R2L: Distilling Neural Radiance Field to Neural Light Field for Efficient Novel View Synthesis

Huan Wang[1,2,*], Jian Ren[1,†], Zeng Huang[1,‡], Kyle Olszewski[1], Menglei Chai[1], Yun Fu[2], Sergey Tulyakov[1]

[1]Snap Inc. [2]Northeastern University, USA

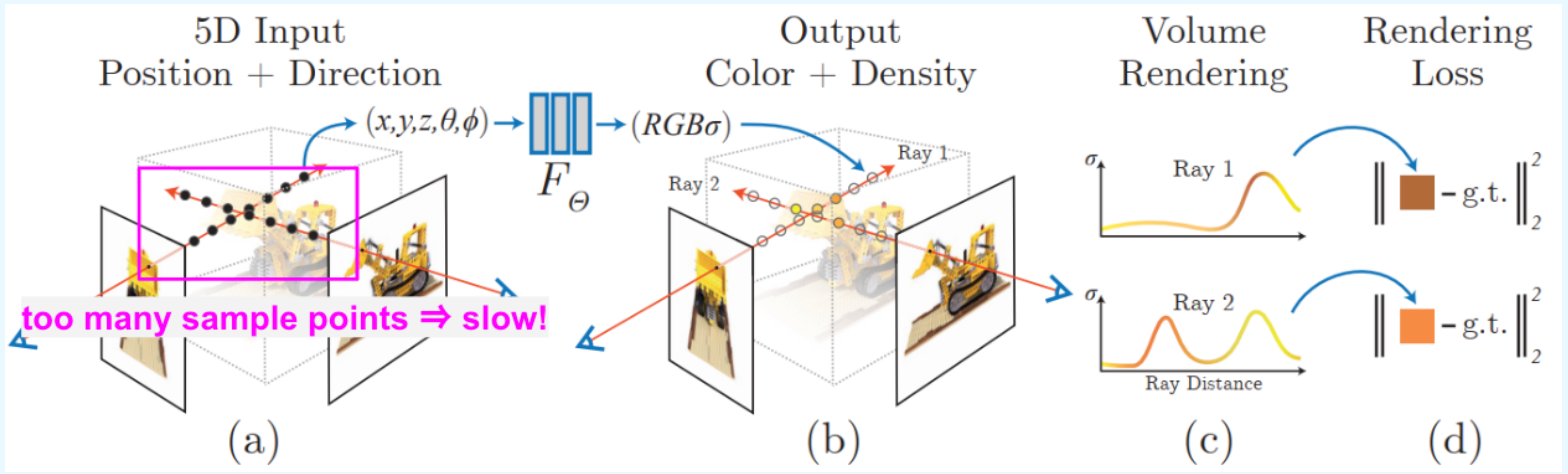
[*]Work done when Huan was an Intern at Snap [†]Corresponding author: jren@snapchat.com [‡]Now at Google

Snap Inc.

SMILE Lab.

