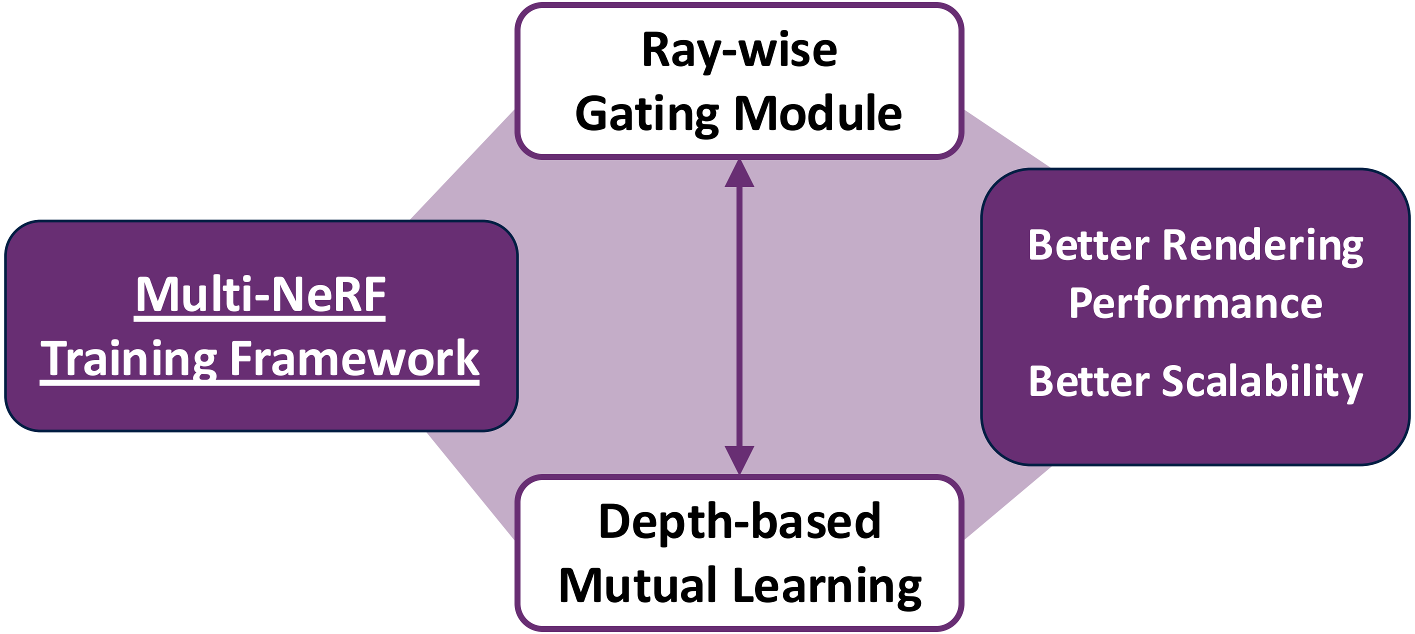
Rad-NeRF: Ray-decoupled Training of Neural Radiance Field

Lidong Guo[1*], Xuefei Ning[1*], Yonggan Fu[2], Tianchen Zhao[3], Zhuoliang Kang[3], Jincheng Yu[1], Yingyan (Celine) Lin[2], Yu Wang[1]

[1]Tsinghua University, [2]Georgia Institute of Technology, [3]Meituan
E-mail: gld21@mails.tsinghua.edu.cn, foxdoraame@gmail.com, yu-wang@tsinghua.edu.cn

Overview

Breaking the limitation of NeRF Scalability in Complex Scenes

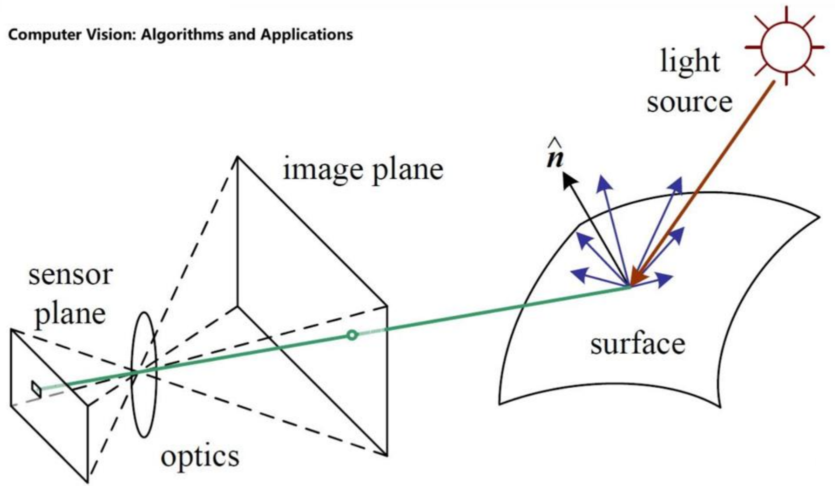


Rendering

- **Render：**

Transform a scene representation (camera, light, surface geometry, etc.) into static images

“Taking pictures with a computer-simulated camera, provided that a 3D representation of the scene already exists”

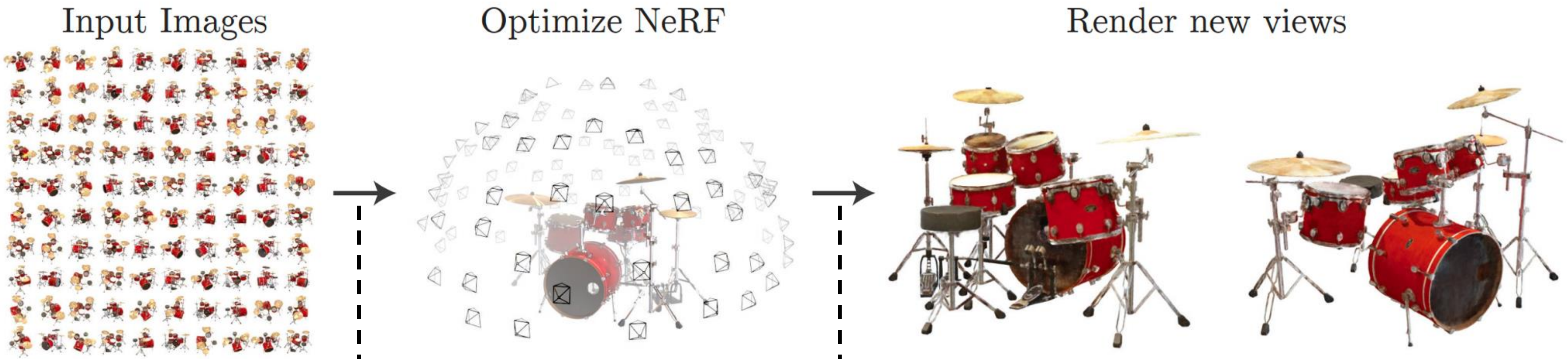


The physical process of taking pictures

Novel View Synthesis

➢ Rendering Task: Novel View Synthesis

Given the source image and source pose, as well as the **unseen target pose**, render and generate the image corresponding to the target pose.



