Efficient Neural Light Fields (ENeLF) for Mobile Devices

Austin Peng
College of Computing
Georgia Institute of Technology
Atlanta, GA, USA
apeng39@gatech.edu

***Abstract*—Novel view synthesis (NVS) is a challenge in computer vision and graphics, focusing on generating realistic images of a scene from unobserved camera poses, given a limited set of authentic input images. Neural radiance fields (NeRF) achieved impressive results in rendering quality by utilizing volumetric rendering. However, NeRF and its variants are unsuitable for mobile devices due to the high computational cost of volumetric rendering. Emerging research in neural light fields (NeLF) eliminates the need for volumetric rendering by directly learning a mapping from ray representation to pixel color. NeLF has demonstrated its capability to achieve results similar to NeRF but requires a more extensive, computationally intensive network that is not mobile-friendly. Unlike existing works, this research builds upon the novel network architecture introduced by MobileR2L and aggressively applies a compression technique (channel-wise structure pruning) to produce a model that runs efficiently on mobile devices with lower latency and smaller sizes, with a slight decrease in performance.**

***Index Terms*—novel view synthesis, neural light field, pruning**

I. Introduction

Impressive breakthroughs in NVS and 3D reconstruction have recently been made. The trained models can generate 3D views from 2D examples, making them highly desirable and versatile (e.g. virtual telepresence, metaverse). However, widespread adoption of such applications requires them to run efficiently on resource-constrained devices. Considering the limitations in resource-constrained devices regarding computing power and storage, there has been an increased interest in research on rendering efficiency. Despite the remarkable rendering quality in the family of NeRF techniques, their slow, repetitive sampling method results in non-real-time performance. Although various approaches have attempted to address this issue, many still rely on high-end GPUs, making them impractical for resource-constrained devices. Emerging research methods rely solely on neural networks that can directly encapsulate the 3D representation from multiple 2D observations.

ENeLF proposes a solution tailored for real-time NVS on mobile devices. Drawing inspiration from previous work like R2L and MobileR2L, ENeLF employs data distillation, an efficient CNN backbone and super-resolution module, and the pruning model compression technique. This design choice enables further real-time rendering at a small sacrifice in quality.

II. Related Works

NeRF. NeRF achieved impressive results in rendering quality for view synthesis by mapping a 5D input (x, y, z, θ, ϕ) to 4D output (R, G, B, σ) [16]. However, a notable drawback of NeRF is that the multi-layer perceptron (MLP) is sampling hundreds of times per pixel to perform accurate volumetric rendering, resulting in slow training and inference times for high-resolution images. Researchers have conducted many additional studies utilizing more efficient radiance field representations [9], [12], [17], [23], or distilling meshes from the radiance field [3], [4], [19], [21]. However, few studies have achieved real-time rendering on mobile devices.

NeLF. NeLF is an emerging research field and can achieve similar rendering quality to NeRF with significant inference speedup. NeLF utilizes a light field (integral of radiance) and can reduce the hundreds of samples per ray to a single sample per ray, resolving the bottleneck sampling issue by eliminating volumetric rendering. However, the light field space is more complex than the radiance field. NeLF requires a larger model and more data to achieve a similar rendering quality. The increased computational cost causes higher training time than NeRF but at significantly lower inference time.

Real-Time NeLF Rendering. In achieving real-time rendering speed of NeLF, three main groups of approaches tackle this issue. **(1)** The first group includes approaches like NeuLF [11] and LFN [18], which frame NVS as directly mapping the camera rays to the pixel color, making inference speed faster by requiring a single forward pass per pixel. However, the light field is more complex than the radiance field, making the learning process more challenging. The intuition for this is that radiance at neighboring spaces does not change significantly (given the radiance field in the physical world is typically continuous). In contrast, two neighboring rays can drastically differ due to occlusion. A second approach of **(2)** altering network design is taken to resolve this issue with light field complexity. R2L [19] modified their MLP to a depth of 88 layers (compared to NeRF's 11-layer MLP) to introduce sufficient network representation space to learn the light field. Due to the significant increase in model size, R2L employs a pre-trained teacher NeRF model to distill data and help train their deeper MLP. With their hardware and software setup, R2L achieved $\sim$30$\times$ speedup and $\sim$2 DB PSNR increase

