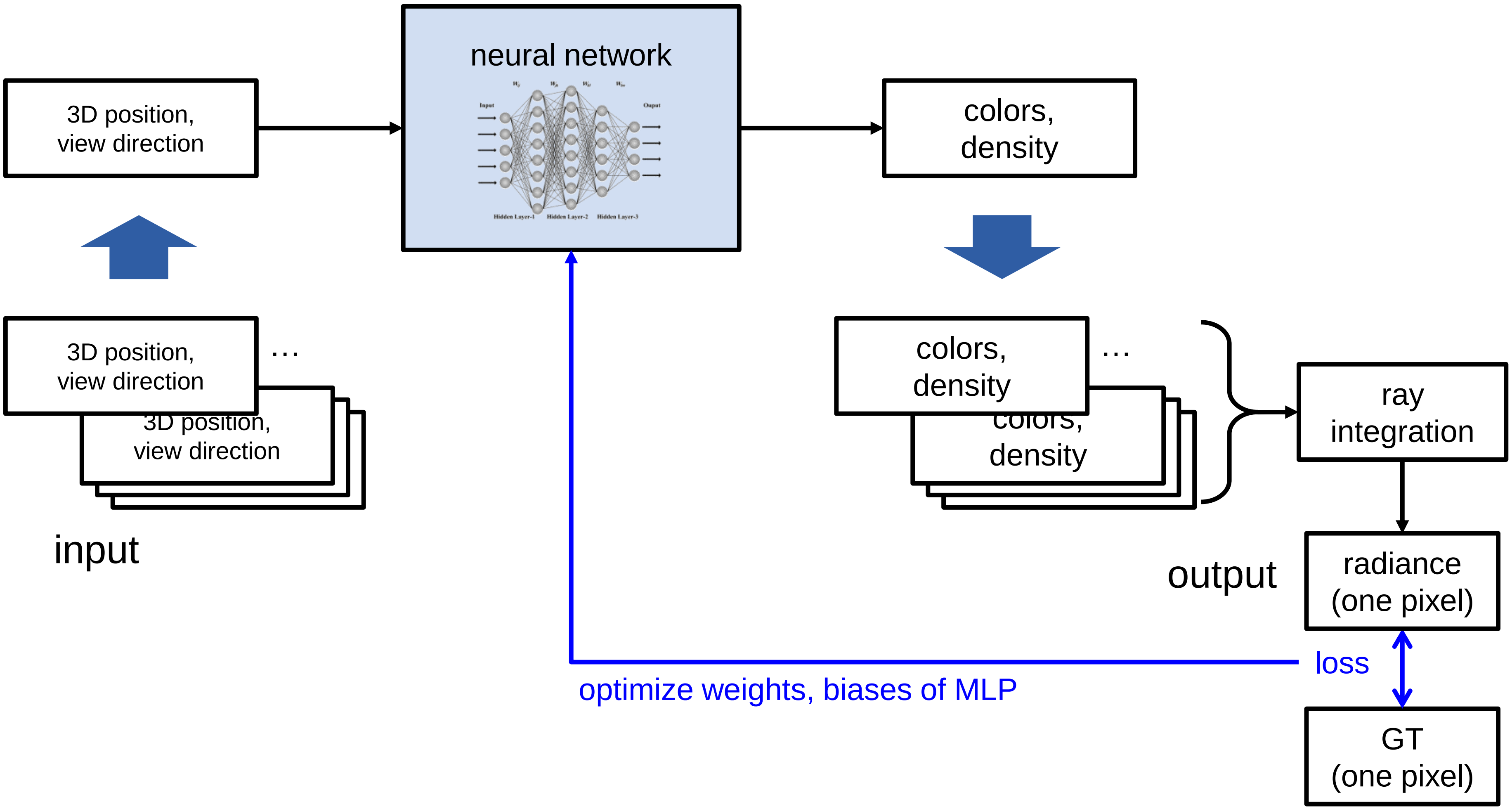
Toward Practical NeRF

(CS580 Student Presentation)

25. APR. 2022.

Kiseok Choi

NeRF (Neural Radiance Fields)



Reference: Ben Mildenhall et al., NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020

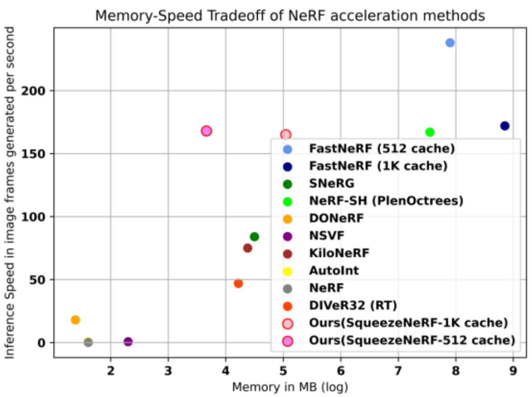
NeRF Variants

NeRF

Speed
- KiloNeRF
- NSVF
- DONeRF
- FastNeRF
- Instant NGP …

Output Quality
- Mip-NeRF
- NeRF-SR
- NSVF
- Plenoxels
- …

Special Features
- NeRF++
- BACON
- NeuS
- GRAF
- pi-GAN
- BARF
- IRON …



Graph reference: Krishna Wadhwani et al., SqueezeNeRF: Further factorized FastNeRF for memory-efficient inference, Arxiv 2022

Motivation

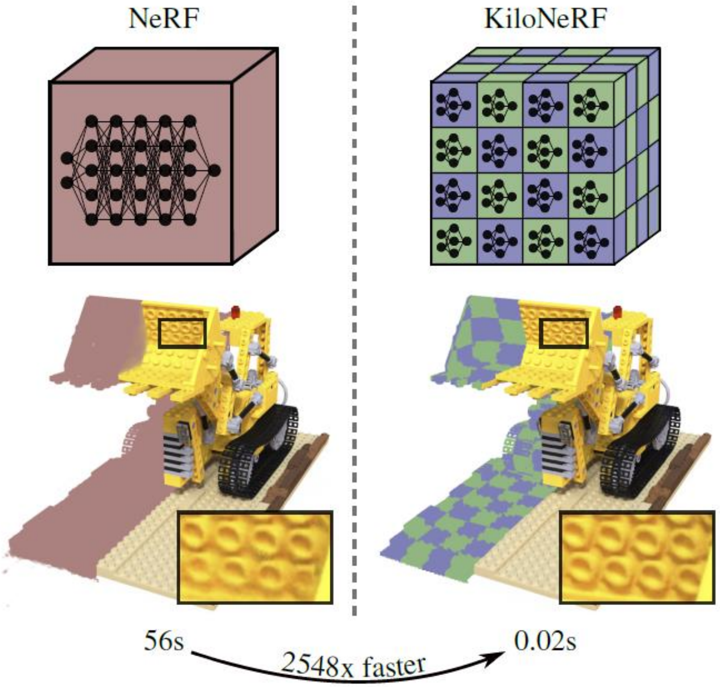
- NeRF is a hot rendering method in computer graphics recently!
- It generates a good image
  without any complicated rendering equation.
- However,
  (1) too slow
  → how to accelerate the method?

  (2) too specialized in a specific field
  → how to be generalized?