Scene Representation

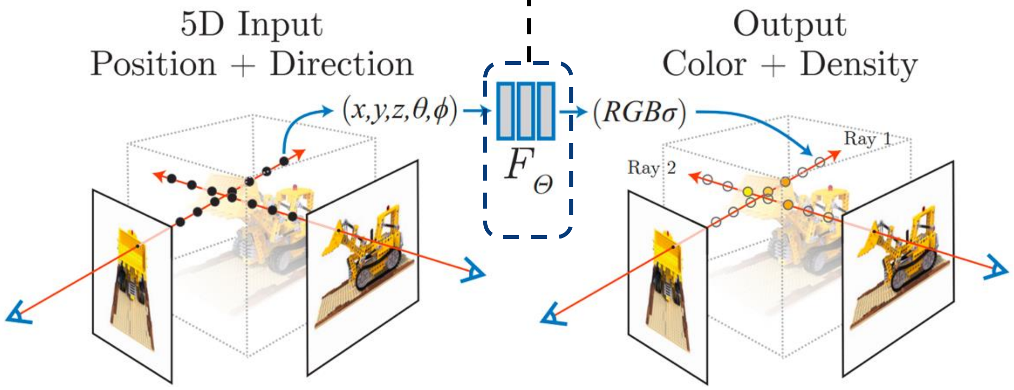
Novel View Rendering

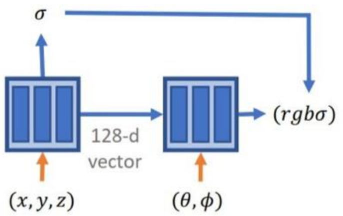
Neural Radiance Field

➢ Neural Network is adopted to fit scene representation

- A ray is made up of infinite number of 3D points in target scene
- The NN <u>encodes</u> the 3-d coordination and 2-d direction, <u>outputs</u> the color/density of each point

MLP Network



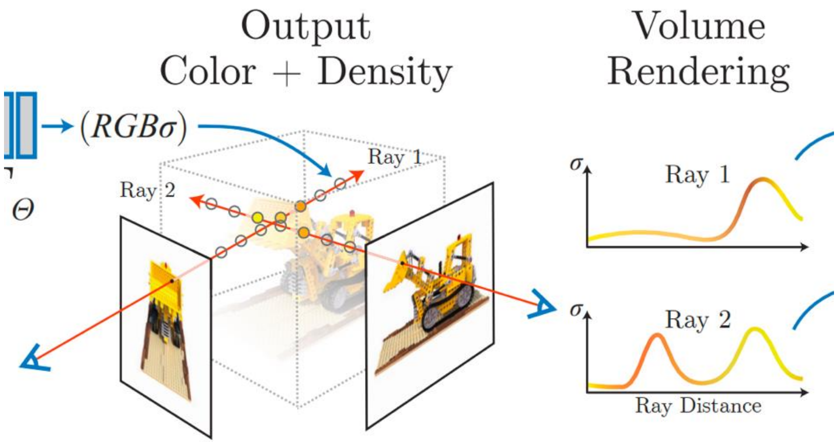


The color is relative to both <u>xyz coordinate</u> and <u>view direction</u>

Neural Radiance Field

➢ Neural Network is adopted to fit scene representation

- A ray is made up of infinite number of 3D points in target scene
- The NN encodes the 3-d coordination and 2-d direction, outputs the color/density of each point
- Integrate the information of each point on the ray to obtain Pixel (Ray) color **(Volume Rendering)**



(b) (c)

$$C(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t), \mathbf{d})dt\,,$$

$$T(t) = \exp\left(-\int_{t_n}^{t} \sigma(\mathbf{r}(s))ds\right)$$ Transmittance

Convert the integral process to discrete summation

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} T_i(1 - \exp(-\sigma_i\delta_i))\mathbf{c}_i\,,$$ Sampling Interval

