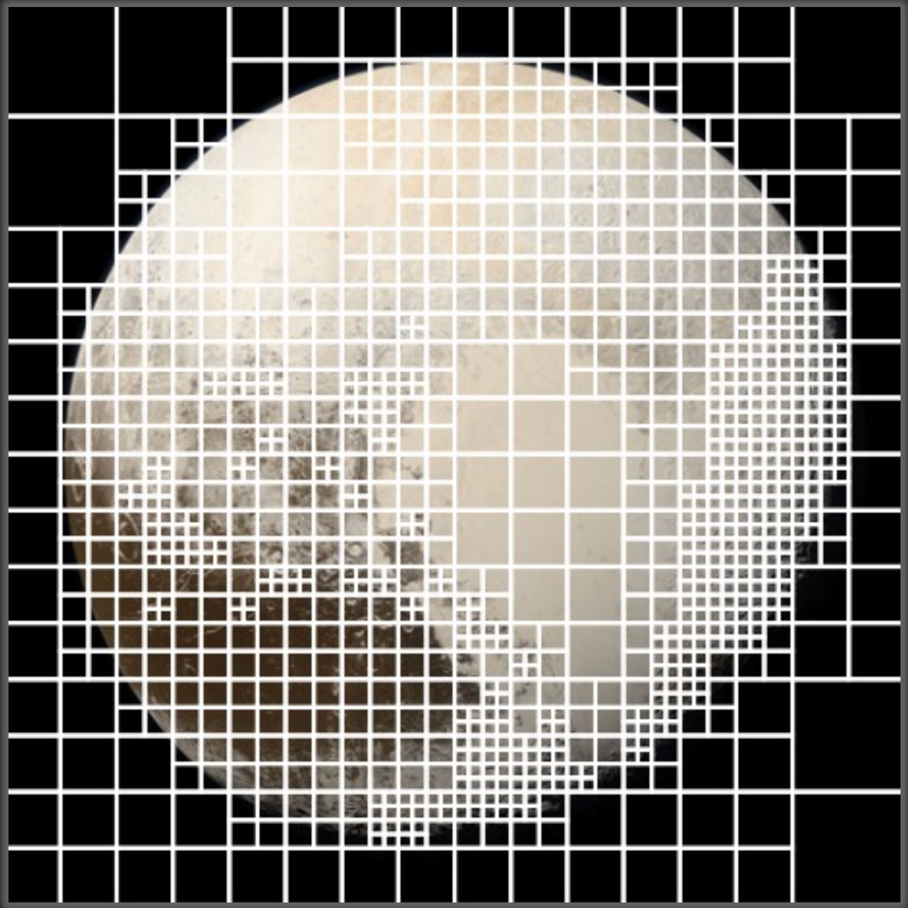
Neural Field Representations

neural fields, neural rendering, applications



CSC2529

David Lindell

University of Toronto

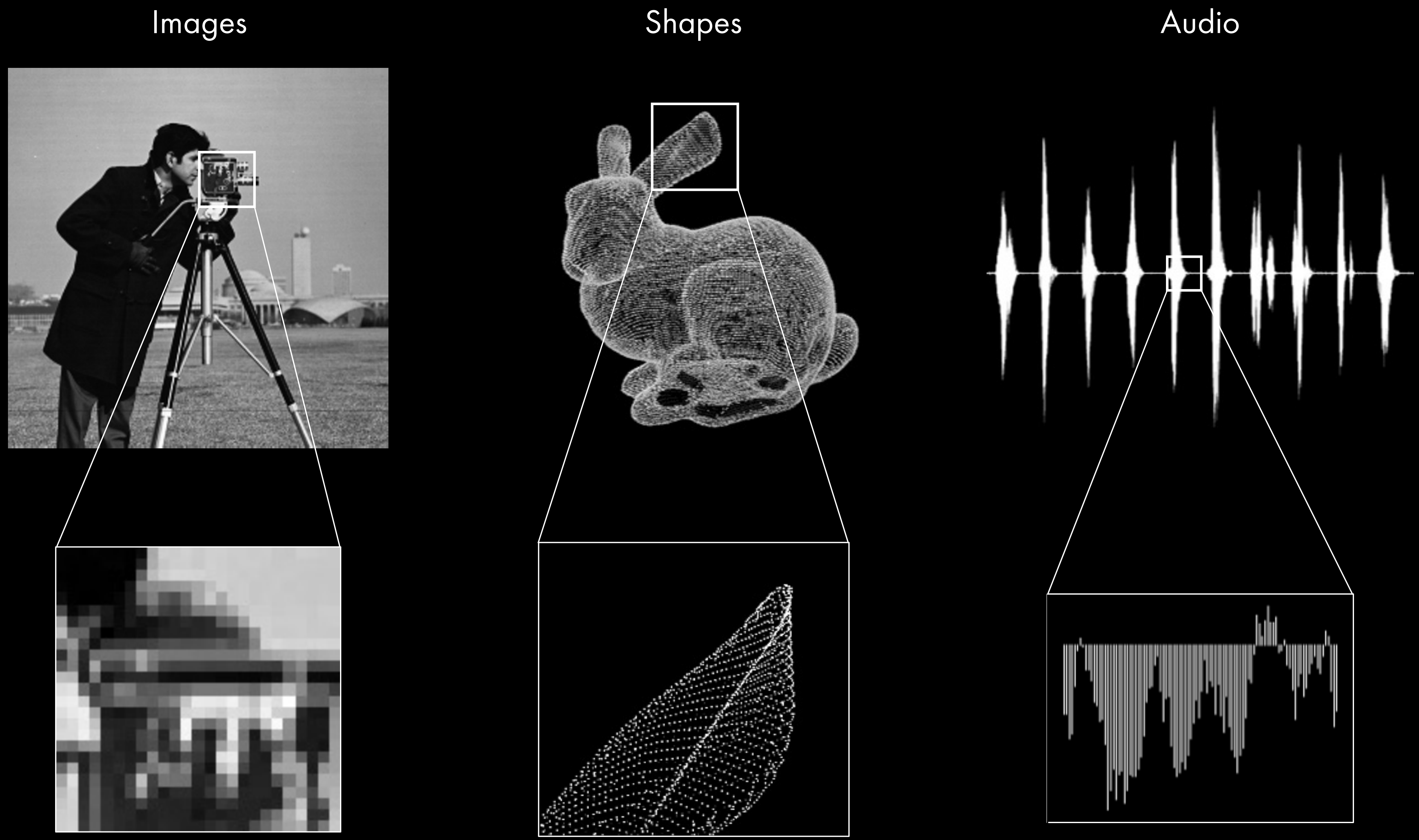
cs.toronto.edu/~lindell/teaching/2529

Announcements

- Poster session is next Thurs. Dec 8!
  - Remember to print your poster ahead of time, follow instructions on Quercus
- Reach out to project mentors with any questions/request feedback
- No in-person lecture next week– guest lecture over zoom

Outline

- Overview of neural fields
- Representing 3D shape
- Neural rendering with NeRF
- Case studies
  - SIREN
  - AutoInt
  - ACORN
  - BACON


Images
Shapes
Audio

Conventional Representations

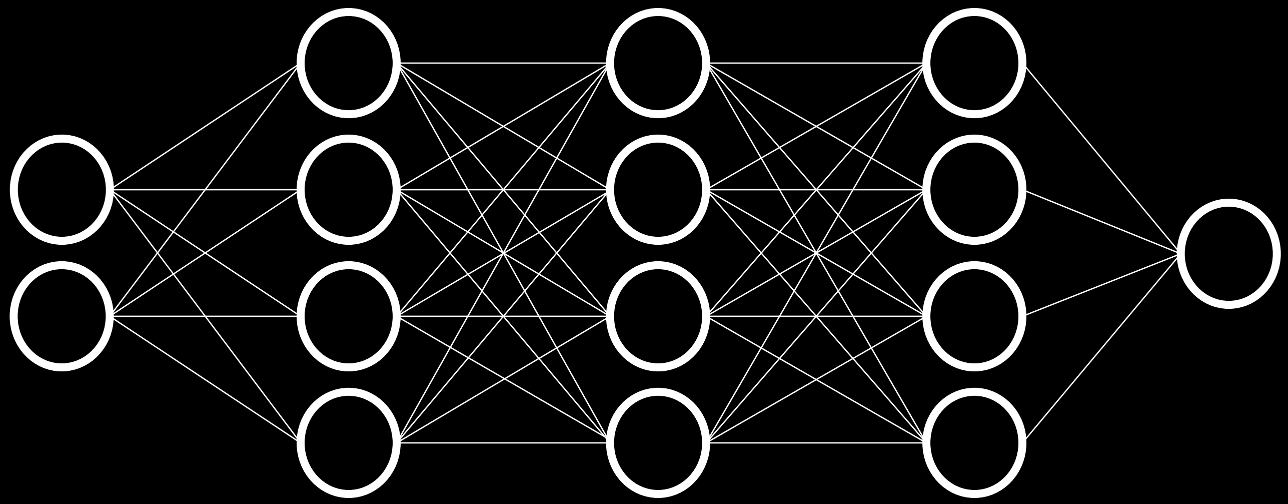
- Number of samples related to global highest frequency (i.e., Nyquist)

- Can be difficult to optimize due to many parameters
  - Need hand-crafted priors for ill-posed problems

- (sparse) basis representations can help, but are not very flexible for high dimensional or multimodal signals

Neural fields

$\mathbf{x}$



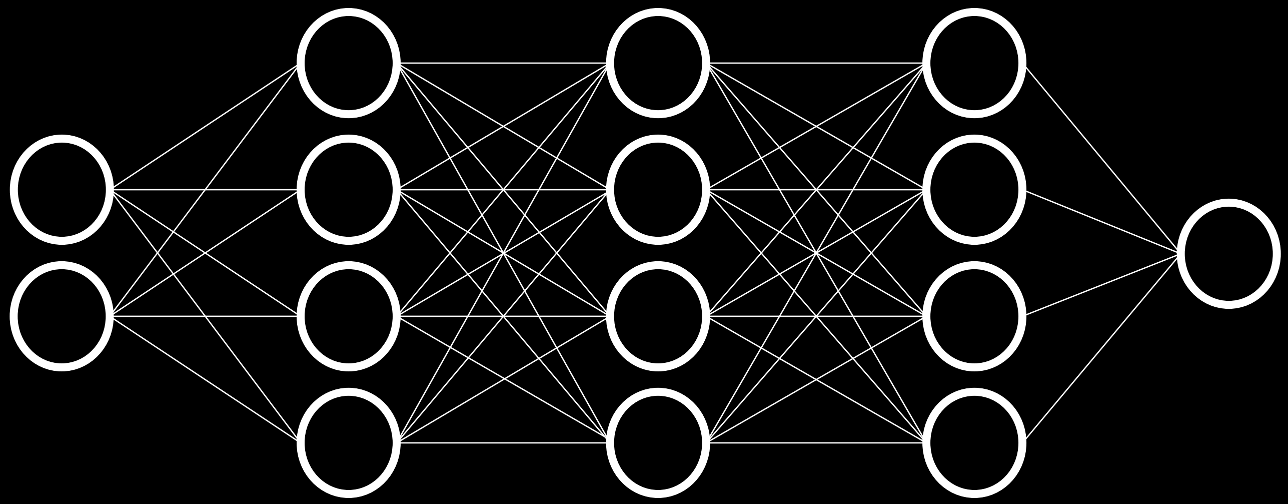
$$\Phi(\mathbf{x}) : \mathbb{R}^2 \mapsto \mathbb{R}$$

Field: a set of numbers along with mathematical operations defined on that set.

CVPR 2022 workshop: “a scalar or vector-valued quantity defined across an input domain”
- Neural field: “a field that is parameterized partly or fully by a neural network”

Neural fields

$\mathbf{x}$



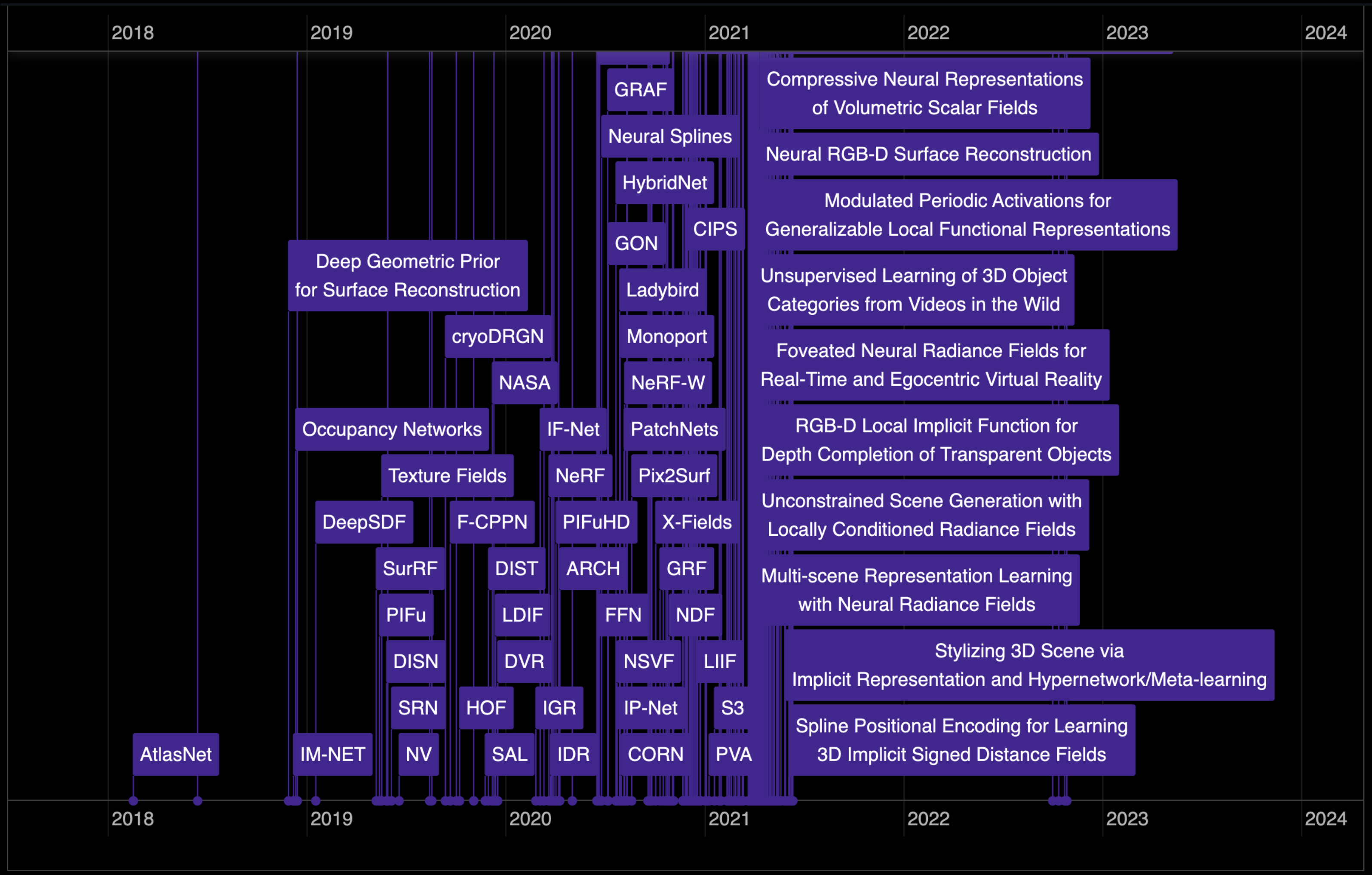
$$\Phi(\mathbf{x}) : \mathbb{R}^2 \mapsto \mathbb{R}$$

pixel coordinates

pixel values

Neural field explosion!

https://neuralfields.cs.brown.edu/



Began around 2019 and accelerated since NeRF (2020)

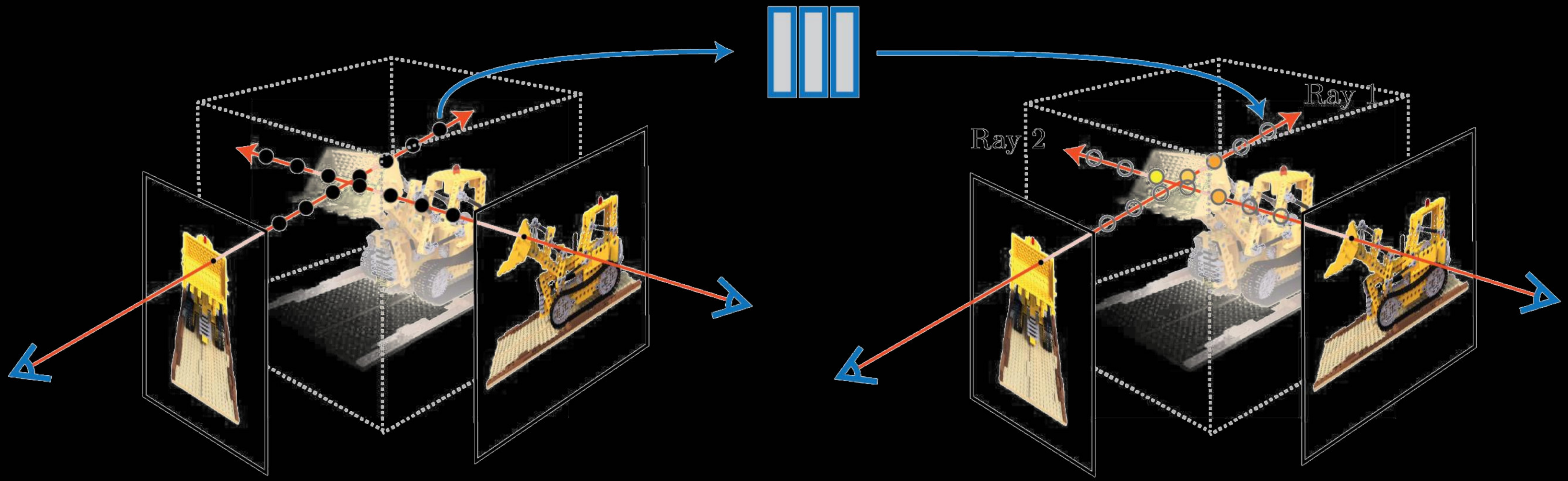
Neural Field Representations

- Compact representations
- Differentiable and easy to optimize (allows learning priors)
- Multi-modal and easy to scale to high dimensions

Compact Representations

This scene is stored in about 30 MB of trained network weights
(optimized from ~2 GB of image pixels)

Differentiable/Easy to Optimize



Becomes trivial to reconstruct 3D appearance and geometry from multiview imagery

slide: Jon Barron

Differentiable/Easy to Optimize

Can learn priors/generative models over a space of signals

[Chan et al. '22]

Multi-modal

[...] an overstuffed pastrami sandwich

[...] a panda wearing a chef's hat and kneading bread dough on a countertop

Luminescent wild horses

A model for text to 3D shape and appearance based on neural fields

[Poole et al. '22]

Outline

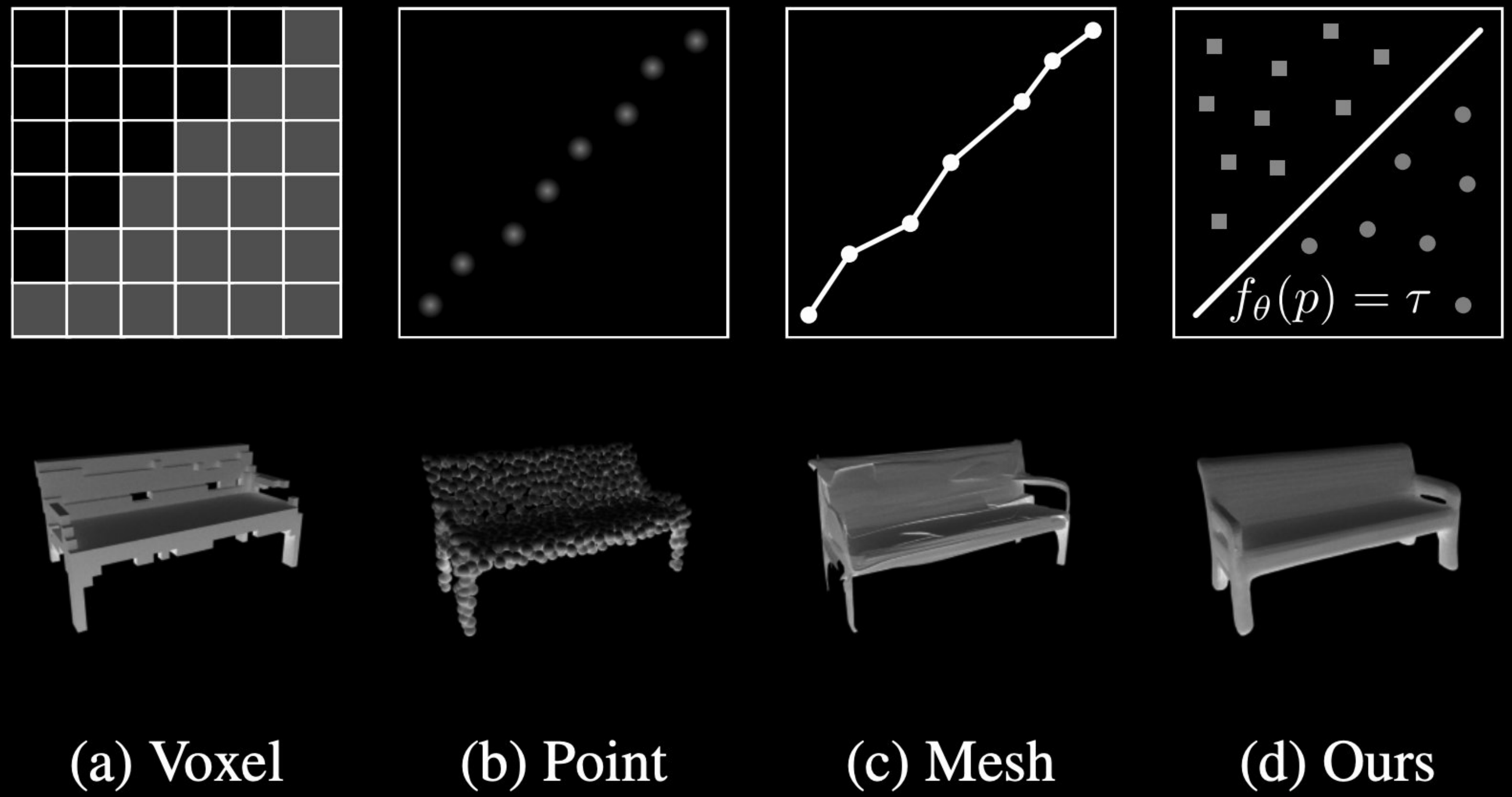
- Overview of neural fields
- Representing 3D shape
- Neural rendering with NeRF
- Case studies
  - SIREN
  - AutoInt
  - ACORN
  - BACON

3D Representations

- Meshes
- Point clouds
- Voxel grids
- ...

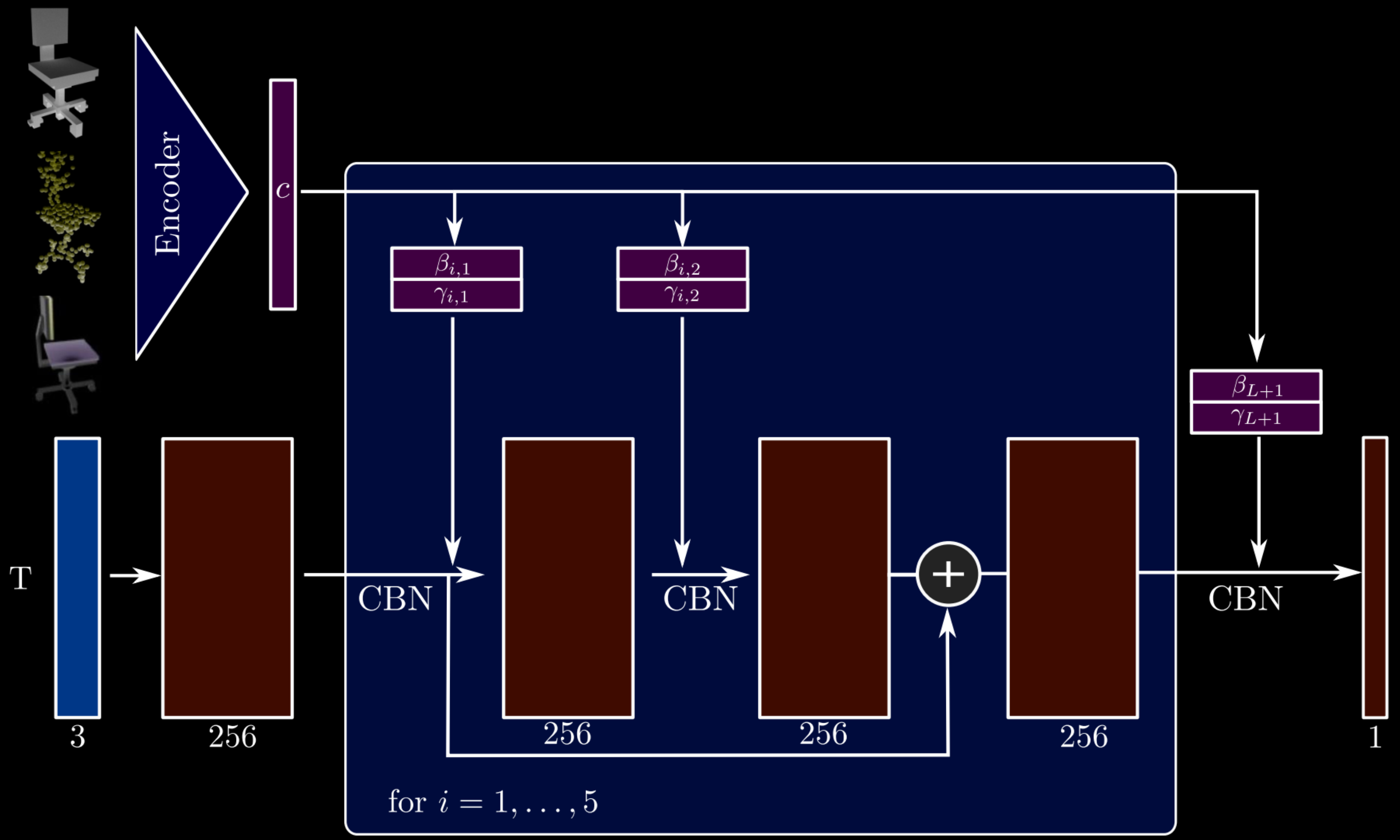
- Cumbersome to represent with a neural network!

Occupancy Fields



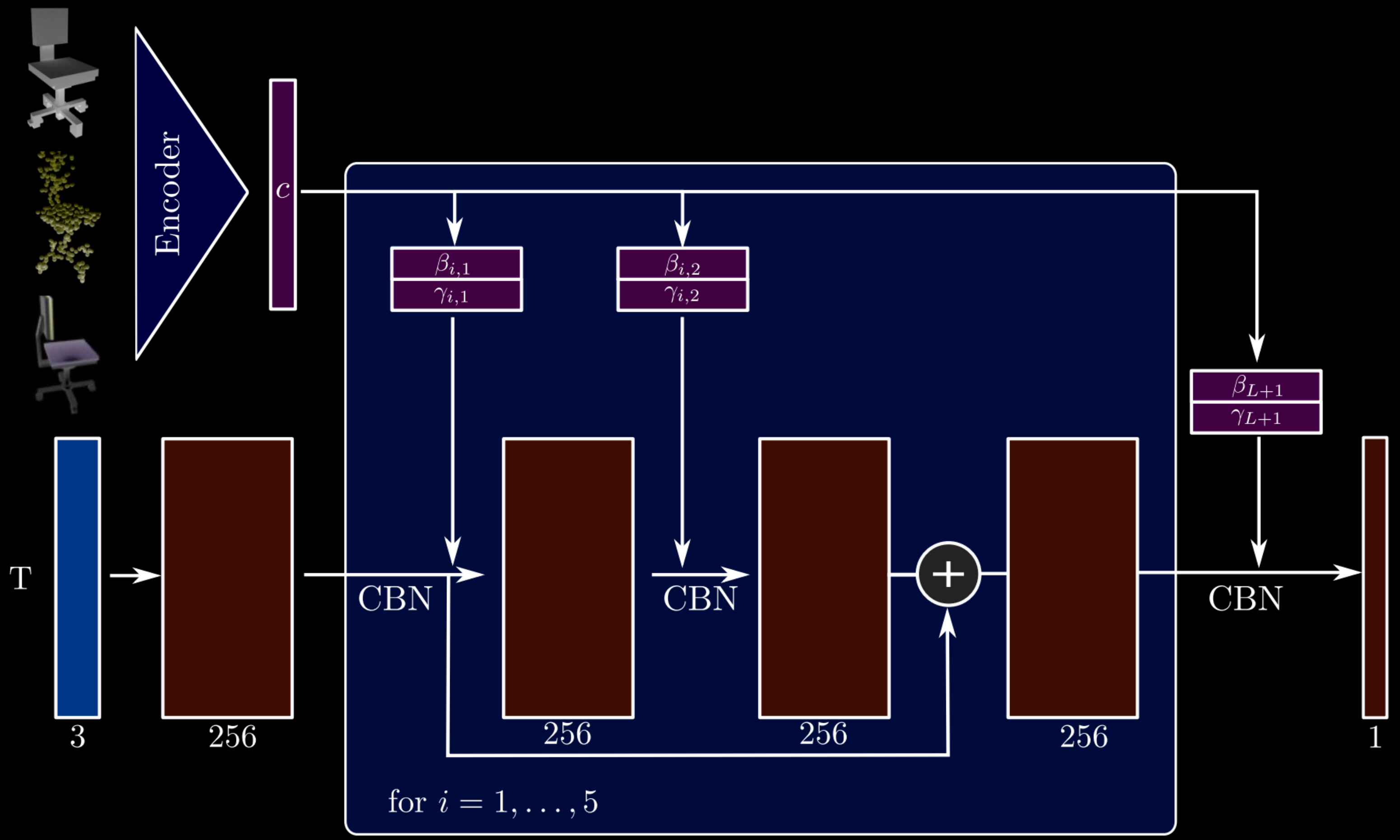
[Mescheder et al. '19]

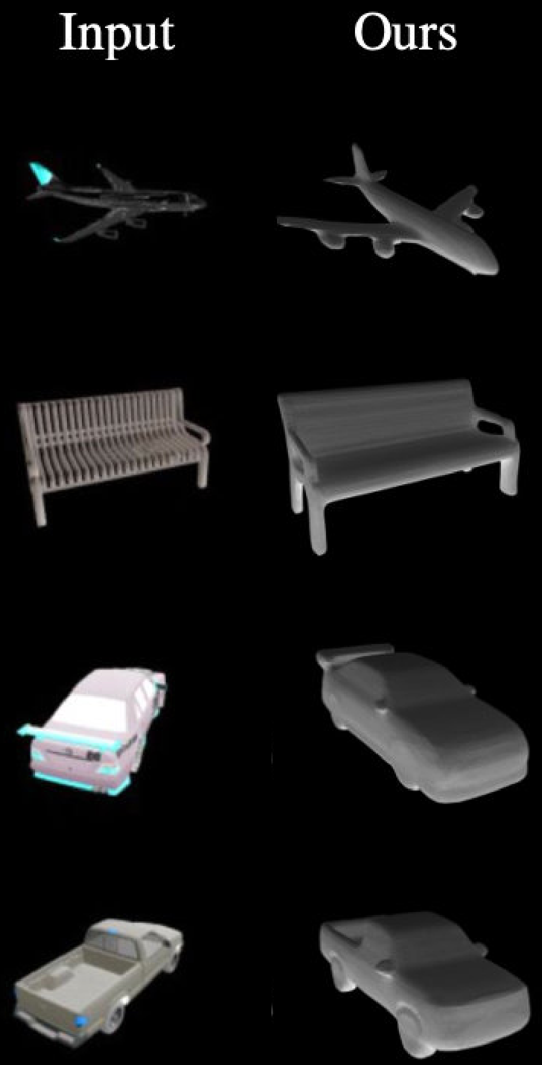
Occupancy Fields



[Mescheder et al. '19]

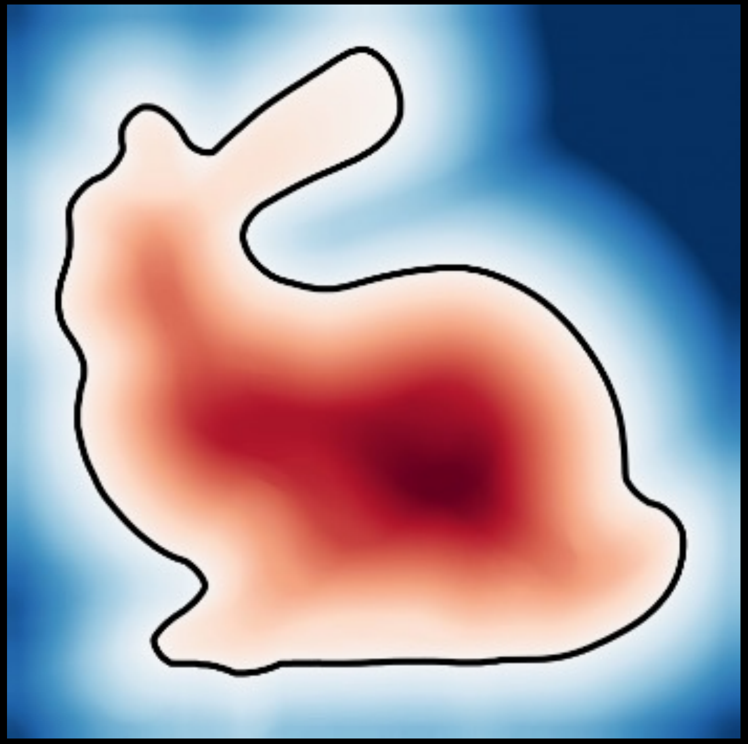
Occupancy Fields



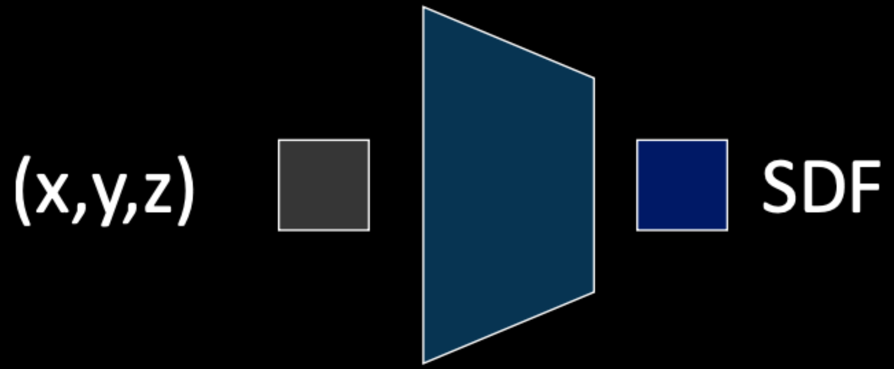


[Mescheder et al. '19]