KiloNeRF: Speeding up Neural Radiance Fields with Thousands of Tiny MLPs

Overview

- Title: “KiloNeRF: Speeding up Neural Radiance Fields with Thousands of Tiny MLPs”
- Conference: ICCV 2021
- Authors: Christian Reiser, Songyou Peng, Yiyi Liao, Andreas Geiger



https://ps.is.mpg.de/publications/reiser2021iccv

Method

- For speed-up of rendering time
  - Decomposition of MLP into thousands of tiny MLPs
  - Empty space skipping (ESS)
  - Early ray termination (ERT)
  - Parallel processing optimization

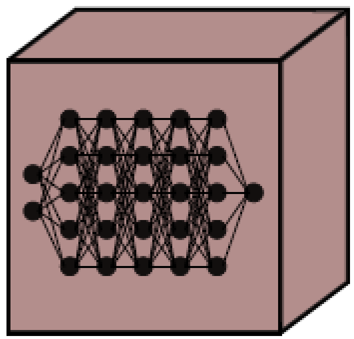
- For similar quality of output as NeRF
  - Teacher-student distillation

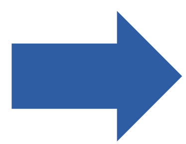
Method

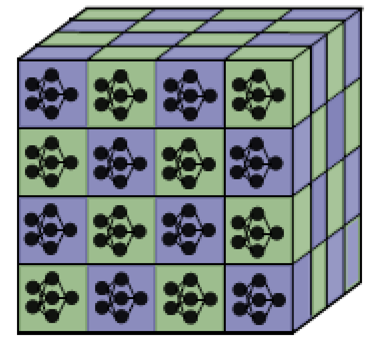
- Decomposition of MLP into thousands of tiny MLPs

$$g(\mathbf{x}) = \lfloor (\mathbf{x} - \mathbf{b}_{min}) / ((\mathbf{b}_{max} - \mathbf{b}_{min}) / \mathbf{r}) \rfloor$$

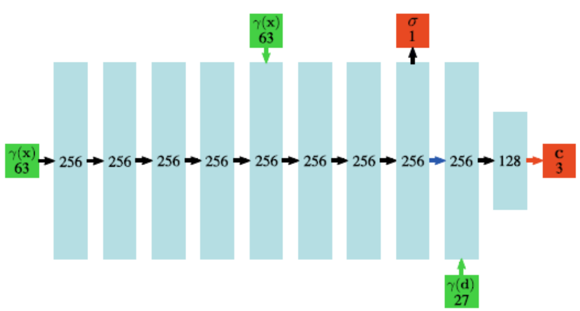
$$(\mathbf{c}, \sigma) = f_{\theta(g(\mathbf{x}))}(\mathbf{x}, \mathbf{d})$$



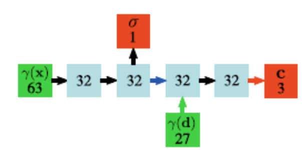


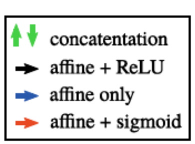


NeRF: ~1056 kFLOPs



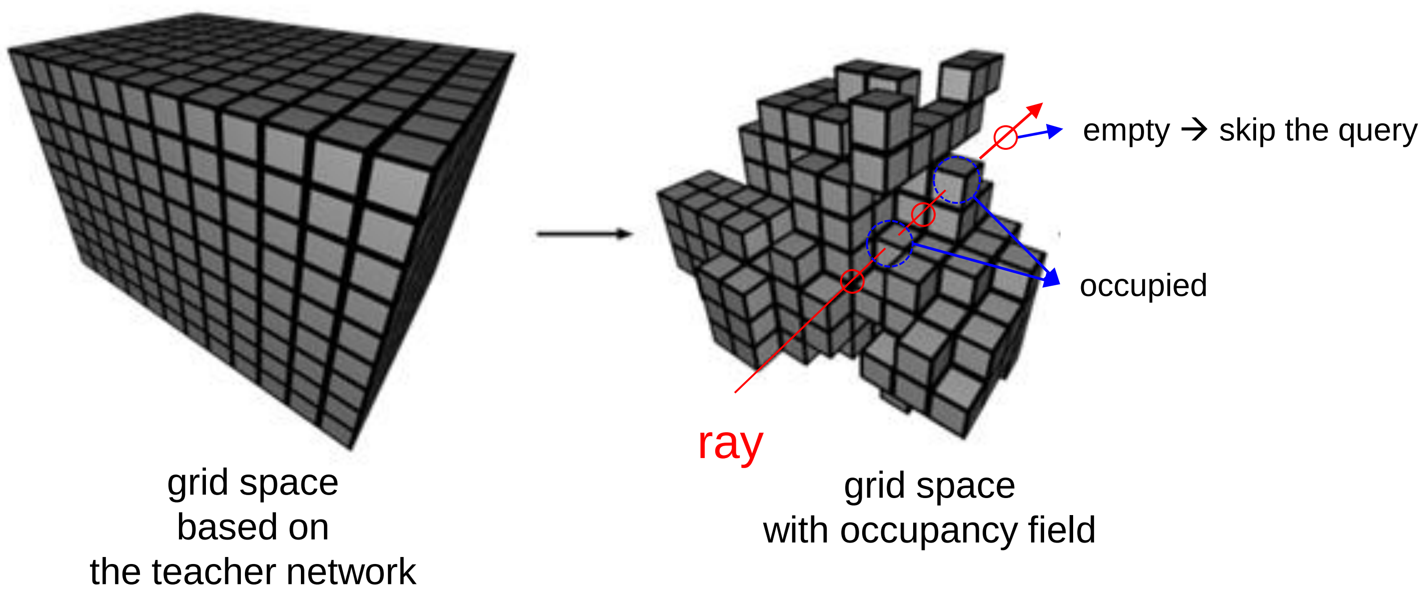
KiloNeRF: ~12 kFLOPs





Method

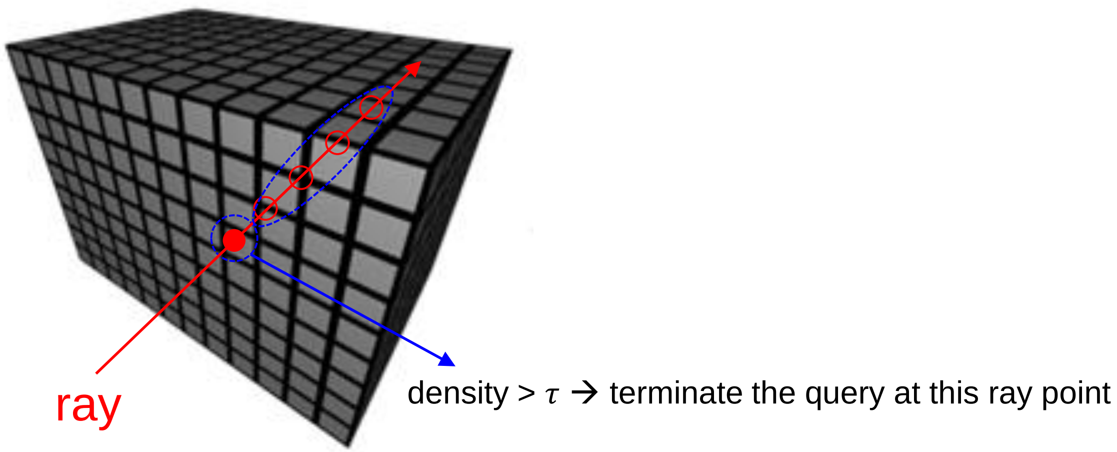
- Empty space skipping (ESS)



Reference: Lingjie Liu et al., Neural sparse voxel fields, NeurIPS 2020

Method

- Early ray termination (ERT)



grid space
based on
the teacher network

Method	Render time ↓	Speedup ↑
NeRF	56185 ms	–
NeRF + ESS + ERT	788 ms	71
KiloNeRF	**22** ms	**2548**