compared to NeRF [16]. **(3)** The third group found ways to represent the light field space in alternate ways. NeLF with Ray-Space Embeddings (RSE) [2] maps 4D ray-space into an intermediate, interpolable latent space. SIGNET [5] utilizes a variety of Fourier-inspired input transformation strategies to represent the light field. SIGNET's main limitation is that it doesn't guarantee a photorealistic reconstruction [8]. As a result, this research will mainly focus on utilizing the findings of MobileR2L [3], which introduces a super-resolution (SR) module into the MLP so the network can learn the pixel colors of missing rays through upsampling. Focus is placed on the techniques utilized by MobileR2L as their SR module introduces significant efficiencies by only needing to pass a fraction of the rays through the model and learning the representation of the remaining pixels via SR. Throughout the paper, comparisons will be made to MobileR2L [3], the current state-of-the-art (SOTA) method for real-time rendering on mobile devices, achieving the highest PSNR among all existing techniques.

III. Methodology

A. Prerequisites: R2L, MobileR2L, Pruning

R2L. NeLF maps the camera ray directly to a pixel color, but because the light field is much harder to learn than the radiance field, R2L [19] proposes an 88-layer deep residual MLP architecture to learn the mapping from ray to pixel color. Using residuals allows the model architecture to be much deeper and capture the complex light field space. However, a larger model needs more data to learn the representation sufficiently. R2L tackles this issue by pre-training a NeRF model as a teacher model to synthesize additional pseudo data. Then, they adopt a two-stage training process. First, the distilled data is used to train the R2L model. Then, the original data is used to finetune the R2L model. Their technique allowed the R2L model to achieve performance comparable to the teacher NeRF model. Fig. 1 illustrates the training and inference pipeline.

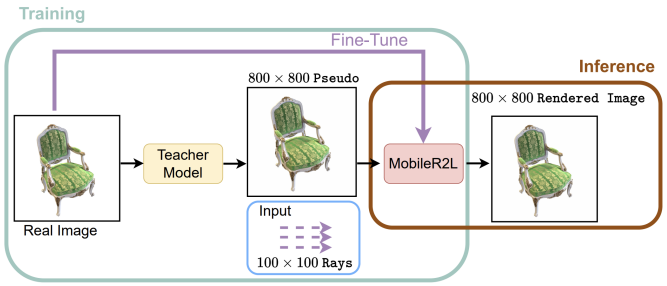

Fig. 1. R2L/MobileR2L training and inference network design.

MobileR2L. R2L network design introduced the possibility of real-time rendering and efficiency gains, but it still had high memory costs. For example, an 800×800 image still needs the prediction of 640,000 rays, which causes out-of-memory issues even with high-end GPUs (e.g. A100) [3]. MobileR2L introduced an SR module that allows the forward pass of a fraction of the rays and learns the rest of the pixels through upsampling with transposed convolutions. In addition, MobileR2L converts the R2L MLP model into a CNN with residual blocks, making the model backbone more efficient and better performing. These two discoveries allowed MobileR2L to set SOTA on the realistic synthetic 360° and real-world forward-facing datasets.

Pruning. Neural network pruning is widely used to reduce the high inference cost of deep neural networks in resource-constrained settings. Generally, the pruning process consists of three stages: (1) training, (2) pruning, and (3) finetuning. Standard pruning criteria include magnitude-based and scaling factor-based methods. During pruning, extra and unnecessary weights are removed following the selected criterion. Removing redundant weights and retaining high-importance weights allow for a reduction in model size while preserving performance. Though the pruned model can suffer from worse performance, this is the typical tradeoff encountered when dealing with efficiency and performance [14].