Signed Distance Field



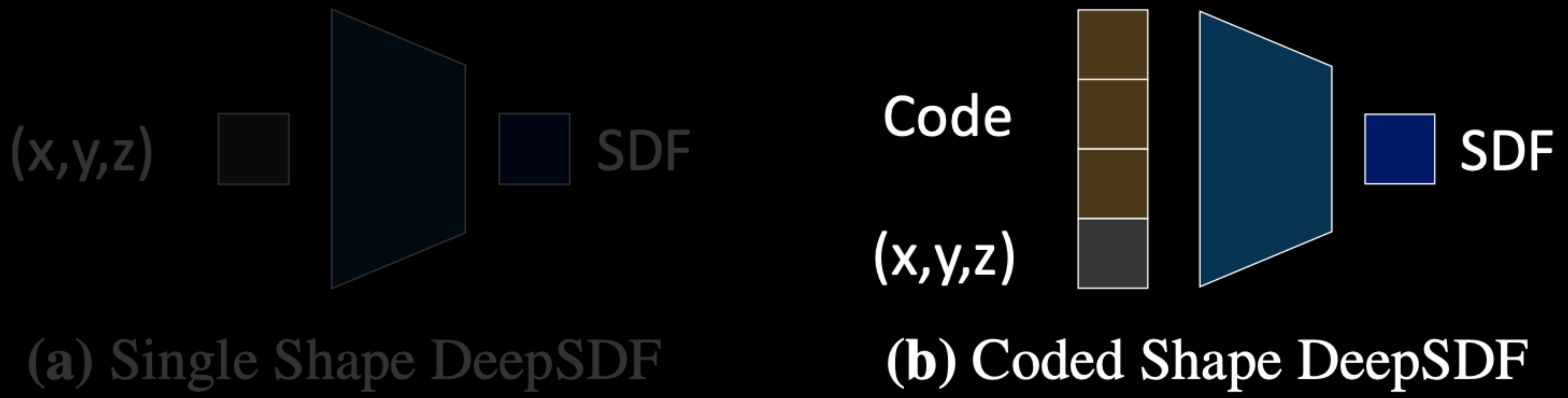
Signed Distance Field



(a) Single Shape DeepSDF

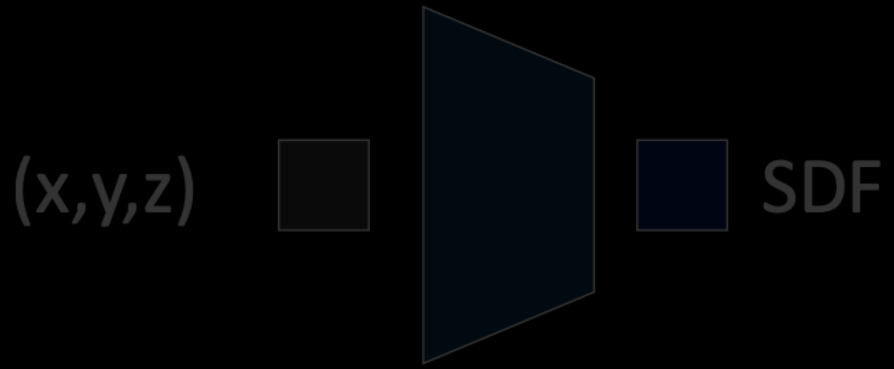
[Park et al. '19]

Signed Distance Field

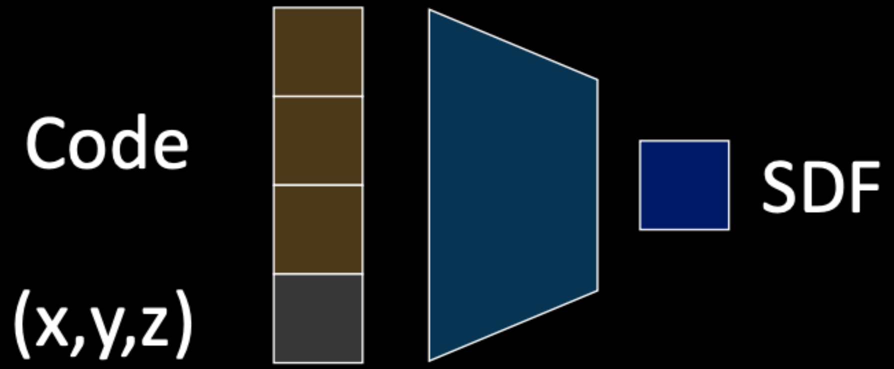


[Park et al. '19]

Signed Distance Field



(a) Single Shape DeepSDF



(b) Coded Shape DeepSDF

(a) Noisy Input Point Cloud

(b) Shape Completion

[Park et al. '19]

NeuS: Learning Neural Implicit Surfaces by Volume Rendering for Multi-view Reconstruction

Peng Wang[†], Li

†{pwar

We preser
structing
neural sur
and IDR [
trapped in
with sever
for novel
use volum
optimizati
surfaces f
not suffici
represent
develop a
observe th
errors (i.e
tion that i
accurate s
on the DT
state-of-th
scenes wi

Deep Local Shapes: Learning Local SDF Priors for Detailed 3D Reconstruction

Rohan Chabra
Str

Fig. 1: Reconstru
of Calais scene [5
codes in a voxel g
SDF function, wh

PIFu: Pixel-Aligned Implicit Function for High-Resolution Clothed Human Digitization

Shunsuke Saito[1,2*] Zeng Huang[1,2*] Ryota Natsume[3*]
Shigeo Morishima[3] Angjoo Kanazawa[4] Hao Li[1,2,5]

[1]University of Southern California [2]USC Institute for Creative Technologies
[3]Waseda University [4]University of California, Berkeley [5]Pinscreen

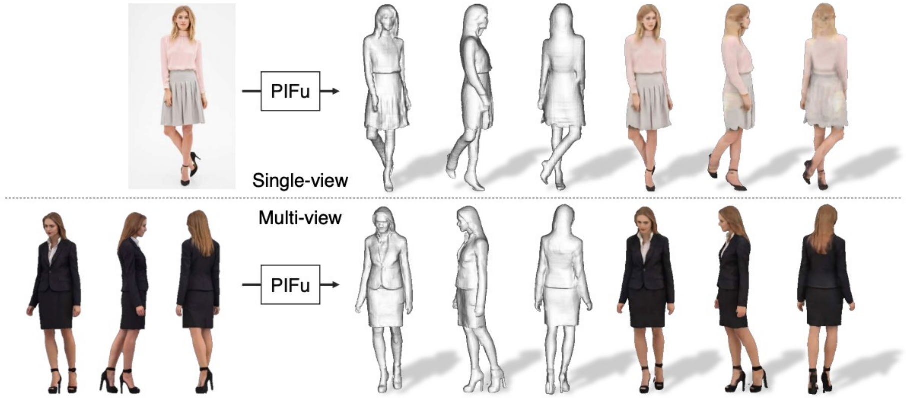


Figure 1: **Pixel-aligned Implicit function (PIFu):** We present pixel-aligned implicit function (PIFu), which allows recovery of high-resolution 3D textured surfaces of clothed humans from a single input image (top row). Our approach can digitize intricate variations in clothing, such as wrinkled skirts and high-heels, including complex hairstyles. The shape and textures can be fully recovered including unseen regions such as the back of the subject. PIFu can be also extended to multi-view input images (bottom row).

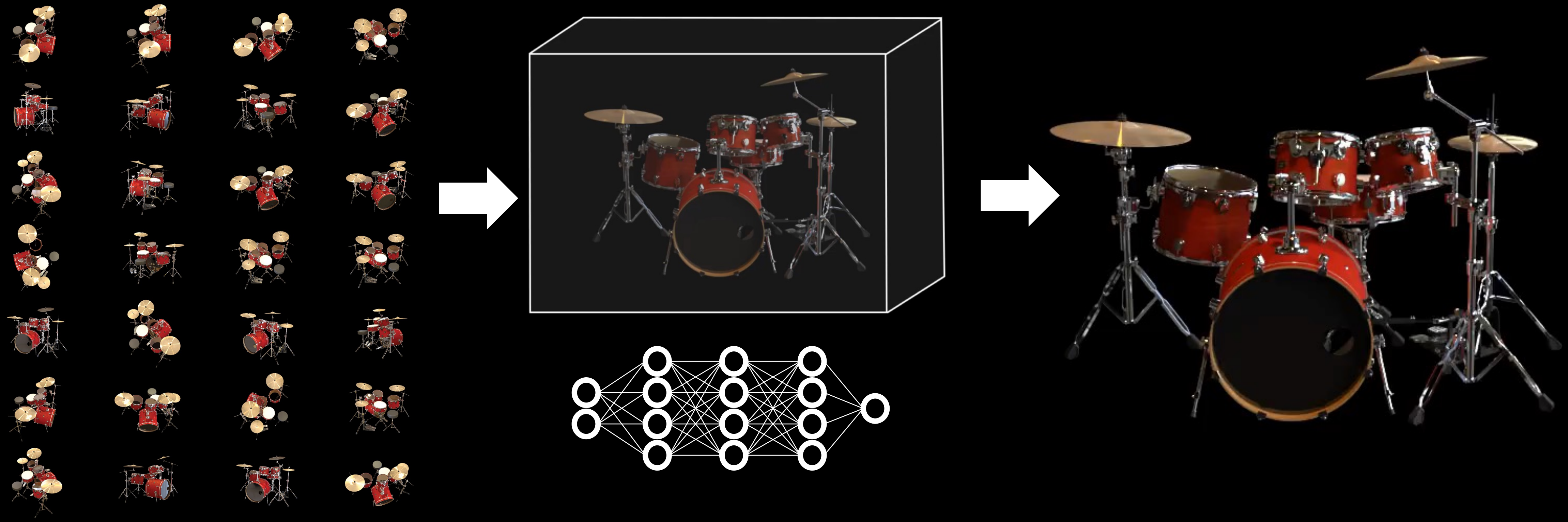
Outline

- Overview of neural fields
- Representing 3D shape
- Neural rendering with NeRF
- Case studies
    - SIREN
    - AutoInt
    - ACORN
    - BACON

Neural Radiance Fields

[Mildenhall et al. '20]

Mildenhall et al. ’20
Liu et al. ’20
Schwarz et al. ’20
...



Training Set

Learned Volume

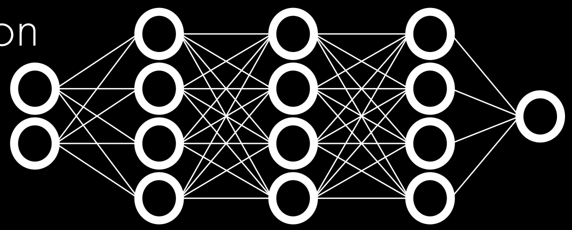
Novel Views (from VRE)

Volume Rendering Equation (VRE)

$$\underbrace{\mathbf{C}(\mathbf{r})}_{\text{color of rendered ray}} = \int_{t_n}^{t_f} \underbrace{\sigma(\mathbf{r}(t))}_{\substack{\text{absorption}\\ \text{coefficient}}} e\underbrace{^{\left\{-\int_{t_n}^{t}\sigma(\mathbf{r}(s))\,\mathrm{d}s\right\}}}_{\text{transmittance}} \underbrace{\mathbf{c}(\mathbf{r}(t))}_{\substack{\text{emissive}\\ \text{radiance}}}\,\mathrm{d}t$$

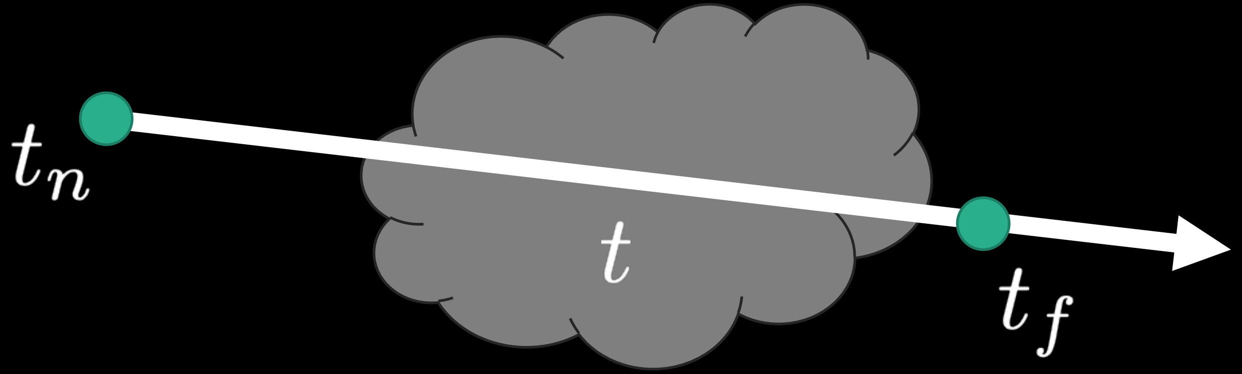
position,
direction

$\mathbf{r}(t)$



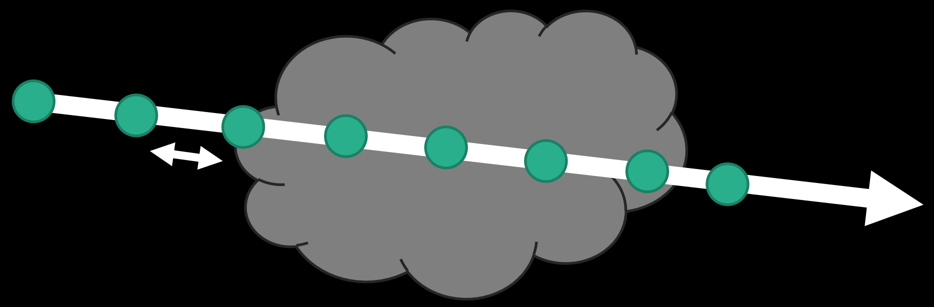
Volume Rendering Equation (VRE)

$$\underbrace{\mathbf{C}(\mathbf{r})}_{\text{color of rendered ray}} = \int_{t_n}^{t_f} \underbrace{\sigma(\mathbf{r}(t))}_{\substack{\text{absorption}\\\text{coefficient}}} \underbrace{e^{\left\{-\int_{t_n}^{t} \sigma(\mathbf{r}(s))\,\mathrm{d}s\right\}}}_{\text{transmittance}} \underbrace{\mathbf{c}(\mathbf{r}(t))}_{\substack{\text{emissive}\\\text{radiance}}} \,\mathrm{d}t$$



Discretizing the VRE

$$\underbrace{\mathbf{C}(\mathbf{r})}_{\text{color of rendered ray}} = \int_{t_n}^{t_f} \underbrace{\sigma(\mathbf{r}(t))}_{\text{absorption coefficient}} \underbrace{e^{\left\{-\int_{t_n}^{t} \sigma(\mathbf{r}(s))\,\mathrm{d}s\right\}}}_{\text{transmittance}} \underbrace{\mathbf{c}(\mathbf{r}(t))}_{\text{emissive radiance}}\,\mathrm{d}t$$



How much light is emitted along a single section? Assuming constant radiance...

Discretizing the VRE

$$\underbrace{\mathbf{C}(\mathbf{r})}_{\text{color of rendered ray}} = \int_{t_n}^{t_f} \underbrace{\sigma(\mathbf{r}(t))}_{\substack{\text{absorption}\\ \text{coefficient}}} \underbrace{e^{\left\{-\int_{t_n}^{t} \sigma(\mathbf{r}(s))\,\mathrm{d}s\right\}}}_{\text{transmittance}} \underbrace{\mathbf{c}(\mathbf{r}(t))}_{\substack{\text{emissive}\\ \text{radiance}}}\,\mathrm{d}t$$

- **Assume constant radiance**

$$\approx \mathbf{c} \int_{t_n}^{t_f} \sigma(t) e^{-\int_{t_n}^{t} \sigma(s)\mathrm{d}s} \mathrm{d}t$$

Discretizing the VRE

$$\approx \mathbf{c} \int_{t_n}^{t_f} \sigma(t) e^{-\int_{t_n}^{t} \sigma(s) \mathrm{d}s} \mathrm{d}t$$

Discretizing the VRE

$$\approx \mathbf{c} \int_{t_n}^{t_f} \sigma(t) e^{-\int_{t_n}^{t} \sigma(s) \mathrm{d}s} \mathrm{d}t$$

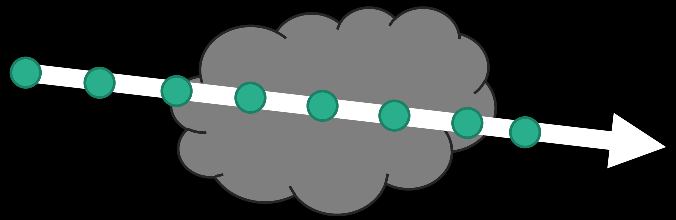
$$= \mathbf{c} \int_{t_n}^{t_f} -\frac{\mathrm{d}}{\mathrm{d}t} e^{-\int_{t_n}^{t} \sigma(s) \mathrm{d}s} \mathrm{d}t$$

Discretizing the VRE

$$\approx \mathbf{c} \int_{t_n}^{t_f} \sigma(t) e^{-\int_{t_n}^{t} \sigma(s)\mathrm{d}s} \mathrm{d}t$$

$$= \mathbf{c} \int_{t_n}^{t_f} -\frac{\mathrm{d}}{\mathrm{d}t} e^{-\int_{t_n}^{t} \sigma(s)\mathrm{d}s} \mathrm{d}t$$

$$= \mathbf{c} \left(1 - e^{-\int_{t_n}^{t} \sigma(s)\mathrm{d}s}\right)$$



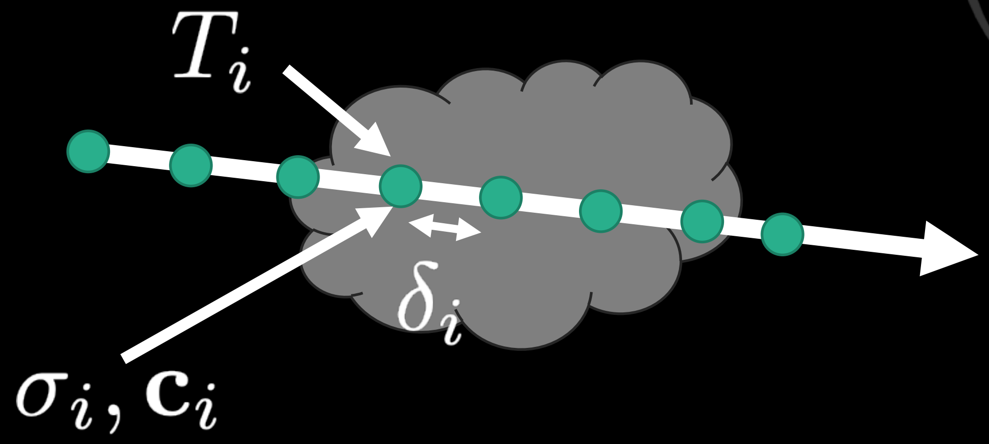
Light emitted along one section

Discretizing the VRE

$$= \mathbf{c}\left(1 - e^{-\int_{t_n}^{t} \sigma(s)\mathrm{d}s}\right)$$

discretize and
sum over sections

$$\approx \sum_{i=1}^{N} (1 - \exp(-\sigma_i \delta_i)) \underbrace{\exp\left(-\sum_{j=1}^{i-1} \sigma_j \delta_j\right)}_{\text{transmittance}} \mathbf{c}_i$$



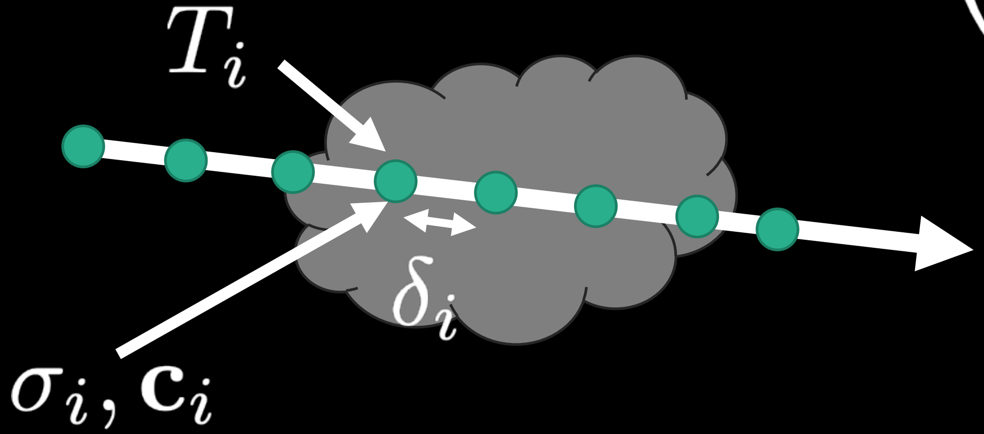
Discretizing the VRE

$$= \mathbf{c}\left(1 - e^{-\int_{t_n}^{t} \sigma(s)\mathrm{d}s}\right)$$

discretize and
sum over sections

$$\approx \sum_{i=1}^{N} (1 - \exp(-\sigma_i \delta_i)) \exp\left(-\sum_{j=1}^{i-1} \sigma_j \delta_j\right) \mathbf{c}_i$$

transmittance



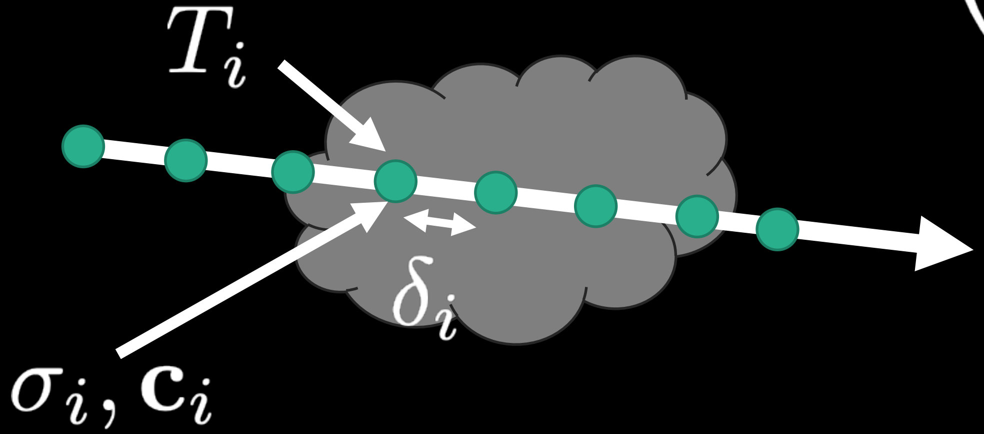
Discretizing the VRE

$$= \mathbf{c}\left(1 - e^{-\int_{t_n}^{t}\sigma(s)\mathrm{d}s}\right)$$

discretize and
sum over sections

$$\approx \sum_{i=1}^{N}(1-\exp(-\sigma_i\delta_i))\exp\left(-\sum_{j=1}^{i-1}\sigma_j\delta_j\right)\mathbf{c}_i$$

transmittance T_i

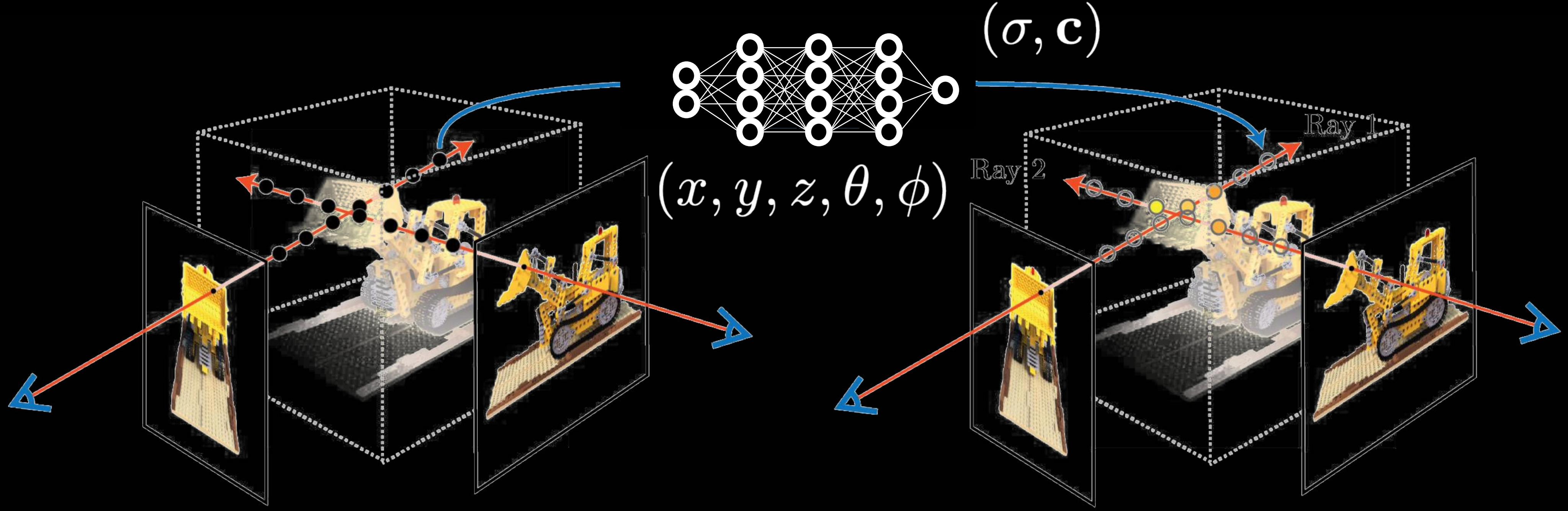


Discretizing the VRE

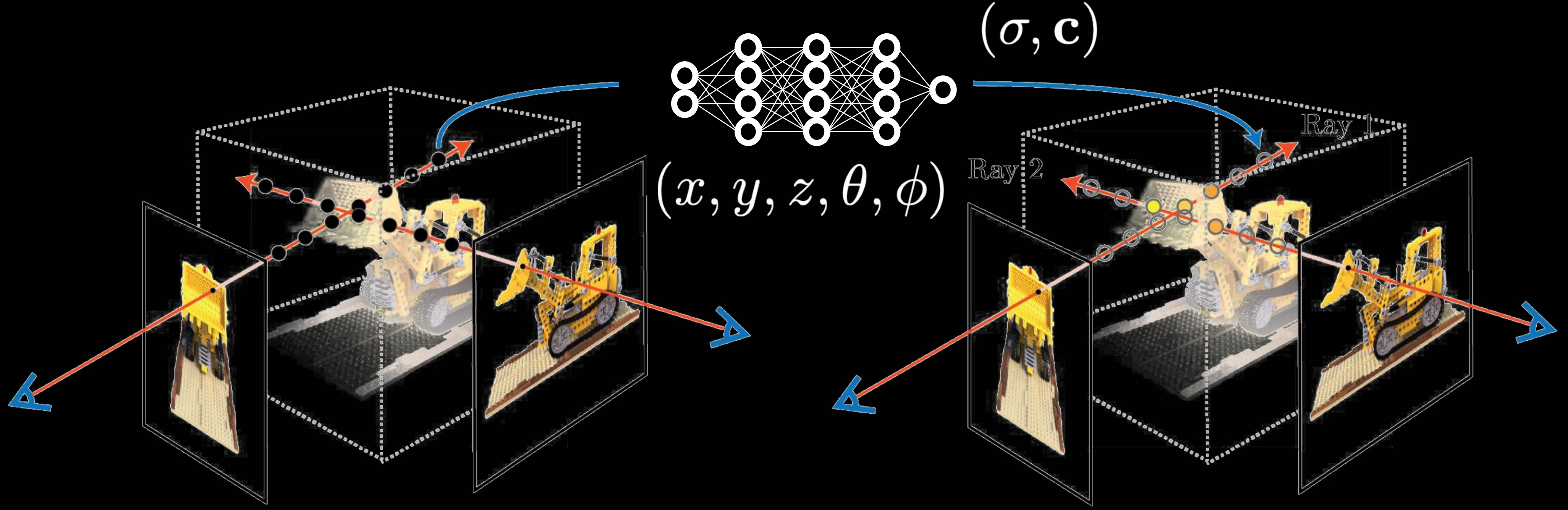
$$\underbrace{\mathbf{C}(\mathbf{r})}_{\text{color of rendered ray}} = \int_{t_n}^{t_f} \underbrace{\sigma(\mathbf{r}(t))}_{\text{absorption coefficient}} \underbrace{e^{\left\{-\int_{t_n}^{t} \sigma(\mathbf{r}(s))\,\mathrm{d}s\right\}}}_{\text{transmittance}} \underbrace{\mathbf{c}(\mathbf{r}(t))}_{\text{emissive radiance}}\,\mathrm{d}t$$

$$\approx \sum_{i=1}^{N} (1 - \exp(-\sigma_i \delta_i)) \exp\left(-\sum_{j=1}^{i-1} \sigma_j \delta_j\right) \mathbf{c}_i$$

Neural Radiance Fields



Neural Radiance Fields



$$\min_{\sigma,\mathbf{c}} \sum_i \|\text{Render}(I^{(i)}) - I_{\text{GT}}^{(i)}\|$$

- Given images with known camera positions
- Sample along rays
- Optimize the absorption and radiance to minimize photometric error!

Positional Encoding

Naïve NeRF produces blurry results...

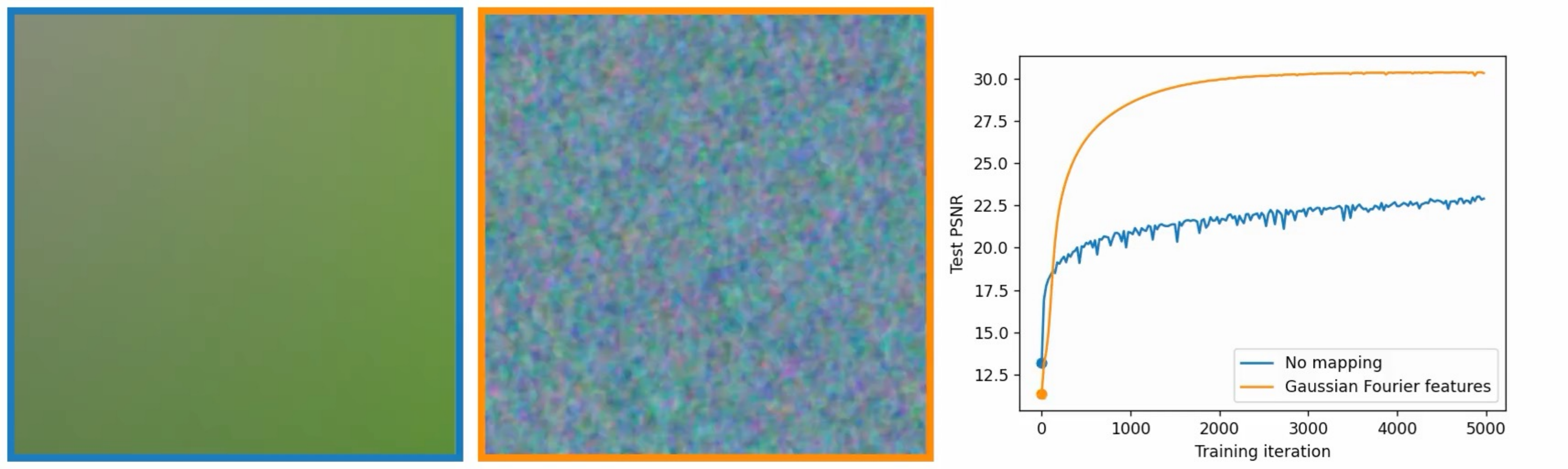
Positional Encoding

Naïve NeRF produces blurry results...

Result with positional encoding

Positional Encoding

Toy example of image fitting

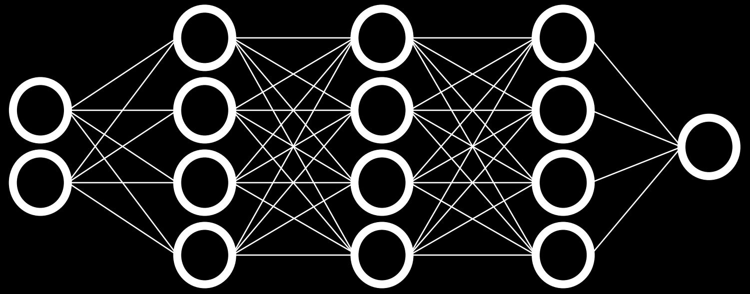


Without Pos. Enc. With Pos. Enc

Positional Encoding

- Simple trick!
  - Instead of passing in v =(x, y) into the network we pass

$$\begin{bmatrix} \cos(\pi v) & \sin(\pi v) \\ \cos(2\pi v) & \sin(2\pi v) \\ \cos(4\pi v) & \sin(4\pi v) \\ \vdots & \\ \cos(2^{L-1}\pi v) & \sin(2^{L-1}\pi v) \end{bmatrix} \Rightarrow$$



Positional Encoding

- Why does this work?