Reference: Lingjie Liu et al., Neural sparse voxel fields, NeurIPS 2020

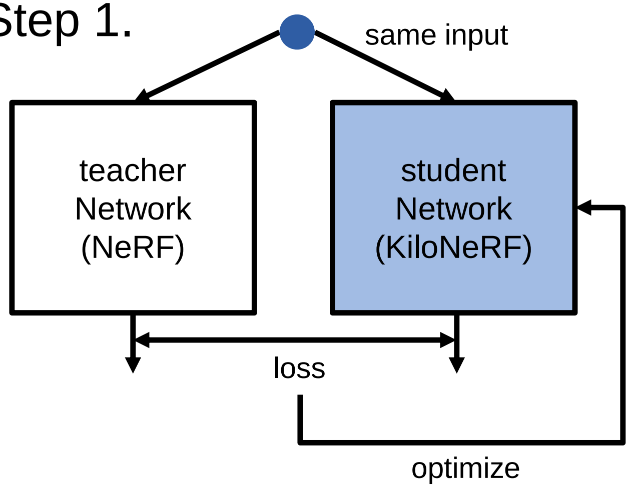
Method

- Parallel processing optimization
  - PyTorch: deep learning package for CUDA
  - MAGMA: linear algebra acceleration package for CUDA
  - Thrust: parallel algorithm library for CUDA and OpenMP
  - Custom CUDA kernels

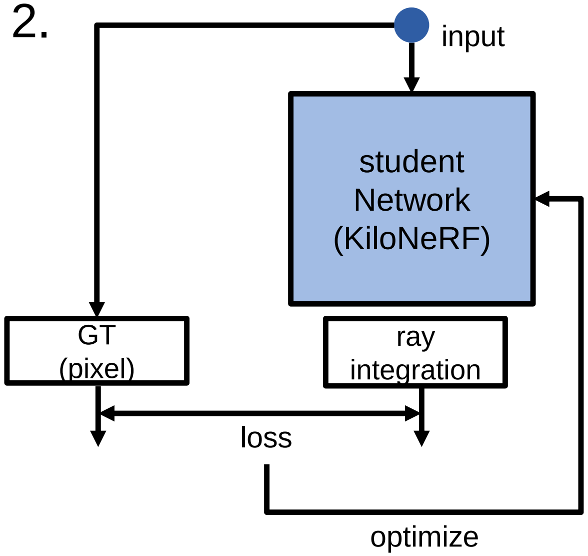
Method

- Teacher-student distillation
  - Assumption: there is a pre-trained teacher network (NeRF)
  - Utilizing the teacher network, train the student network with 2 steps as below.

Step 1.



Step 2.



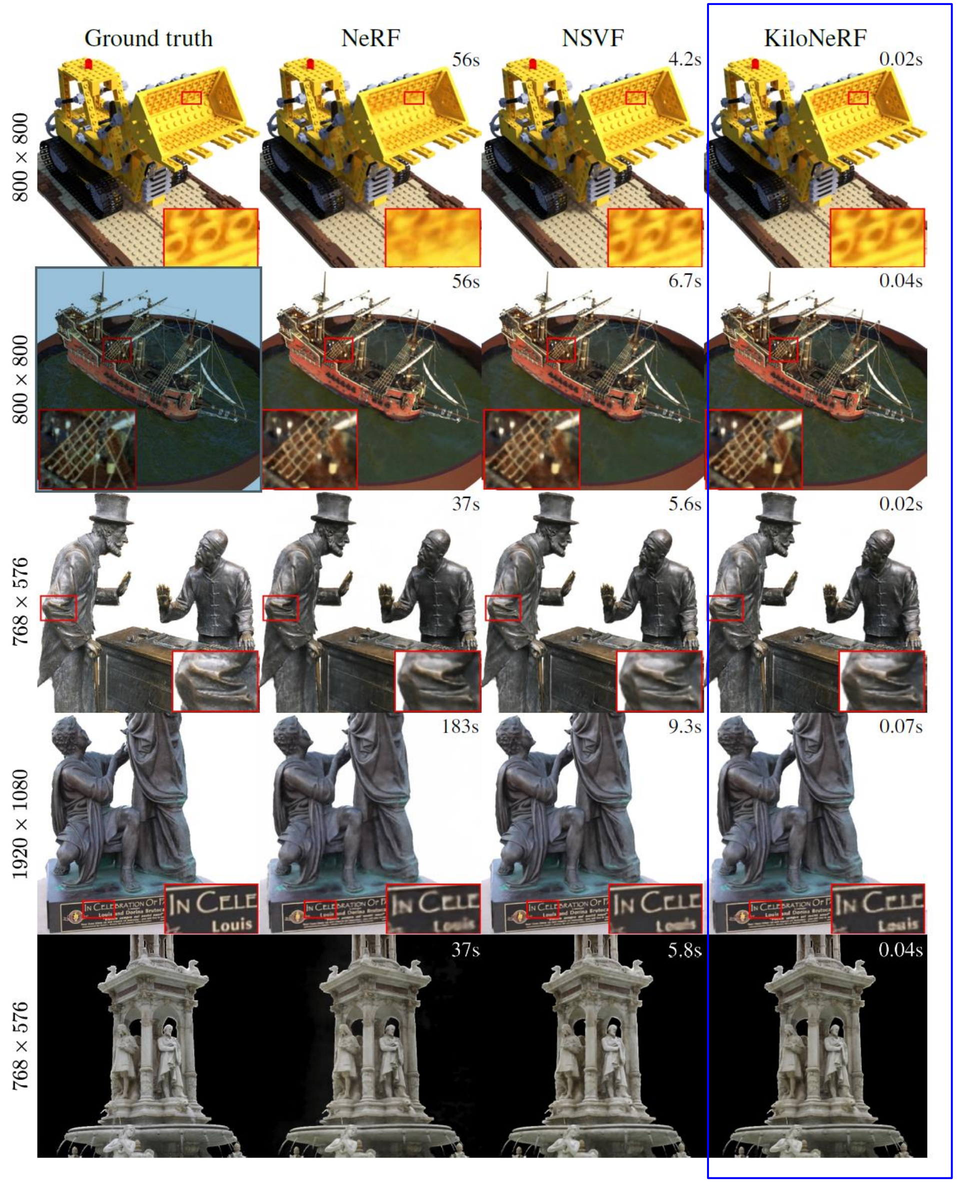
(a) Without Distillation (b) With Distillation

Method	PSNR ↑	SSIM ↑	LPIPS ↓
KiloNeRF w/o Distillation and L_2 Regularization	29.54	0.932	0.051
KiloNeRF w/o Distillation	30.44	0.940	0.048
KiloNeRF	**30.66**	**0.945**	**0.043**