$$T_i = \exp\left(-\sum_{j=1}^{i-1} \sigma_j\delta_j\right)$$

Challenges exist on NeRF Training

NeRF still exhibits rendering defects on complex scenes

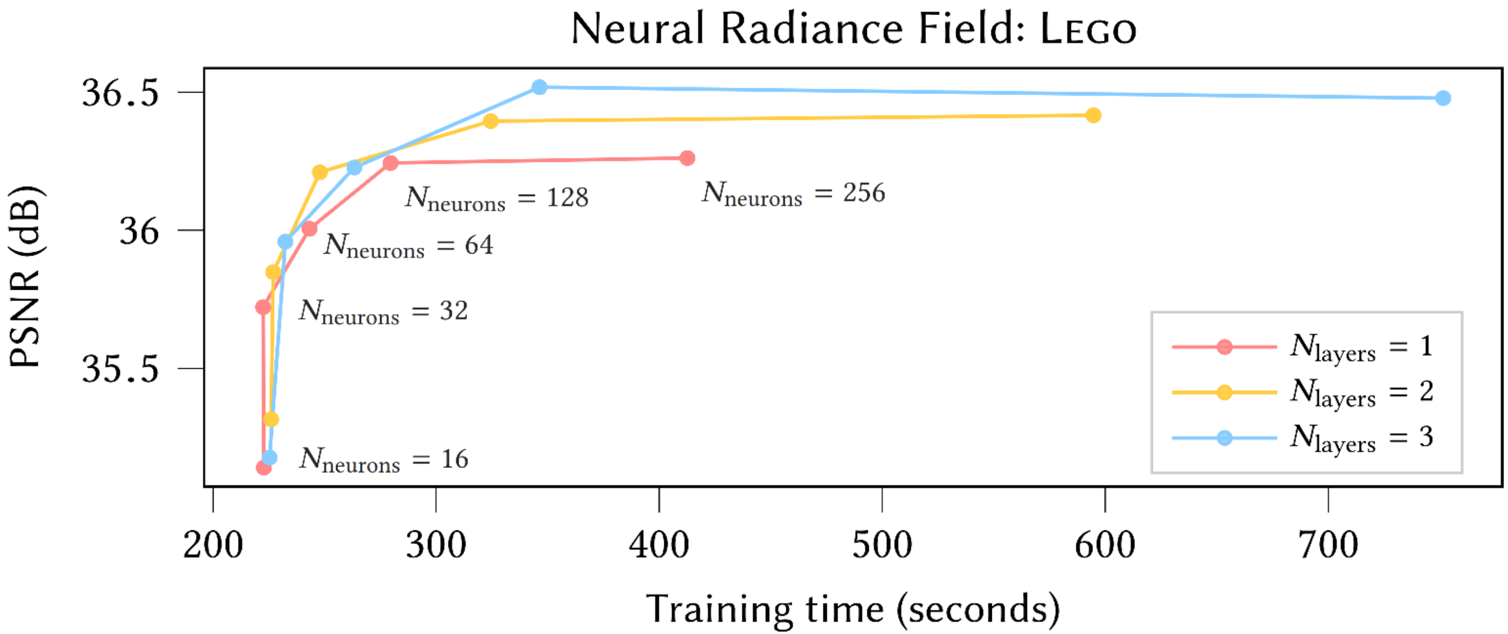
Outdoor free scene

Large indoor scene

Challenges exist on NeRF Training

➢ Challenge-1: Limitation of NeRF Model Capacity

Directly increasing the network's size (width/depth) yields marginal performance improvement

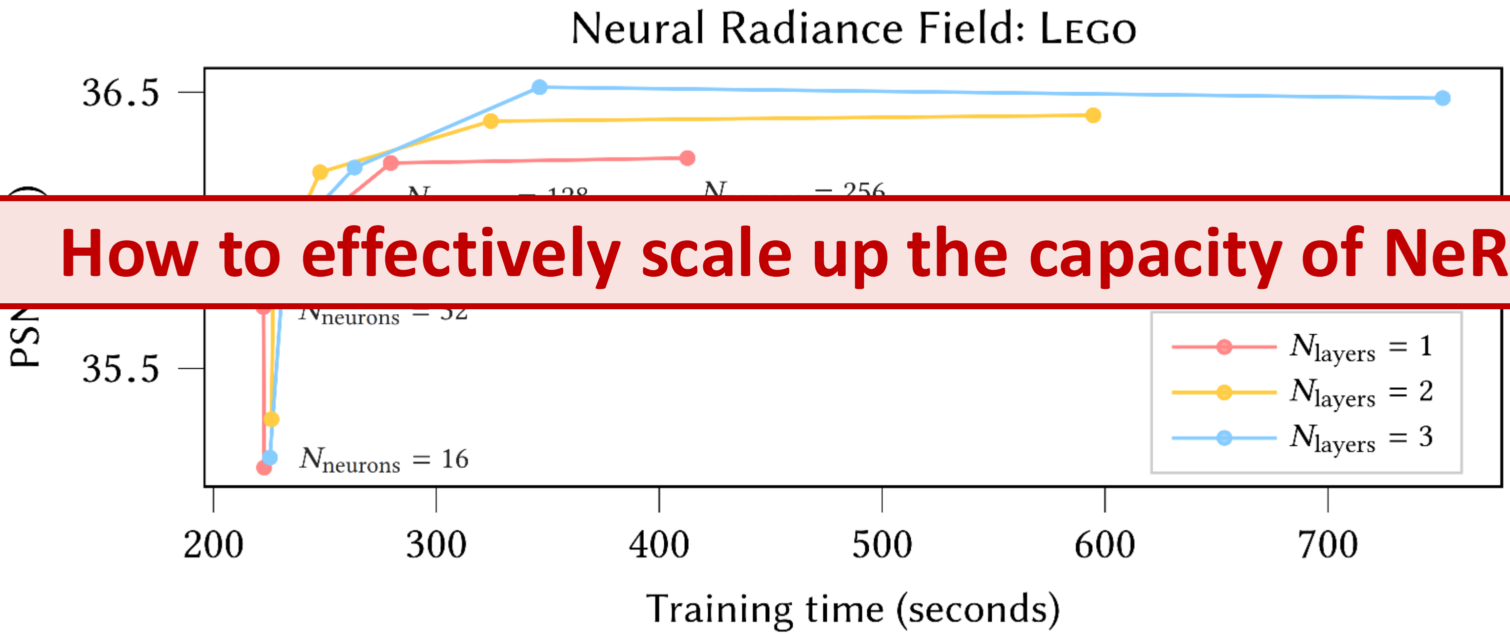


Impact of MLP size on rendering accuracy

Challenges exist on NeRF Training

➢ Challenge-1: Limitation of NeRF Model Capacity

Directly increasing the network's size (width/depth) yields marginal performance improvement



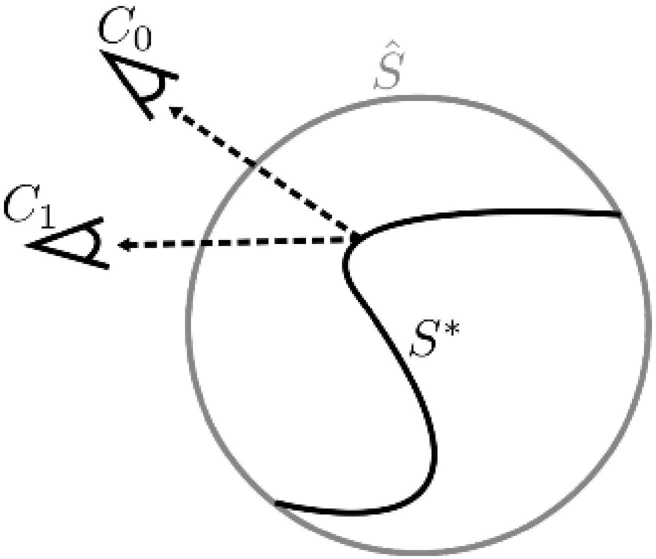
How to effectively scale up the capacity of NeRF ?