Motivation: Towards Faster NeRF Rendering

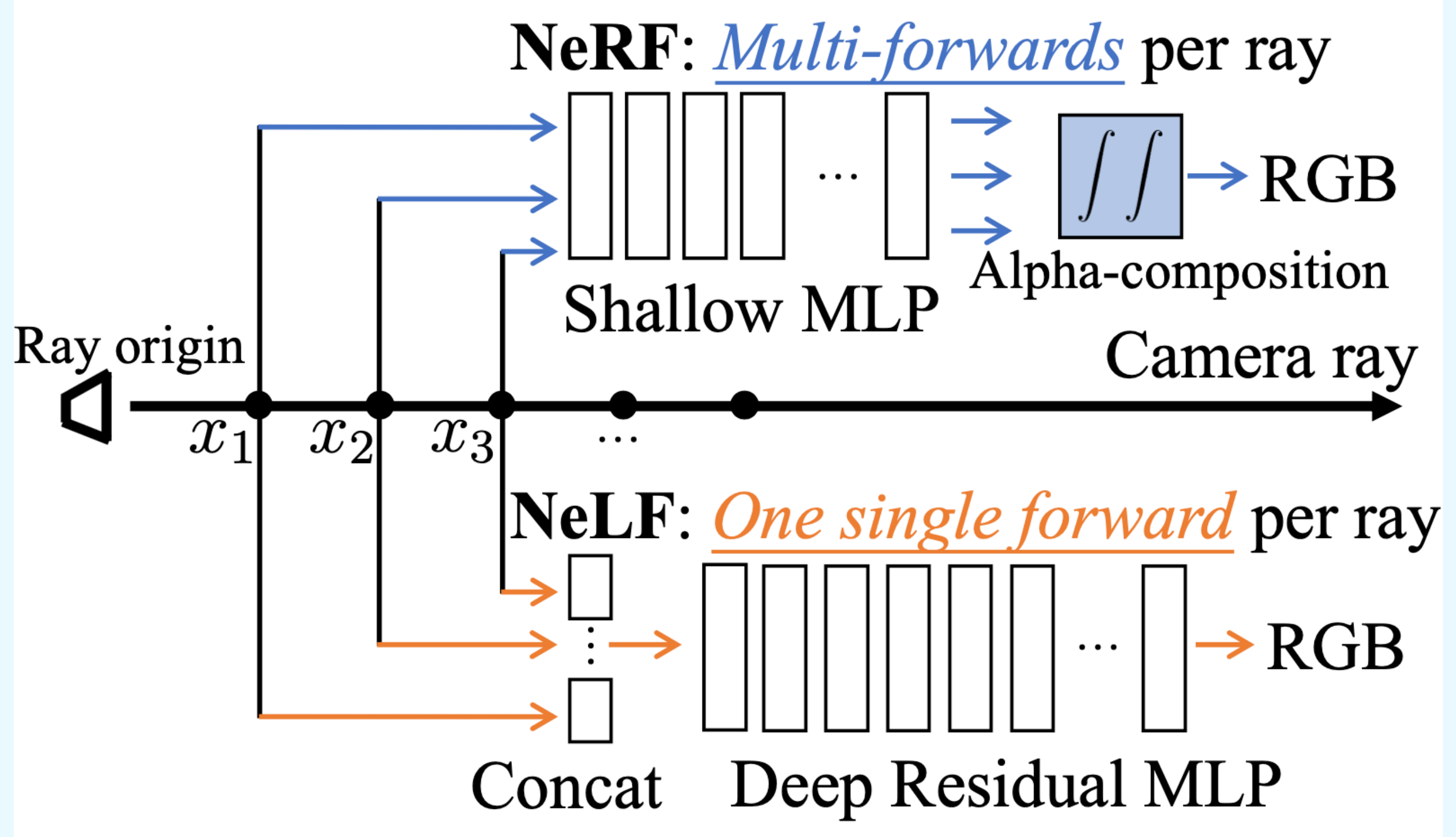
- NeRF (Neural Radiance Field) [Mildenhall et al., ECCV, 2020] opens the new doors of representing scenes with a simple MLP network. Rendering one pixel takes **hundreds of queries** of the MLP network, making **NeRF prohibitively slow in inference**: rendering a 400*400 image with PyTorch on a NVIDIA V100 GPU takes 6.7s.
- Primary cause: NeRF samples too many points along a ray in rendering.