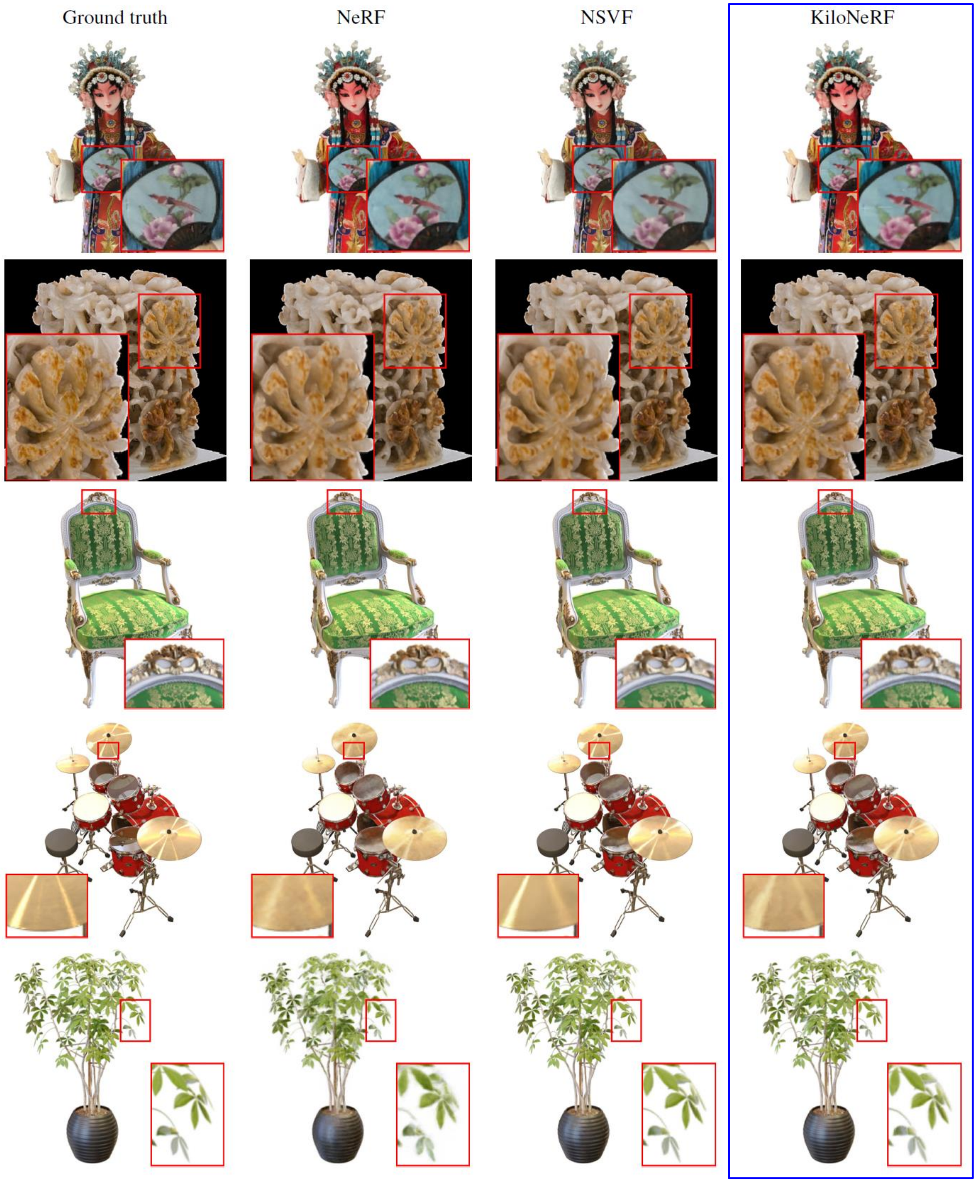
Result 1

Resolution		BlendedMVS 768 × 576	Synthetic-NeRF 800 × 800	Synthetic-NSVF 800 × 800	Tanks & Temples 1920 × 1080
PSNR ↑	NeRF	27.29	31.01	31.55	28.32
	NSVF	26.90	**31.74**	**35.13**	28.40
	KiloNeRF	**27.39**	31.00	33.37	**28.41**
SSIM ↑	NeRF	0.91	**0.95**	0.95	0.90
	NSVF	0.90	**0.95**	**0.98**	0.90
	KiloNeRF	**0.92**	**0.95**	0.97	**0.91**
LPIPS ↓	NeRF	0.07	0.08	0.04	0.11
	NSVF	0.11	0.05	**0.01**	0.15
	KiloNeRF	**0.06**	**0.03**	0.02	**0.09**
Render time (milliseconds) ↓	NeRF	37266	56185	56185	182671
	NSVF	4398	4344	10497	15697
	KiloNeRF	**30**	**26**	**26**	**91**
Speedup over NeRF ↑	NSVF	8	13	5	12
	KiloNeRF	**1258**	**2165**	**2167**	**2002**

Result 2


Ground truth
NeRF
NSVF
KiloNeRF
800 × 800
56s
4.2s
0.02s
800 × 800
56s
6.7s
0.04s
768 × 576
37s
5.6s
0.02s
1920 × 1080
183s
9.3s
0.07s
768 × 576
37s
5.8s
0.04s

Result 3



Conclusion

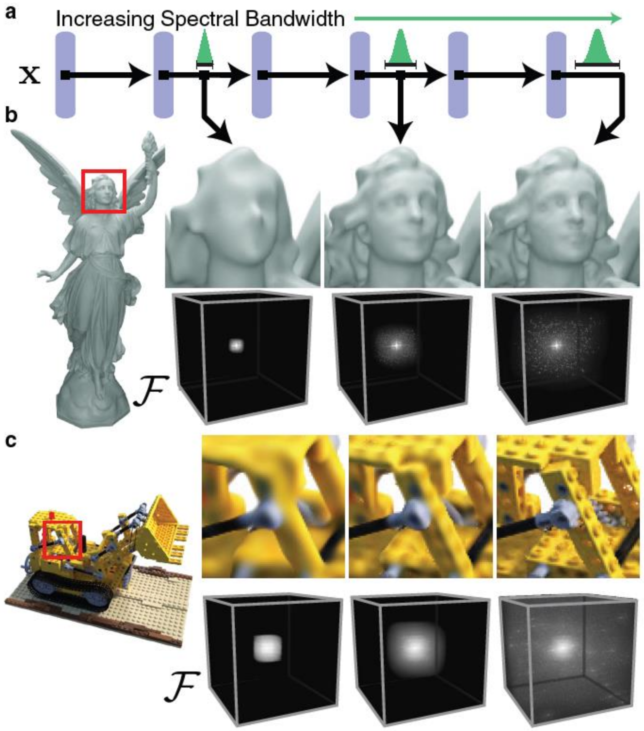
- Strengths
  - Rendering is very fast.
  - Image quality is slightly better than NeRF.

- Weaknesses
  - Pre-trained NeRF is necessary for output quality similar to NeRF.
  - Special accelerating packages for CUDA are necessary.