Impact of MLP size on rendering accuracy

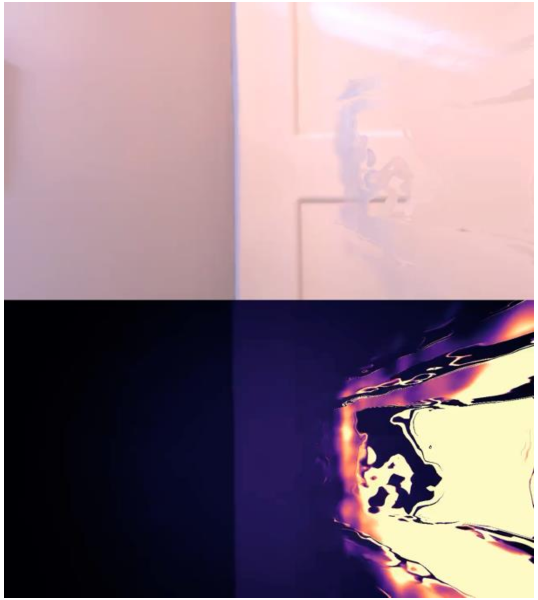
Challenges exist on NeRF Training

➢ Challenge-2: Low Accuracy of Geometric Modeling

The quality of geometric modeling exhibits a significant influence on NeRF's generalization

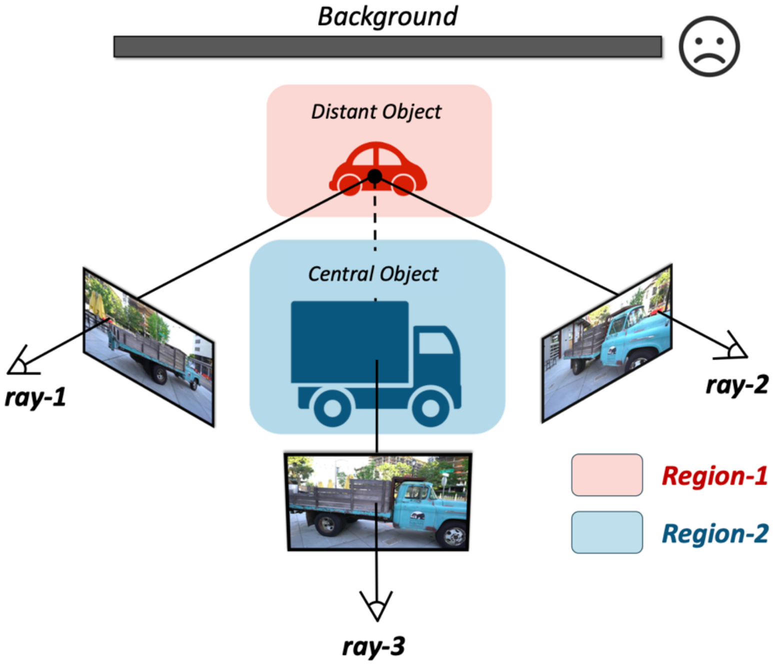


Shape-Radiance Ambiguity

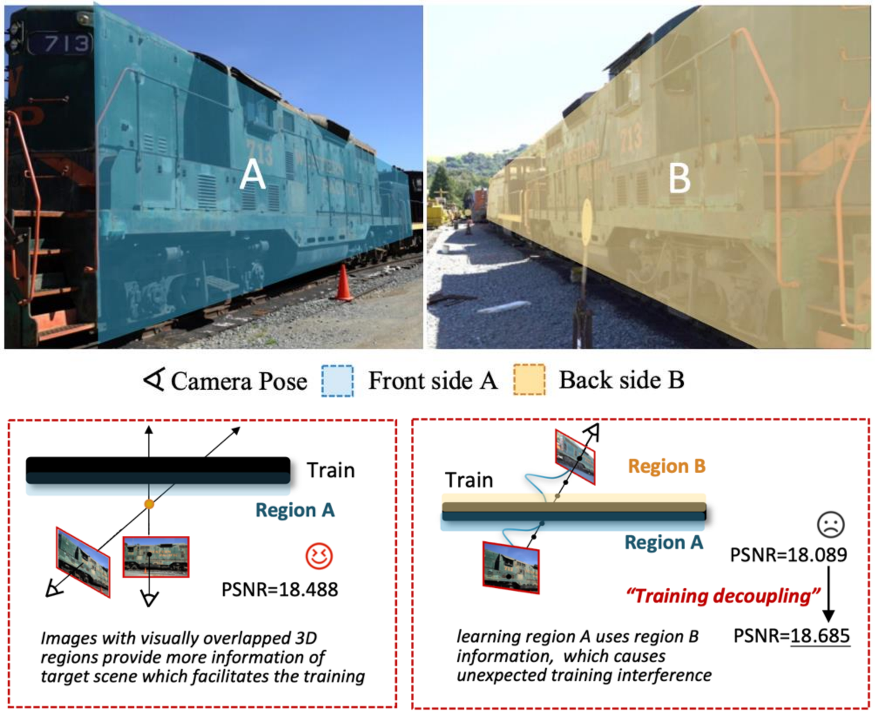


NeRF's Training interference

Ray-3 does not contain valid information about distant object.



The NeRF trained with more invisible rays (two sides of train) performs worse.