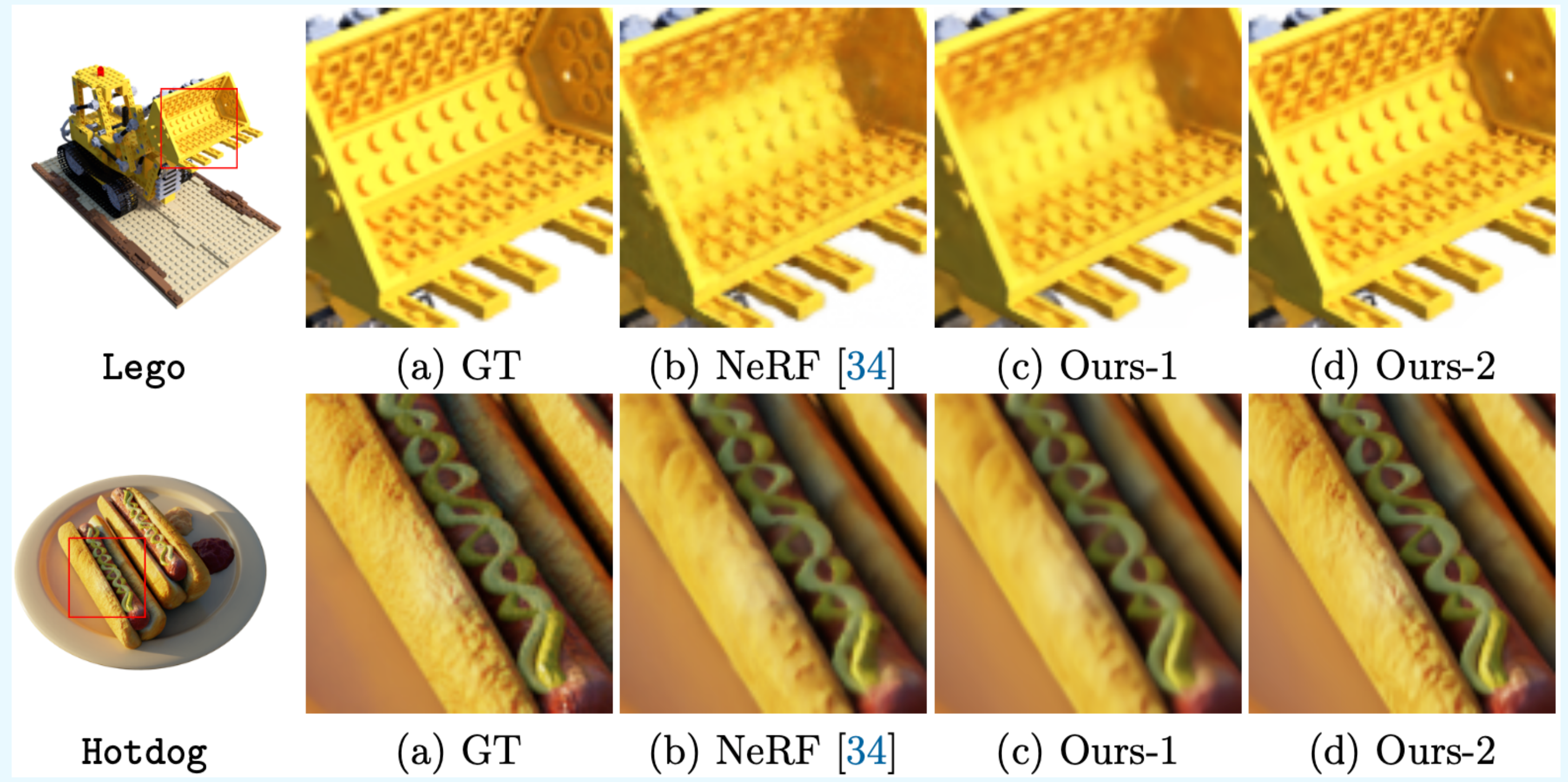
NeRF illustration. Src: [Mildenhall et al., ECCV, 2020], edited

Proposed Method: R2L

- Simply put, the presented method is to convert the scene representation from NeRF (neural radiance field) to NeLF (neural light field), hence the method name R2L.
- What's good with NeLF? Rendering one pixel in NeLF only needs **one** network query vs. **hundreds** of query in NeRF -- **much faster!**