- Explained with theory from Neural Tangent Kernel
  - Training a network is similar to kernel regression (becomes closer as network layers become wider)

Positional Encoding

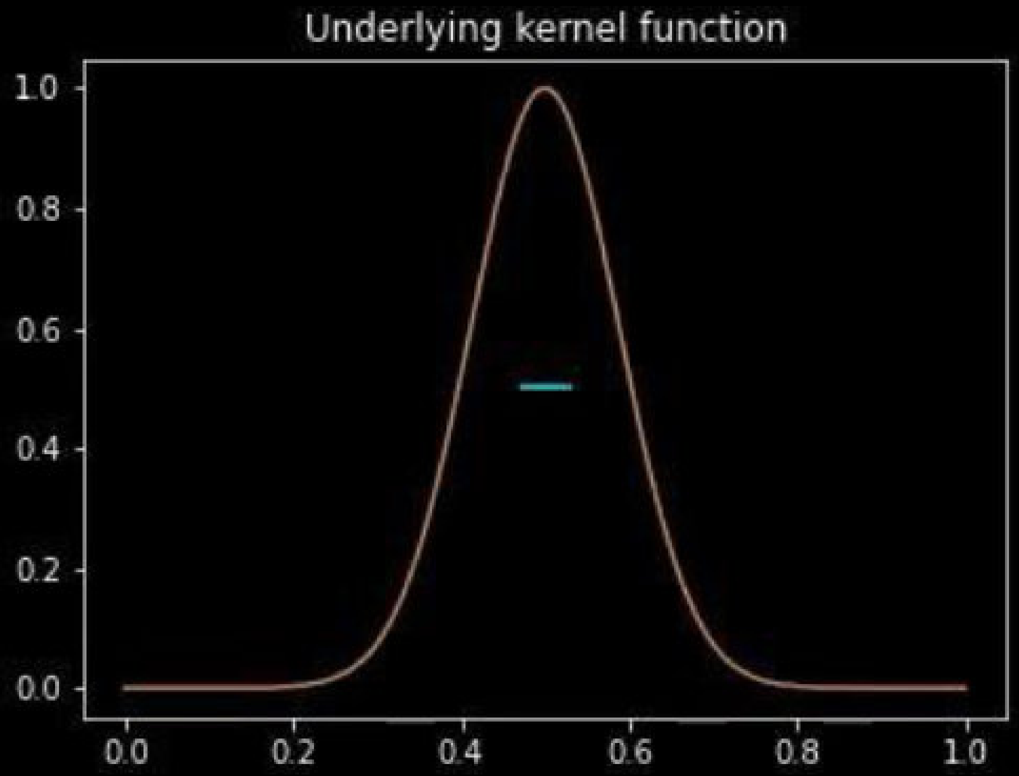
- Kernel Regression

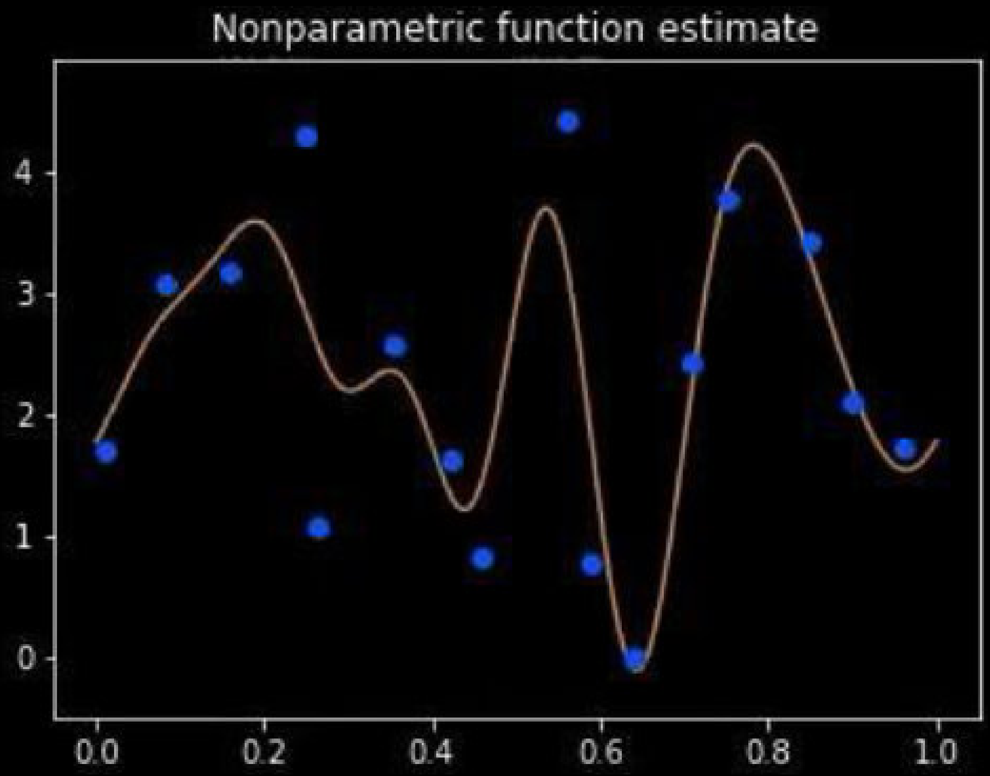
weighting & kernel function computes similarity between input and training points

$$f(x) = \sum_{i=1}^{N} w_i k(x - x_i)$$

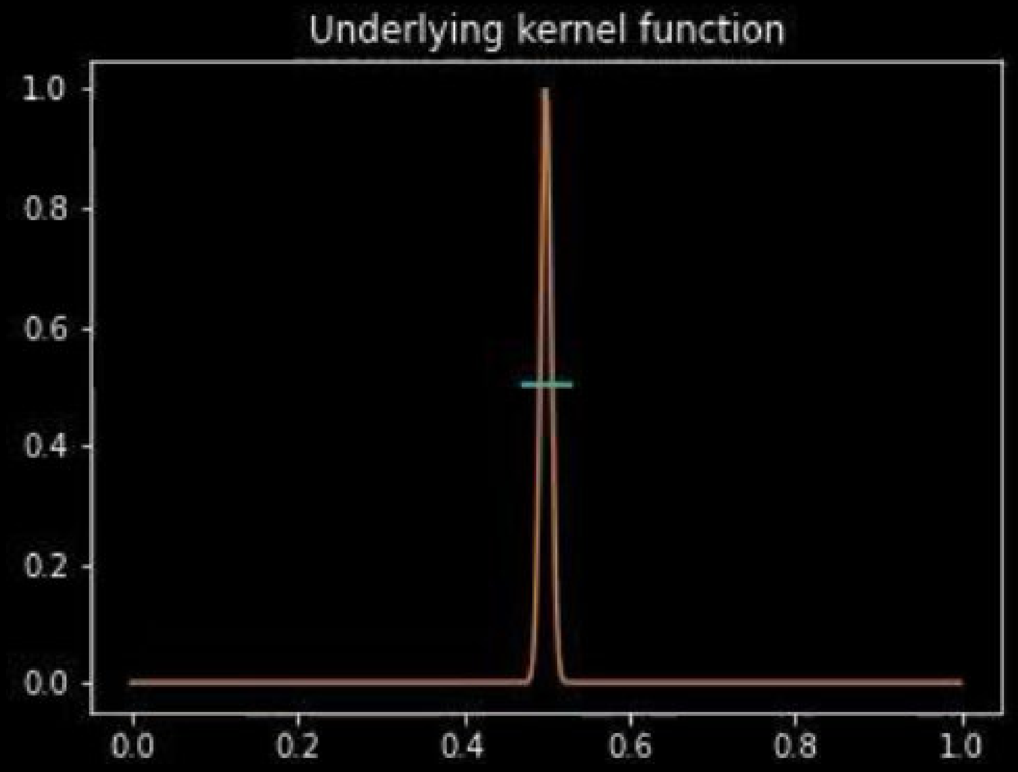
Sum over training points

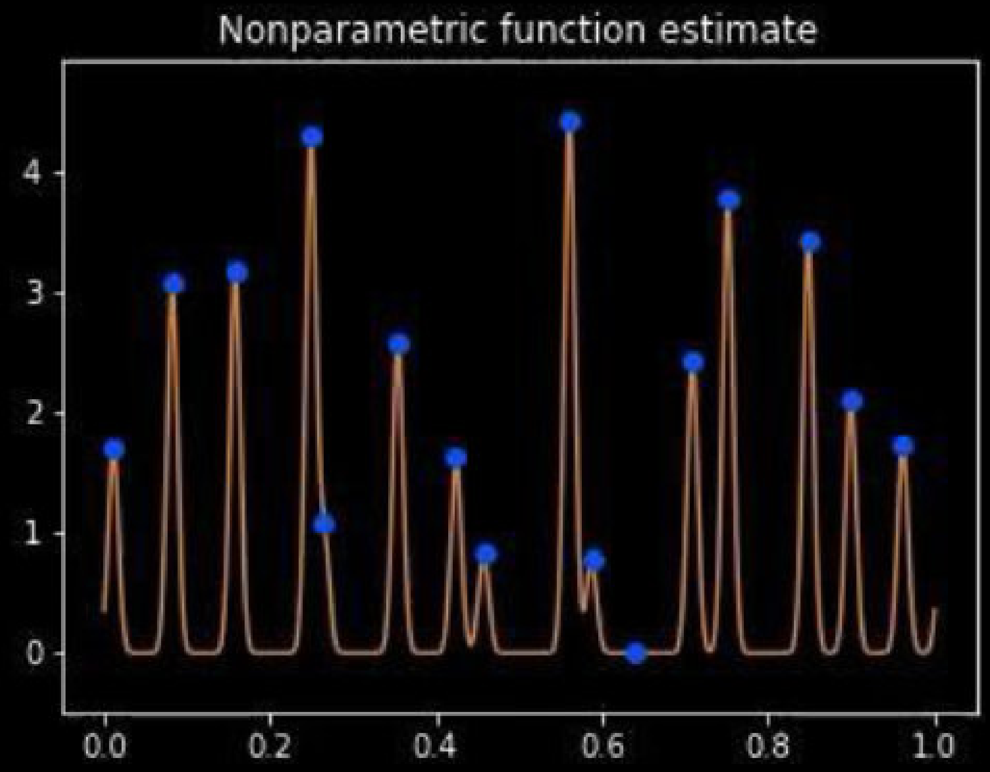
e.g., if the kernel is a Gaussian, this puts a bump at every training data point





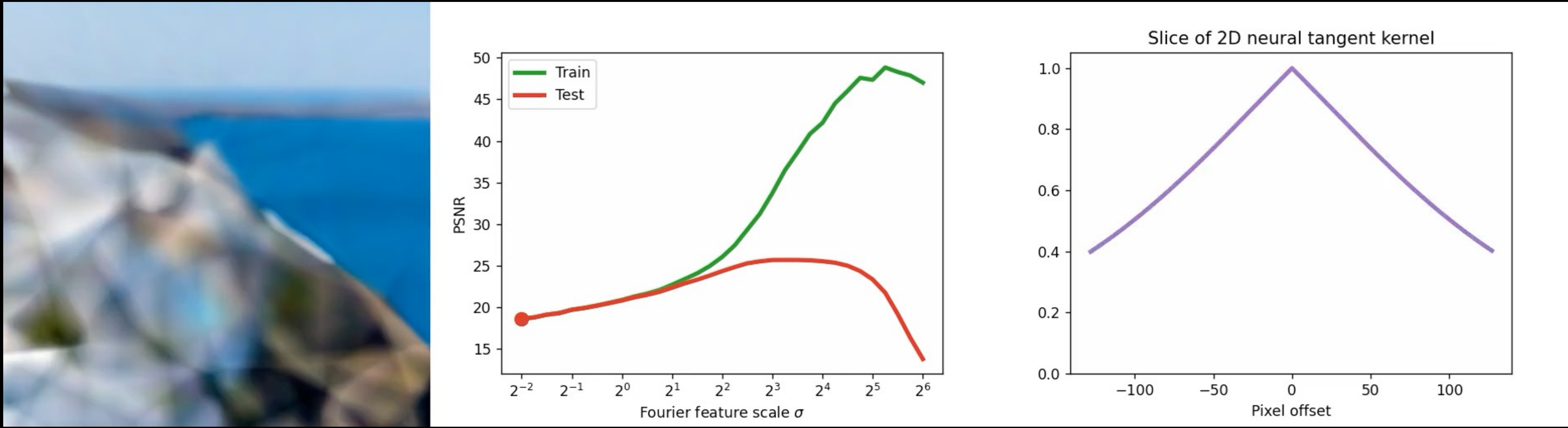
Width of kernel is important to trade off interpolation/overfitting to data!





Width of kernel is important to trade off interpolation/overfitting to data!

NTK Visualization



NeRF Results

Detailed geometry (depth map visualizes the location of the mean of the absorption)

NeRF Results

View dependent effects (right is fixing the ray position, but feeding different ray direction)

Outline

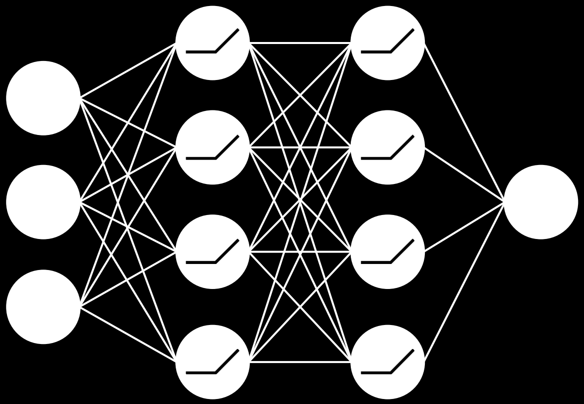
- Overview of neural fields
- Representing 3D shape
- Neural rendering with NeRF
- Case studies
  - SIREN
  - AutoInt
  - ACORN
  - BACON

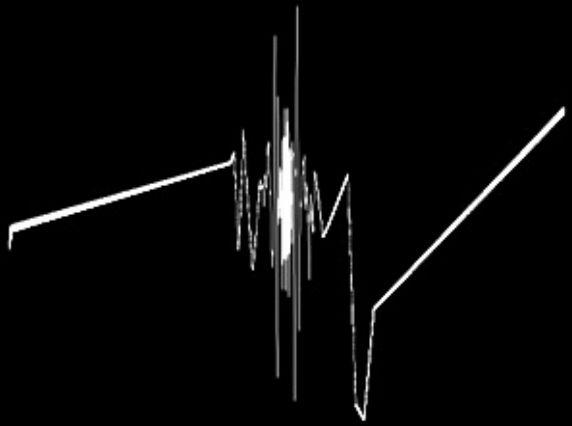
Images

Shapes

Audio

ReLU MLP

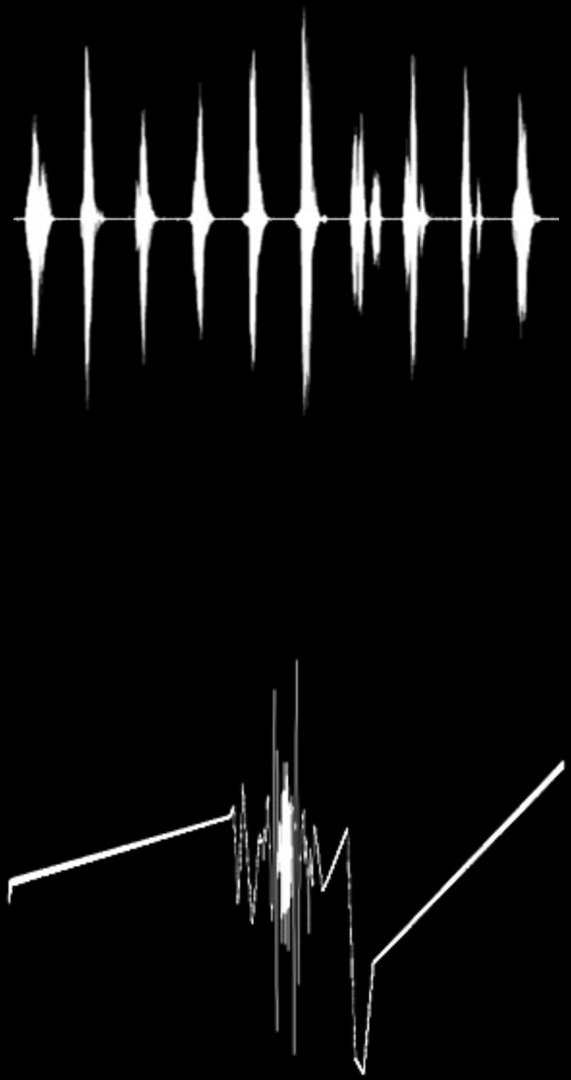




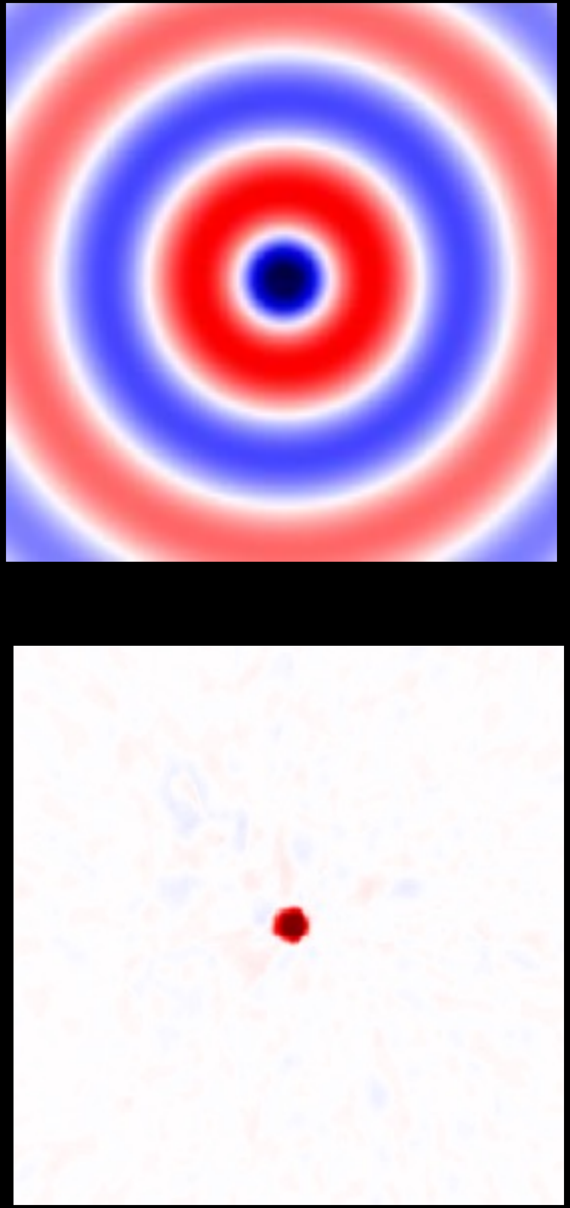
Images

Shapes

Audio

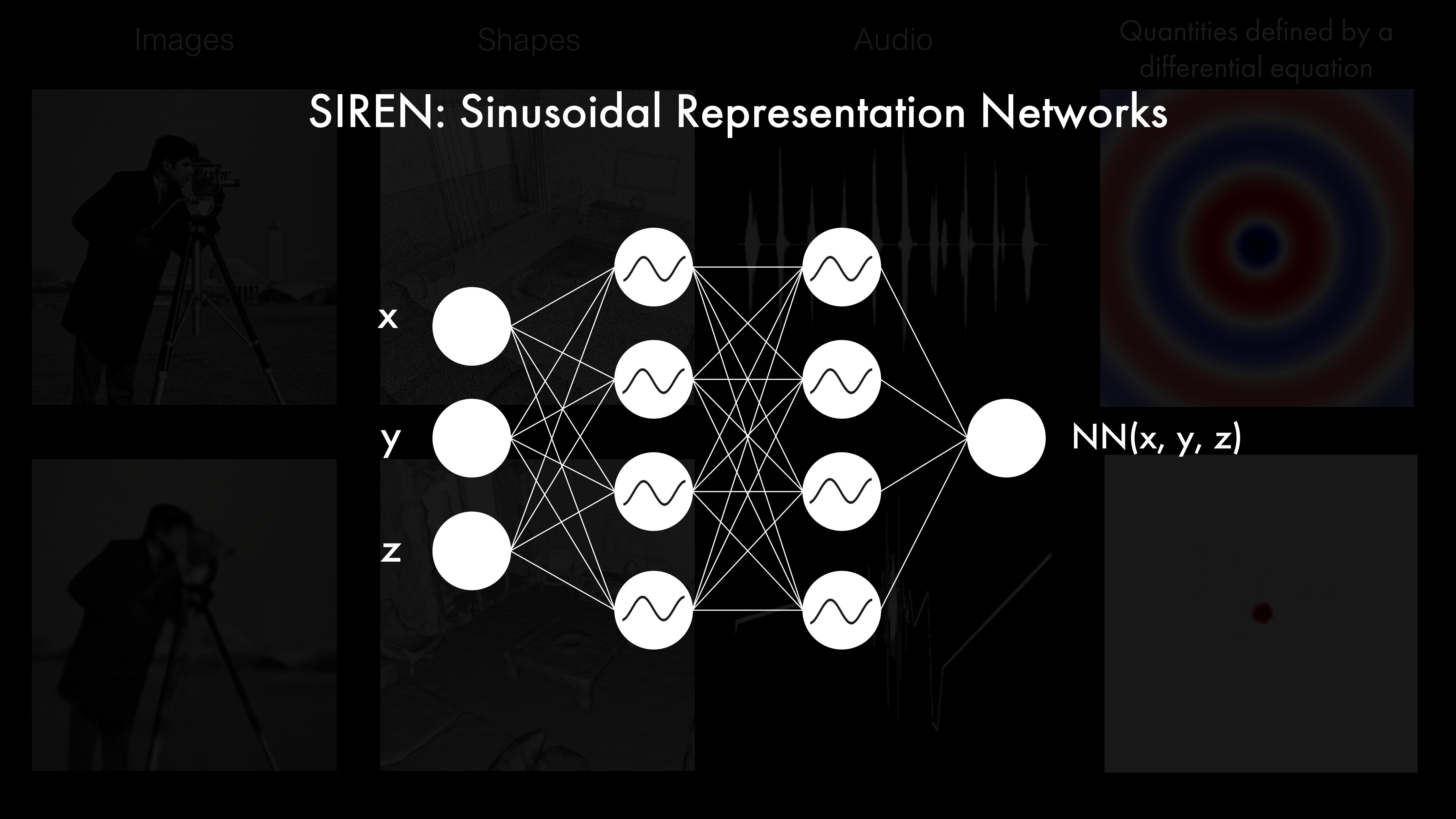


Quantities defined by a differential equation

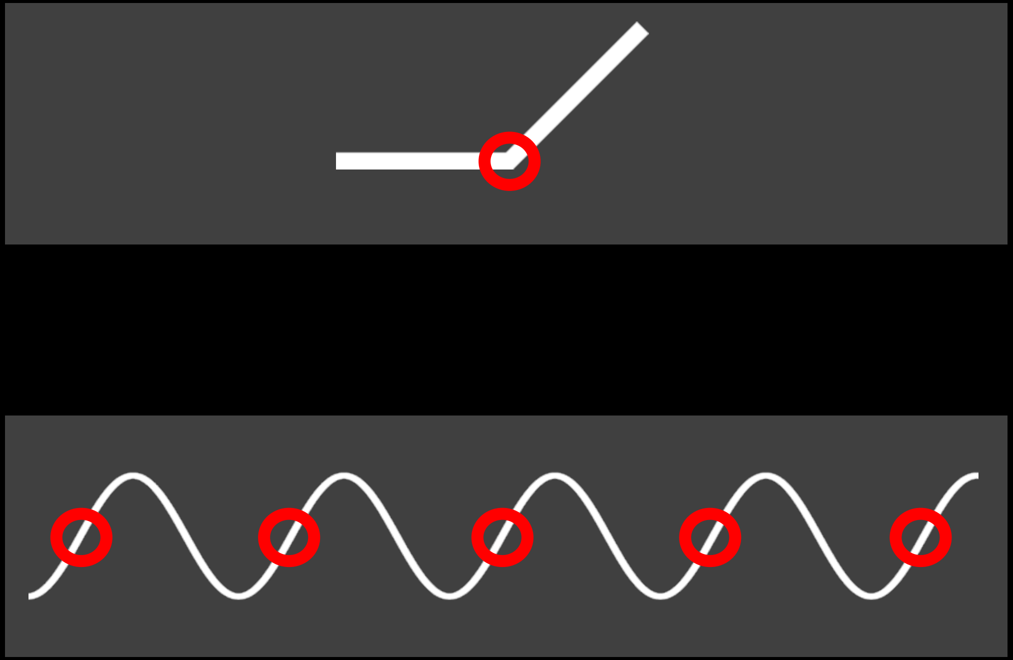


Images Shapes Audio Quantities defined by a differential equation

SIREN: Sinusoidal Representation Networks



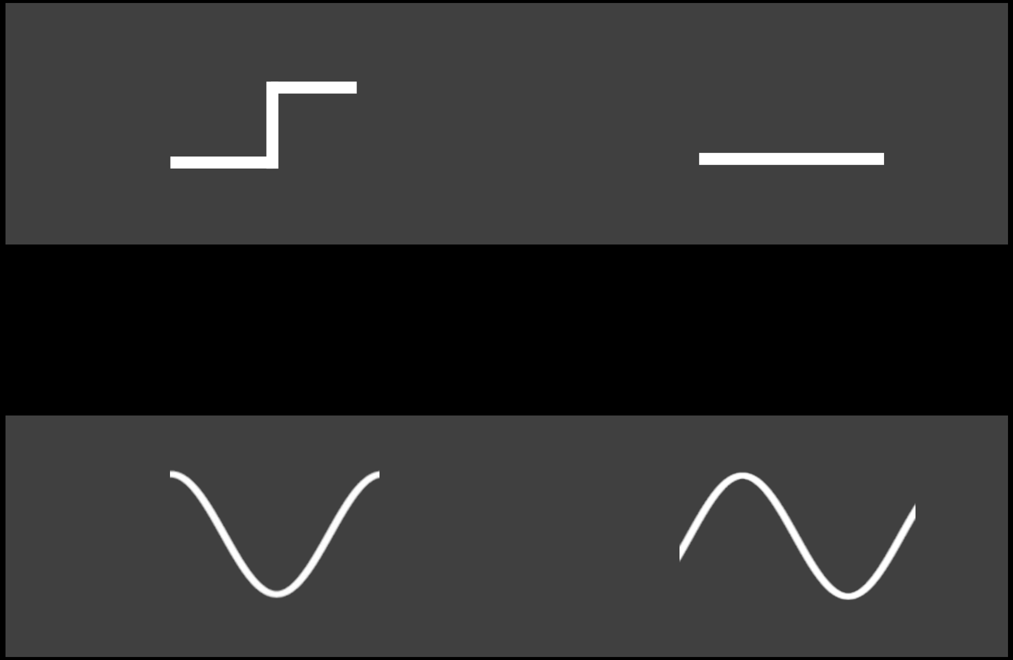
Locality



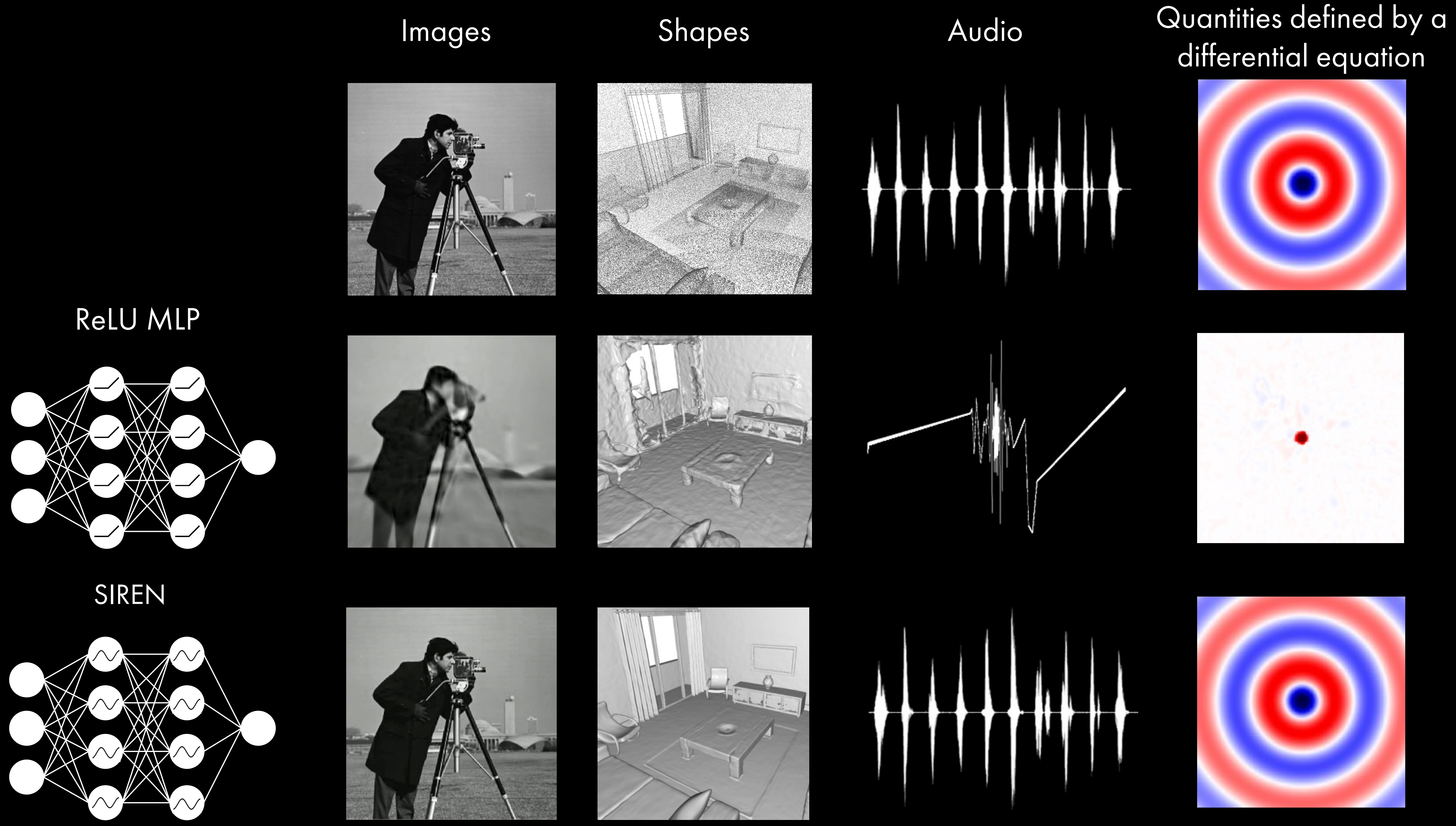
Periodicity allows SIREN to replicate activations across the input domain

Derivatives

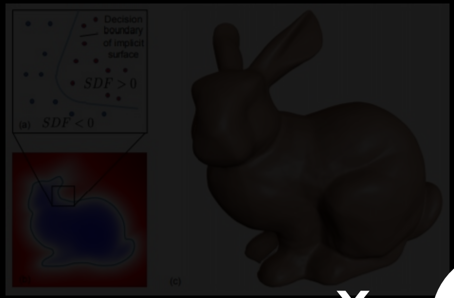
$\partial/\partial x$ $\partial^2/\partial x^2$



All derivatives exist, are nonzero and bounded by 1


Images
Shapes
Audio
Quantities defined by a differential equation
ReLU MLP
SIREN

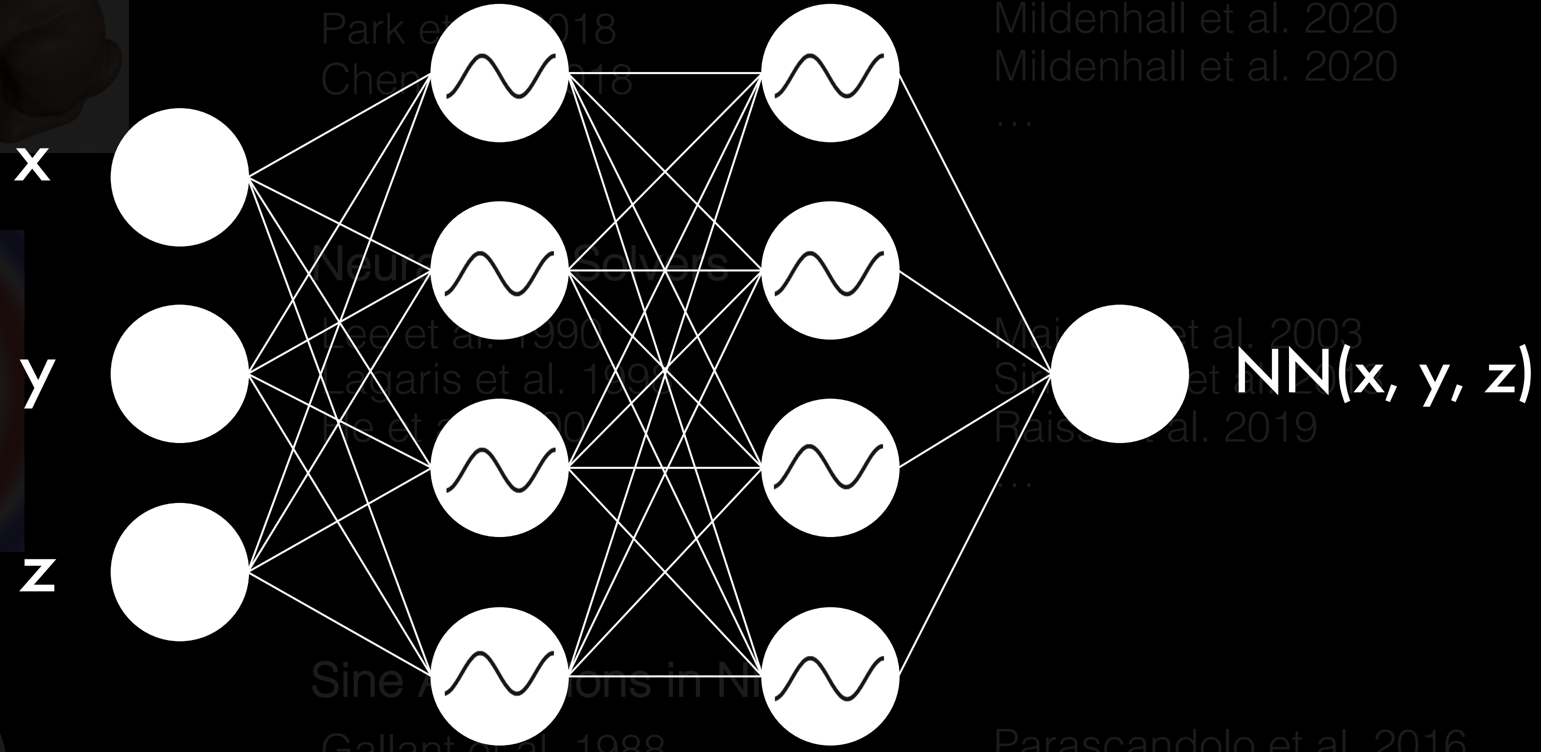
Related Work



Implicit Representations

Mescheder et al. 2018

Sitzmann et al. 2019
Mildenhall et al. 2020
Mildenhall et al. 2020
…



Lee et al. 1990

…

Raise... al. 2019
…

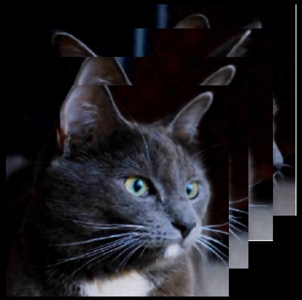
Gallant et al. 1988
Sopena et al. 1999
Candès et al. 1999