BACON: Band-limited Coordinate Networks for Multiscale Scene Representation

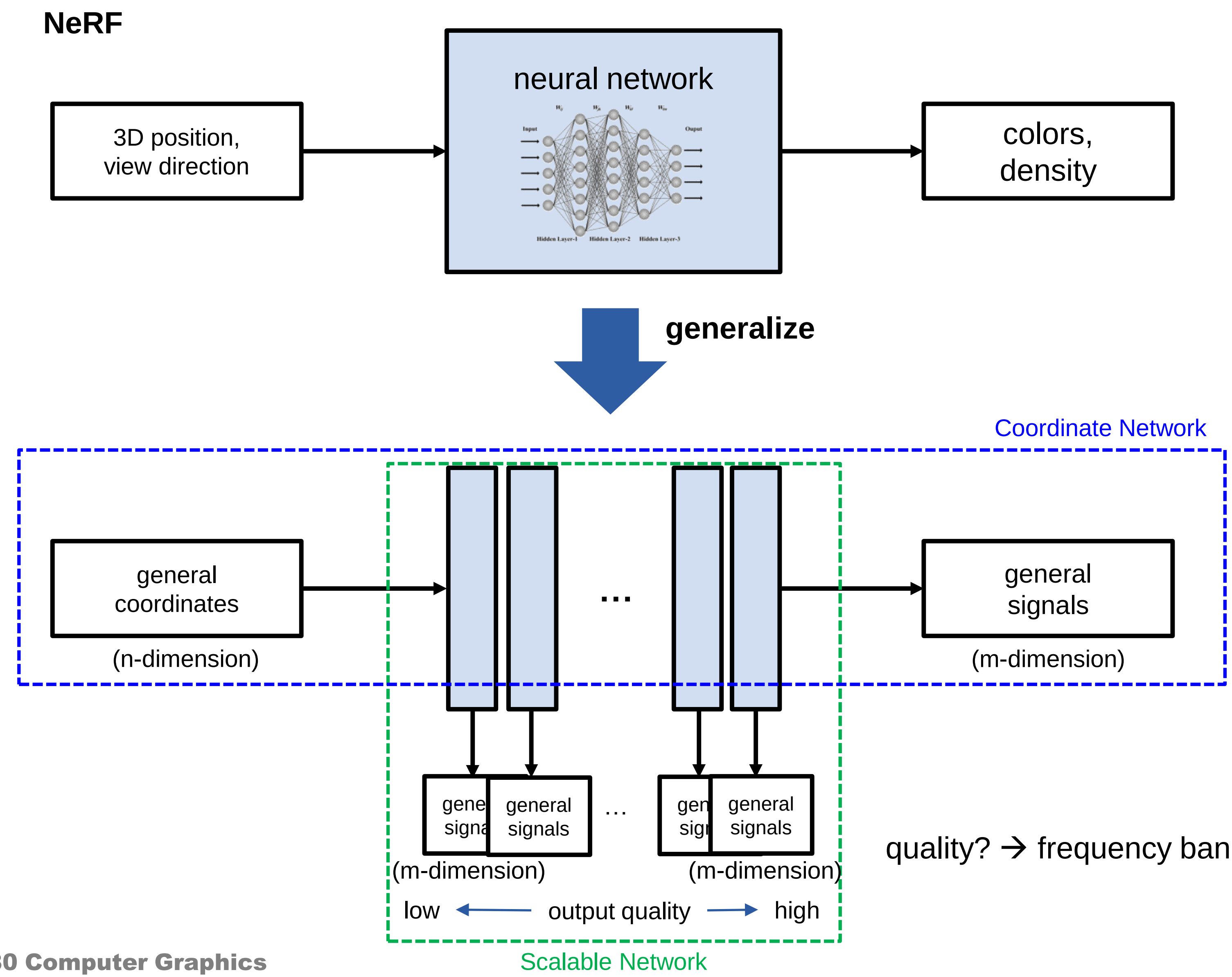
Overview

- Title: “BACON: Band-limited Coordinate Networks for Multiscale Scene Representation”
- Conference: CVPR 2022 (Oral)
- Authors: David B. Lindell, Dave Van Veen, Jeong Joon Park, Gordon Wetzstein



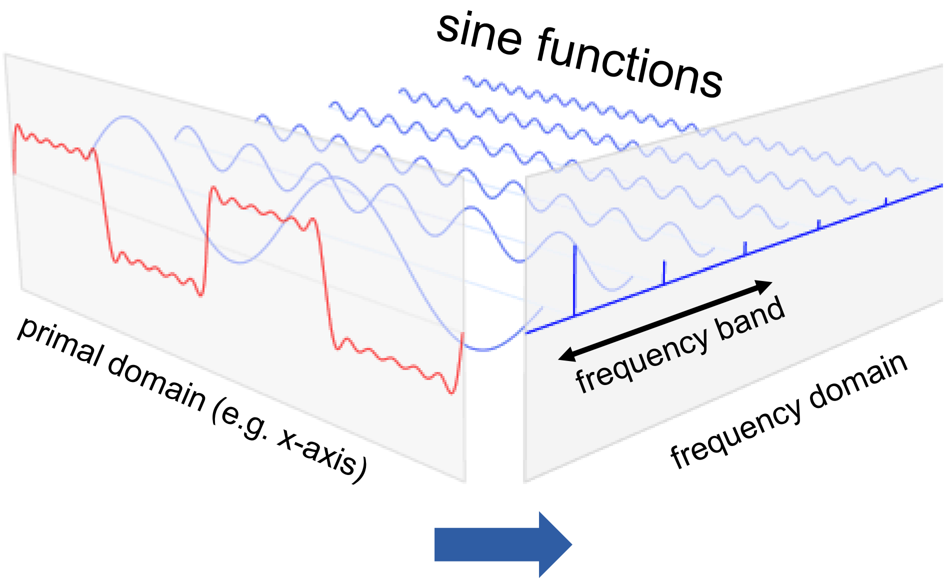
https://www.computationalimaging.org/publications/bacon/