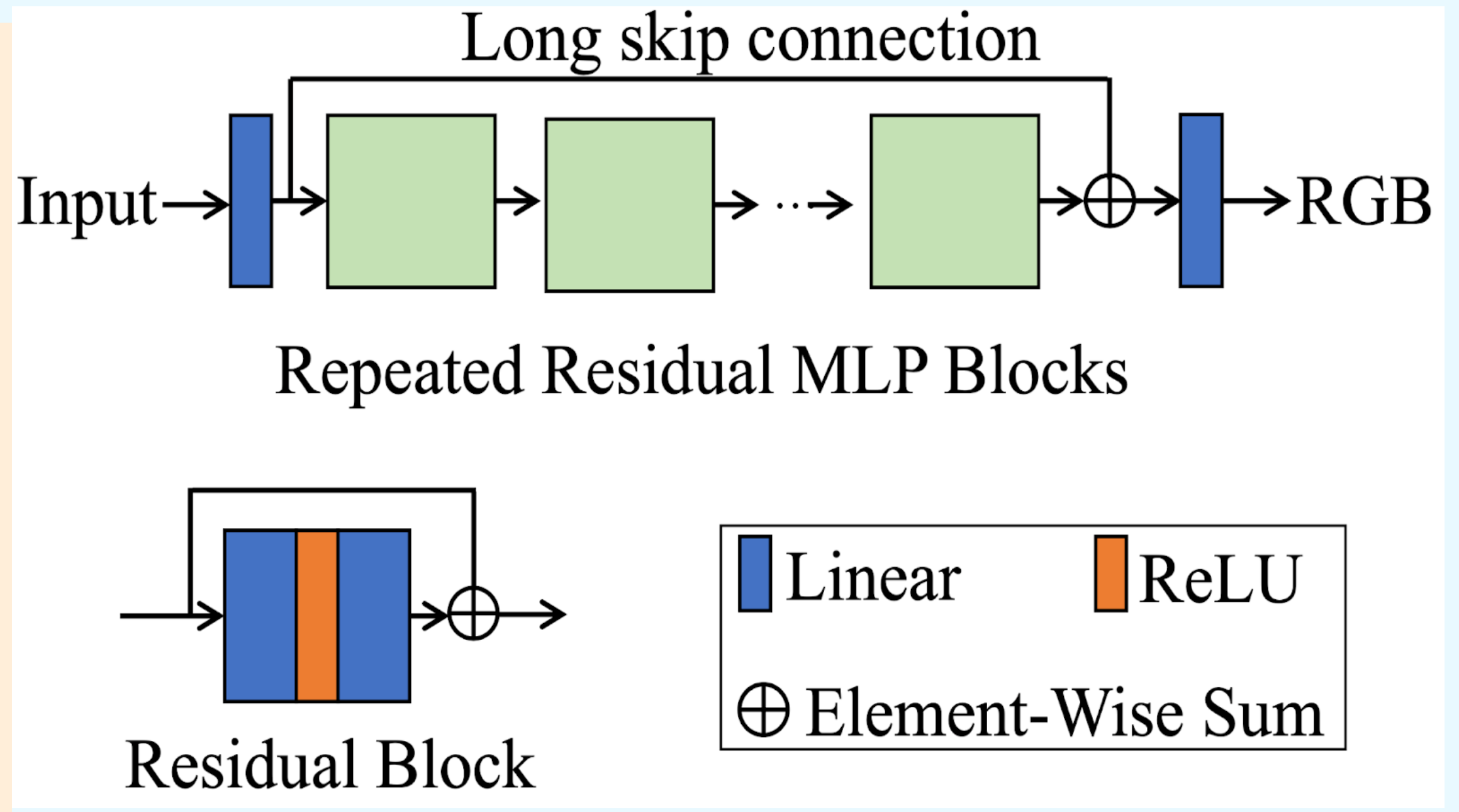


(a) NeRF (top) *vs.* NeLF (ours, bottom)

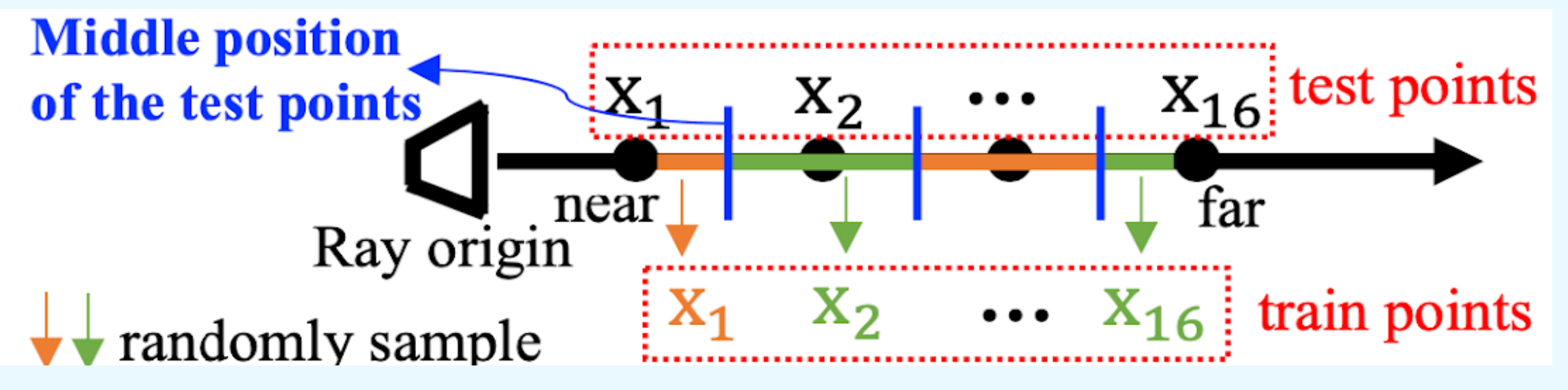


To enable NeLF, we need to resolve **three key problems**:

- NeLF is harder to represent than NeRF. **Solution**: We propose a deeper (88 layers) ResMLP architecture
- How to train the deeper model? **Solution**: Employ a pretrained NeRF to synthesize abundant pseudo data.
- How to represent a ray? **Solution**: Concatenating sampled points of the ray.