Parascandolo et al. 2016
Stanley et al. 2007
…

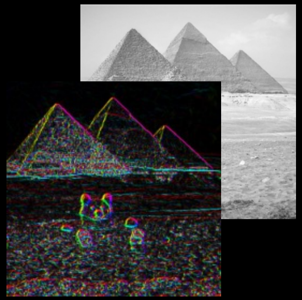
Images

Audio

Videos

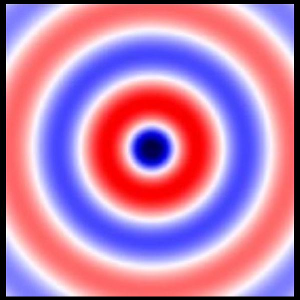


Poisson equation

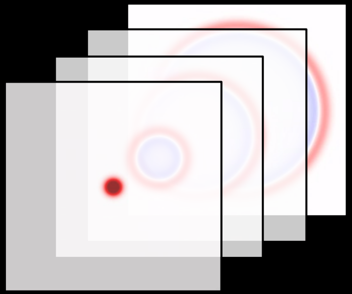


Eikonal equation

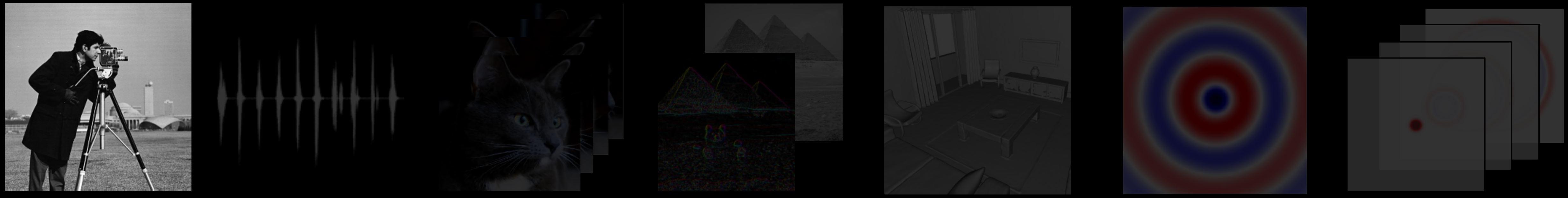
Helmholtz equation



Wave equation

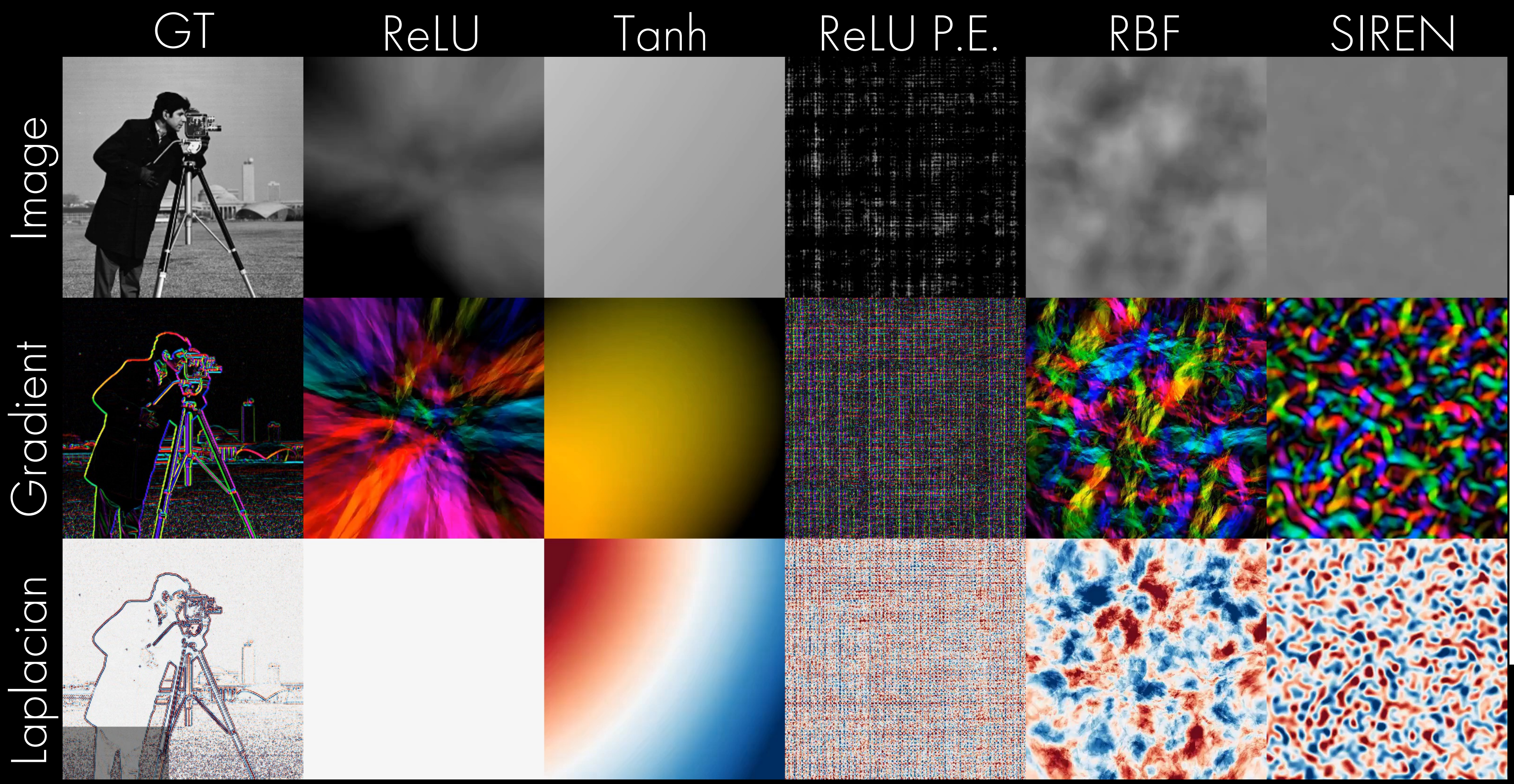


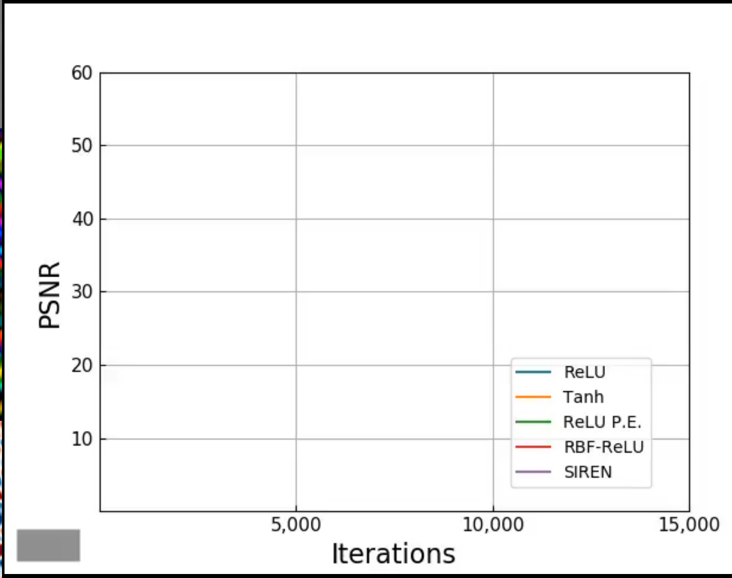
Images Audio Videos Poisson equation Eikonal equation Helmholtz equation Wave equation



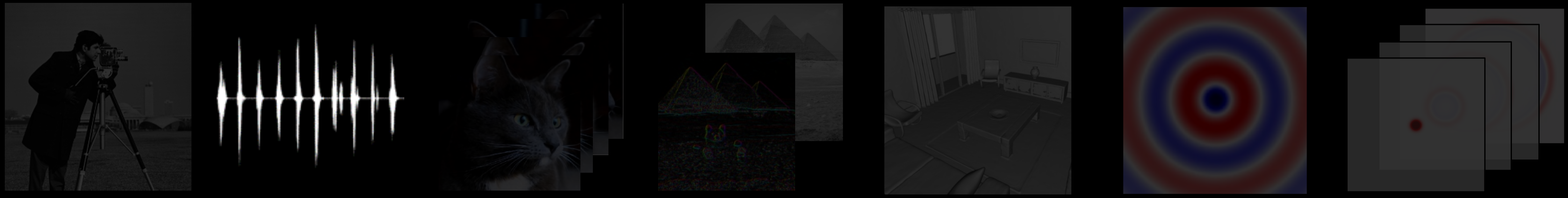
Input	Output supervised by	Implicit Formulation Find Φ that minimizes $\mathcal{L}$
$\mathbf{x} \in \mathbb{R}^2$	$f(\mathbf{x}) \in \mathbb{R}^3$	$\mathcal{L}_{\text{img}} = \int_\Omega \lVert \Phi(\mathbf{x}) - f(\mathbf{x}) \rVert \, d\mathbf{x}$
spatial coords.	*RGB values*	

Representing Images



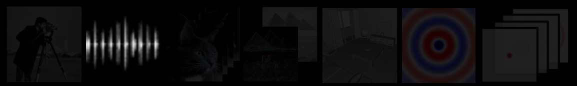


Images | Audio | Videos | Poisson equation | Eikonal equation | Helmholtz equation | Wave equation

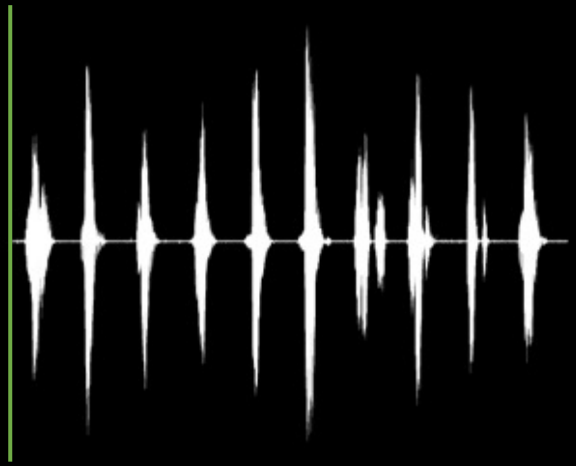


Input	Output supervised by	Implicit Formulation Find $\boldsymbol{\Phi}$ that minimizes $\boldsymbol{\mathcal{L}}$
$t \in \mathbb{R}$	$f(t) \in \mathbb{R}$	$\mathcal{L}_{\mathrm{audio}} = \int_{\Omega} \lVert \Phi(t) - f(t) \rVert \, \mathrm{d}t$

Representing Audio – Voice

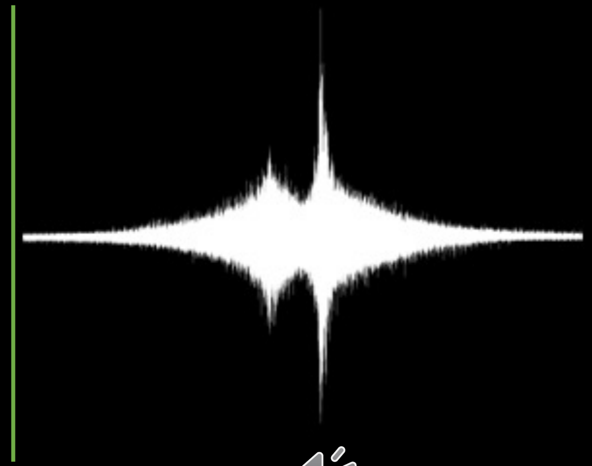


Ground Truth



ReLU MLP

ReLU w/ positional encoding

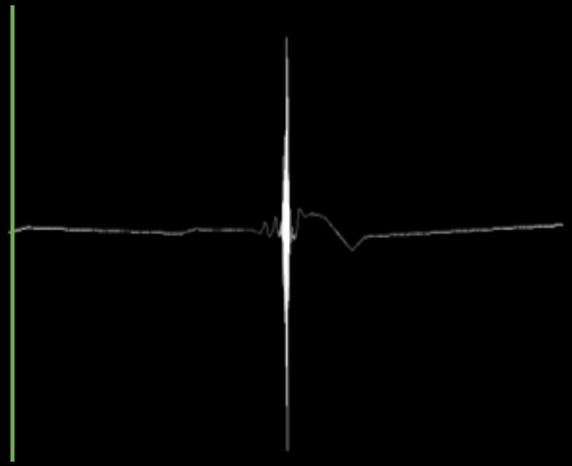


SIREN

Representing Audio – Music

Ground Truth

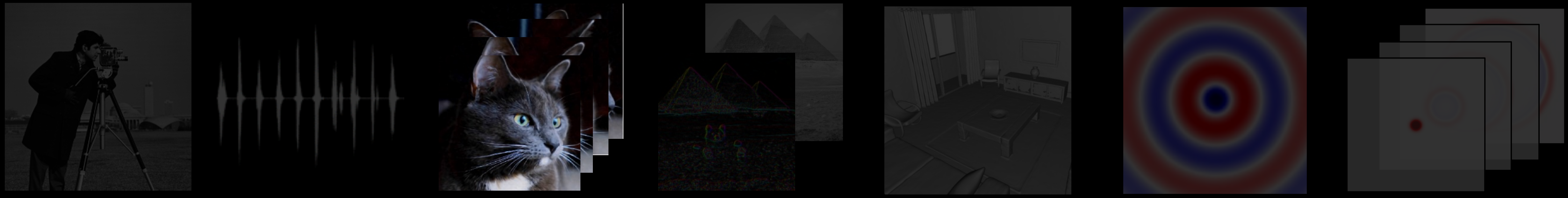
ReLU MLP



ReLU w/ positional encoding

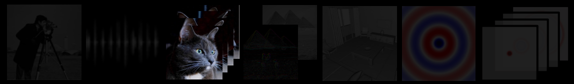
SIREN

Images | Audio | Videos | Poisson equation | Eikonal equation | Helmholtz equation | Wave equation



Input	Output supervised by	Implicit Formulation Find $\boldsymbol{\Phi}$ that minimizes $\mathcal{L}$
$(\mathbf{x}, t) \in \mathbb{R}^3$	$f(\mathbf{x}, t) \in \mathbb{R}^3$	$\mathcal{L}_{\text{video}} = \int_{\Omega} \lVert \Phi(\mathbf{x}, t) - f(\mathbf{x}, t) \rVert \, \mathrm{d}\mathbf{x} \, \mathrm{d}t$
space-time coord.	*RGB value*	

Representing Video

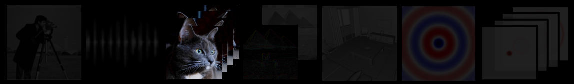


Ground Truth

ReLU MLP

SIREN

Representing Video



Ground Truth

ReLU MLP

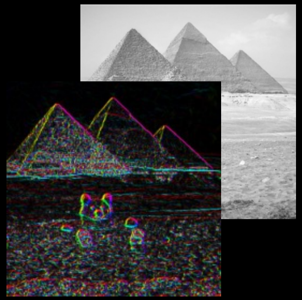
SIREN

Images

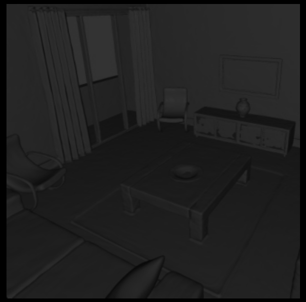
Audio

Videos

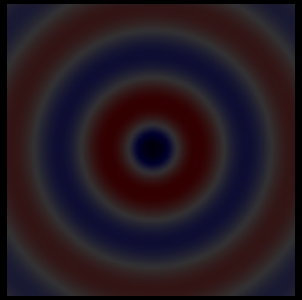
Poisson equation



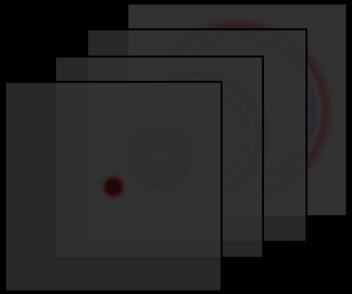
Eikonal equation



Helmholtz equation

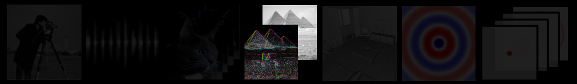


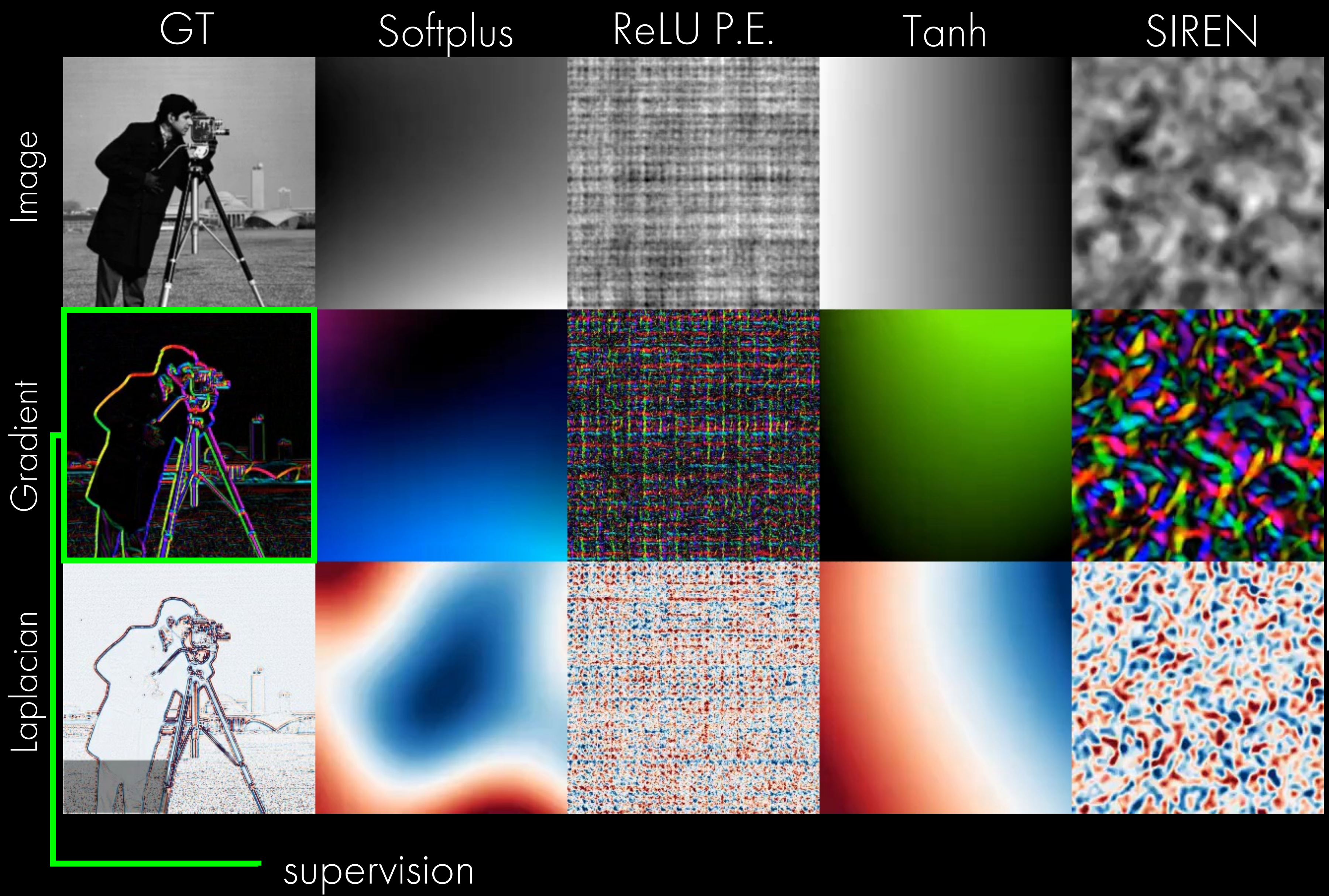
Wave equation

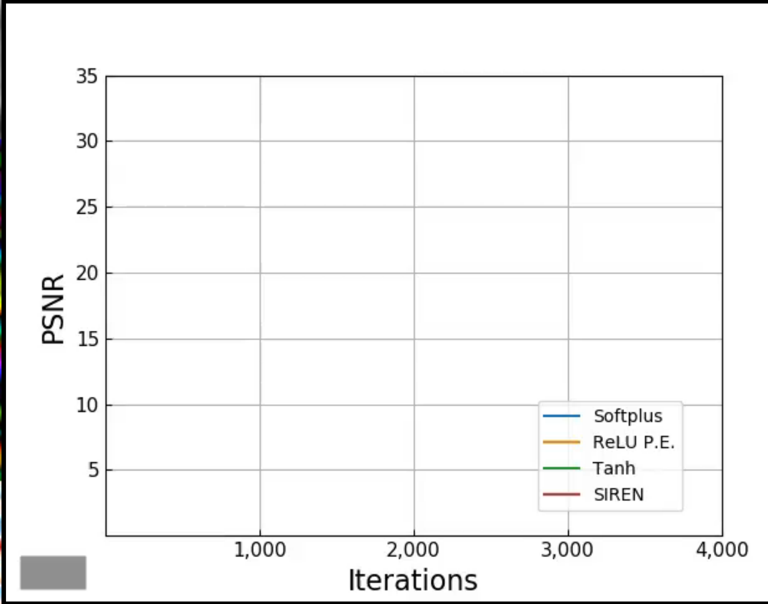


Input	Output supervised by	Implicit Formulation Find Φ that minimizes $\mathcal{L}$
$\mathbf{x} \in \mathbb{R}^2$	$f(\mathbf{x}) \in \mathbb{R}^2$	$\mathcal{L}_{\text{Poisson}} = \int_{\Omega} \lVert \nabla \Phi(\mathbf{x}) - \nabla f(\mathbf{x}) \rVert \, \mathrm{d}\mathbf{x}$
spatial coord.	*gray level*	

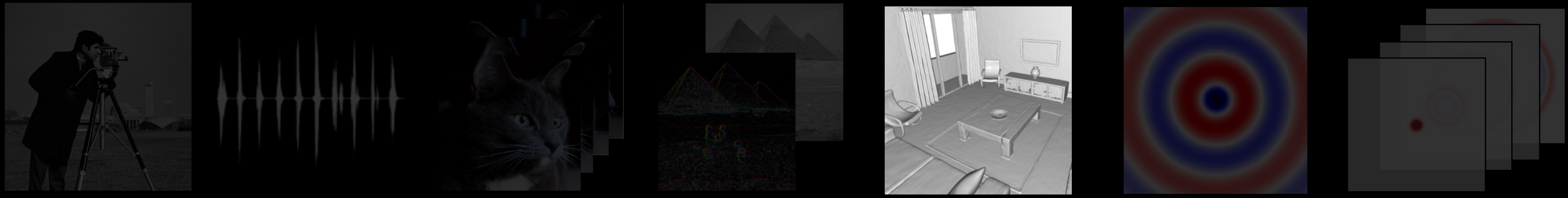
Poisson's Equation





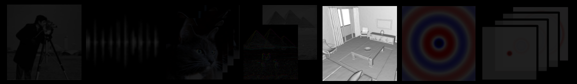


Images | Audio | Videos | Poisson equation | **Eikonal equation** | Helmholtz equation | Wave equation



Input	Output supervised by	Implicit Formulation Find Φ that minimizes $\mathcal{L}$
$\mathbf{x} \in \mathbb{R}^3$ *spatial coord.*	$f(\mathbf{x}) \in \mathbb{R}$ *signed distance*	$\mathcal{L}_{\text{Eikonal}} = \int_{\Omega_0} \lvert\Phi(\mathbf{x})\rvert + (1 - \langle \nabla\Phi(\mathbf{x}), \nabla f(\mathbf{x})\rangle)\,\mathrm{d}\mathbf{x} + \int_{\Omega} \lVert\lvert\nabla\Phi(\mathbf{x})\rvert - 1\rVert\,\mathrm{d}\mathbf{x}$

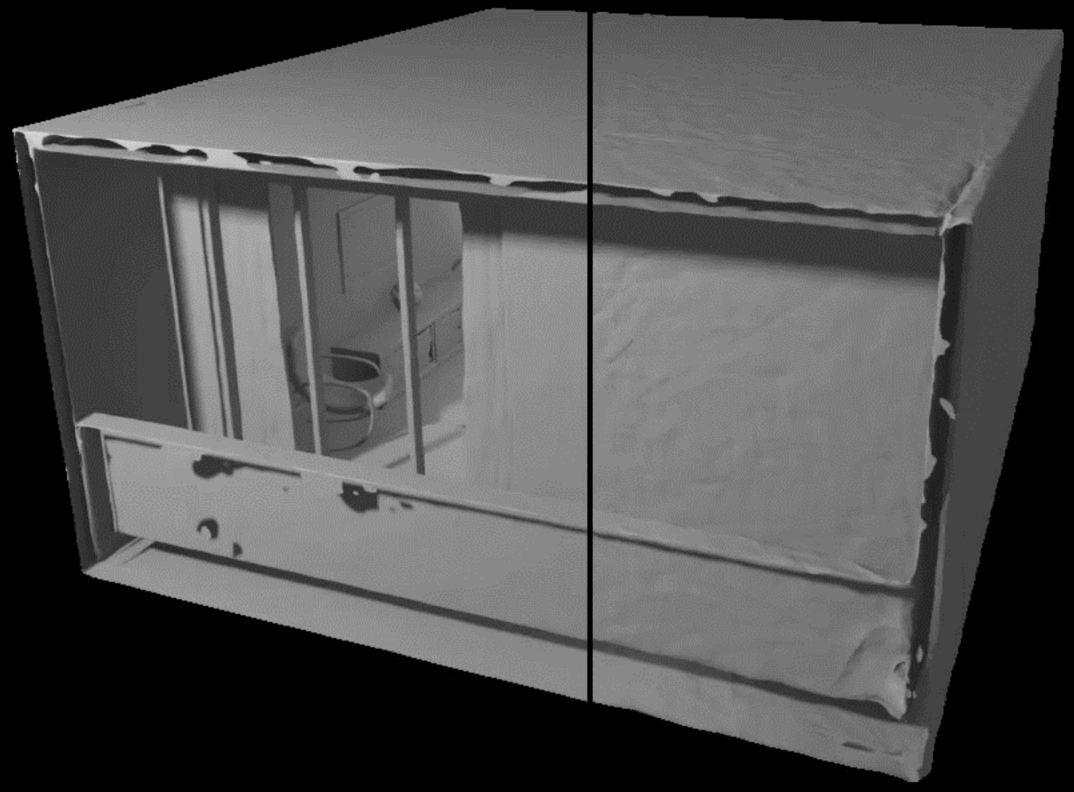
3D Shapes - solving the Eikonal equation



ReLU

SIREN

5 layers,
256 hidden units



ReLU

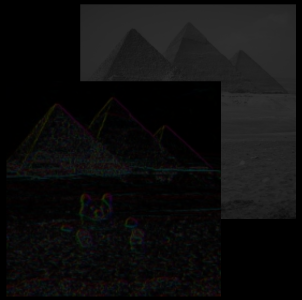
SIREN

Images

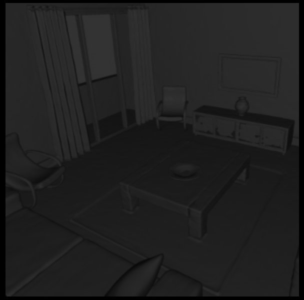
Audio

Videos

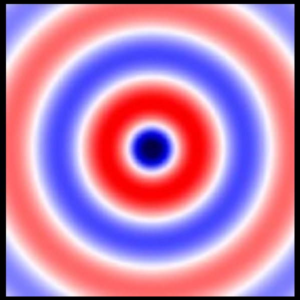
Poisson equation



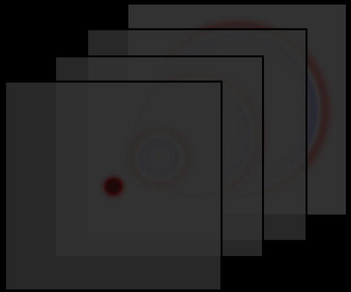
Eikonal equation



Helmholtz equation

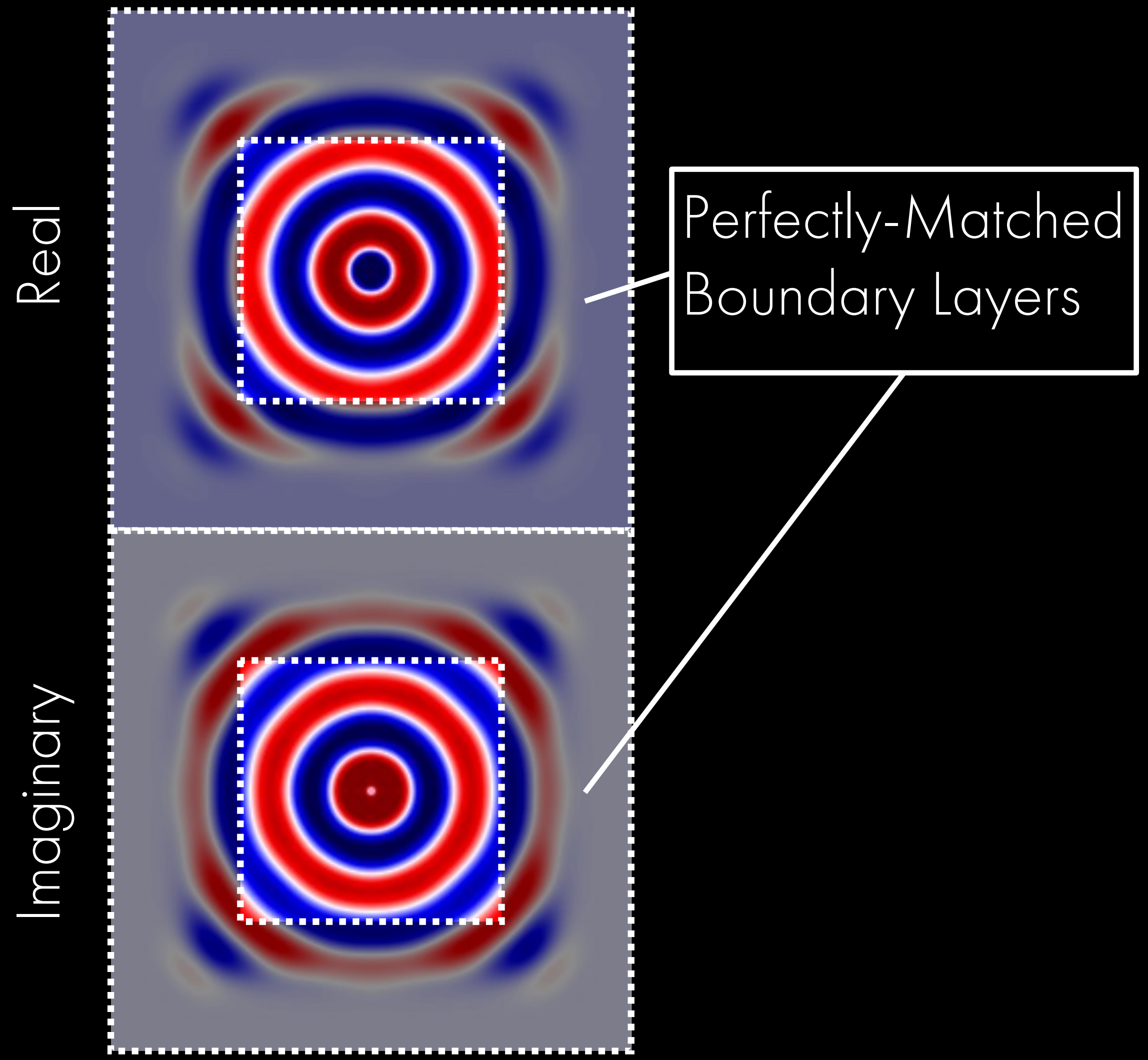


Wave equation

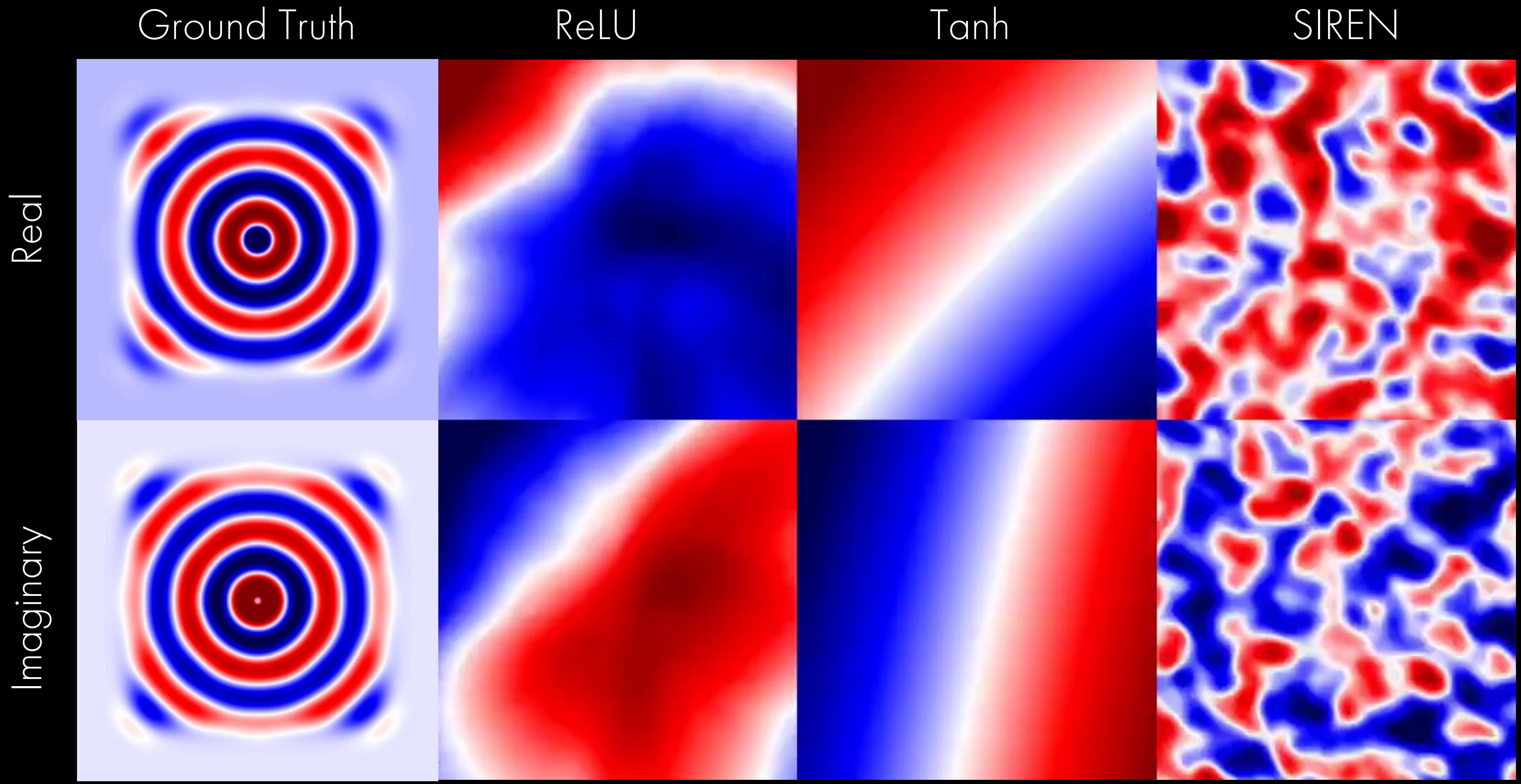


Input	Output supervised by	Implicit Formulation Find $\boldsymbol{\Phi}$ that minimizes $\boldsymbol{\mathcal{L}}$
$\mathbf{x} \in \mathbb{R}^2$	$f(\mathbf{x}) \in \mathbb{C}$	$\mathcal{L}_{\text{Helmholtz}} = \int_\Omega \|(\Delta + m(x)\omega^2)\Phi(\mathbf{x}) - f(\mathbf{x})\| \, d\mathbf{x}$
spatial coord.	*complex wave field*	