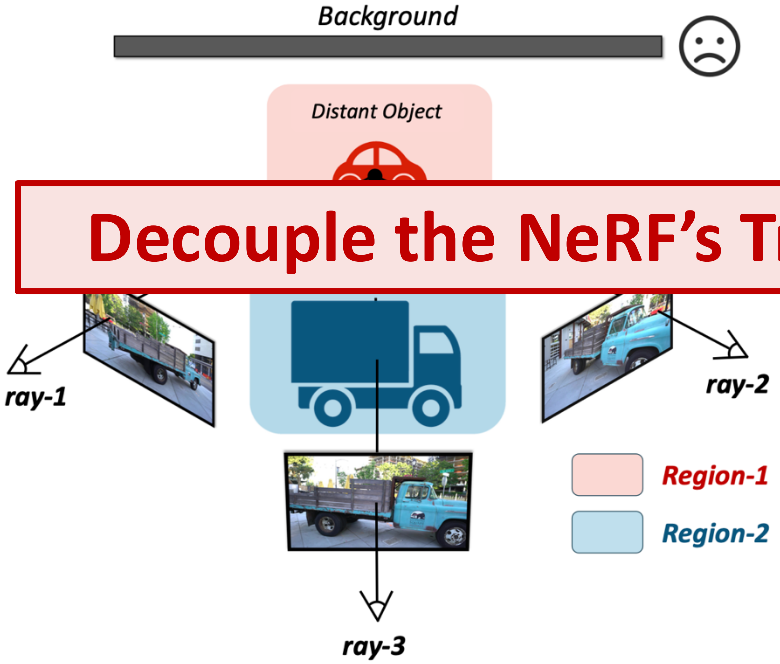


Without training interference

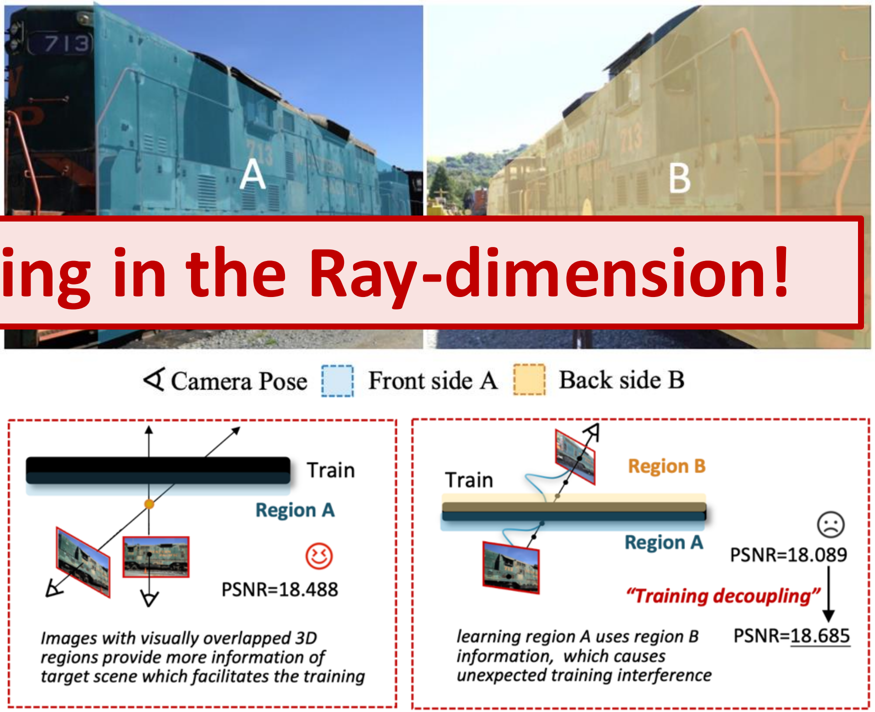
Decoupling mitigates training interference

NeRF's Training interference

Ray-3 does not contain valid information about distant object.



The NeRF trained with more invisible rays (two sides of train) performs worse.



Without training interference

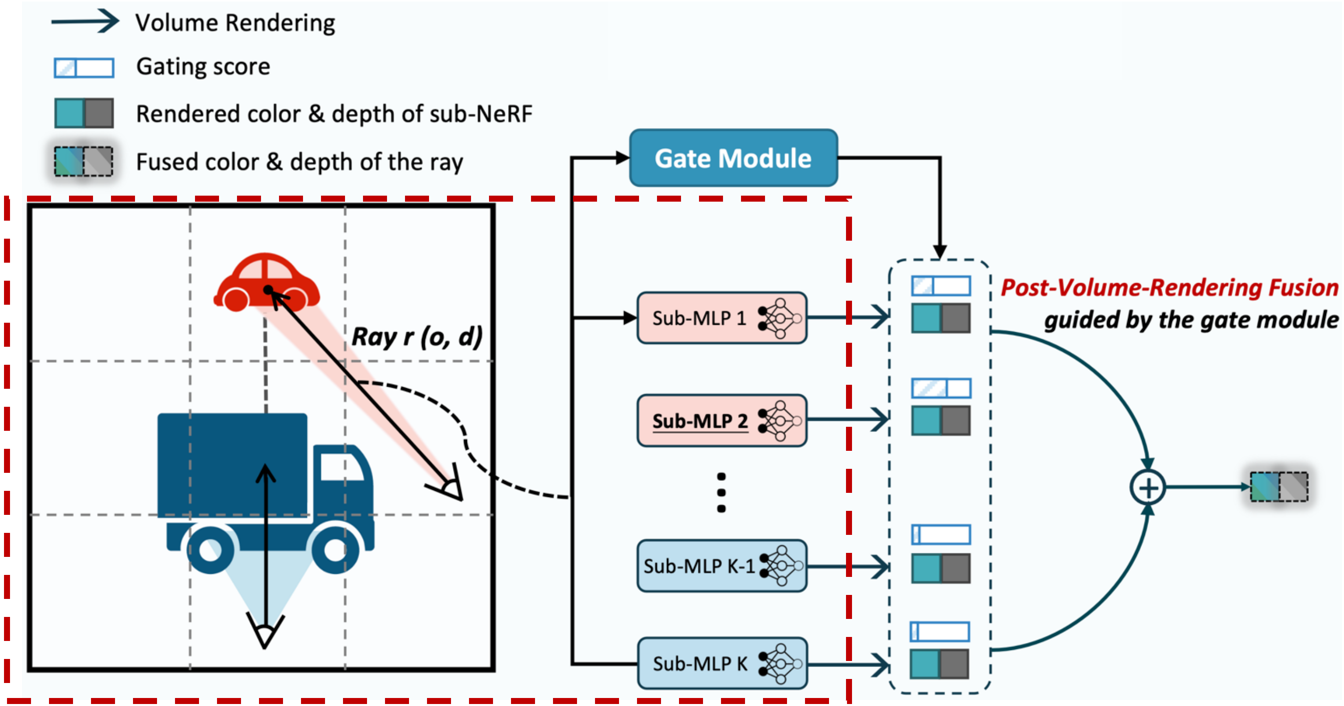
Decoupling mitigates training interference

Decouple the NeRF's Training in the Ray-dimension!