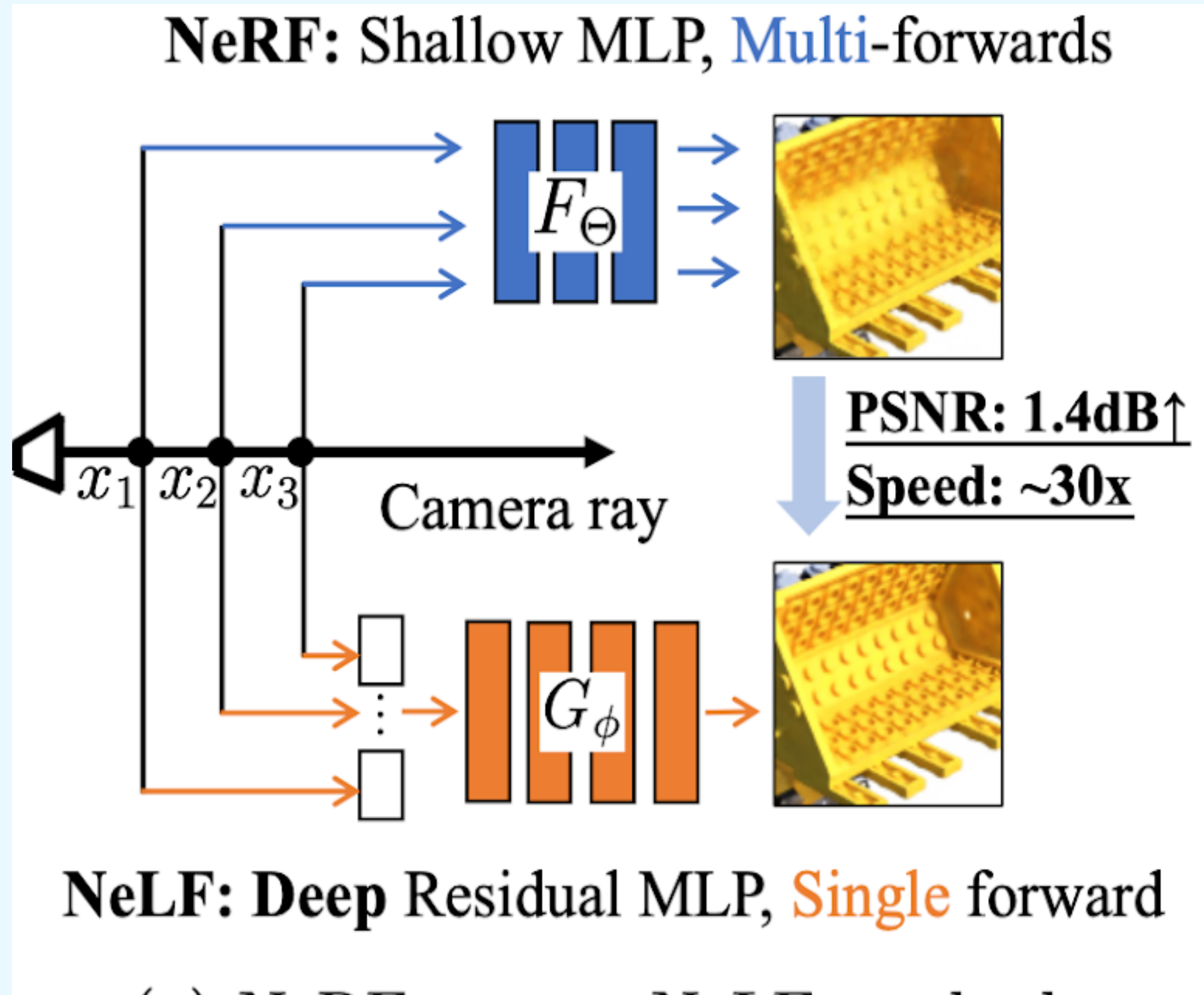
Our ResMLP architecture


Middle position of the test points
x1 x2 ... x16 test points
near
far
Ray origin
randomly sample
x1 x2 ... x16 train points