Deep convolutional neural networks (CNNs) generally have a high computational cost, causing deployment to be an issue for real-world applications. Network slimming is an approach for (1) reducing CNN model size, (2) decreasing run-time memory, and (3) lowering the compute operations, all with minimal effect on accuracy [13].

B. ENeLF

Overview. The experiments conducted follow the training process of R2L [19], using (1) a pre-trained teacher NeRF model [16] and (2) performing data distillation with that teacher model to generate additional data. The efficient CNN backbone and SR modules in MobileR2L [3] are leveraged by ENeLF, significantly reducing the model size and inference speed compared to R2L. Modifications to apply efficiency discoveries from these two papers with channel-wise structure pruning are made to the MobileR2L model architecture, transforming it to the ENeLF architectecture.

Channel-wise Structure Pruning The specific pruning method investigated for CNNs is channel-wise structure pruning, where the convolutional (CONV) layer channels are pruned based on their importance. This approach defines "importance" with the learnable scaling factors parameters in the batch normalization (BN) layers. An L1-norm can be applied to the learnable parameters of BN layers and push them towards 0, enabling the usage of BN scaling factor magnitude to identify the importance of channels. Channels with higher associated scaling factors are more critical and are retained. Channels with smaller associated scaling factors are less important and are pruned. This proposed method requires no special software or hardware accelerator to achieve speedups.

Reorder BN and CONV layers. The previous section mentioned that BN layer scaling factors are directly responsible for determining which channels of the CONV layer to prune. However, MobileR2L placed their BN layers after the CONV layers. In the implementation of ENeLF, the BN layers