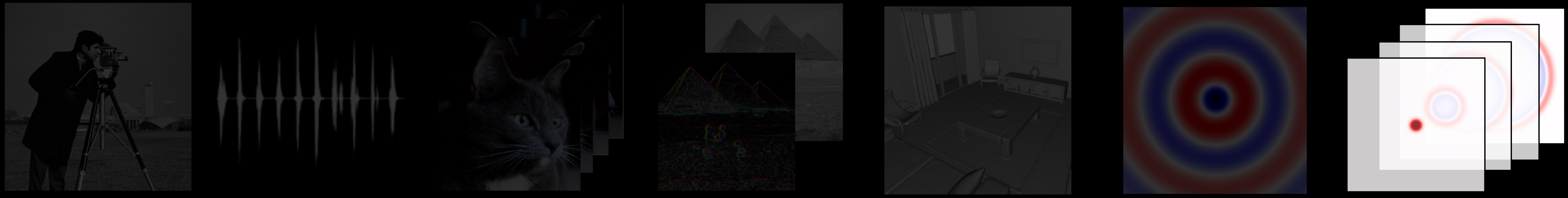
Solving the Helmholtz Equation



Solving the Helmholtz Equation



Images | Audio | Videos | Poisson equation | Eikonal equation | Helmholtz equation | **Wave equation**



Input	Output supervised by	Implicit Formulation Find Φ that minimizes $\mathcal{L}$
$(\mathbf{x}, t) \in \mathbb{R}^3$	$f(\mathbf{x}, t) \in \mathbb{R}$	$\mathcal{L}_{\text{wave}} = \int_\Omega \left\| \frac{\partial^2 \Phi}{\partial t^2} - c^2 \, \Delta \Phi \right\| \, \mathrm{d}\mathbf{x} \, \mathrm{d}t$
spatial coord.	*real wave field*	

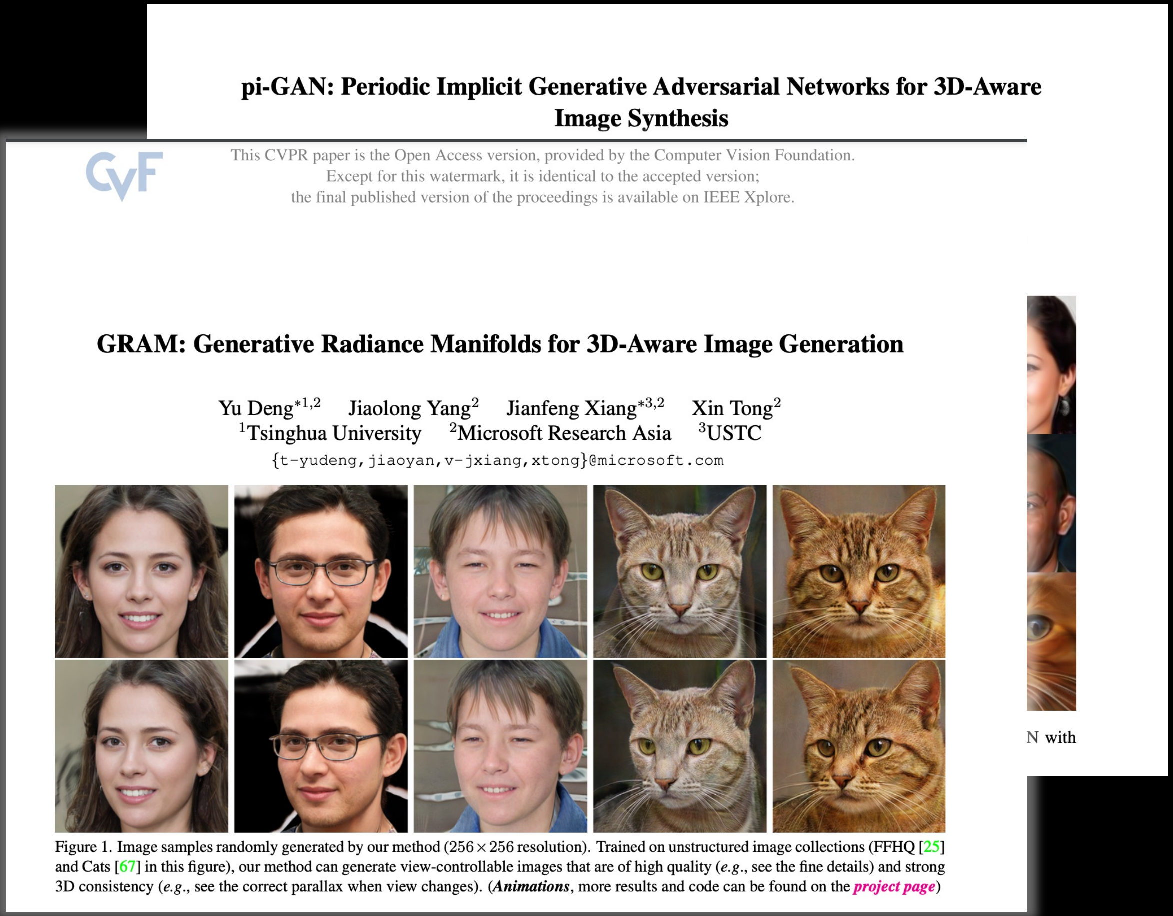
Solving the Wave Equation

GT SIREN Tanh

SIREN in generative models

Uses SIREN-based backbone for generative 3D model synthesis!

Pi-GAN [Chan et al. '21]
GRAM [Deng et al. '22]


pi-GAN: Periodic Implicit Generative Adversarial Networks for 3D-Aware Image Synthesis

GRAM: Generative Radiance Manifolds for 3D-Aware Image Generation

Yu Deng[*1,2] Jiaolong Yang[2] Jianfeng Xiang[*3,2] Xin Tong[2]
[1]Tsinghua University [2]Microsoft Research Asia [3]USTC
{t-yudeng, jiaoyan, v-jxiang, xtong}@microsoft.com

Figure 1. Image samples randomly generated by our method (256 × 256 resolution). Trained on unstructured image collections (FFHQ [25] and Cats [67] in this figure), our method can generate view-controllable images that are of high quality (*e.g.*, see the fine details) and strong 3D consistency (*e.g.*, see the correct parallax when view changes). (***Animations***, more results and code can be found on the ***project page***)

N with

SIREN in physics solvers

Uses SIREN for efficient physics solvers (NVIDIA SimNet)

[Hennigh et al. '20]

NVIDIA SimNet™: an AI-accelerated multi-physics simulation framework

Preprint- December 16, 2020

Oliver Hennigh ohennigh@nvidia.com
Susheela Narasimhan susheelan@nvidia.com
Mohammad Amin Nabian mnabian@nvidia.com
Akshay Subramaniam asubramaniam@nvidia.com
Kaustubh Tangsali ktangsali@nvidia.com
Max Rietmann mrietmann@nvidia.com
Jose del Aguila Ferrandis jaguila@mit.edu
Wonmin Byeon wbyeon@nvidia.com
Zhiwei Fang zhiweif@nvidia.com
Sanjay Choudhry* schoudhry@nvidia.com

NVIDIA
developer.nvidia.com/simnet

Abstract

We present SimNet, an AI-driven multi-physics simulation framework, to accelerate simulations across a wide range of disciplines in science and engineering. Compared to traditional numerical solvers, SimNet addresses a wide range of use cases - coupled forward simulations without any training data, inverse and data assimilation problems. SimNet offers fast turnaround time by enabling parameterized system representation that solves for multiple configurations simultaneously, as opposed to the traditional solvers that solve for one configuration at a time. SimNet is integrated with parameterized constructive solid geometry as well as STL modules to generate point clouds. Furthermore, it is customizable with APIs that enable user extensions to geometry, physics and

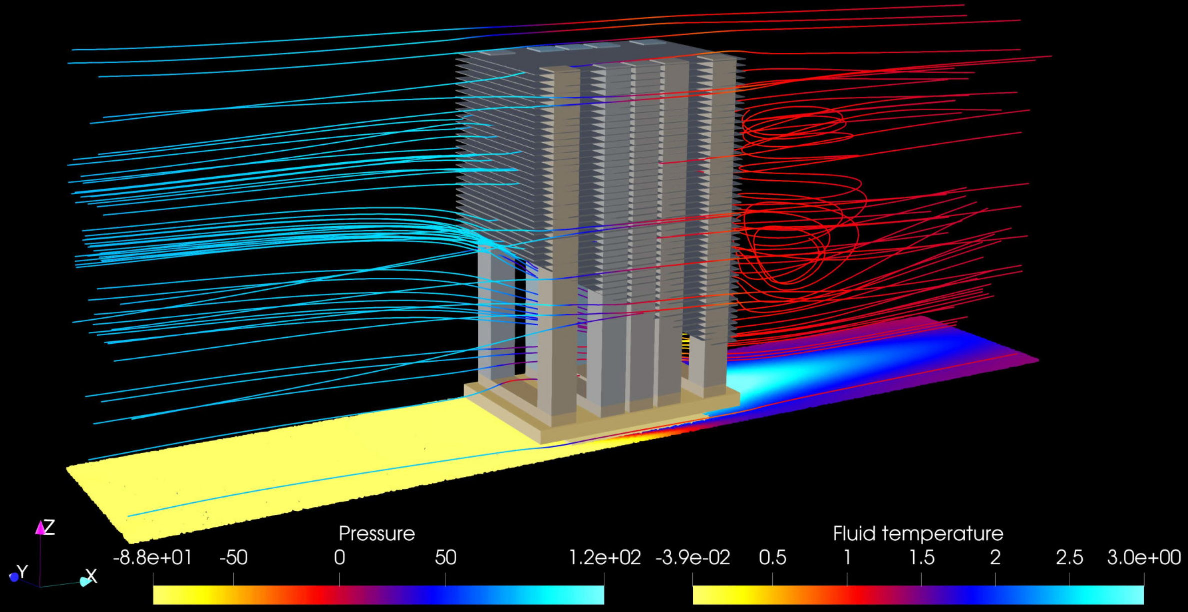


Figure 11: Streamlines colored with pressure and temperature profile in the fluid for optimal NVSwitch geometry.

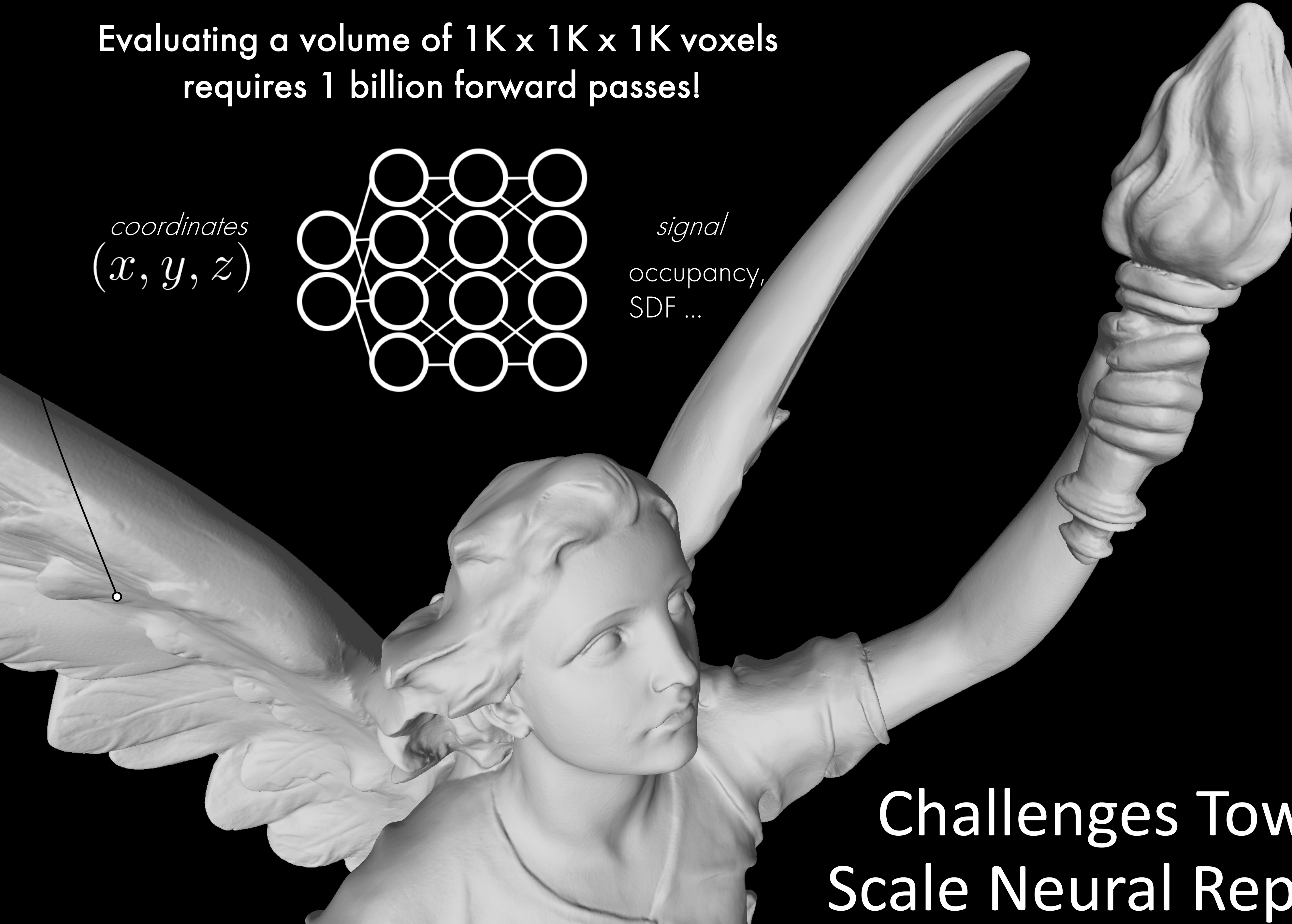
Table 3: Total compute time needed for different solvers for the NVSwitch heat sink design optimization

Solver	OpenFOAM	Commercial Solver	SimNet
Compute Time (x 1000 hrs.)	405935	137494	3

Outline

- Overview of neural fields
- Representing 3D shape
- Neural rendering with NeRF
- Case studies
    - SIREN
    - ACORN
    - AutoInt
    - BACON

Evaluating a volume of 1K x 1K x 1K voxels
requires 1 billion forward passes!

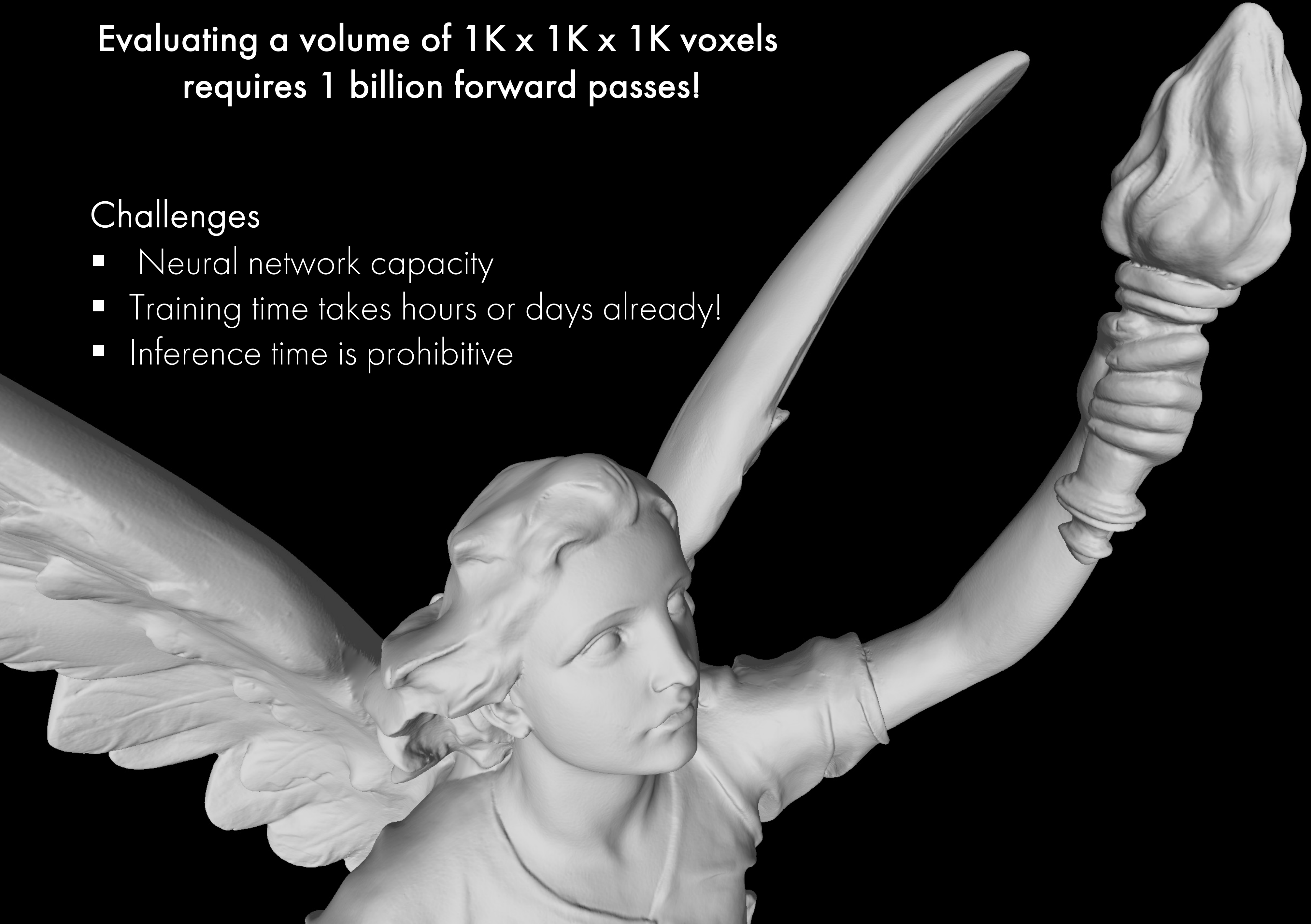


Challenges Towards Large Scale Neural Representations

Evaluating a volume of 1K x 1K x 1K voxels requires 1 billion forward passes!

Challenges

- Neural network capacity
- Training time takes hours or days already!
- Inference time is prohibitive



	Comp. Efficiency	Mem. Efficiency	Online multiscale	Pruning
Explicit	✓	✗	✗	✗
Global implicit	✗	✓	✗	✗
Local implicit	✗✓	✗✓	✗	✓✗
Hybrid Implicit-Explicit (ours)	✓	✓	✓	✓

Explicit methods

Input: features
Output: signal

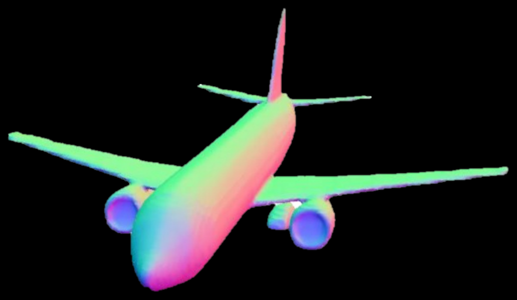
DeepVoxels

Neural Volumes

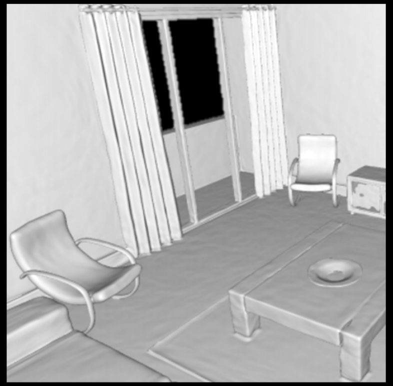
	Comp. Efficiency	Mem. Efficiency	Online multiscale	Pruning
Explicit	✓	✗	✗	✗
Global implicit	✗	✓	✗	✗
Local implicit	✗✓	✗✓	✗	✓✗
Hybrid Implicit-Explicit (ours)	✓	✓	✓	✓

Global implicit methods

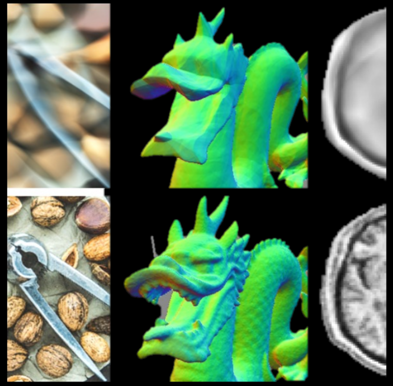
Input: coordinates
Output: signal



DeepSDF



SIREN

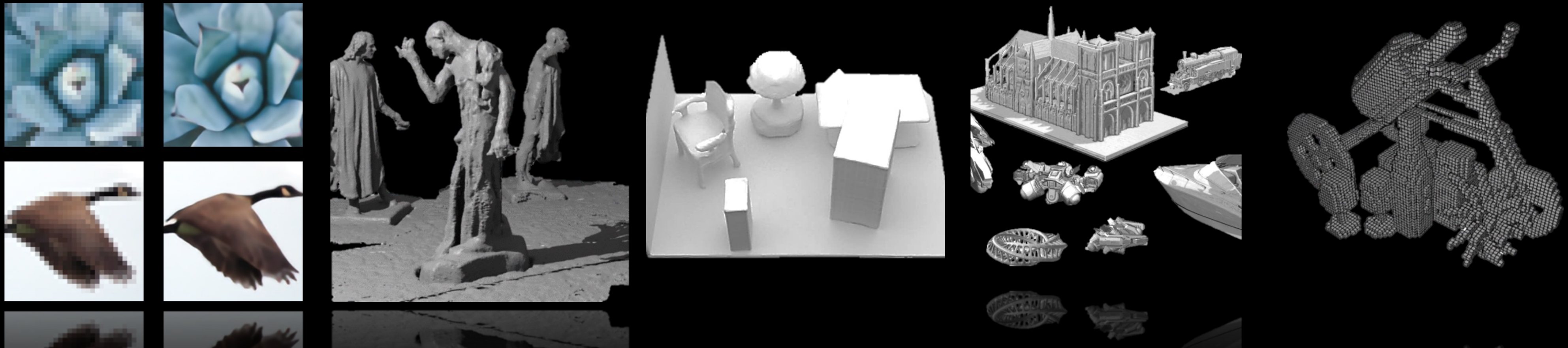


Fourier Features

	Comp. Efficiency	Mem. Efficiency	Online multiscale	Pruning
Explicit	✓	✗	✗	✗
Global implicit	✗	✓	✗	✗
Local implicit	✗✓	✗✓	✗	✓✗
Hybrid Implicit-Explicit (ours)	✓	✓	✓	✓

Local implicit methods

Input: features & coordinates
Output: signal



Local Implicit Image Functions

Deep Local Shapes

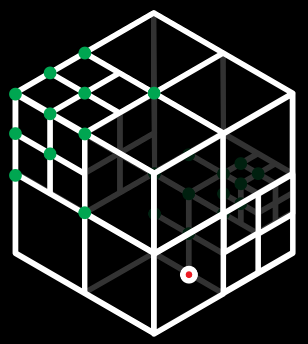
Convolutional Occupancy Nets

Neural Geometric Level of Detail

Neural Sparse Voxel Fields

	Comp. Efficiency	Mem. Efficiency	Online multiscale	Pruning
Explicit	✓	✗	✗	✗
Global implicit	✗	✓	✗	✗
Local implicit	✗✓	✗✓	✗	✓✗
Hybrid Implicit-Explicit (ours)	✓	✓	✓	✓

ACORN: an hybrid implicit-explicit architecture



Global coordinate encoder

Input: global coordinates
Output: features

Local coordinate decoder

Input: local coordinates
& features
Output : signal

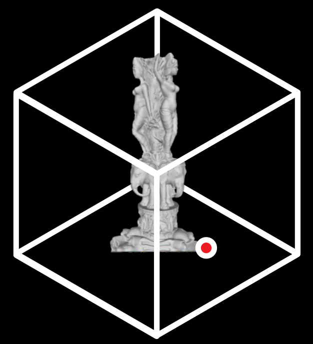
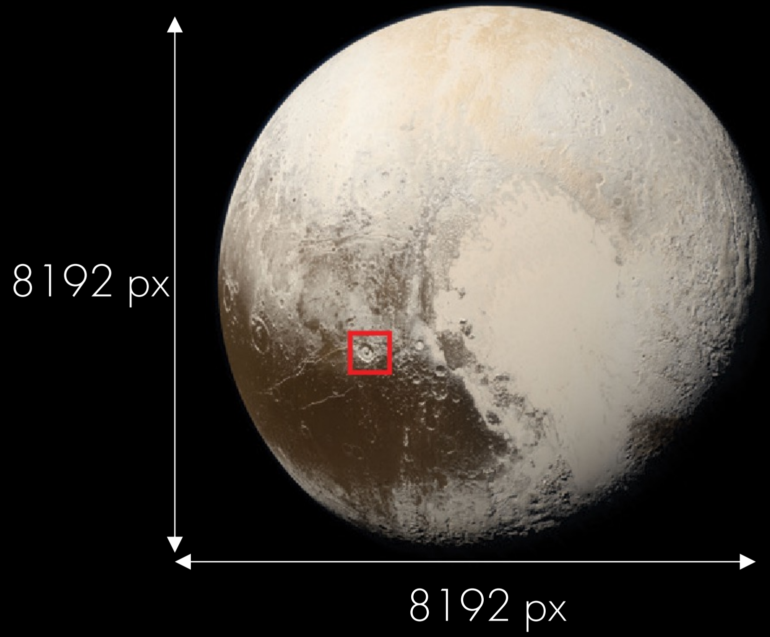


Image Fitting Example (16 MP)

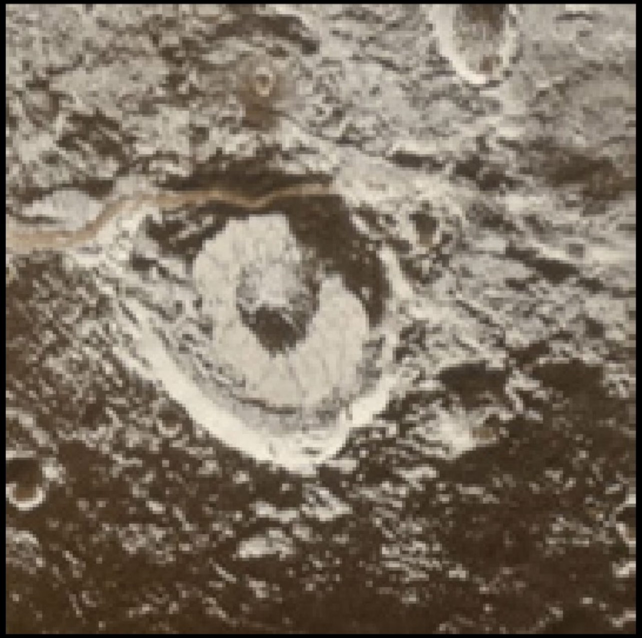
Pluto image: NASA/JHU

Scaling Up

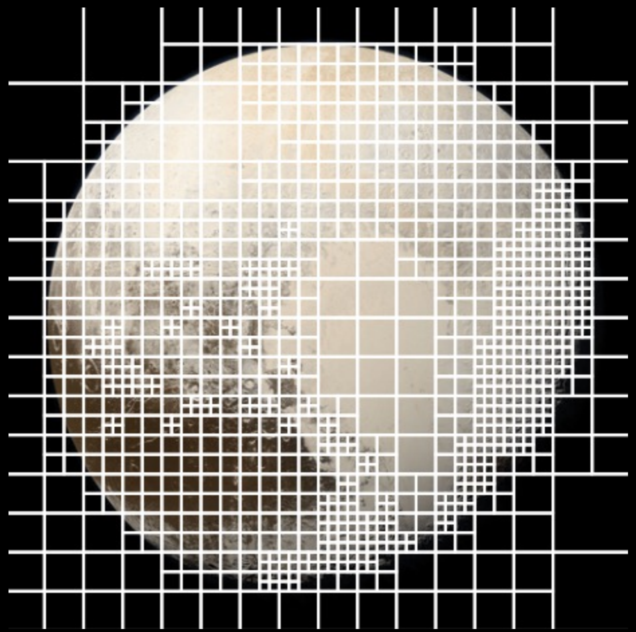
Ground Truth (64 MP)