Gate-guided Multi-NeRF Fusion

➢ Multi-NeRF Structure based on Hybrid Representation

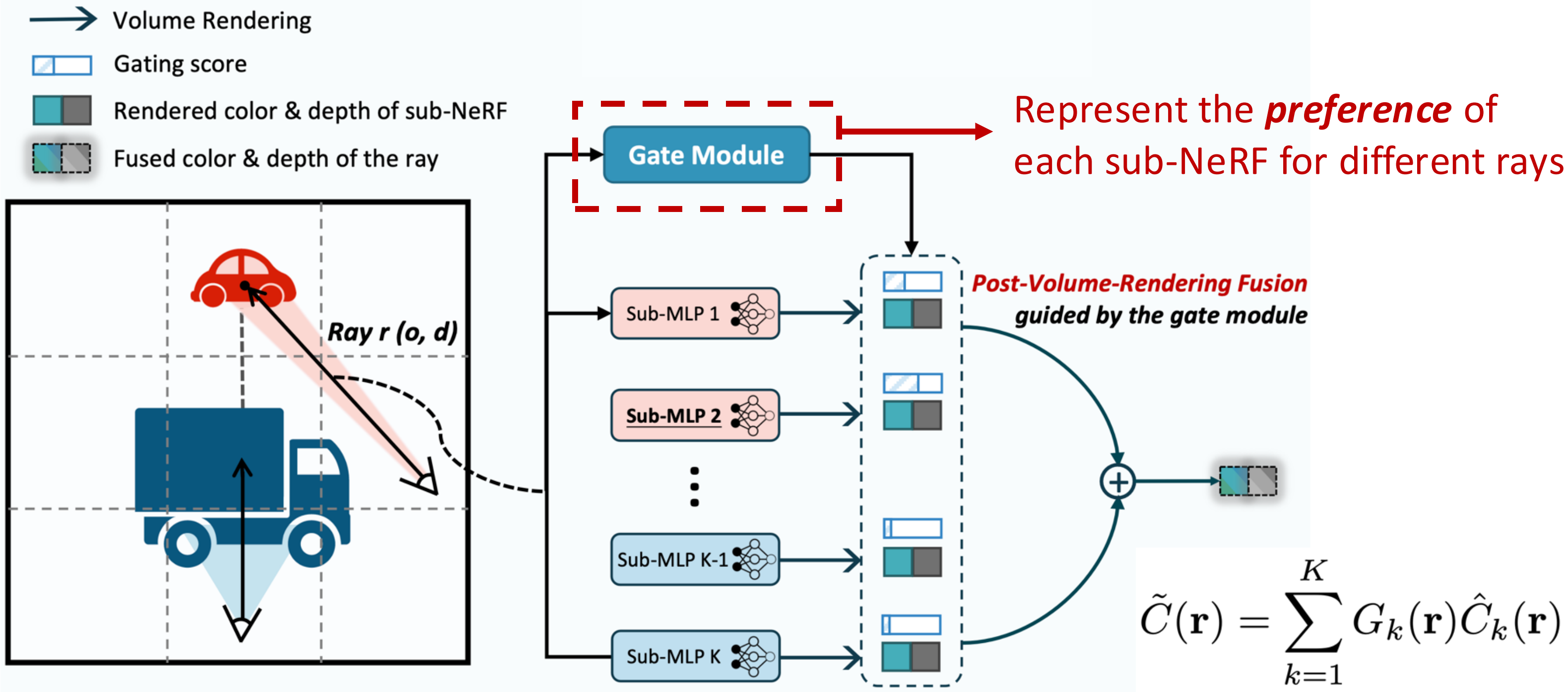
- The feature grid is shared for all sub-NeRFs and the MLP decoders are independent



Gate-guided Multi-NeRF Fusion

➢ Ray-wise Soft Gating Module Design

- A soft gating module is adopted to assign gating scores to the sub-NeRFs for each ray.

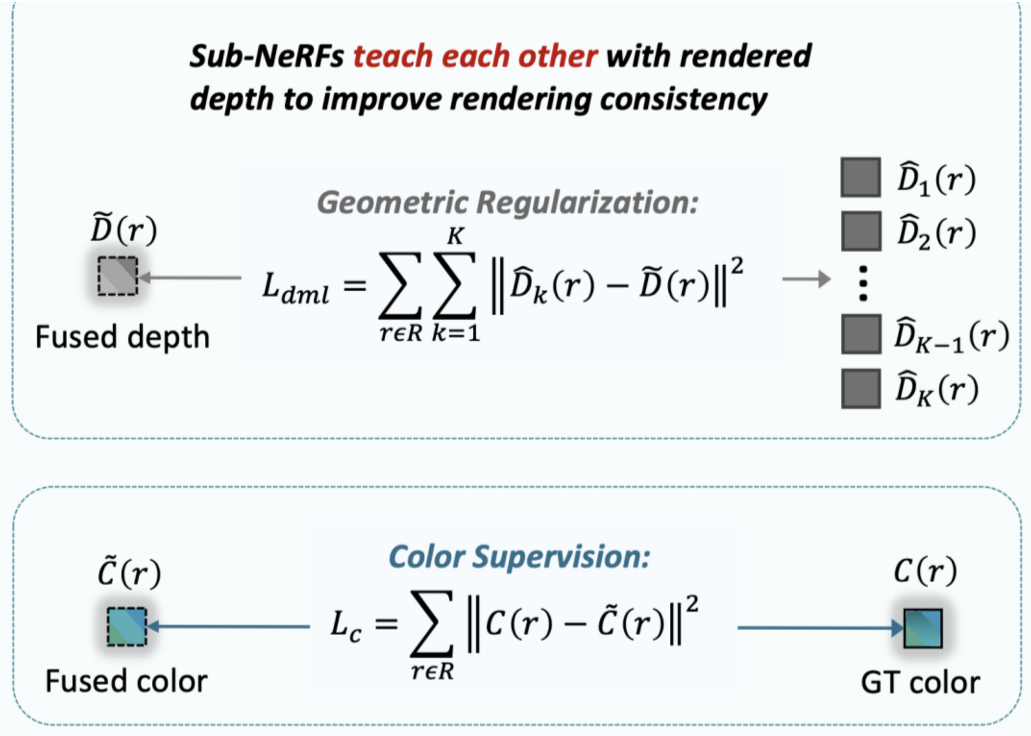


$$\tilde{C}(\mathbf{r}) = \sum_{k=1}^{K} G_k(\mathbf{r})\hat{C}_k(\mathbf{r})$$

Depth-based Mutual Learning

➢ Mutual Learning:

- each sub-NeRF not only learns from ground truth but also learns from each other.



serves as geometric regularization

↓

helps the model find more robust geometric solutions

Total Training Loss Function

Color rendering loss Depth regularization CV balancing loss

$$L = L_c + \lambda_1 L_{dml} + \lambda_2 L_{cv}$$

$$L_{cv} = \frac{\mathrm{Var}(\overline{G}(\mathcal{R}))}{\left(\sum_{k=1}^{n} \overline{G_k}(\mathcal{R})/n\right)^2}$$

$$\overline{G_k}(\mathcal{R}) = \sum_{\mathbf{r}\in\mathcal{R}} G_k(\mathbf{r}),$$