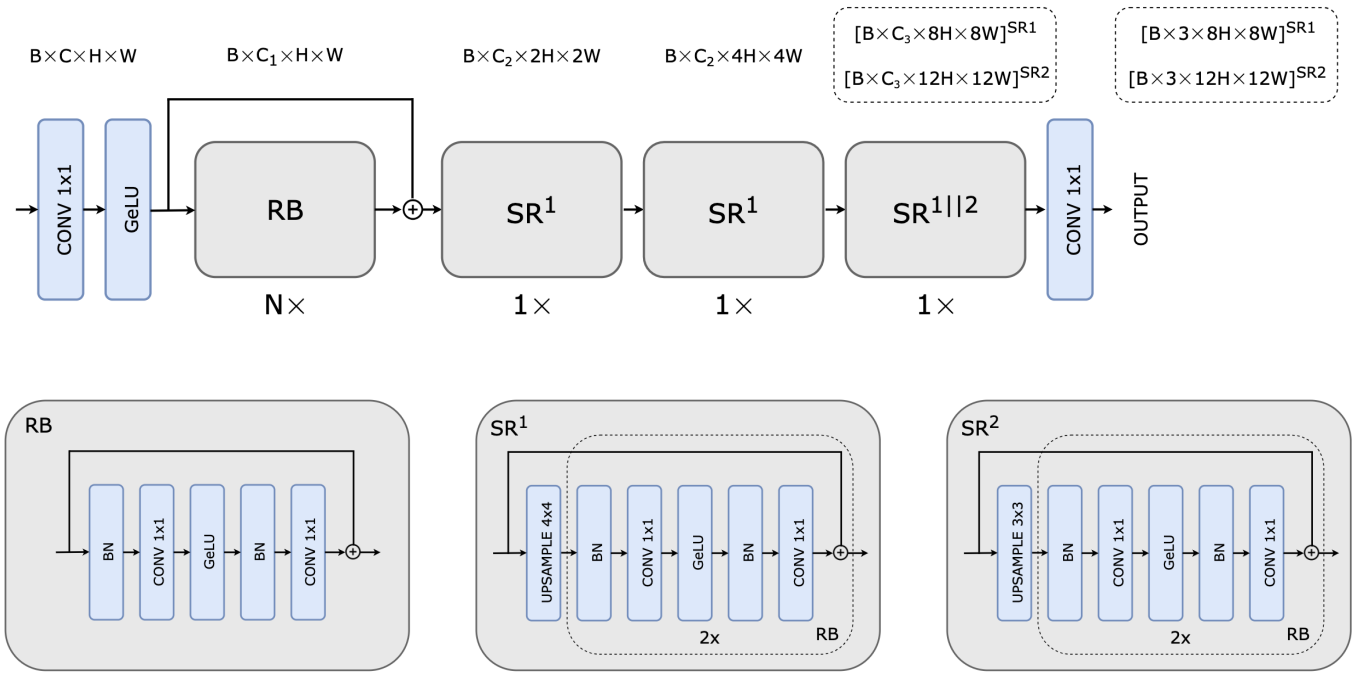

Fig. 2. Modified MobileR2L model architecture to support channel-wise structure pruning.

are placed before the CONV layers. Experimental results on ENeLF showed that the reordering of these layers did not significantly affect model performance.

IV. Experiments

Datasets. Most studies perform experiments on two datasets: realistic synthetic 360° [16] and real-world forward-facing [15]. The realistic synthetic 360° dataset contains 8-path traced scenes, including 100 images for training and 200 for testing for each scene. Due to GPU constraints, experiments are performed on select scenes (e.g. chair, drums, ficus, lego) from the realistic synthetic 360° dataset. Following research conducted in MobileR2L [3], the experiments on ENeLF for realistic synthetic 360° will also be conducted on the original resolution of the dataset (800 × 800).

Implementation. Leveraging the best-performing architecture from MobileR2L, D60-SR3 (60 layers, 3 SR blocks), the experimental procedure is as follows: (1) train the teacher NeRF model on select scenes from realistic synthetic 360°, (2) distill 5K examples per scene (50% less compared to MobileR2L), (3) modify MobileR2L to a channel-wise pruning friendly architecture ENeLF, (4) train ENeLF for 200K iterations (33% less compared to MobileR2L) with batch size of 10, (5) perform channel-wise structure pruning, and (6) retrain ENeLF for 60K iterations. Significant dataset and training time reductions are made due to academic GPU and storage space constraints.

The ENeLF model accepts an input of size 100×100 for the realistic synthetic 360° dataset and upsamples it by 8× to render an 800×800 image. For the real-world forward-facing dataset, an input of size 84×63 is upsampled 12× to render a 1008×756 image. Note that CONV layer kernel size and padding are adjusted in the final SR block to achieve the desired 8× and 12× upsampling. Realistic synthetic 360° uses 3 SR^1 to achieve 8× upsampling while real-world forward-facing uses 2 SR^1 and 1 SR^2 to achieve 12× upsampling.

Model Performance. Due to the many adaptations made to the experimental procedure in MobileR2L for ENeLF, an attempt is made to reproduce the work of MobileR2L with reduced scenes, 50% distilled data, 33% less training epochs, and smaller batch size. The reasoning is to ensure fair and comparable results despite different environments. To understand the rendering quality of MobileR2L and ENeLF, three standard metrics are reported (PSNR [10], SSIM [20], and LPIPS [24]) on realistic synthetic 360° dataset.

TABLE I
Realistic Synthetic 360°

Pruning Ratio	PSNR ↑	SSIM ↑	LPIPS ↓
0% - MobileR2L [3]	**31.34**	**0.993**	0.051
0% - Mobile R2L Reproduced	29.54	**0.993**	**0.048**
30%	27.79	0.990	0.0738
50%	26.71	0.989	0.094
70%	24.81	0.984	0.137

Model parameters, FLOPs, size (MB), and latency are often used when evaluating efficiency. Although ENeLF achieved worse results on PSNR, SSIM, and LPIPS (see Table I), it achieved better results on all efficiency metrics (see Table II). Latency measurements are profiled with an iPhone 15 (iOS 17) using CoreMLTools [1].