Background



quality? → frequency band of signals

Background

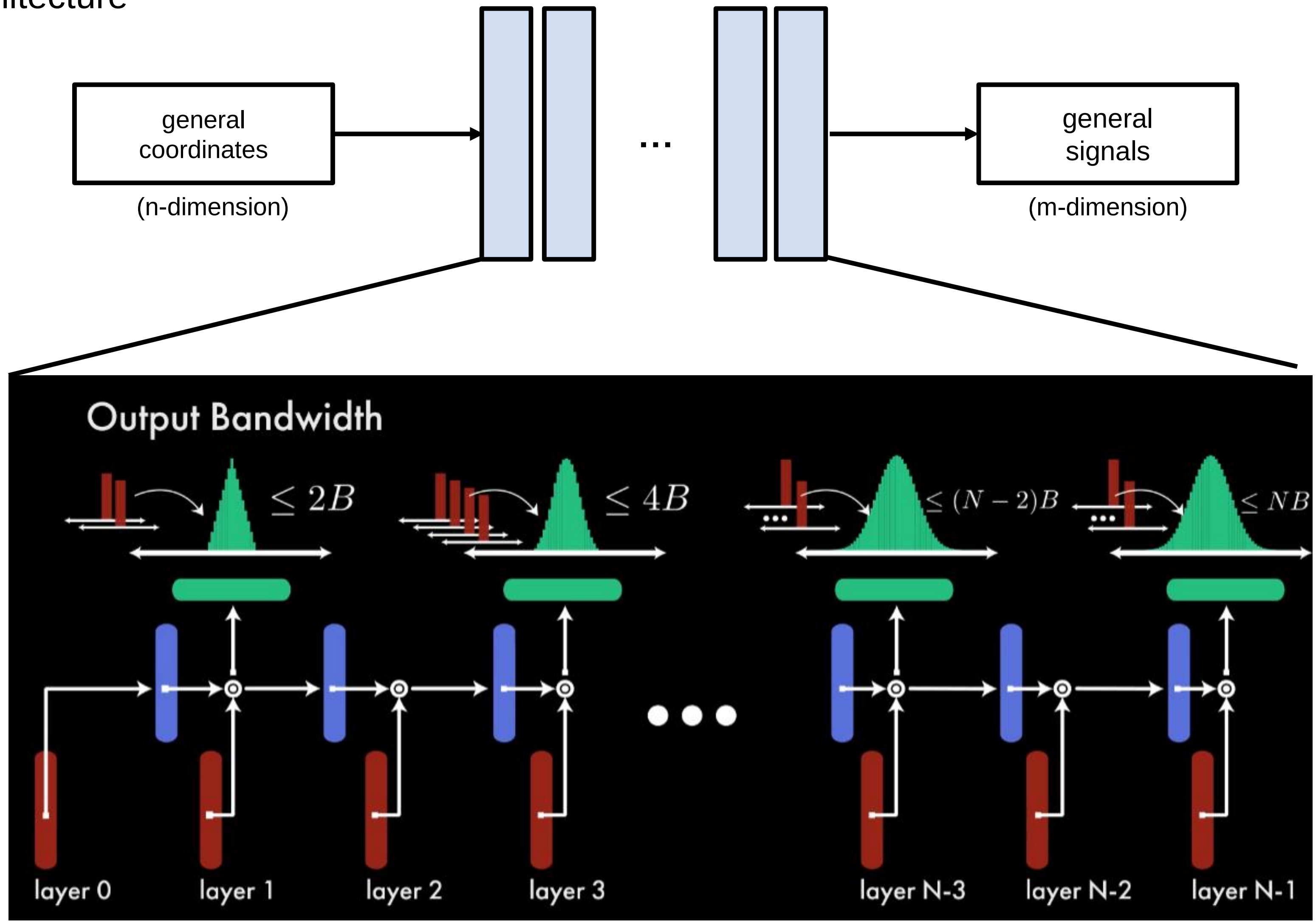


Fourier analysis

reference: Wikipedia

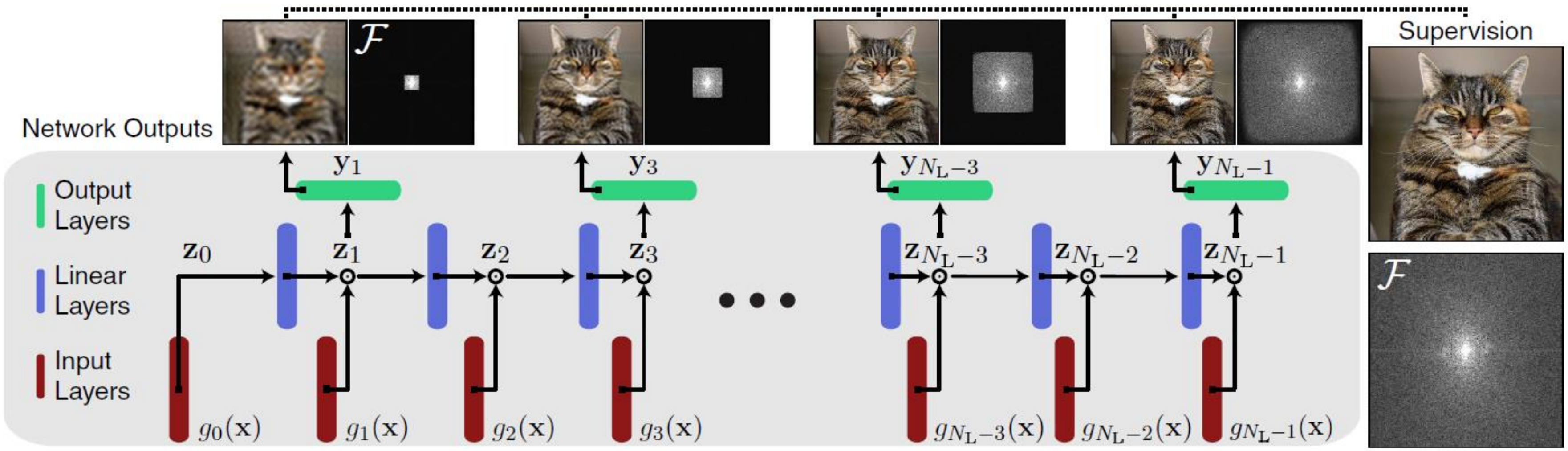
Method

- Architecture



Method

- Equations



low frequency encoding

high frequency encoding

$$g_i(\mathbf{x}) = \sin(\boldsymbol{\omega}_i \mathbf{x} + \phi_i)$$

$$\mathbf{z}_0 = g_0(\mathbf{x})$$

$$\mathbf{z}_i = g_i(\mathbf{x}) \circ (\mathbf{W}_i \mathbf{z}_{i-1} + \mathbf{b}_i), \quad 0 \leq i < N_L$$

Hadamard (element-wise) product

$$\mathbf{y}_i = \mathbf{W}_i^{\text{out}} \mathbf{z}_i + \mathbf{b}_i^{\text{out}},$$

$$\mathbf{y}_i = \sum_{j=0}^{N_{\text{sine}}^{(i)}-1} \bar{\alpha}_j \sin(\bar{\boldsymbol{\omega}}_j \mathbf{x} + \bar{\phi}_j)$$

$$N_{\text{sine}}^{(N_L)} = \sum_{i=0}^{N_L-1} 2^i d_h^{i+1}$$

number of sine terms

$$B = \sum_{i=0}^{N_L-1} B_i$$

Method

- Initialization

$$B = \sum_{i=0}^{N_\mathrm{L}-1} B_i$$

0.5 cycles per coordinate resolution
(Nyquist's theorem)