Encourages a balanced allocation of model parameters for training rays.

prevents the gate module from collapsing onto a specific sub-NeRF

Evaluation: Setup

➢ Experiment Setup

- **Datasets: five datasets from different types of scenes**
  (1) Object dataset: Masked Tanks-And-Temples (MaskTAT)
  (2) 360-degree inward/outward-facing dataset:
  Tanks-And-Temples (TAT) & NeRF-360-v2 dataset
  (3) Free-shooting-trajectory dataset:
  Free-dataset & ScanNet dataset
- **Baselines: different NeRF training frameworks**
  (1) Grid-based single-NeRF: PlenOctrees, DVGO, Instant-NGP & F2-NeRF
  (2) MLP-based single-NeRF: NeRF, NeRF++, MipNeRF & MipNeRF360
  (3) Multi-NeRF frameworks: NGP-version of Block-NeRF, Switch-NeRF & Rad-NeRF

Higher Rendering Performance

- **Rad-NeRF achieves higher rendering quality than other single/multi-NeRF methods**

Table 1: Quantitative results in complex scenes.

Methods	TAT			NeRF-360-v2			Free-Dataset		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
NeRF++	20.419	0.663	0.451	27.211	0.728	0.344	24.592	0.648	0.467
MipNeRF360	**22.061**	**0.731**	**0.357**	**28.727**	**0.799**	**0.255**	**27.008**	**0.766**	0.295
$\text{MipNeRF360}_{short}$*	20.078	0.617	0.508	25.484	0.631	0.452	24.711	0.648	0.466
DVGO	19.750	0.634	0.498	25.543	0.679	0.380	23.485	0.633	0.479
Instant-NGP	20.722	0.657	0.417	27.309	0.756	0.316	25.951	0.711	0.312
F2-NeRF	–	–	–	26.393	0.746	0.361	26.320	**0.779**	**0.276**
Switch-NGP†	20.512	0.654	0.432	26.524	0.740	0.331	25.755	0.694	0.341
Block-NGP†	20.783	0.659	0.415	27.436	0.761	0.298	26.015	0.702	0.325
Rad-NeRF	**21.708**	**0.672**	**0.398**	**27.871**	**0.769**	**0.298**	**26.449**	0.719	**0.285**

* MipNeRF360 requires nearly one day for training. For a fair comparison, we also report its results with one-hour of training.
† We adapt Switch-NeRF and Block-NeRF to the Instant-NGP fast training framework.

Higher Rendering Performance

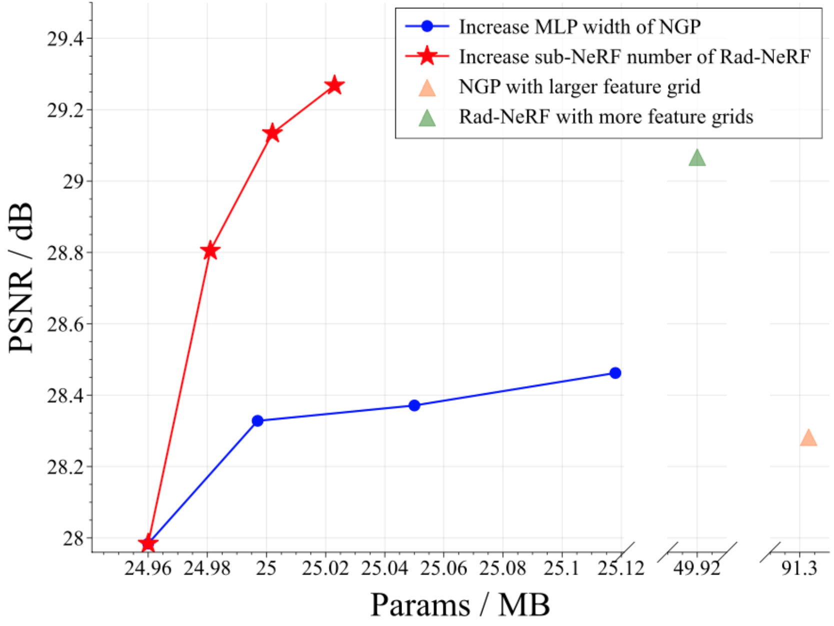
➢ **Rad-NeRF achieves better recovery of scene details**