As for qualitative results (see Fig. 3), the pruned models

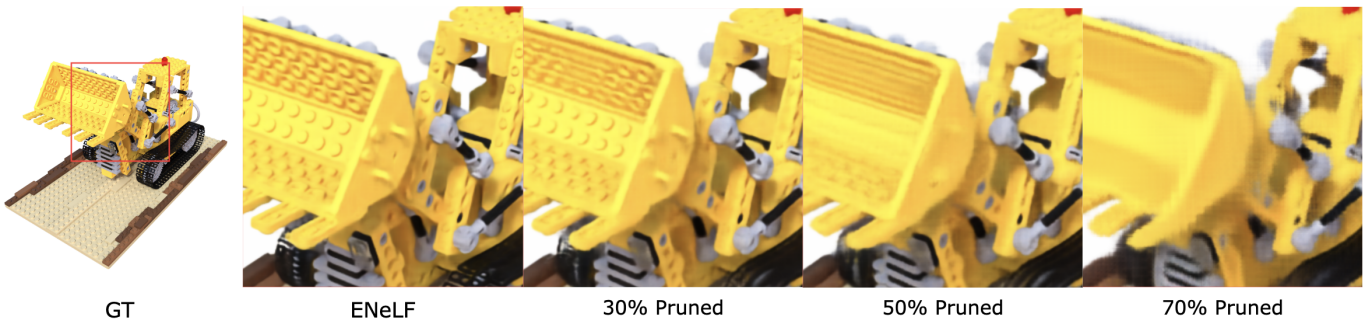

Fig. 3. Qualitative visual comparison between ENeLF and ground truth on the realistic synthetic 360° lego scene of size 800×800. It is best viewed in color.

TABLE II
REALISTIC SYNTHETIC 360°

Pruning Ratio	# Params ↓	FLOPs ↓	Size ↓	Latency ↓
0% - MobileR2L [3]	4.3 M	148.52 G	8.3 MB	**22.65 ms**
0% - Reproduced	4.3 M	148.52 G	8.8 MB	175.06 ms
30%	3.11 M	124.72 G	6.3 MB	142.62 ms
50%	2.29 M	108.52 G	4.8 MB	134.39 ms
70%	**1.55 M**	**93.8 G**	**3.3 MB**	116.37 ms

all lose some capability to render fine-grained details in the resulting views. This is likely due to the reduced parameters in the ENeLF model, which reduces the model's capability to represent the light field space accurately.

V. LIMITATIONS

The main limitation of this work is training time. Following the procedure from R2L [19], 10k additional images were distilled by the teacher NeRF model to train ENeLF. The additional training examples were increased ∼10× compared to the original dataset (up to 100 examples), resulting in longer training times than the NeRF approaches. There are existing approaches that apply pruning at different stages of training to identify winning tickets (small but critical subnetworks) for dense, randomly initialized deep networks [6]. However, identifying winning tickets still requires a costly train, prune, and retrain process. A newer work, Early-Bird Ticket (EBT) [22], offers a more efficient approach to finding "winning tickets" by determining an early stopping point for initial training with pruned mask distance. The authors of EBT also found that EBTs consistently emerge in ResNet-like models at early epoch ranges, even when training at a lower precision.

Another limitation is using a widely shared high-traffic, high-performance computing cluster at Georgia Institute of Technology. GPU availability is scarce, and training time for each ENeLF model is high (i.e., 16 GPU hours on H100 for only 200k iterations). Consequently, the results are not as flushed out as they could be. However, these results show significant improvements in model efficiency with minor performance degradations, which could be improved with additional training.

Finally, ENeLF fails to generate high-frequency details in the image, a similar issue to MobileR2L. This is likely due to the reduced representation a smaller pruned model can capture. A larger model may reduce this problem but will impact latency and deployment to mobile devices.

Future research should focus on further compression of ENeLF (e.g. quantization-aware training with static quantization or cyclic precision training [7]) and improving the pruning process (train, prune, retrain) to help boost the performance and efficiency of the current model.

