forward-facing dataset between NeRF [33], NeRF-Pytorch [44], and our approach.

Method	Room	Fern	Leaves	Fortress	Orchids	Flower	T-Rex	Horns	Average
NeRF [33]	32.70	25.17	20.92	31.16	20.36	27.40	26.80	27.45	26.50
NeRF-Pytorch [44]	32.10	24.80	20.50	31.20	20.45	27.50	26.48	27.05	26.26
MobileR2L (Ours)	32.09	24.39	20.52	30.81	20.06	27.61	26.71	27.01	26.15

Table 11. Per-scene SSIM↑ comparison on the Forward-facing dataset between NeRF [33], NeRF-Pytorch [44], and our approach.

Method	Room	Fern	Leaves	Fortress	Orchids	Flower	T-Rex	Horns	Average
NeRF [33]	0.948	0.792	0.690	0.881	0.641	0.827	0.880	0.828	0.811
NeRF-Pytorch [44]	0.989	0.976	0.921	0.995	0.920	0.968	0.972	0.983	0.965
MobileR2L (Ours)	0.995	0.973	0.923	0.995	0.916	0.971	0.973	0.982	0.966

Table 12. Per-scene LPIPS↓ comparison on the Forward-facing dataset between NeRF [33], NeRF-Pytorch [44], and our approach.

Method	Room	Fern	Leaves	Fortress	Orchids	Flower	T-Rex	Horns	Average
NeRF [33]	0.178	0.280	0.316	0.171	0.321	0.219	0.249	0.268	0.250
NeRF-Pytorch [44]	0.089	0.210	0.921	0.995	0.920	0.968	0.972	0.983	0.153
MobileR2L (Ours)	0.088	0.239	0.280	0.103	0.296	0.150	0.121	0.217	0.187

Table 13. Power usage on the Synthetic 360° dataset and the Forward-facing dataset between MobileNeRF [10] and our approach.

Synthetic 360°	Chair	Drums	Ficus	Hotdog	Lego	Material	Mic	Ship	Avg↓
MobileNeRF	1.7W	1.6W	1.4W	4.3W	2.6W	2.1W	1.2W	7.3W	2.8W
Ours	2.5W	2.5W	2.5W	2.5W	2.5W	2.5W	2.5W	2.5W	2.5W
Forward-facing	**Fern**	**Flower**	**Fortress**	**Horn**	**Leaves**	**Orchids**	**Room**	**Trex**	**Avg↓**
MobileNeRF	12.3W	13.0W	12.4W	12.8W	15.1W	14.5W	12.8W	12.9W	13.2W
Ours	1.7W	1.7W	1.7W	1.7W	1.7W	1.7W	1.7W	1.7W	1.7W

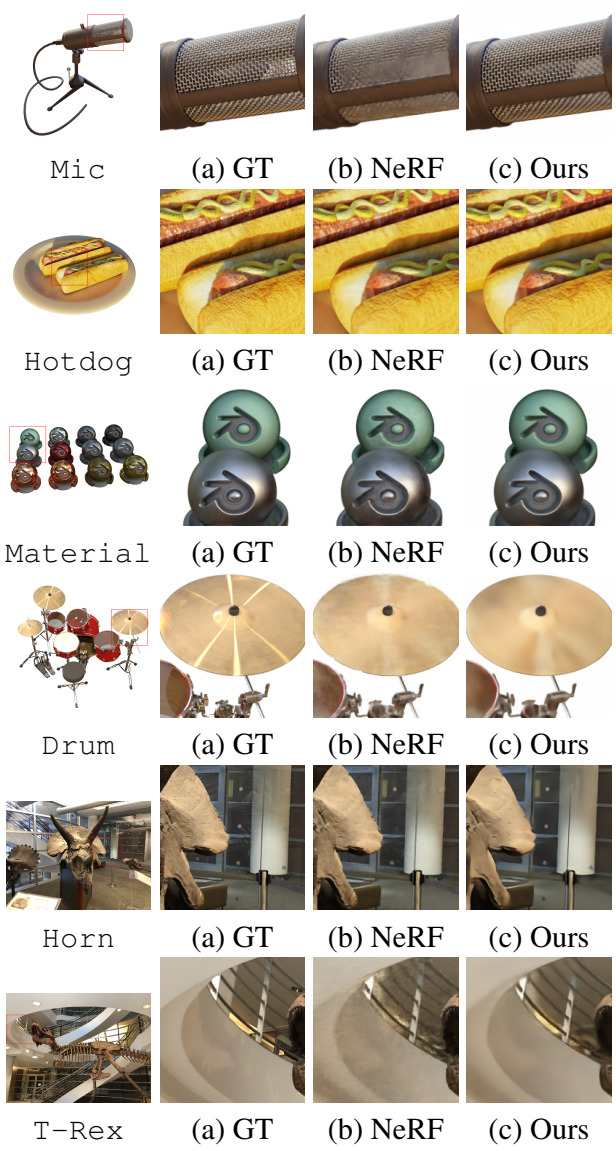

Figure 8. More visual comparisons between our method and NeRF [33] (trained via NeRF-Pytorch [44]) on the synthetic 360° (size: $800 \times 800 \times 3$) and real-world forward-facing scenes (size: $1008 \times 756 \times 3$). *Best viewed in color and zoomed in.*