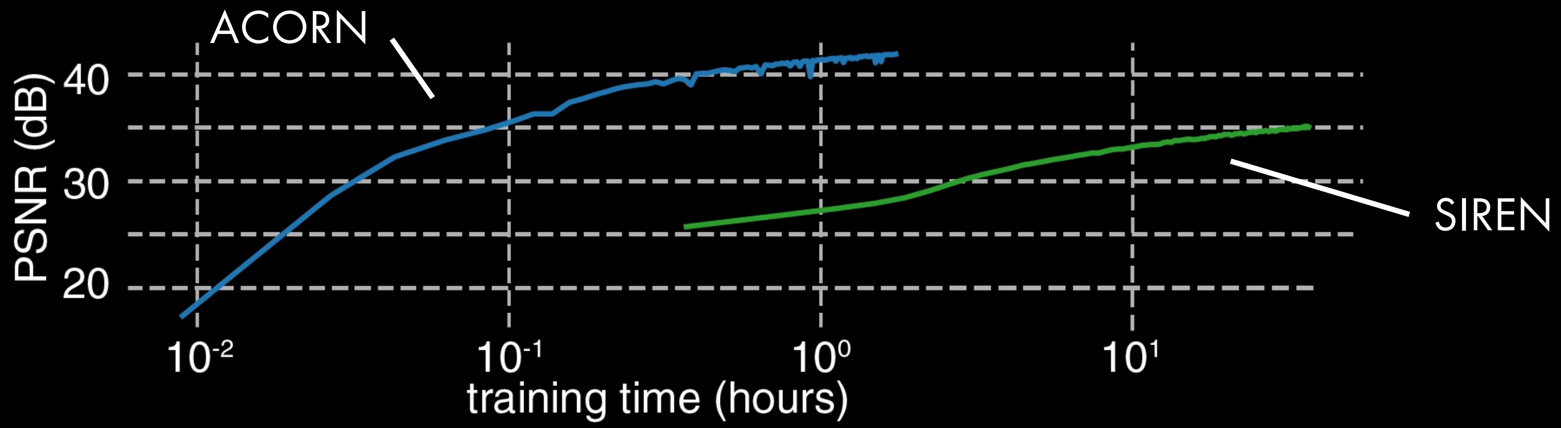


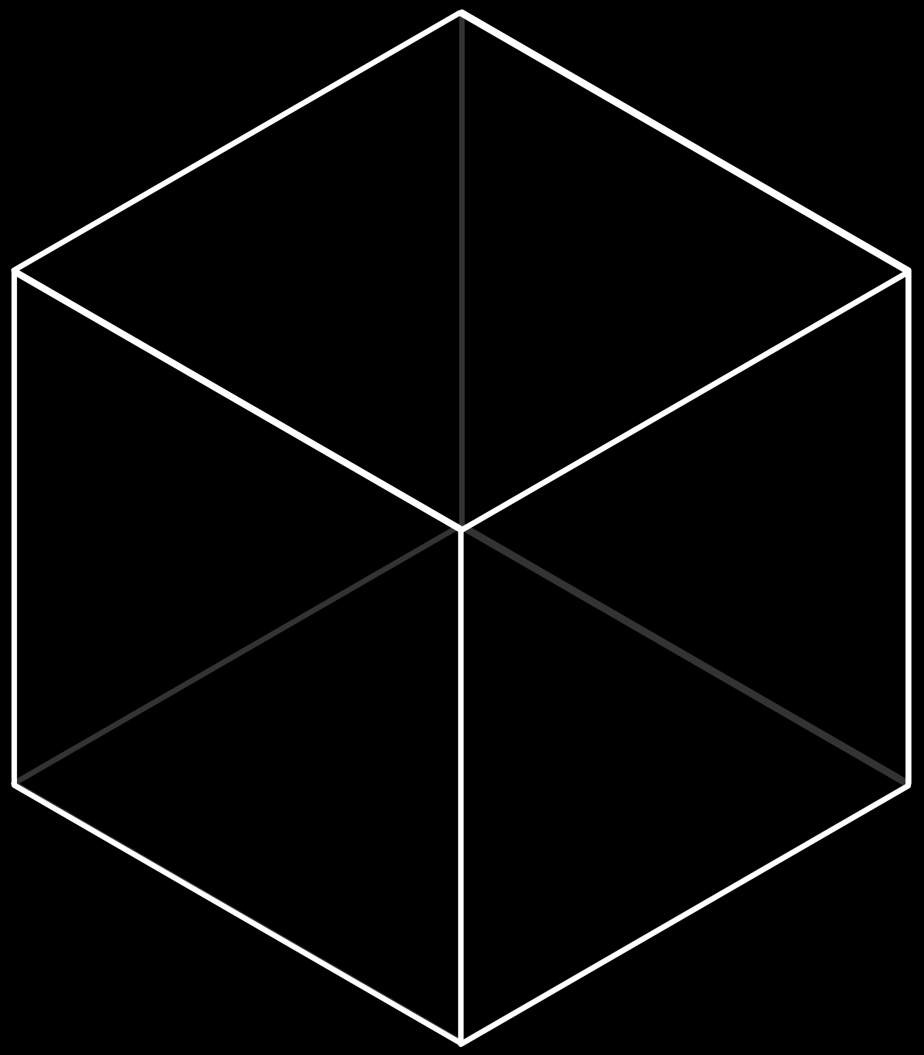
Closeup



Optimized ACORN

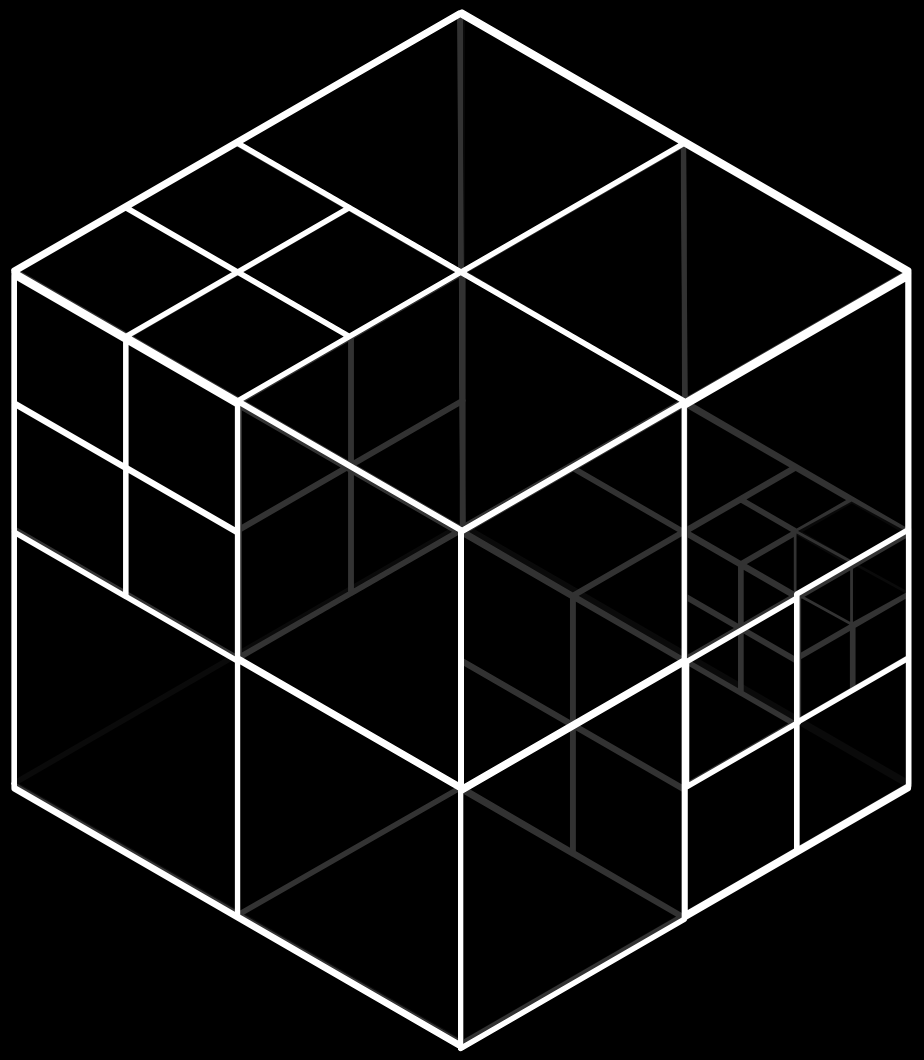


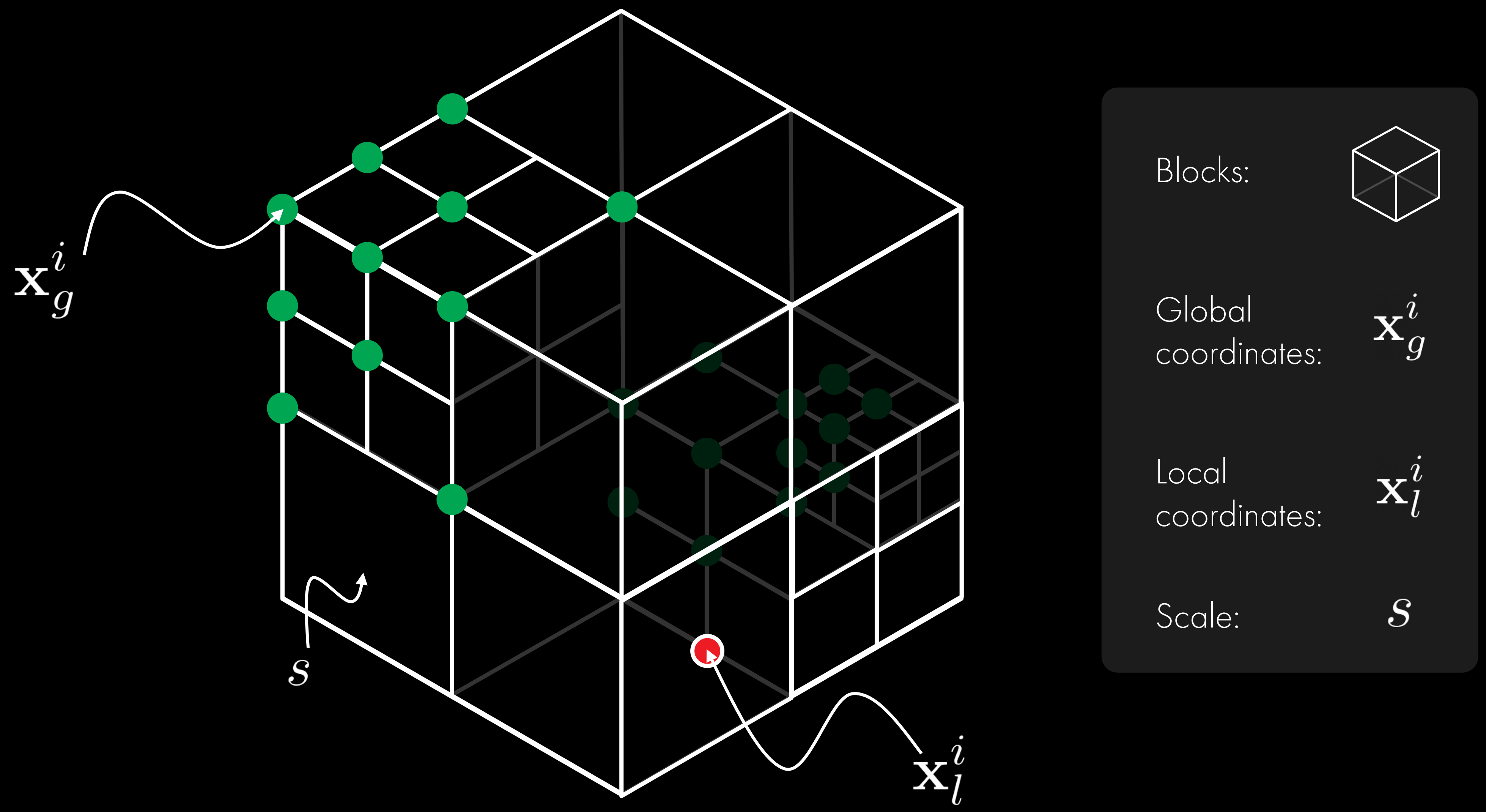




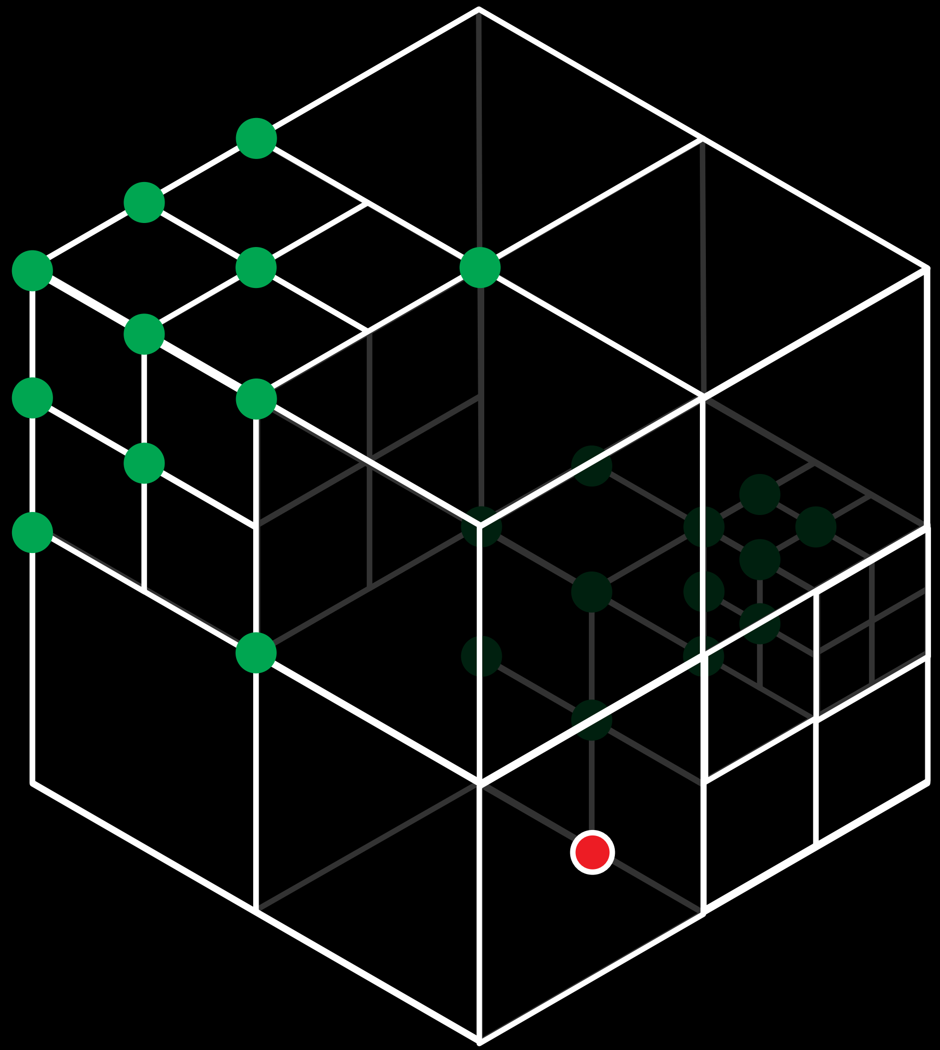
The partition is optimized **online**, using an Integer Linear Program.

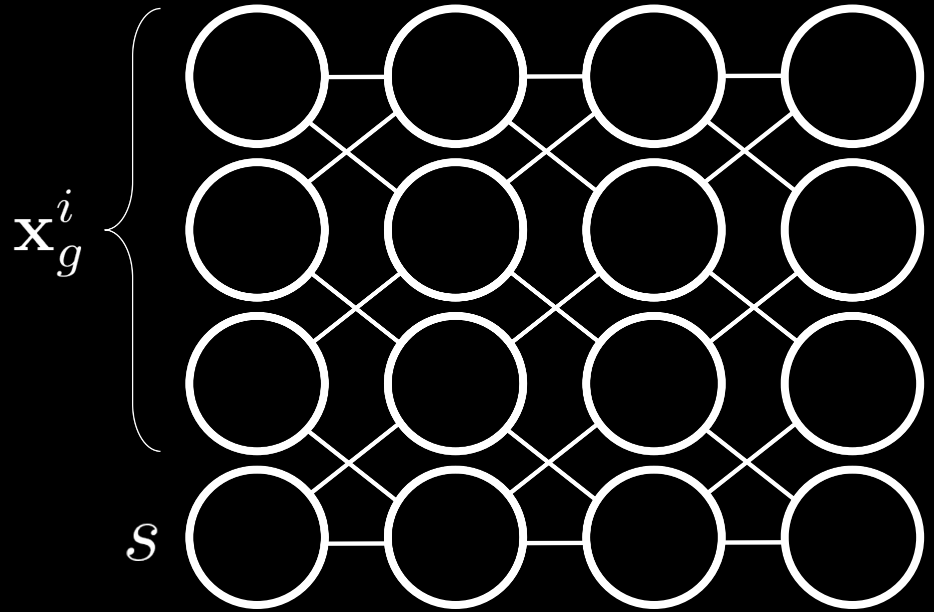
Blocks are split and merged based on a feedback of the training error.

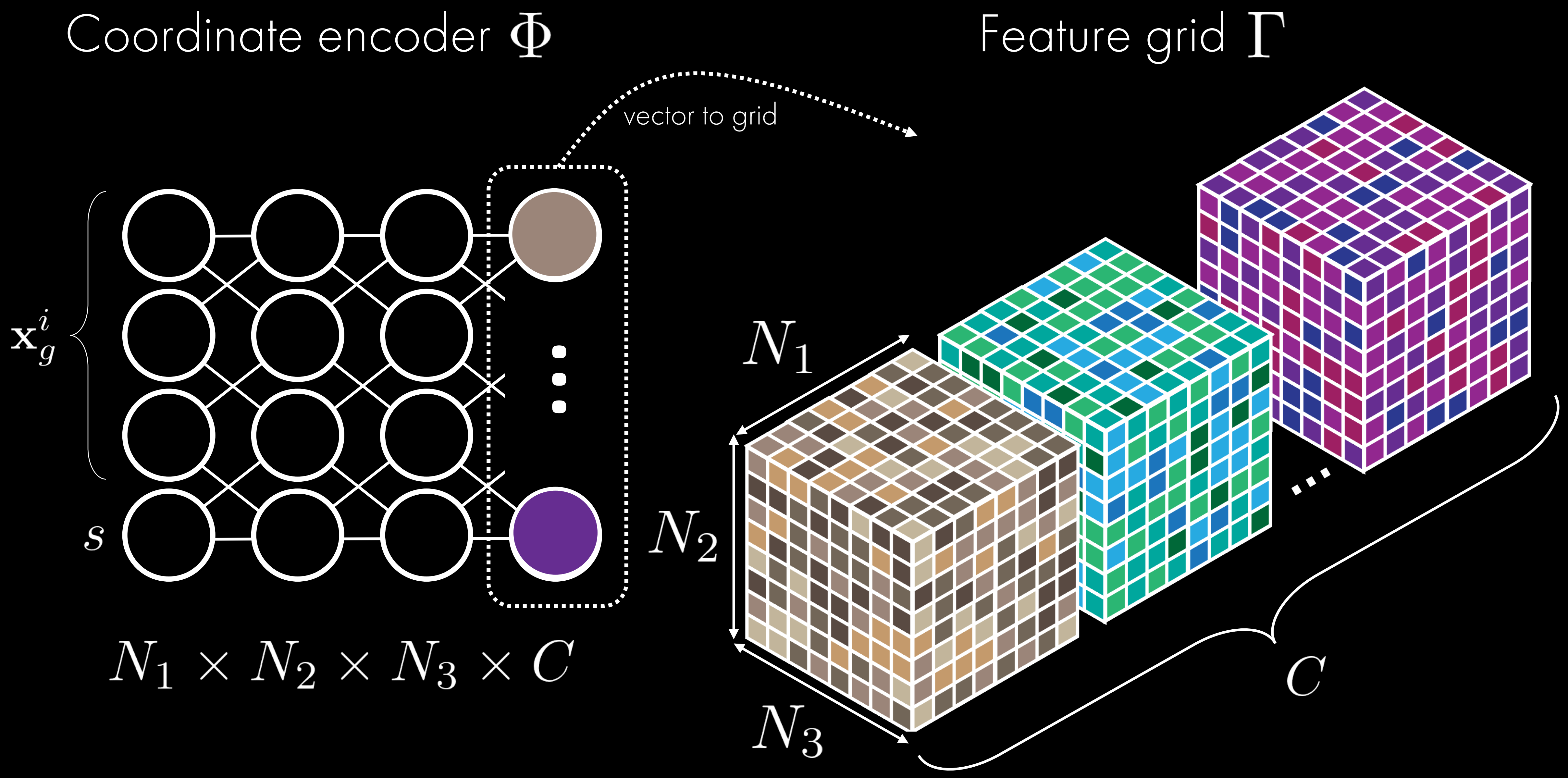


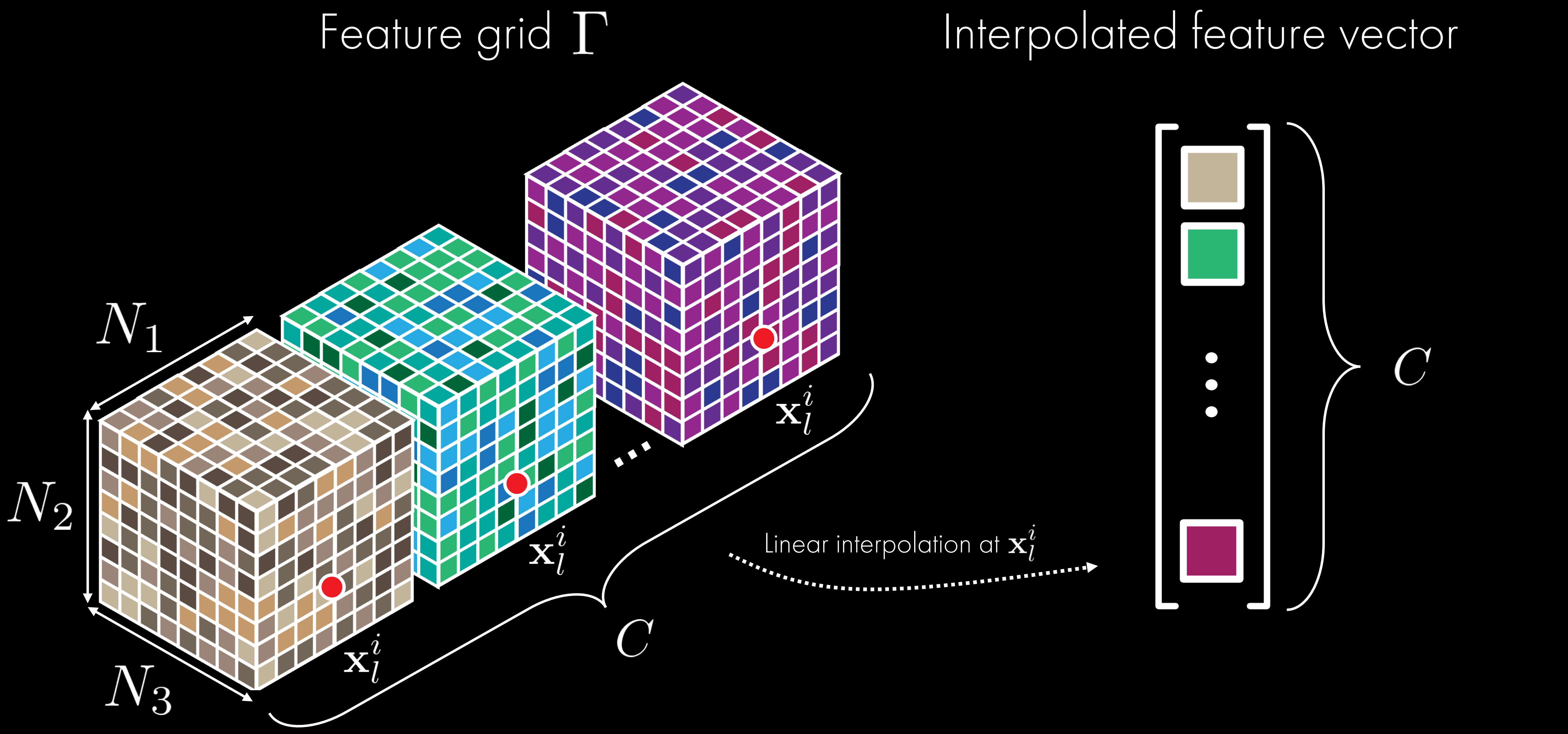

$\mathbf{x}_g^i$
s
$\mathbf{x}_l^i$
Blocks:
Global coordinates: $\mathbf{x}_g^i$
Local coordinates: $\mathbf{x}_l^i$
Scale: s

Coordinate encoder Φ

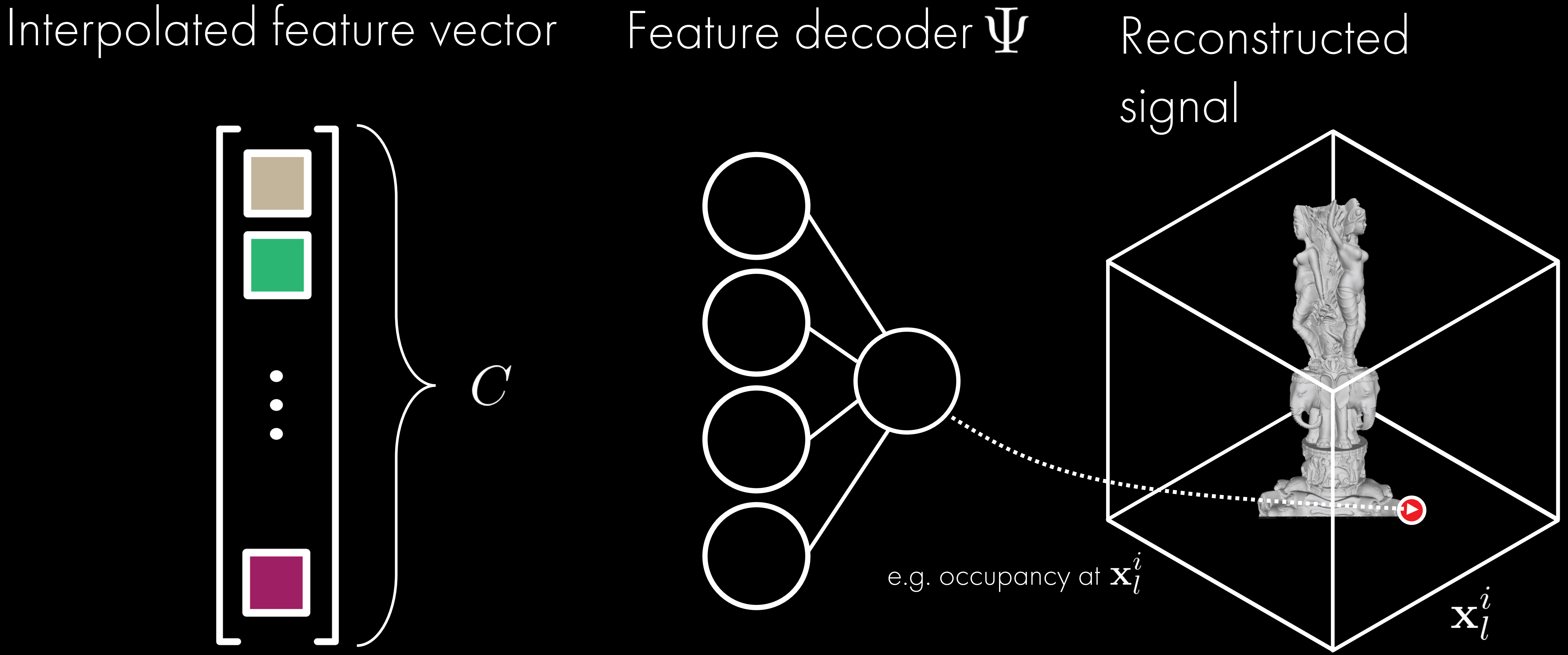




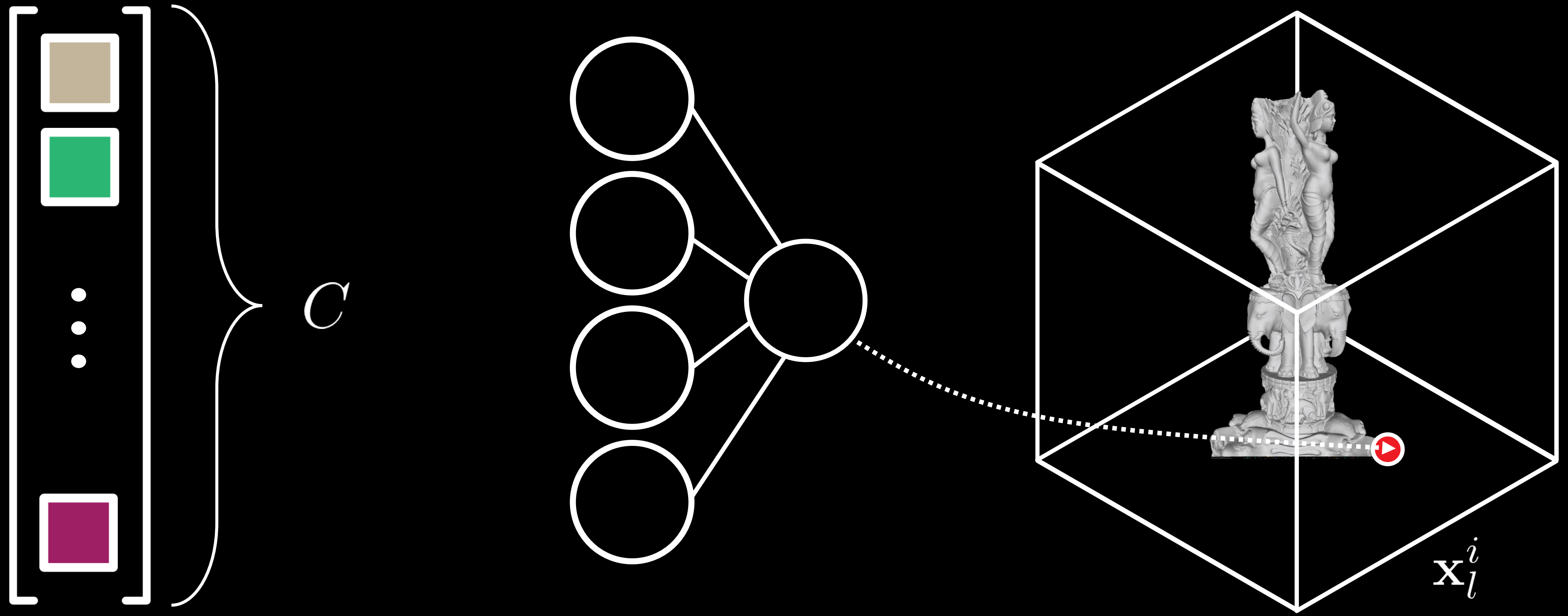

Coordinate encoder Φ
Feature grid Γ
vector to grid
$\mathbf{x}_g^i$
s
$N_1 \times N_2 \times N_3 \times C$
N_1
N_2
N_3
C



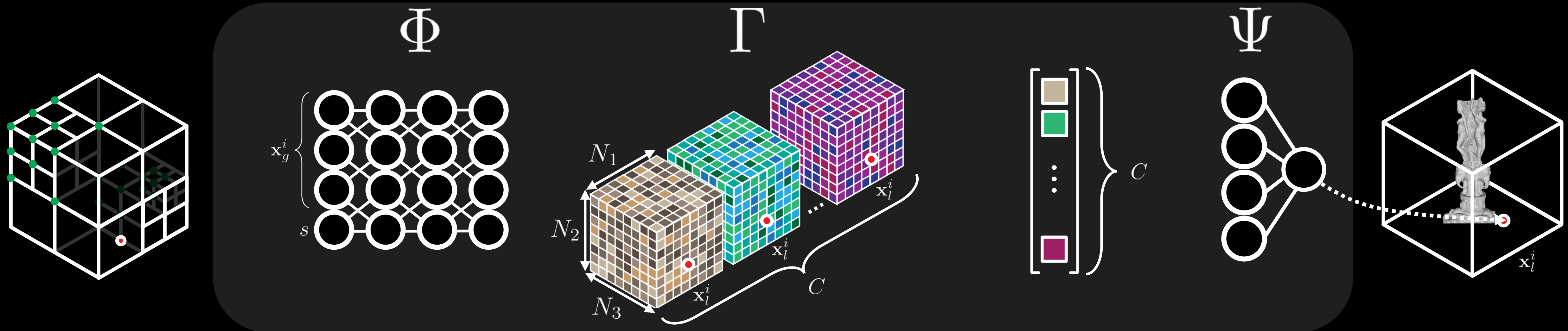
Feature grid Γ
Interpolated feature vector
N_1
N_2
N_3
$\mathbf{x}_l^i$
$\mathbf{x}_l^i$
$\mathbf{x}_l^i$
C
Linear interpolation at $\mathbf{x}_l^i$
C



Interpolated feature vector
Feature decoder Ψ
Reconstructed signal
C
e.g. occupancy at $\mathbf{x}_l^i$
$\mathbf{x}_l^i$


C
$\mathbf{x}_l^i$

ACORN

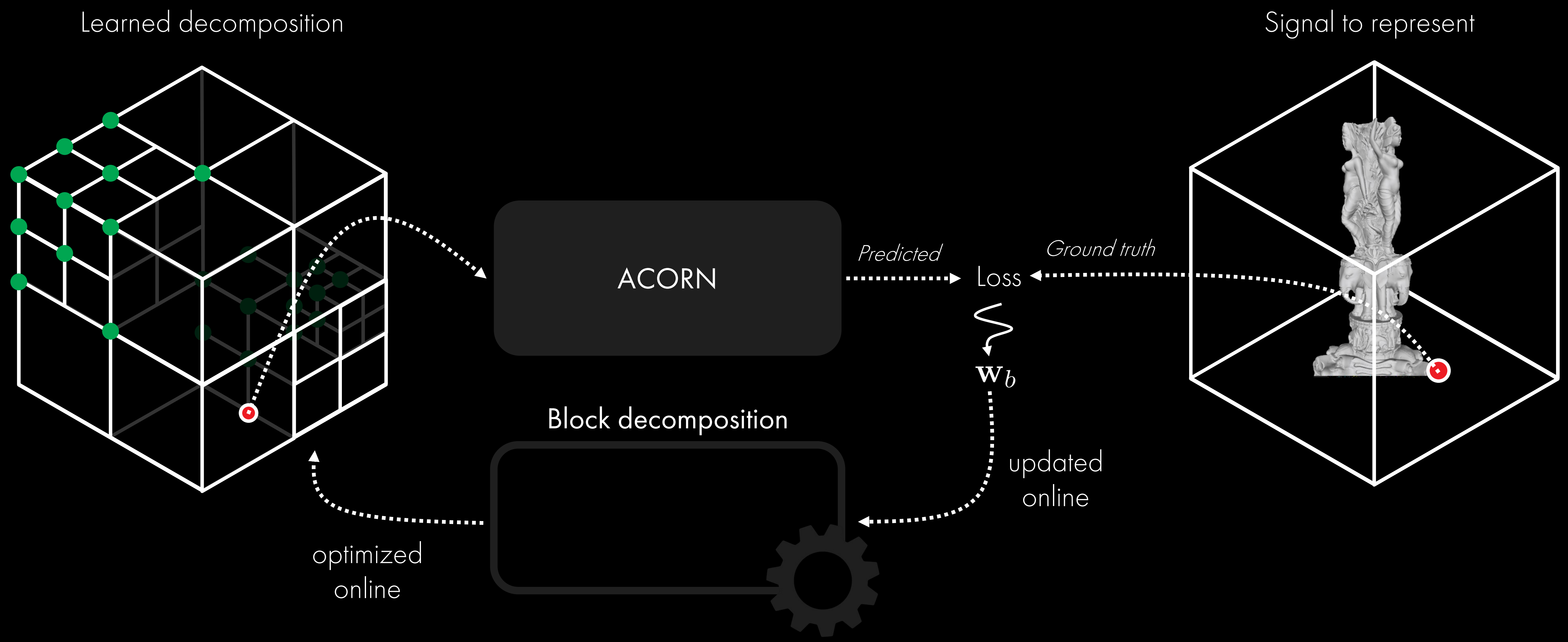


Coordinate Encoder

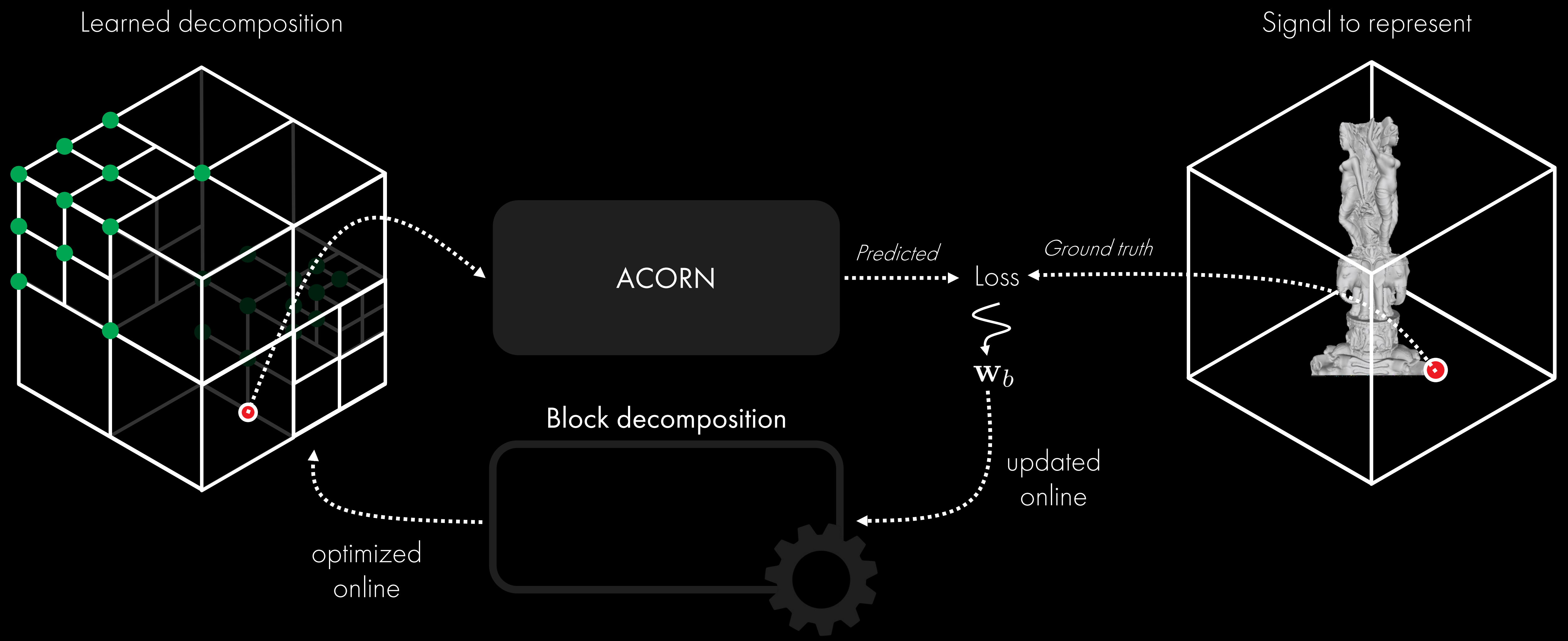
Feature grid

Interpolated feature vector

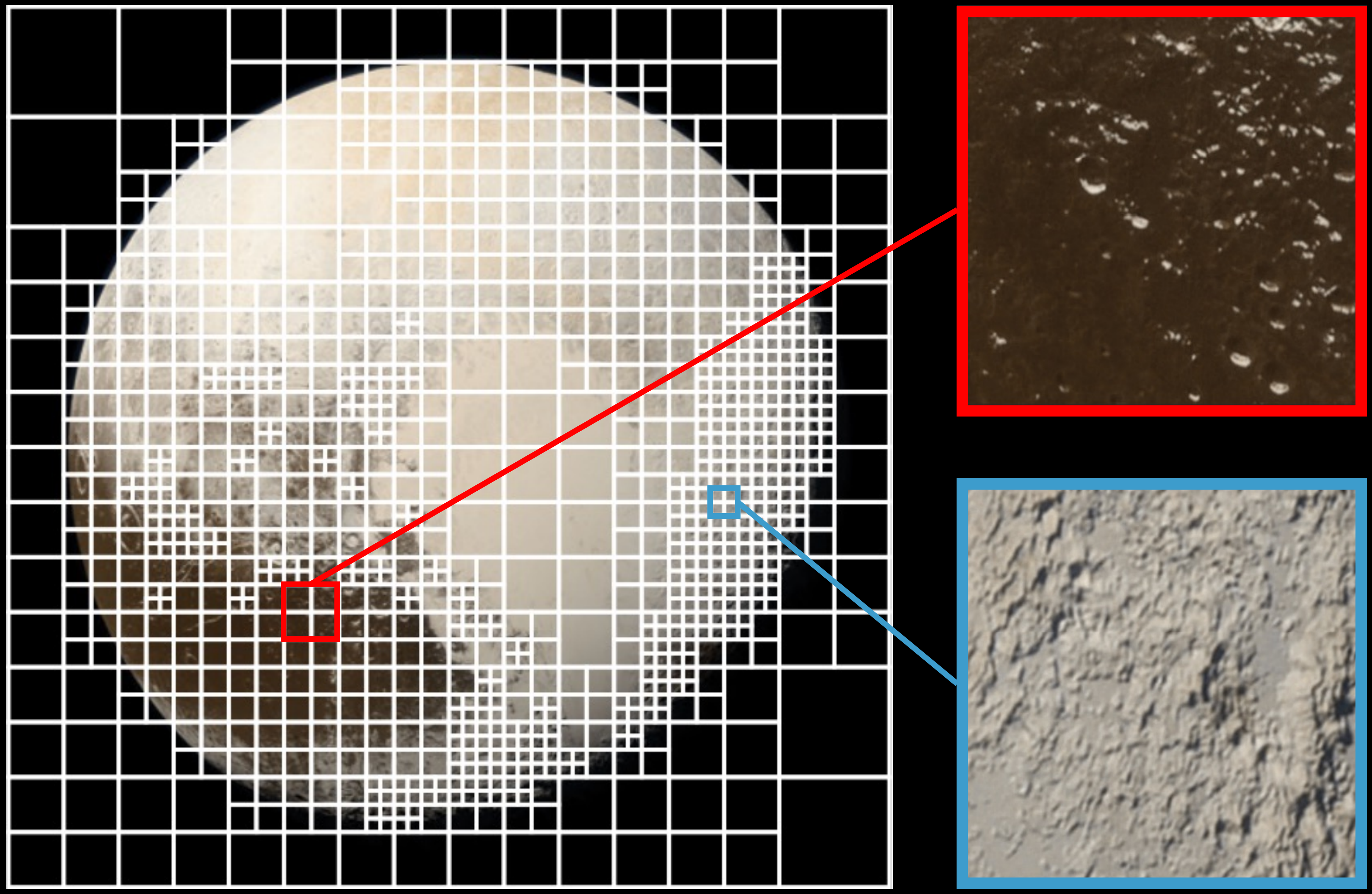
Feature decoder



Learned decomposition
Signal to represent
ACORN
Predicted
Loss
Ground truth
$\mathbf{w}_b$
Block decomposition
updated
online
optimized
online