uniform
distribution
in band limit

$$\boldsymbol{\omega}_i \sim \mathcal{U}(-B_i, B_i)$$

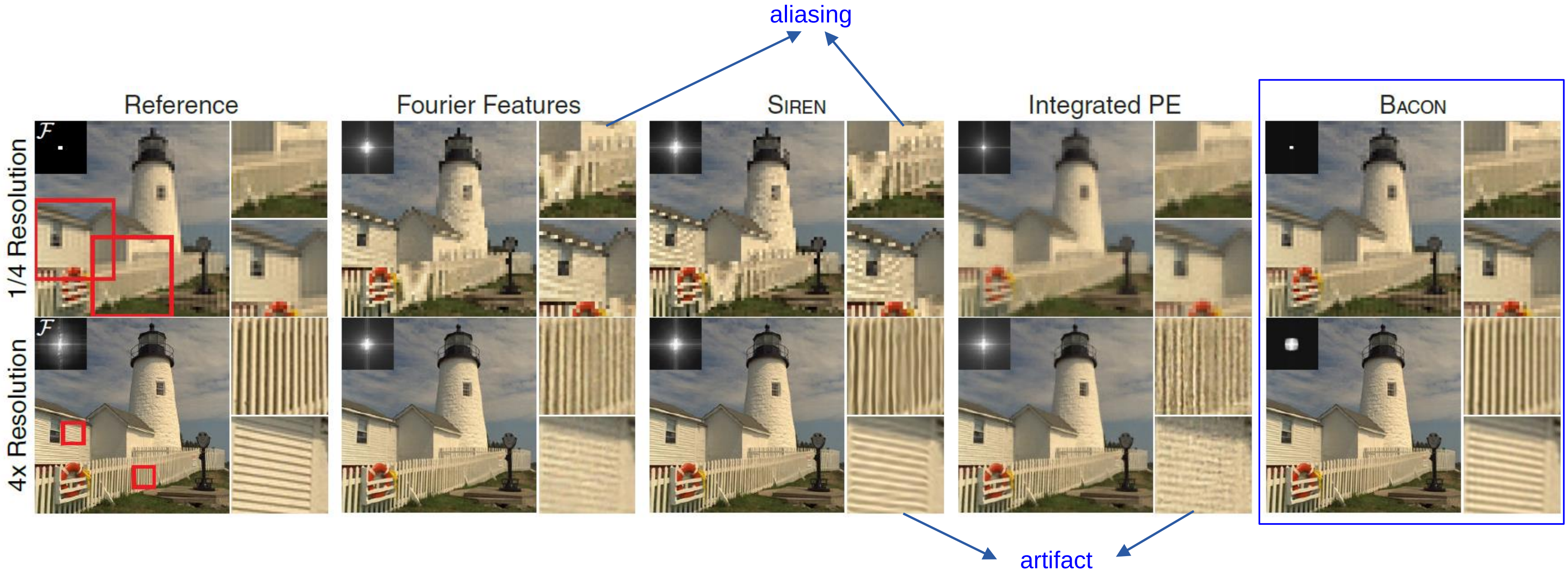
uniform
distribution
in phase limit

$$\phi_i \sim \mathcal{U}(-\pi, \pi)$$

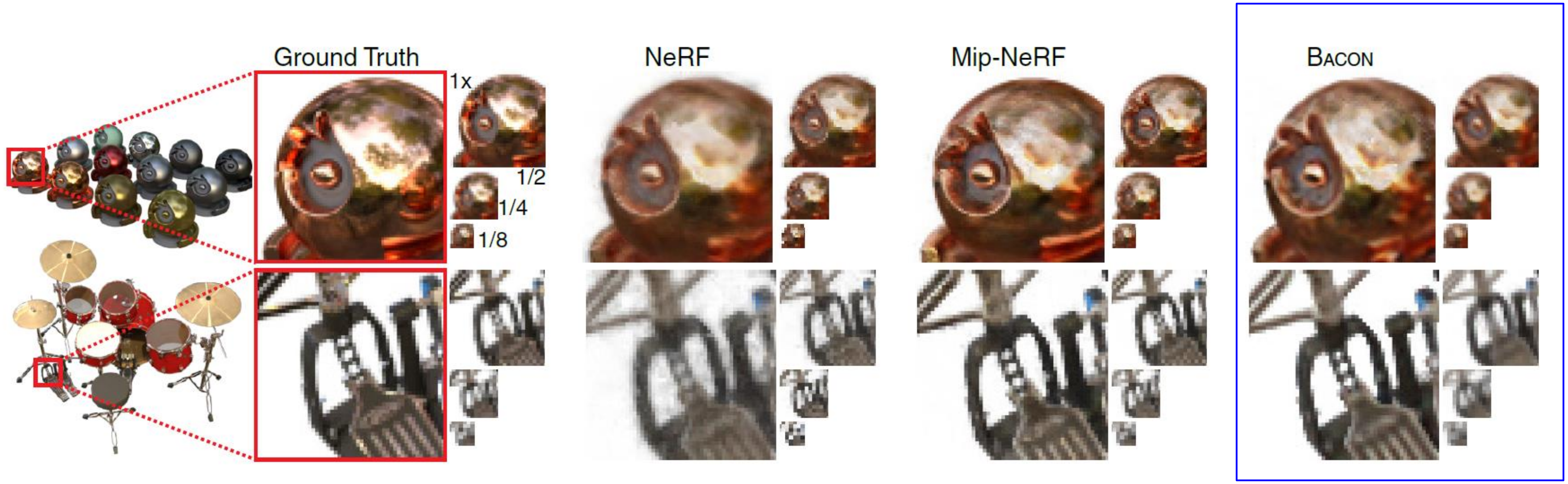
$$g_i(\mathbf{x}) = \sin(\boldsymbol{\omega}_i \mathbf{x} + \phi_i)$$

Result 1: 2D Image Fitting

- Input: 256 x 256 image (1x Resolution)
- Output: 64 x 64 image (1/4 Resolution)
- Output: 1024 x 1024 image (4x Resolution)



Result 2: Neural Rendering



- NeRF, Mip-NeRF
  input: 512 x 512 images (for 1x)
  256 x 256 images (for 1/2x)
  … 64 x 64 images (for 1/8x)
  output: 512 x 512 images (for 1x)
  256 x 256 images (for 1/2x)
  … 64 x 64 images (for 1/8x)

- BACON
  input: 512 x 512 images
  output: 512 x 512 images (for 1x)
  256 x 256 images (for 1/2x)
  … 64 x 64 images (for 1/8x)

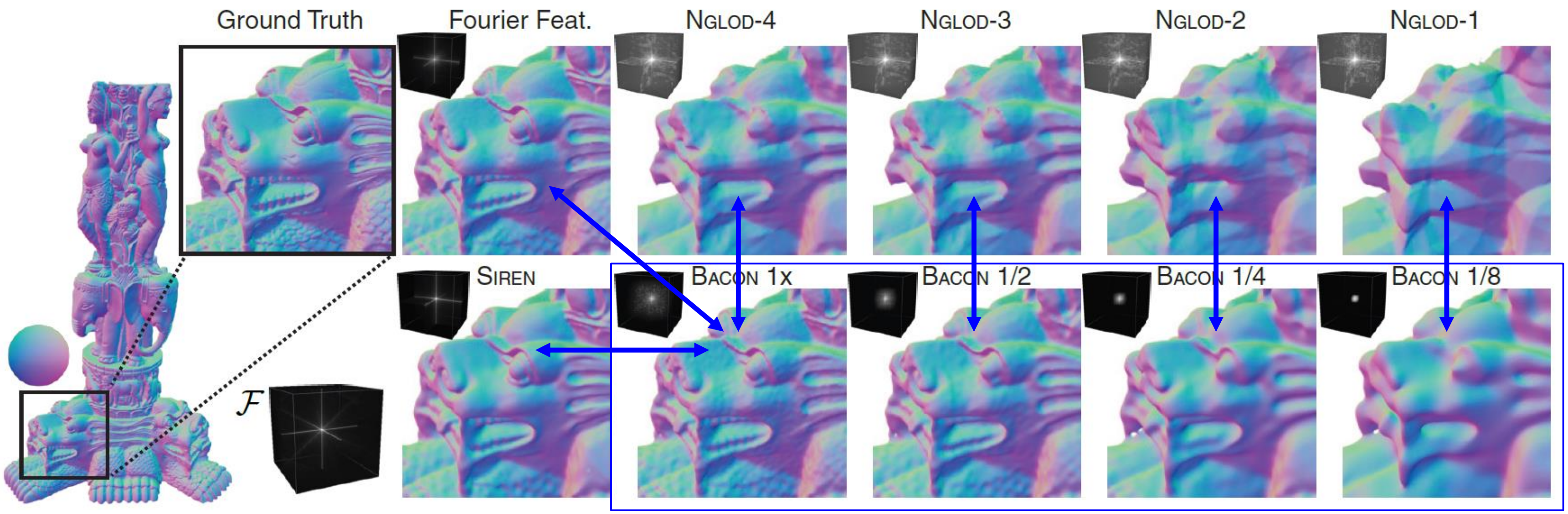
	PSNR ↑					# Params.			
	1×	1/2	1/4	1/8	Avg.	1×	1/2	1/4	1/8
NeRF	26.734	28.941	29.297	26.464	27.859	511K			
Mip-NeRF	29.874	31.307	32.093	32.832	**31.526**	511K			
BACON	27.430	28.066	28.520	28.475	<u>28.123</u>	531K	398K	266K	133K

output quality: NeRF < BACON < Mip-NeRF

	Inference Times (s)			
	1×	1/2	1/4	1/8
NeRF	4.4	1.1	0.28	0.073
Mip-NeRF	4.5	1.1	0.28	0.073
BACON	10.2	2.1	0.39	0.065

Rendering time of BACON is longer.