VI. CONCLUSION

This work presents ENeLF, the first instance of applying compression techniques to a NeLF network that renders images with slight performance degradations compared to MobileR2L [3] and NeRF [16] while running on model devices. Experiments are performed to prune and compress an optimized network architecture trained using data distillation to render high-resolution images.

REFERENCES

[1] Apple. (2017, September 1). Core ML Tools. Core ML Tools - Guide to Core ML Tools. https://apple.github.io/coremltools/docs-guides/
[2] Attal, Benjamin, et al. "Learning neural light fields with ray-space embedding." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022.
[3] Cao, Junli, et al. "Real-time neural light field on mobile devices." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023.
[4] Chen, Zhiqin, et al. "Mobilenerf: Exploiting the polygon rasterization pipeline for efficient neural field rendering on mobile architectures." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023.
[5] Feng, Brandon Yushan, and Amitabh Varshney. "Signet: Efficient neural representation for light fields." Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021.
[6] Frankle, Jonathan, and Michael Carbin. "The lottery ticket hypothesis: Finding sparse, trainable neural networks." arXiv preprint arXiv:1803.03635 (2018).
[7] Fu, Yonggan, et al. "Cpt: Efficient deep neural network training via cyclic precision." arXiv preprint arXiv:2101.09868 (2021).
[8] Gupta, Aarush, et al. "LightSpeed: Light and Fast Neural Light Fields on Mobile Devices." Advances in Neural Information Processing Systems 36 (2024).
[9] Hedman, Peter, et al. "Baking neural radiance fields for real-time view synthesis." Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021.
[10] Huynh-Thu, Quan, and Mohammed Ghanbari. "Scope of validity of PSNR in image/video quality assessment." Electronics letters 44.13 (2008): 800-801.

[11] Li, Zhong, et al. "Neulf: Efficient novel view synthesis with neural 4d light field." arXiv preprint arXiv:2105.07112 (2021).
[12] Liu, Lingjie, et al. "Neural sparse voxel fields." Advances in Neural Information Processing Systems 33 (2020): 15651-15663.
[13] Liu, Zhuang, et al. "Learning efficient convolutional networks through network slimming." Proceedings of the IEEE international conference on computer vision. 2017.
[14] Liu, Zhuang, et al. "Rethinking the value of network pruning." arXiv preprint arXiv:1810.05270 (2018).
[15] Mildenhall, Ben, et al. "Local light field fusion: Practical view synthesis with prescriptive sampling guidelines." ACM Transactions on Graphics (TOG) 38.4 (2019): 1-14.
[16] Mildenhall, Ben, et al. "Nerf: Representing scenes as neural radiance fields for view synthesis." Communications of the ACM 65.1 (2021): 99-106.
[17] Reiser, Christian, et al. "Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps." Proceedings of the IEEE/CVF international conference on computer vision. 2021.
[18] Sitzmann, Vincent, et al. "Light field networks: Neural scene representations with single-evaluation rendering." Advances in Neural Information Processing Systems 34 (2021): 19313-19325.
[19] Wang, Huan, et al. "R2l: Distilling neural radiance field to neural light field for efficient novel view synthesis." European Conference on Computer Vision. Cham: Springer Nature Switzerland, 2022.
[20] Wang, Zhou, et al. "Image quality assessment: from error visibility to structural similarity." IEEE transactions on image processing 13.4 (2004): 600-612.
[21] Yariv, Lior, et al. "Bakedsdf: Meshing neural sdfs for real-time view synthesis." ACM SIGGRAPH 2023 Conference Proceedings. 2023.
[22] You, Haoran, et al. "Drawing early-bird tickets: Towards more efficient training of deep networks." arXiv preprint arXiv:1909.11957 (2019).
[23] Yu, Alex, et al. "Plenoctrees for real-time rendering of neural radiance fields." Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021.
[24] Zhang, Richard, et al. "The unreasonable effectiveness of deep features as a perceptual metric." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.