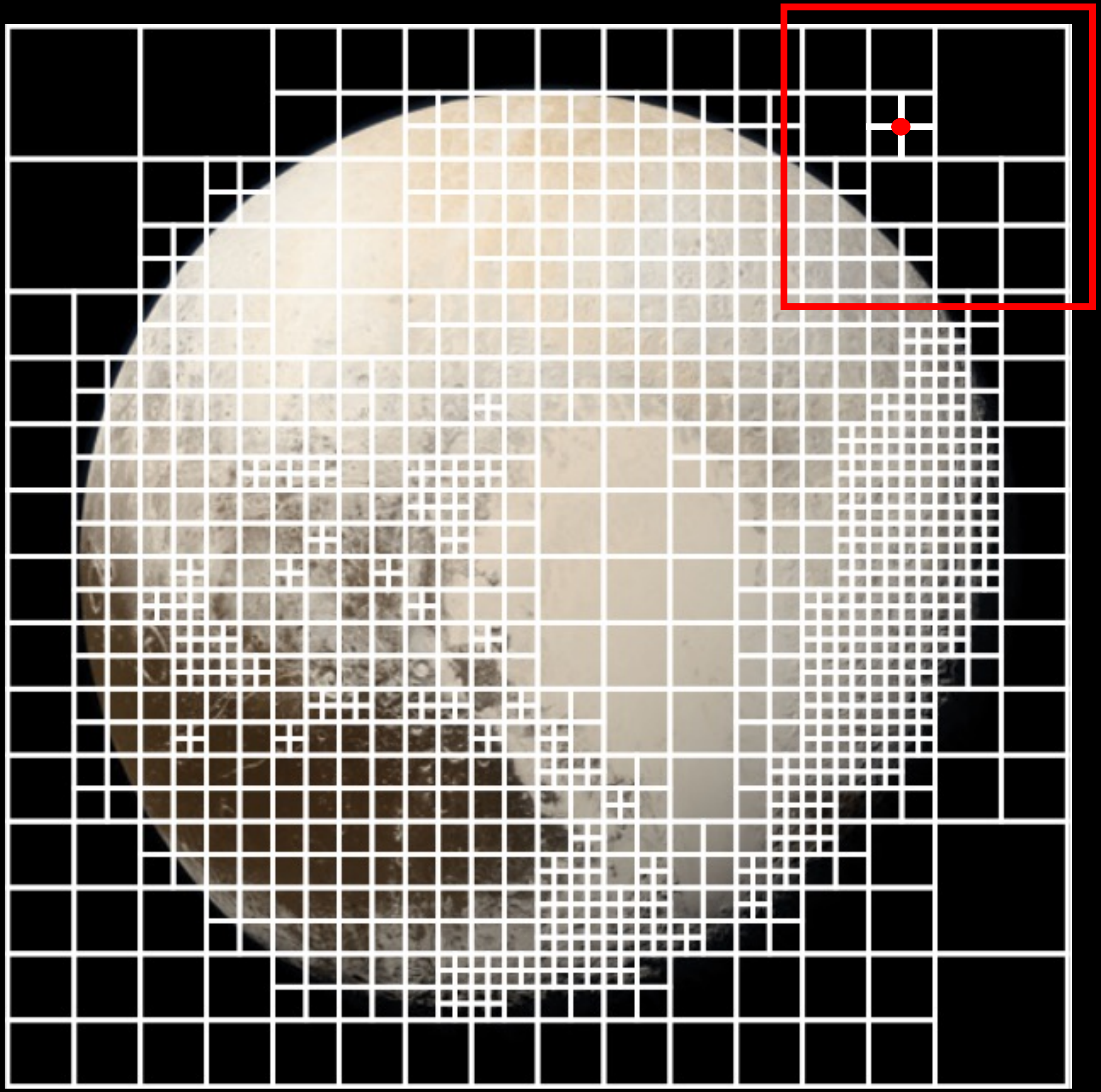


Learned decomposition
Signal to represent
ACORN
Predicted
Loss
Ground truth
$\mathbf{w}_b$
Block decomposition
updated
online
optimized
online

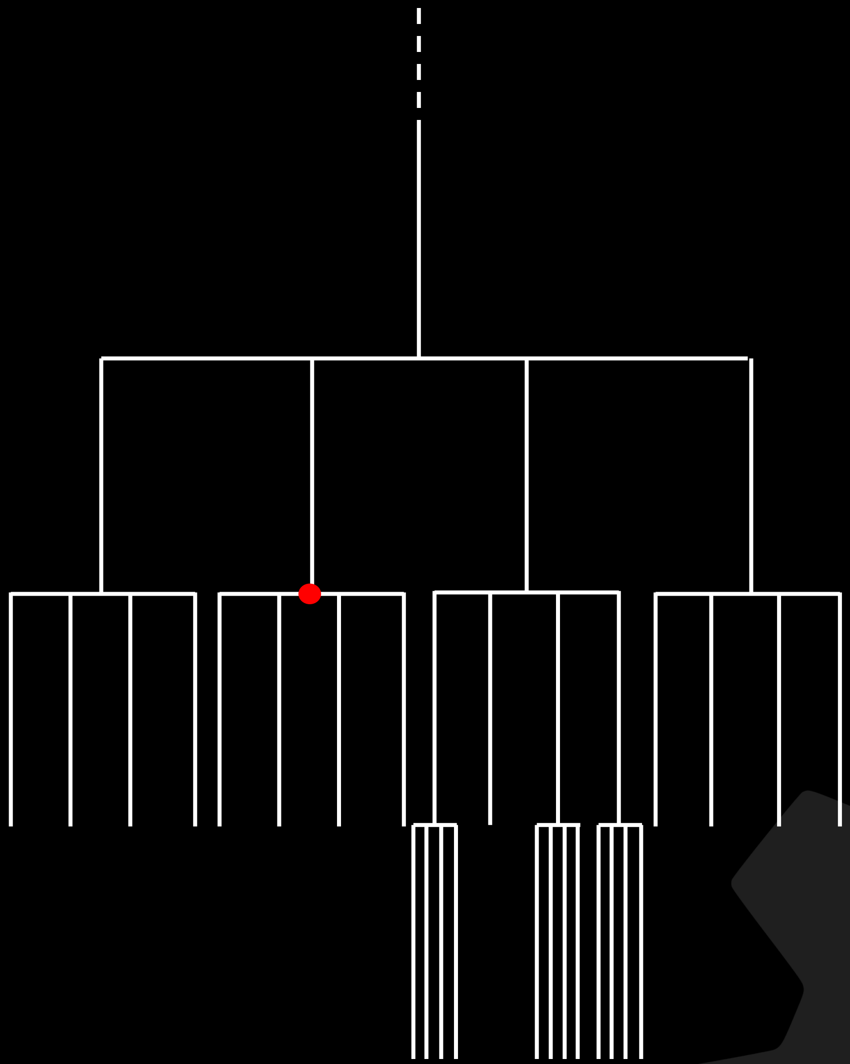


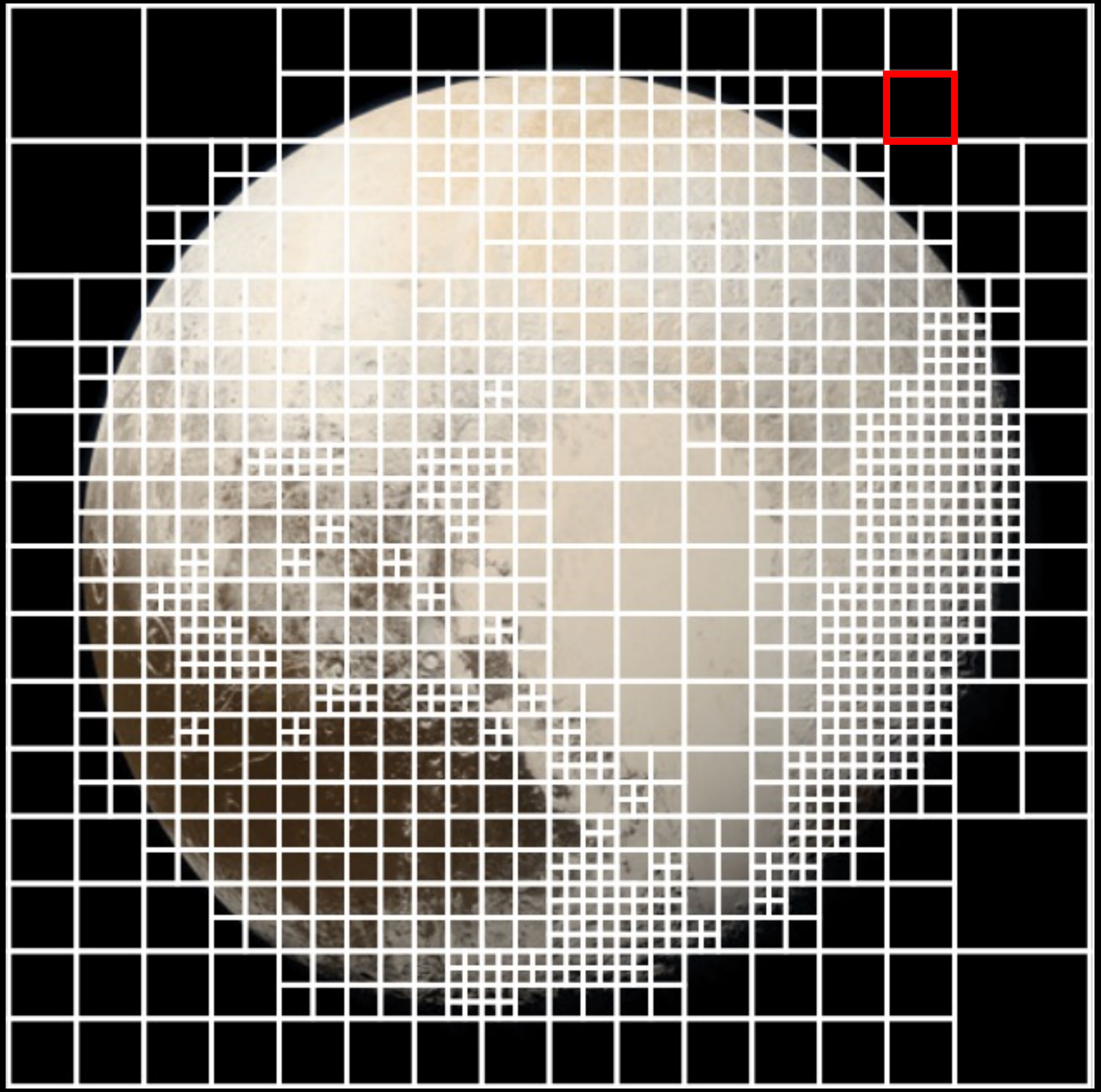
Coarser grid for low level of details

Finer grid for high level of details



Quadtree representing $\boxed{}$ region

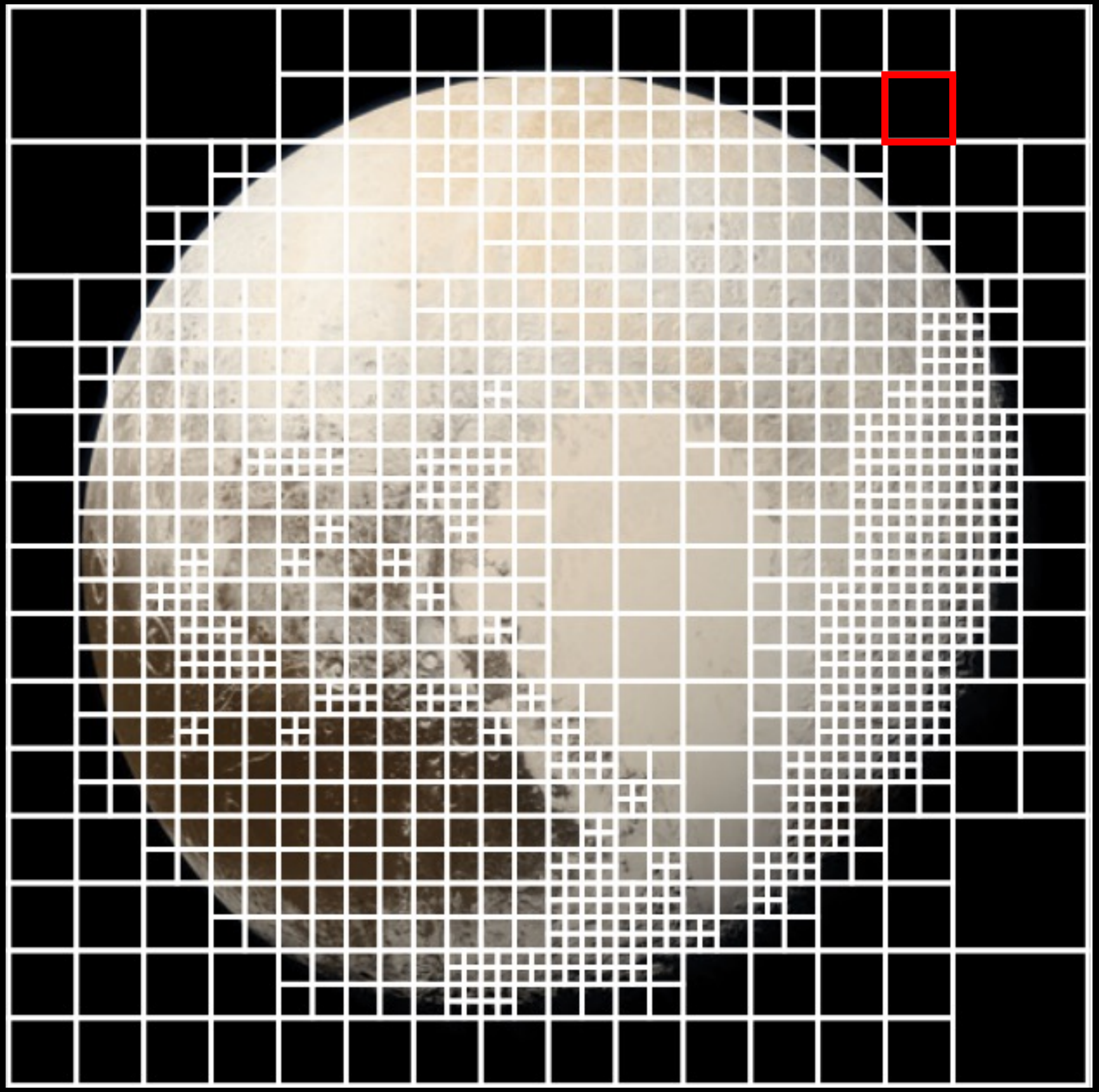




1. should $\boxed{}$ split, stay as is, or merge?

$$(I_b^{\downarrow}, I_b^{=}, I_b^{\uparrow}) \in \{0,1\}^3$$

$$I_b^{\downarrow} + I_b^{=} + \frac{1}{8}\left(I_b^{\uparrow} + \sum_{b' \in \mathcal{S}(b)} I_{b'}^{\uparrow}\right) = 1$$

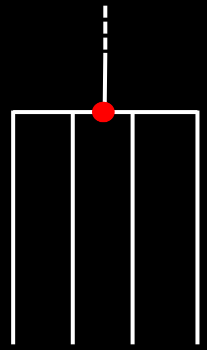


2. error to split, stay as is, merge

$$\mathbf{w}_b = (w_b^{\downarrow}, w_b^{=}, w_b^{\uparrow})^{\mathsf{T}}$$

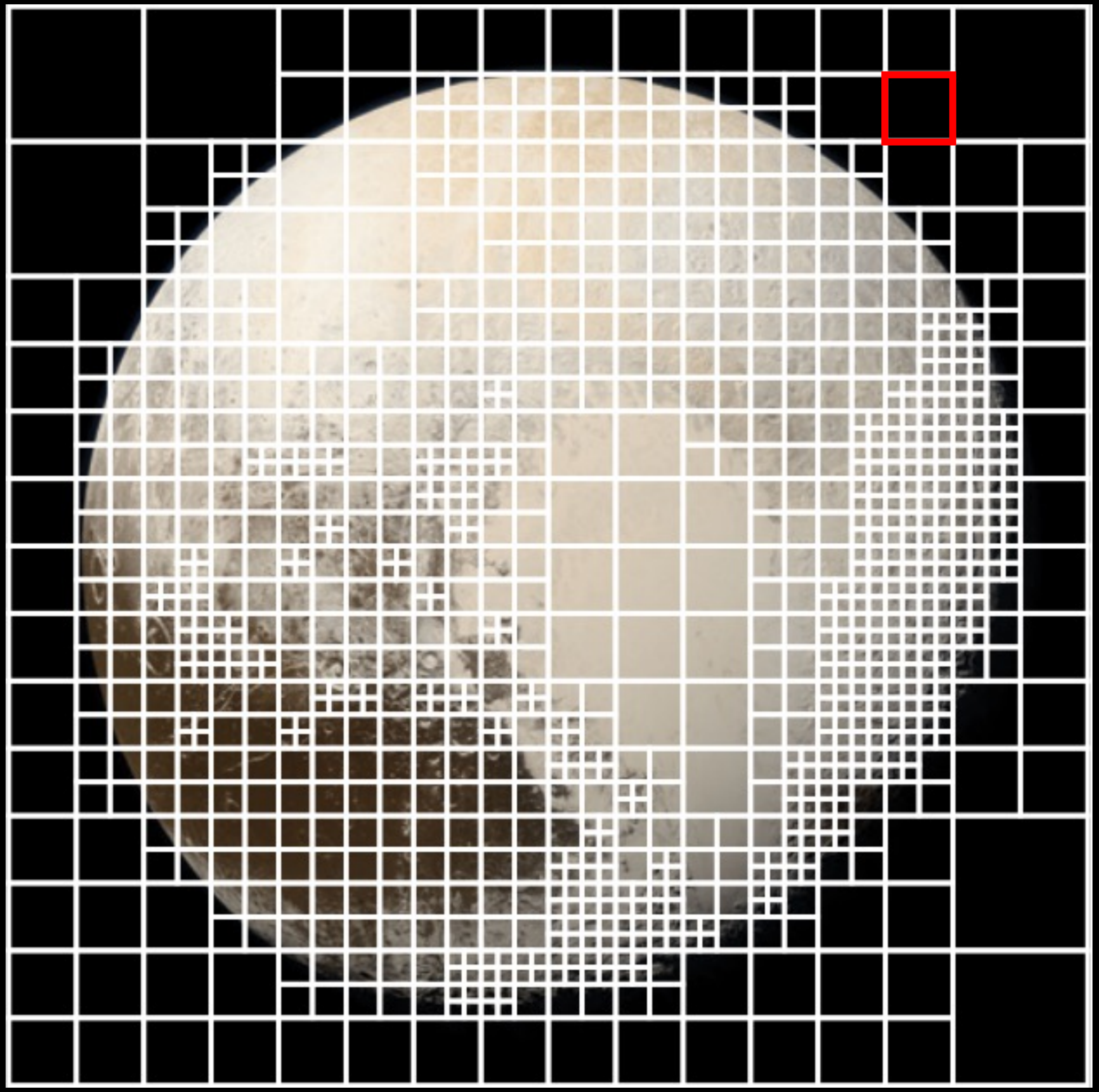
with

$$\mathbf{I}_b = (I_b^{\downarrow}, I_b^{=}, I_b^{\uparrow})^{\mathsf{T}}$$



What is the best partition?

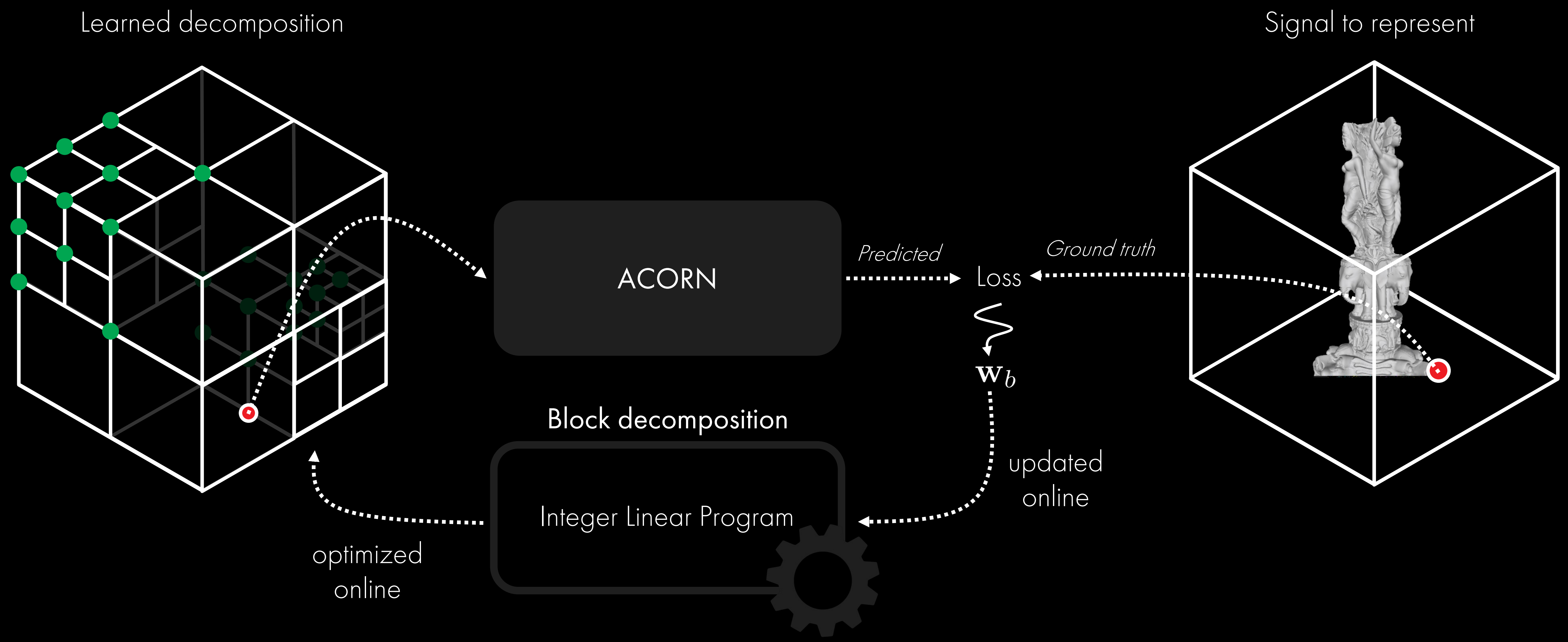
minimize $\sum_b \mathbf{w}_b^{\mathsf{T}} \mathbf{I}_b$



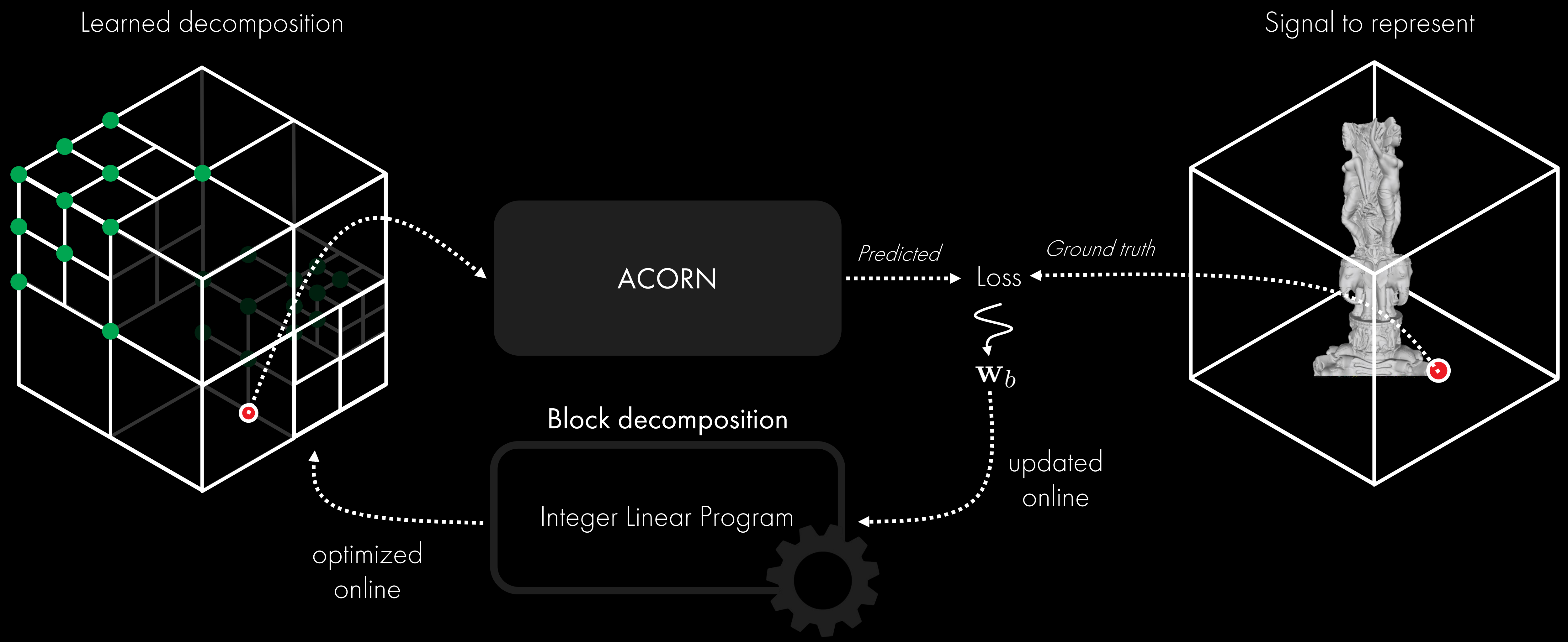
3. Limit the total number of blocks in the partition

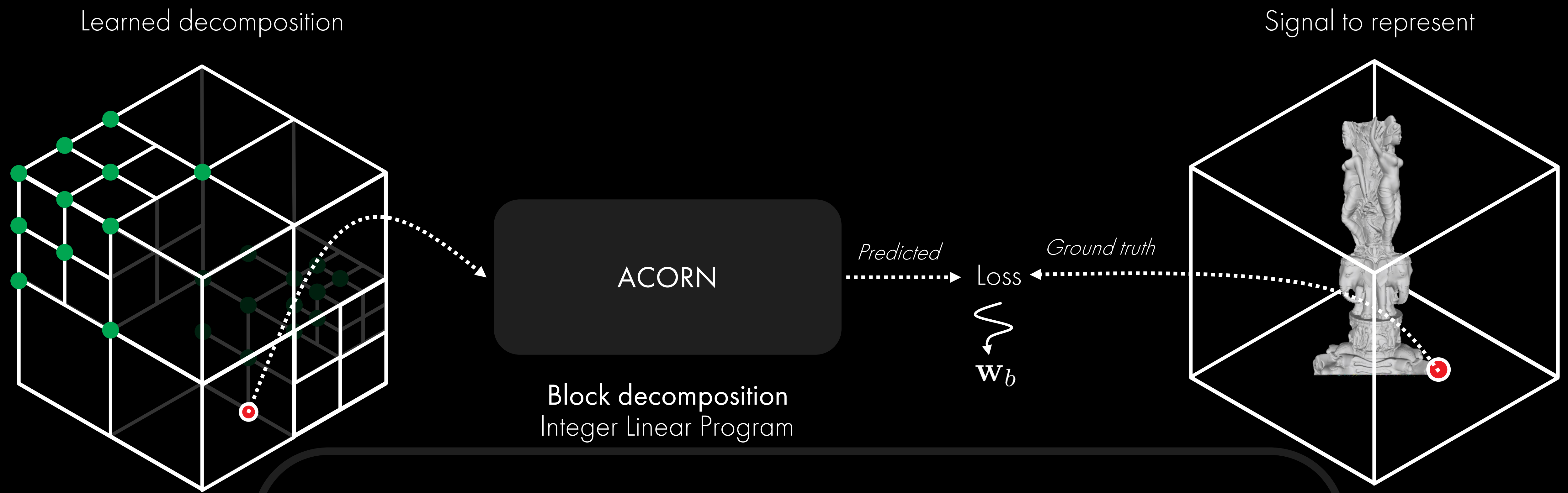
Assumption: the network has a given capacity, thus it can only fit a given number of blocks

$$\sum_b \frac{1}{8} I_b^{\uparrow} + I_b^{=} + 8\, I_b^{\downarrow} \leq \text{Max. Num. Blocks}$$



Learned decomposition
Signal to represent
ACORN
Predicted
Loss
Ground truth
$\mathbf{w}_b$
Block decomposition
Integer Linear Program
updated online
optimized online


Learned decomposition
Signal to represent
ACORN
Predicted
Ground truth
Loss
$\mathbf{w}_b$
updated online
Block decomposition
Integer Linear Program
optimized online



minimize total loss in the partition

$$\sum_b \mathbf{w}_b^\intercal \mathbf{I}_b \text{ , with } \quad \mathbf{I}_b = (I_b^\downarrow, I_b^=, I_b^\uparrow)^\intercal \text{ and } \mathbf{w}_b = (w_b^\downarrow, w_b^=, w_b^\uparrow)^\intercal$$

subject to: a maximum number of blocks

$$\sum_b \frac{1}{8} I_b^\uparrow + I_b^= + 8\, I_b^\downarrow \le \text{Max. Num. Blocks}$$

each block can either be split or merged, and the partition remains valid

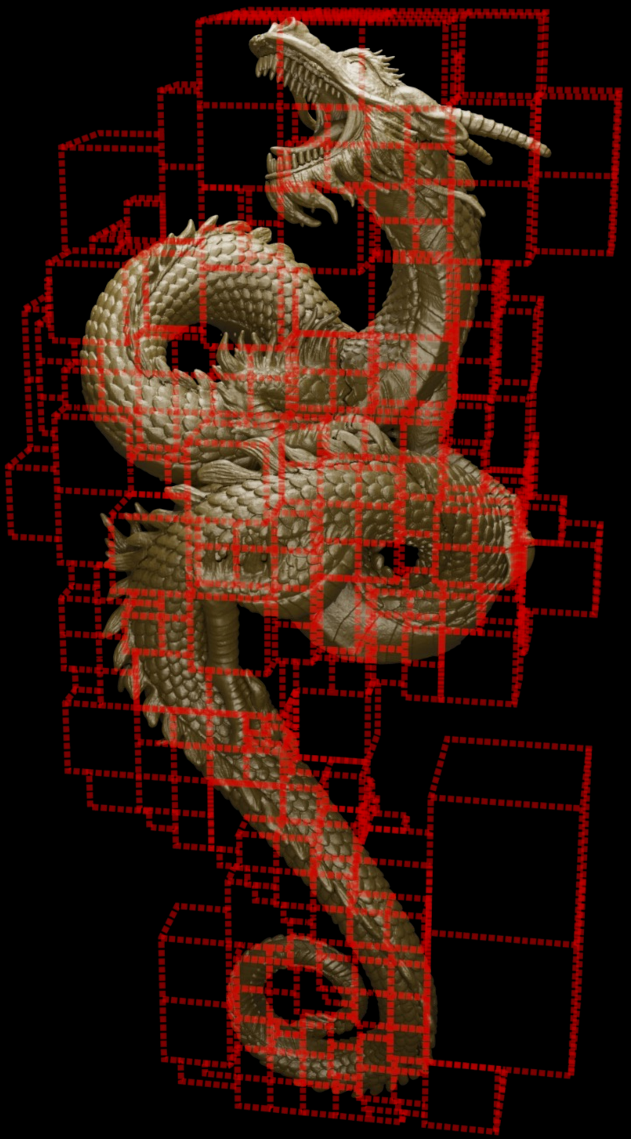
$$I_b^\downarrow + I_b^= + \frac{1}{8}\Big(I_b^\uparrow + \sum_{b' \in \mathcal{S}(b)} I_{b'}^\uparrow\Big) = 1$$