Result 3: 3D Shape Fitting



1x, 1/2x, 1/4x, 1/8x image quality: BACON > NGOLD

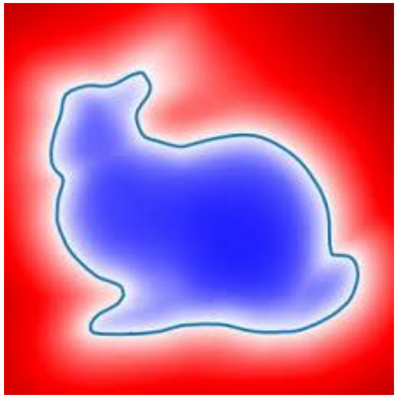
- input: SDF (Signed Distance Function) from mesh + Laplacian noise
- output: SDF (Signed Distance Function) estimated

	FF	SIREN	NGLOD-4	NGLOD-5	BACON 1×
# Params.	527K	528K	1.35M	10.1M	531K
Chamfer↓	**2.166e-6**	2.780e-6	8.358e-6	2.422e-6	2.198e-6
IOU ↑	**9.841e-1**	9.751e-1	9.479e-1	9.811e-1	9.833e-1

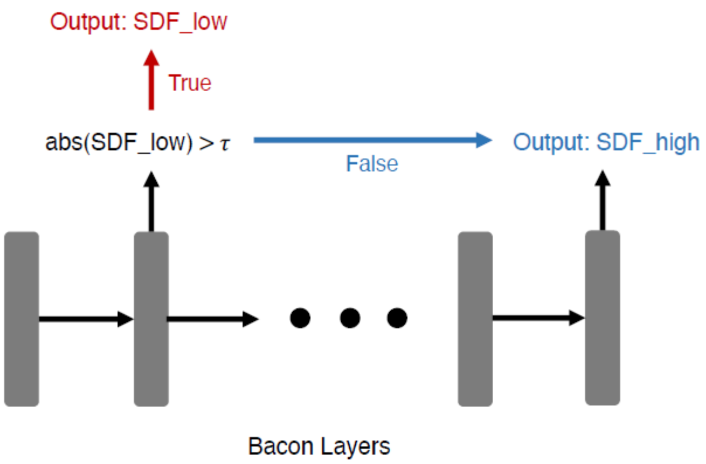
1x image quality: FF > BACON 1x > SIREN > NGOLD-4

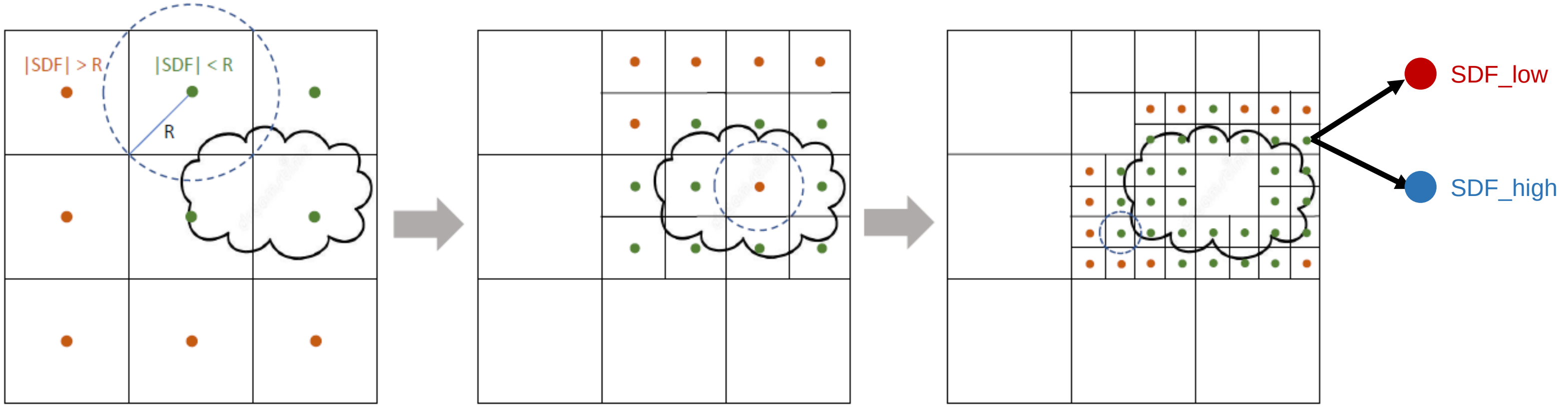
Result 4: Mesh Extraction

- Adaptive multi-scale SDF evaluation process



SDF: Signed Distance Function





Result 4: Mesh Extraction

	Dense Grid (512^3)	Adaptive-Frequency	Adaptive + Multiscale (Proposed)
Time (s)	17.91	5.50	**0.222**

Conclusion

- Strengths
  - BACON can be applied to various applications.
    - It is a generalized coordinate network.
  - Single scale supervision generates multiple scale outputs.
    - We can select a part of network
      according to the trade-off between
      quality vs. resource(memory & computation power)

- Weaknesses
  - It is difficult to understand the fundamental principles.
  - Inference time of BACON is longer than NeRF.
  - Additional initialization process is necessary.
    - Frequency parameters should be initialized carefully.

Quiz

1. KiloNeRF focuses on improving the speed of NeRF.
   (True / False)

2. BACON is a (   ), (   ) network for signal representation.
   (1) detection, non-scalable
   (2) classification, scalable
   (3) segmentation, non-scalable
   (4) coordinate, non-scalable
   (5) coordinate, scalable

Thank you.

Back-up Slide

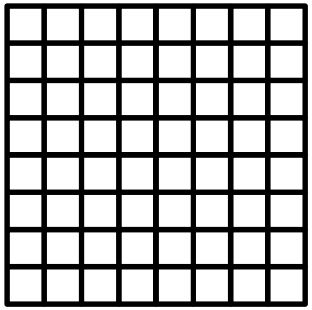
- Comparison of NeRF, Mip-NeRF, BACON
  with similar size of parameters in low resolution

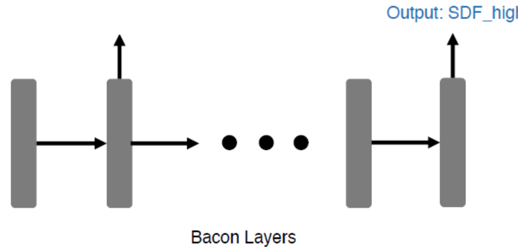
	# Params.	PSNR 1×	PSNR 1/2	PSNR 1/4	PSNR 1/8	SSIM 1×	SSIM 1/2	SSIM 1/4	SSIM 1/8
NeRF	157K	27.144	30.050	31.554	27.309	0.903	0.949	0.971	0.940
Mip-NeRF	157K	30.136	32.067	32.901	32.798	0.939	0.965	0.977	0.980
BACON	133K	N/A	N/A	N/A	29.161	N/A	N/A	N/A	0.954

Back-up Slide

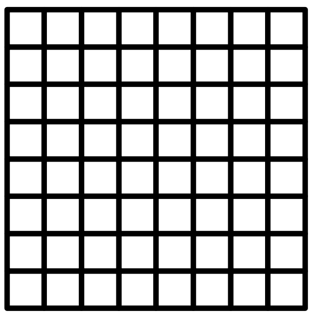
- Mash extraction methods (SDF evaluation)

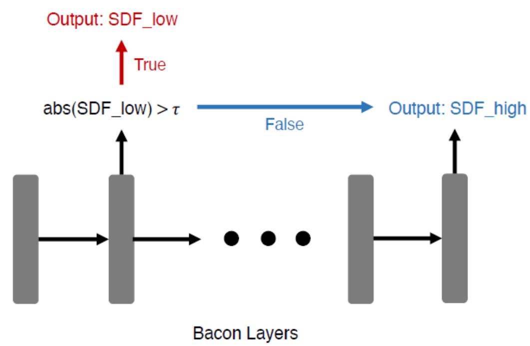
Dense Grid





Adaptive
Frequency





Adaptive
+ Multiscale
(Proposed)