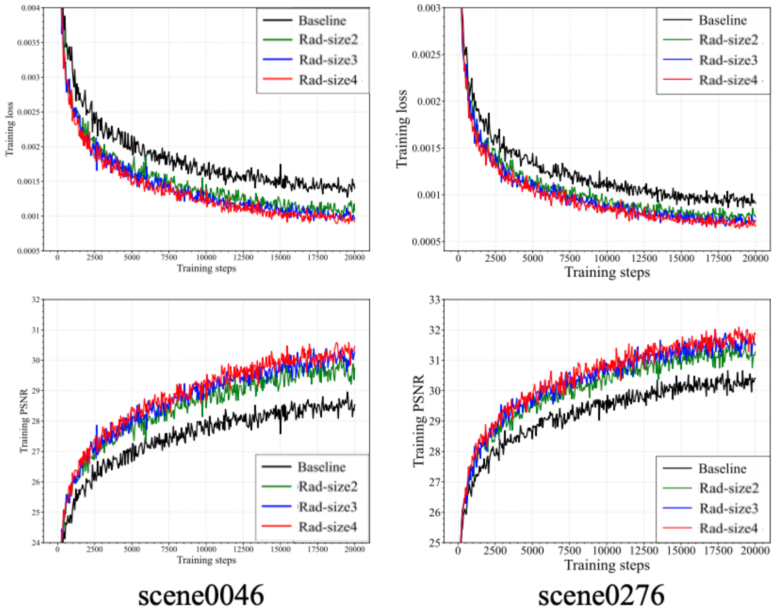
Instant-NGP

Rad-NeRF

Better Model Scalability

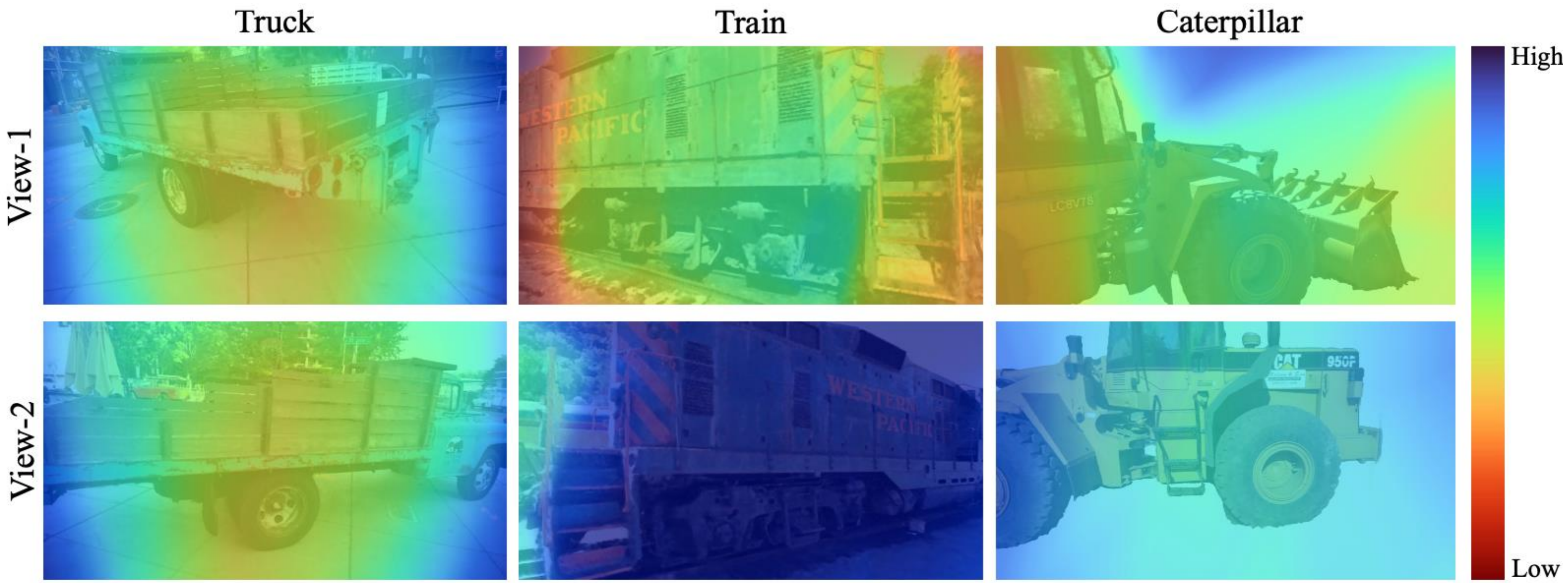
- Rad-NeRF achieves better **performance-parameter scalability**





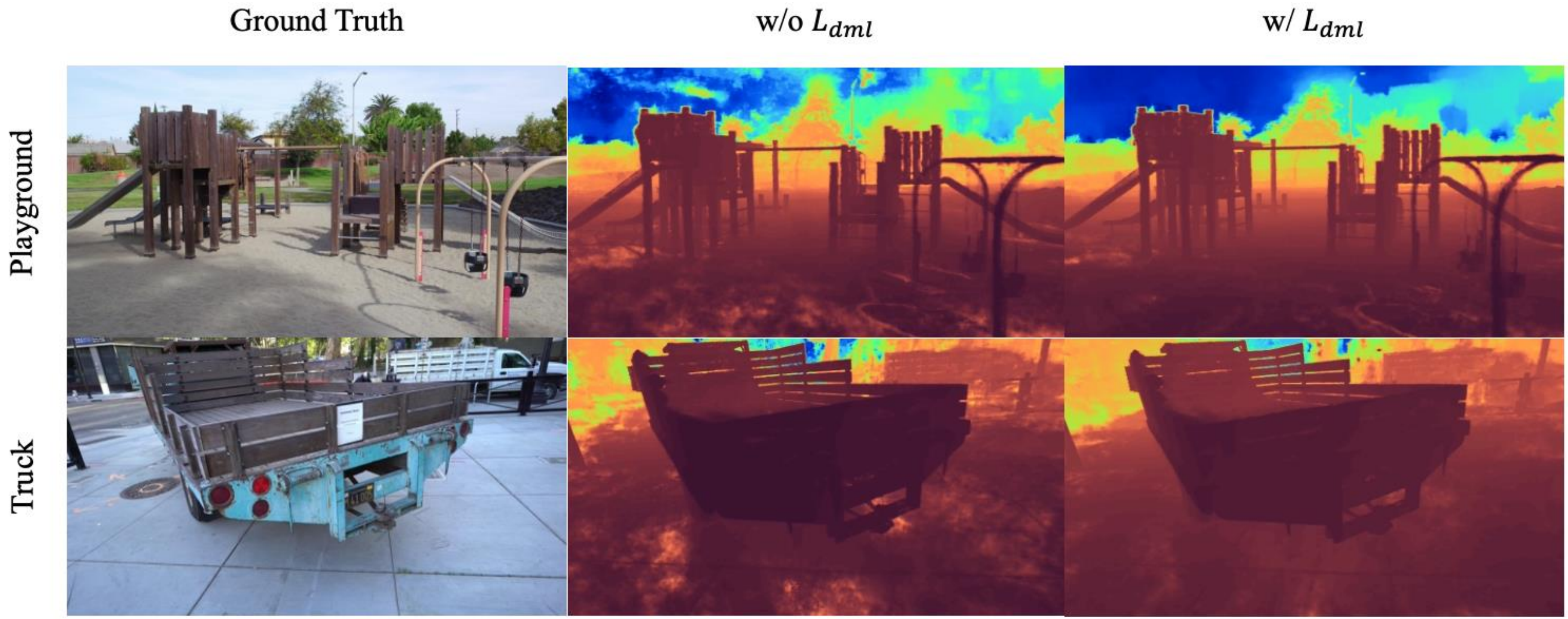
Reasonable Gating Scores

➢ **Rad-NeRF learns reasonable ray allocations, matching training interference "intuition"**



Ablation Study of DML

- **DML enables a smooth and reasonable depth prediction**



Thanks for your attention!
Q&A

<u>Lidong Guo[1*],</u> Xuefei Ning[1*], Yonggan Fu[2], Tianchen Zhao[3], Zhuoliang Kang[3], Jincheng Yu[1], Yingyan (Celine) Lin[2], Yu Wang[1]

[1]Tsinghua University, [2]Georgia Institute of Technology, [3]Meituan
E-mail: gld21@mails.tsinghua.edu.cn, foxdoraame@gmail.com, yu-wang@tsinghua.edu.cn