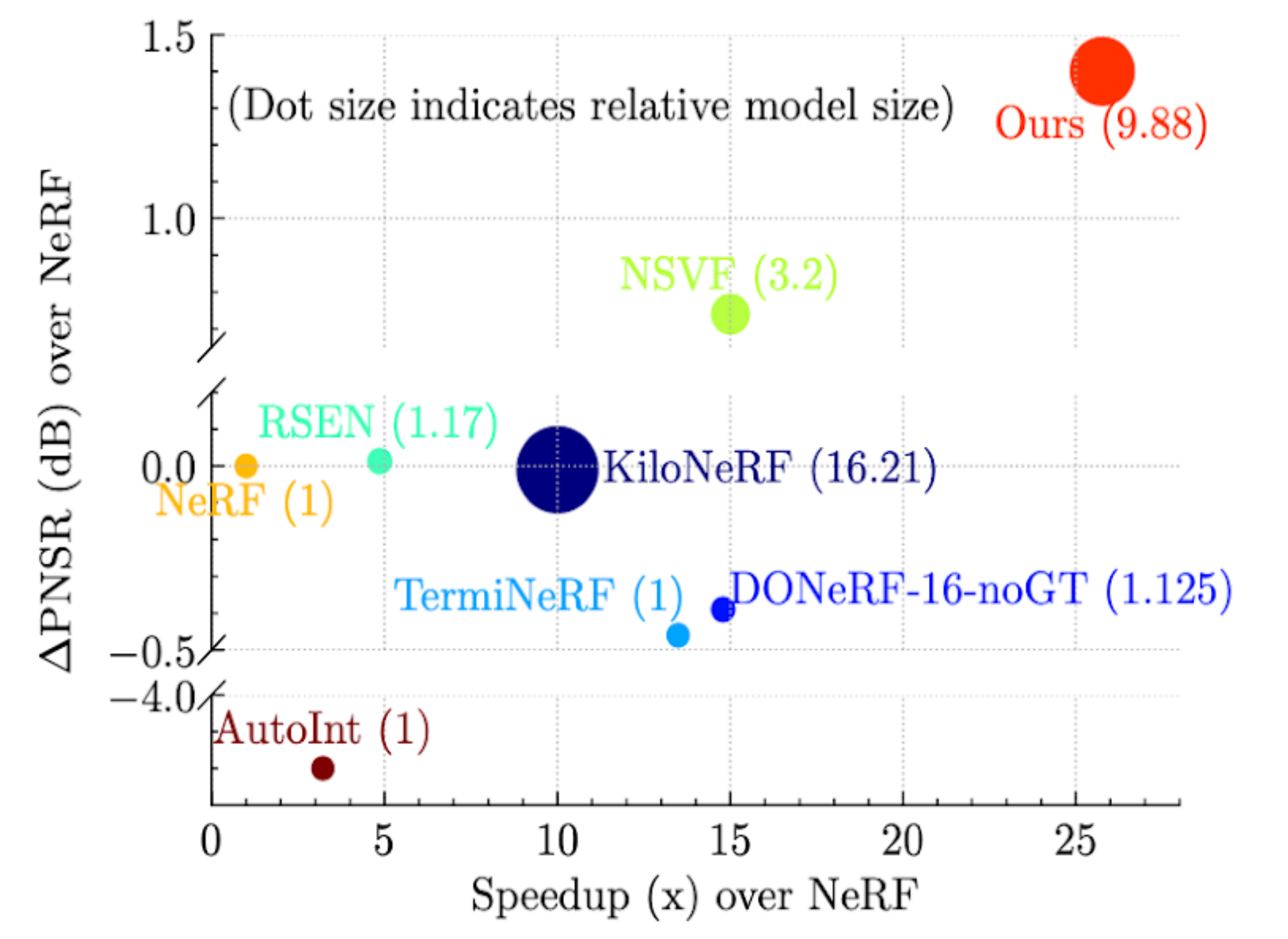
Point sampling in our method

Experimental Results:

- On the NeRF synthetic dataset (400x400 resolution), R2L achieves around **30x faster** than NeRF while enjoying **1.4dB PSNR boost,** reporting (one of) the SOTA speedup-PSNR-model size tradeoff.
- On the NeRF realistic dataset, we maintain the PSNR with only **1/26 FLOPs**. (Please refer to our paper for more results.)


NeRF: Shallow MLP, Multi-forwards
F_Θ
PSNR: 1.4dB↑
Speed: ~30x
x1 x2 x3
Camera ray
G_ϕ
NeLF: Deep Residual MLP, Single forward

(a) NeRF *vs.* our NeLF method



(b) Speedup-PSNR-Model Size comparison

Code & trained models are released at: https://github.com/snap-research/R2L

SCAN ME