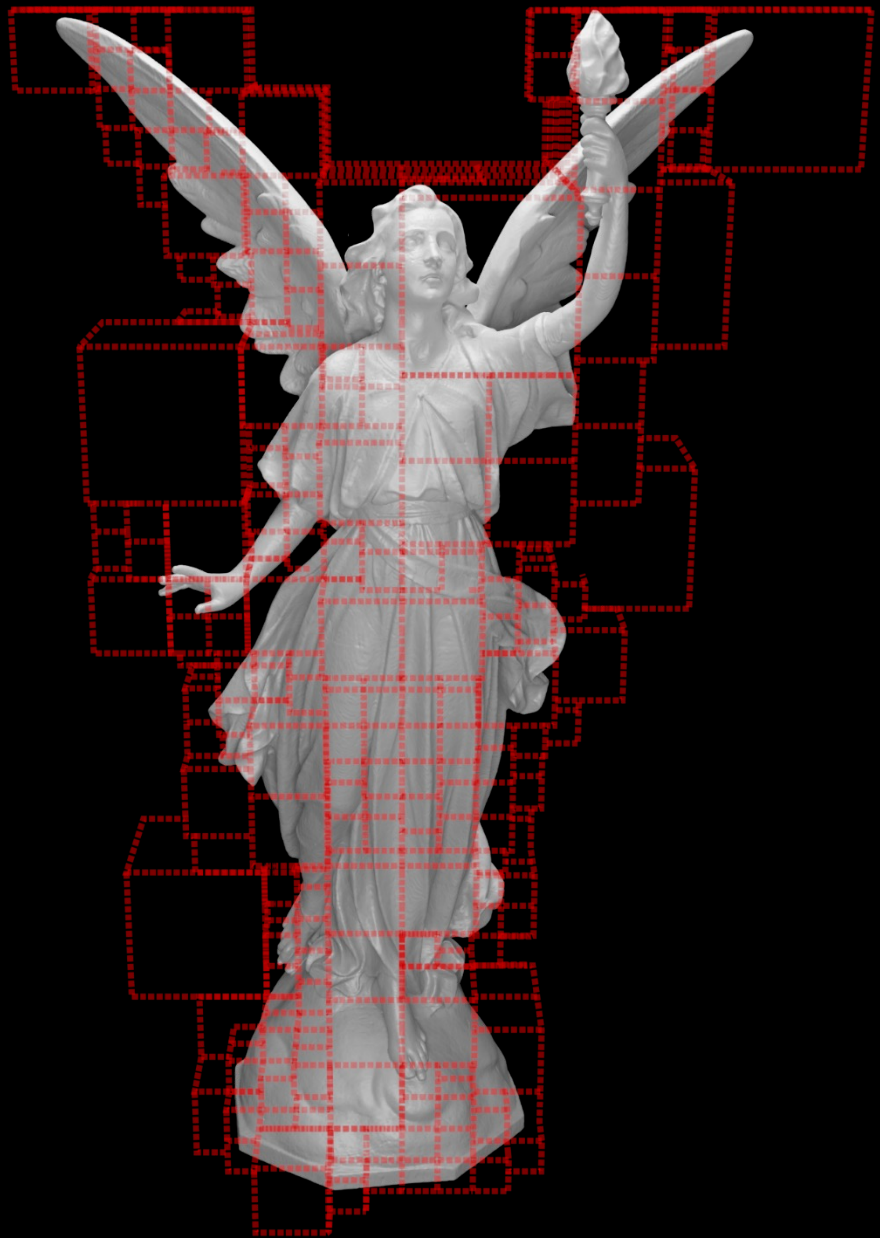
Results

ACORN
Output

Photo: Trevor Dobson, https://creativecommons.org/licenses/by-nc-nd/2.0/

3D Shapes



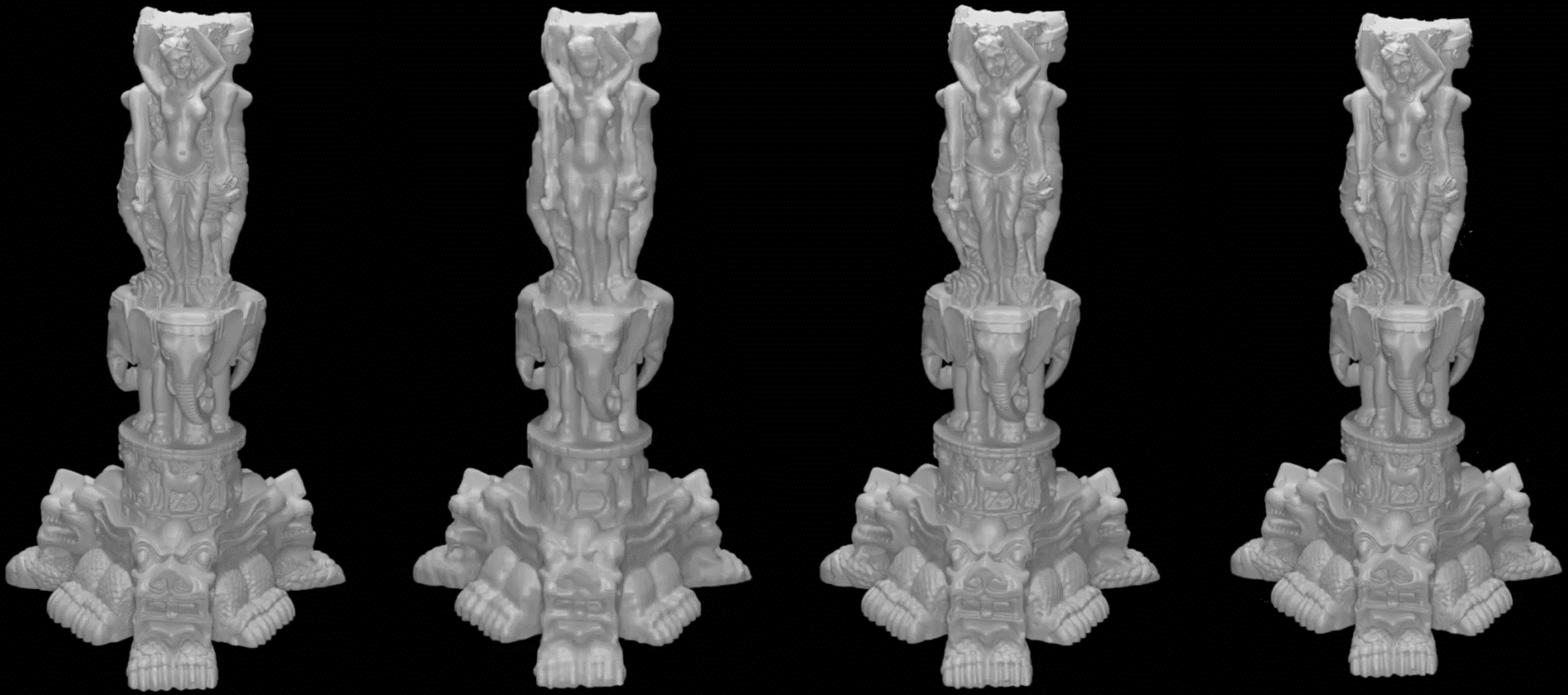


Lucy Statue: Stanford 3D Scanning Repository

3D Shapes

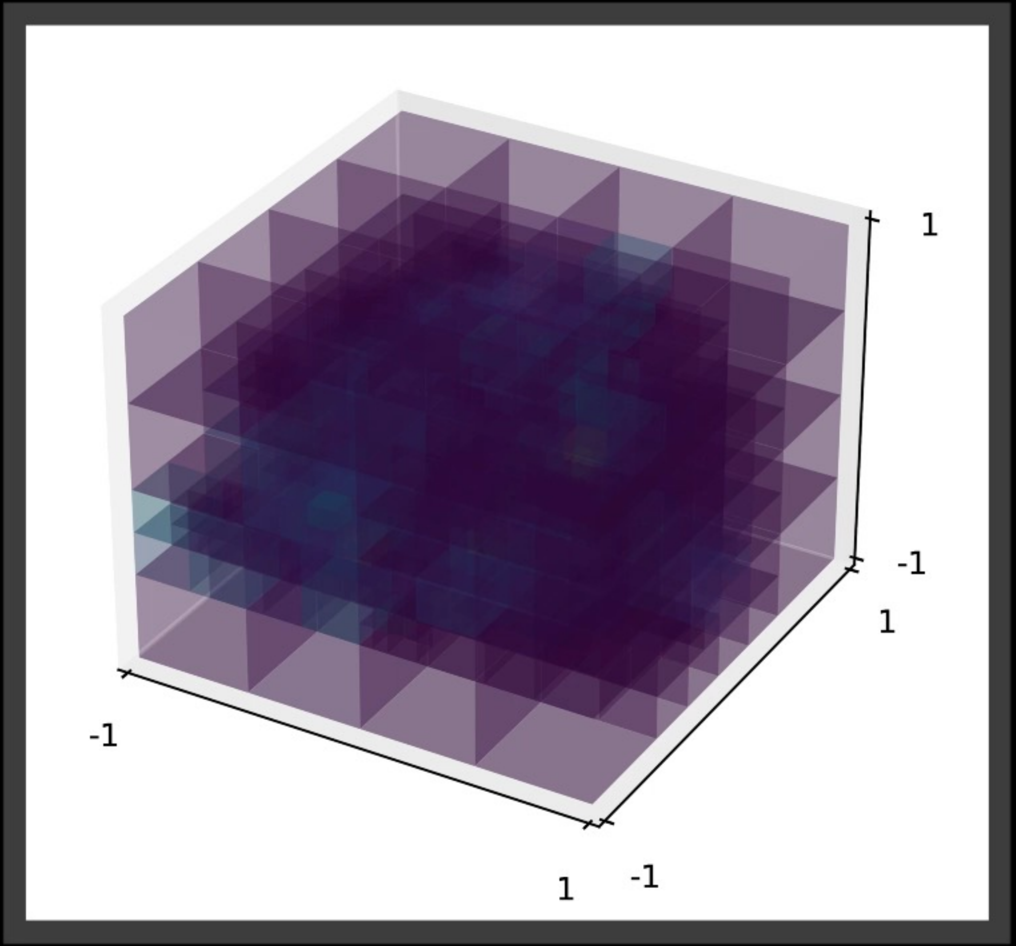
Ground Truth	Conv. Occ. Nets	SIREN	ACORN
Chamfer↓/F1-Score↑	3.98e-05/0.947	1.18e-05/0.990	1.04e-05/0.997

3D Shapes

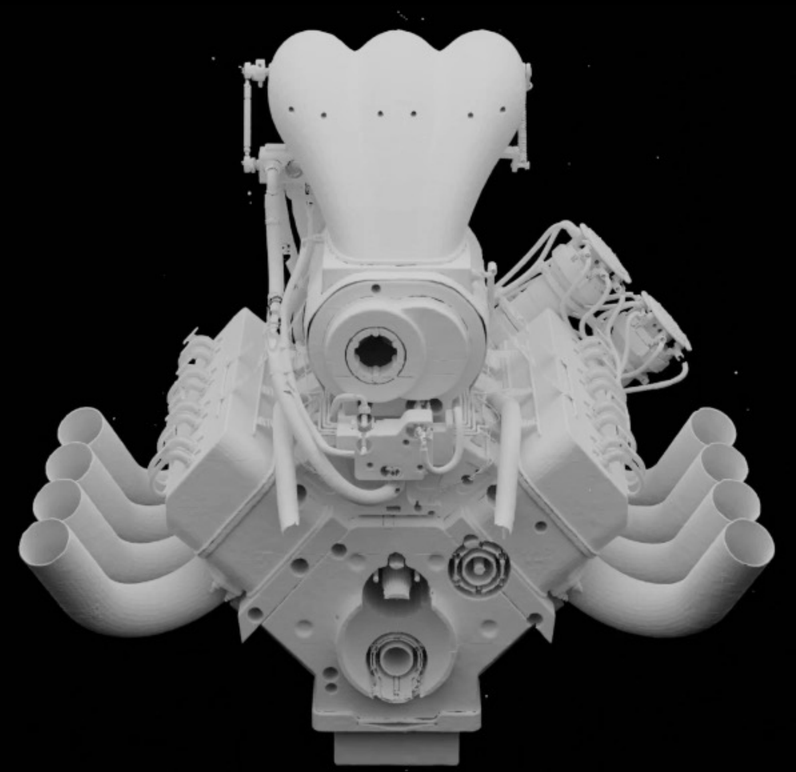


Ground Truth	Conv. Occ. Nets	SIREN	ACORN
Chamfer↓/F1-Score↑	2.16e-05/0.982	2.06e-05/0.996	1.55e-05/0.999

Adaptive Block Decomposition

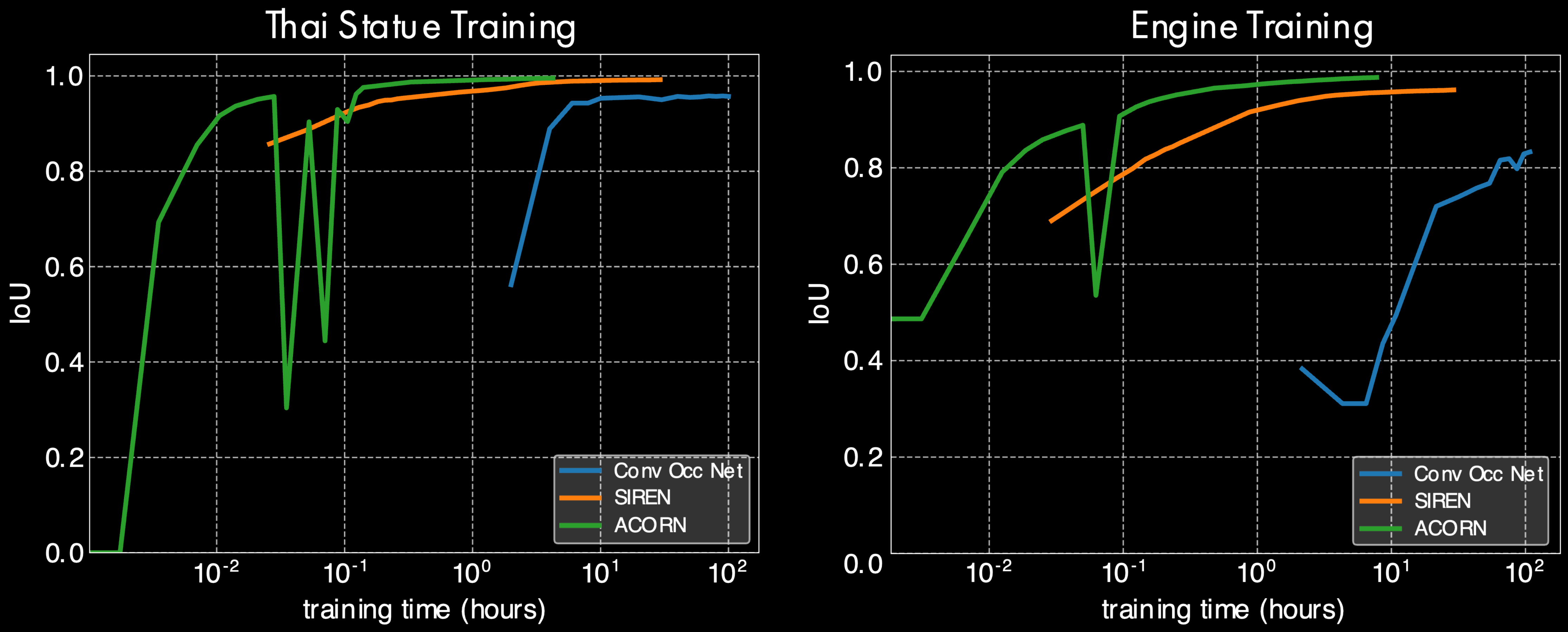


Block Decomposition Training Progression



Final ACORN Reconstruction

Training Time



Recent followup: Instant NGP

Real-time Gigapixel fitting!

[Müller et al. '22]

Recent followup: Instant NGP

...and NeRF fitting!

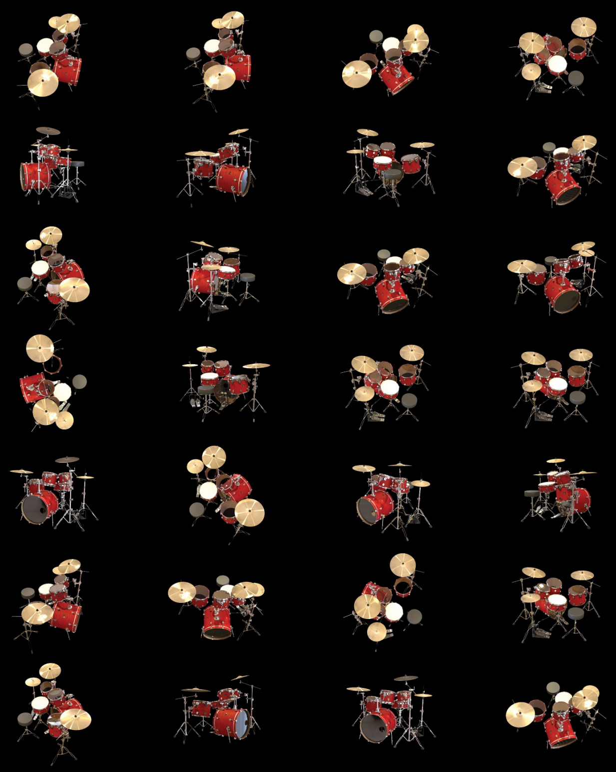
[Müller et al. '22]

Outline

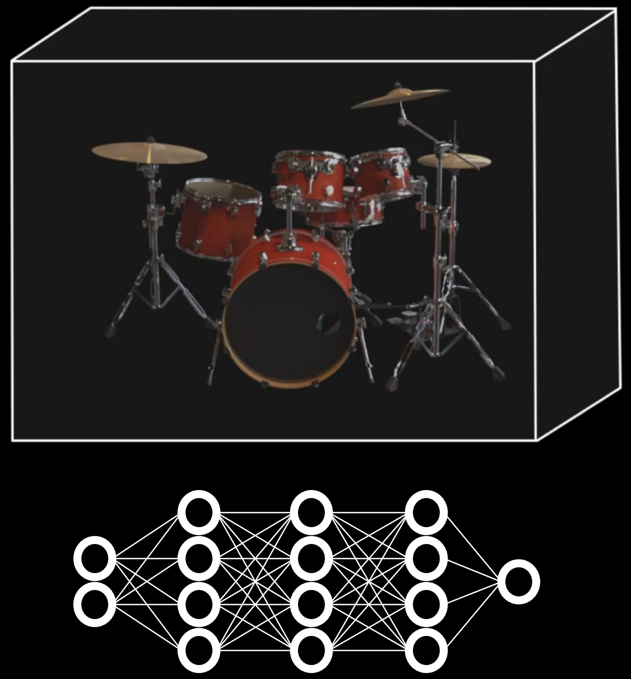
- Overview of neural fields
- Representing 3D shape
- Neural rendering with NeRF
- Case studies
    - SIREN
    - ACORN
    - AutoInt
    - BACON

$$\Phi(x) = \int \Psi(x)\,\mathrm{d}x$$

Mildenhall et al. '20
Liu et al. '20
Schwarz et al. '20
...

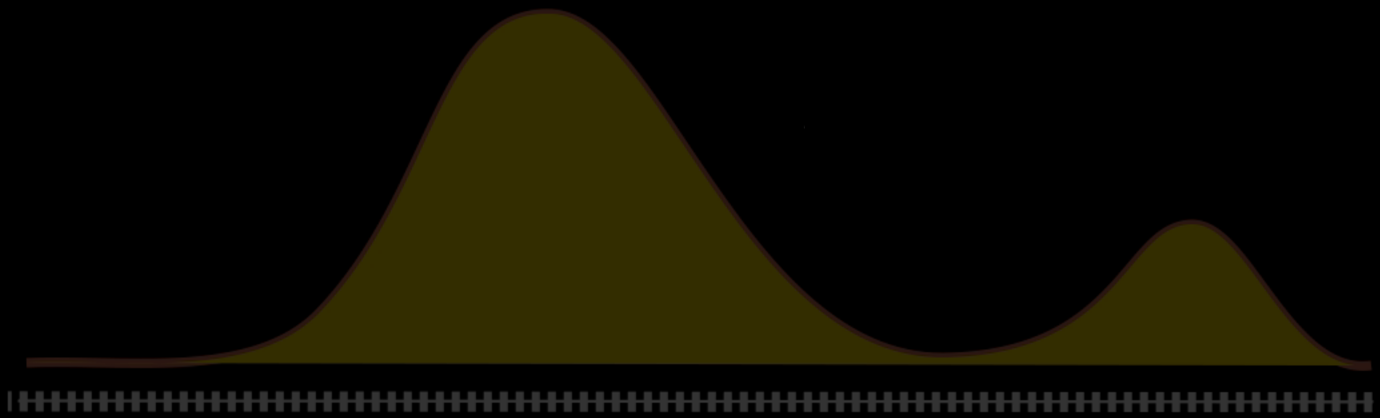


Training Set

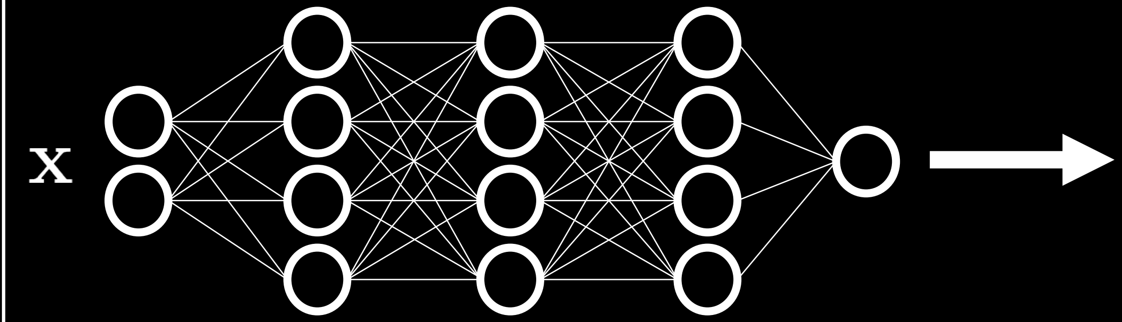


Learned Volume

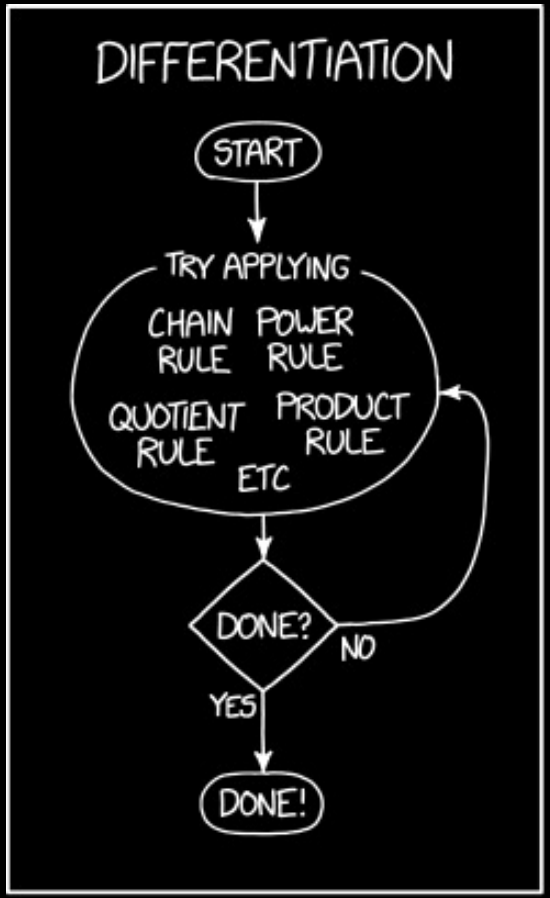
Novel Views (from VRE)

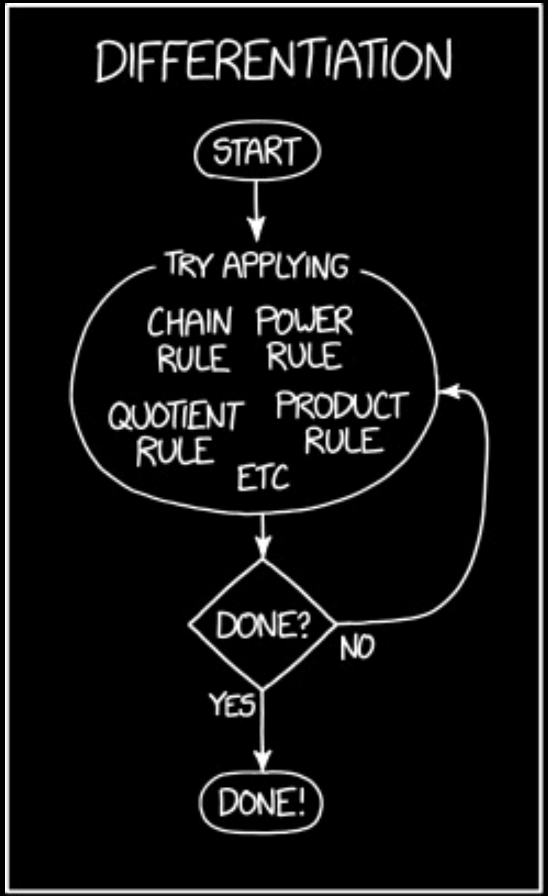


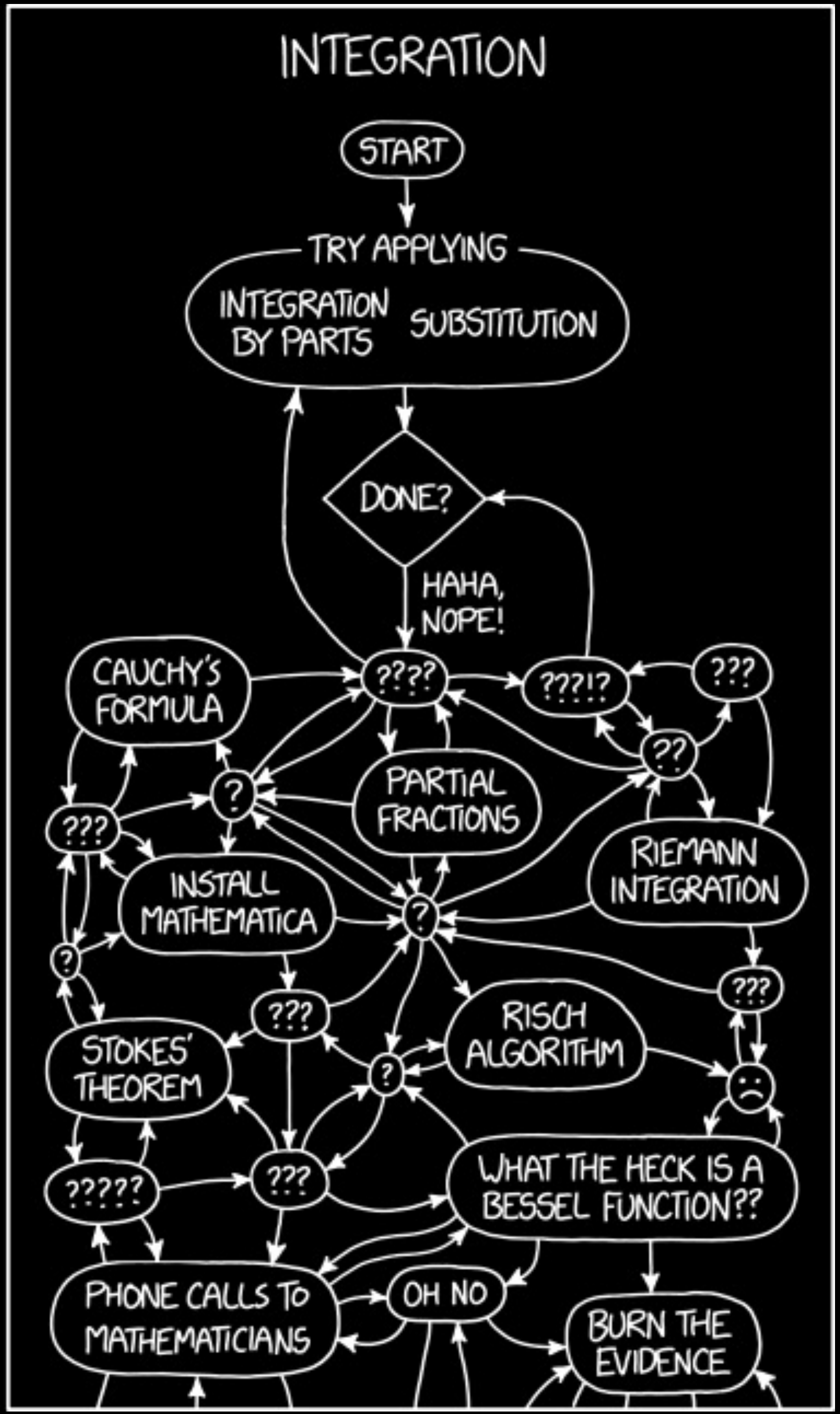
Closed-form Solution



$$\int f(x)\,\mathrm{d}x$$

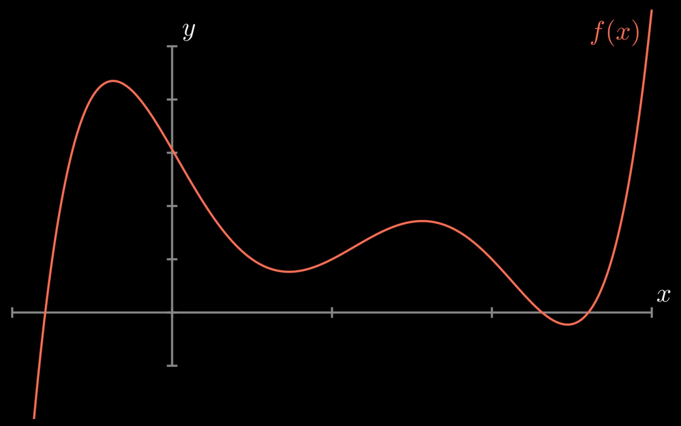

DIFFERENTIATION
START
TRY APPLYING
CHAIN RULE
POWER RULE
QUOTIENT RULE
PRODUCT RULE
ETC
DONE?
NO
YES
DONE!


DIFFERENTIATION
START
TRY APPLYING
CHAIN RULE
POWER RULE
QUOTIENT RULE
PRODUCT RULE
ETC
DONE?
NO
YES
DONE!

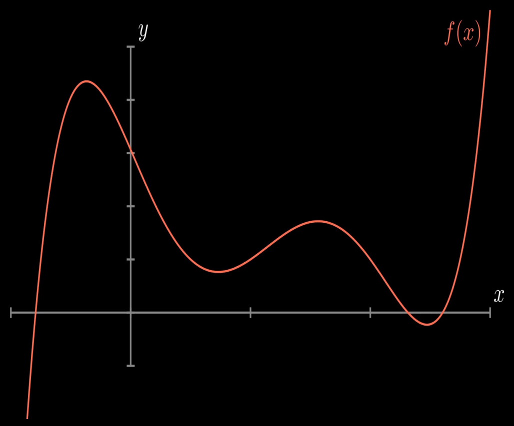

INTEGRATION
START
TRY APPLYING
INTEGRATION BY PARTS
SUBSTITUTION
DONE?
HAHA, NOPE!
CAUCHY'S FORMULA
????
???!?
???
??
?
???
PARTIAL FRACTIONS
RIEMANN INTEGRATION
INSTALL MATHEMATICA
?
?
???
RISCH ALGORITHM
???
STOKES' THEOREM
?
?????
???
WHAT THE HECK IS A BESSEL FUNCTION??
PHONE CALLS TO MATHEMATICIANS
OH NO
BURN THE EVIDENCE

Numerical Integration Techniques

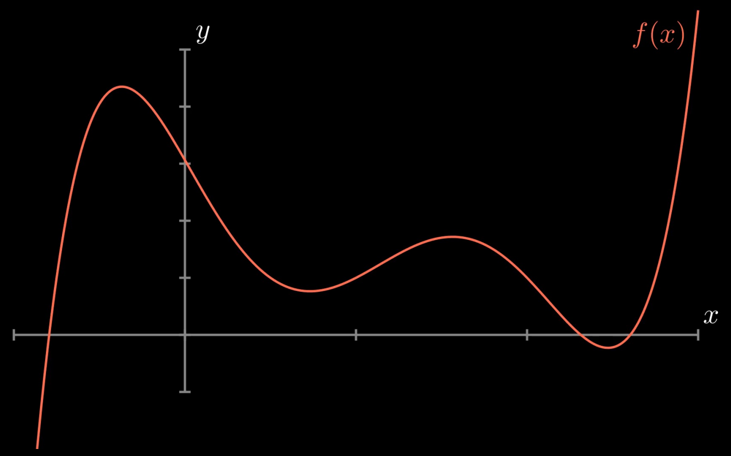
Riemann Sums



Quadrature

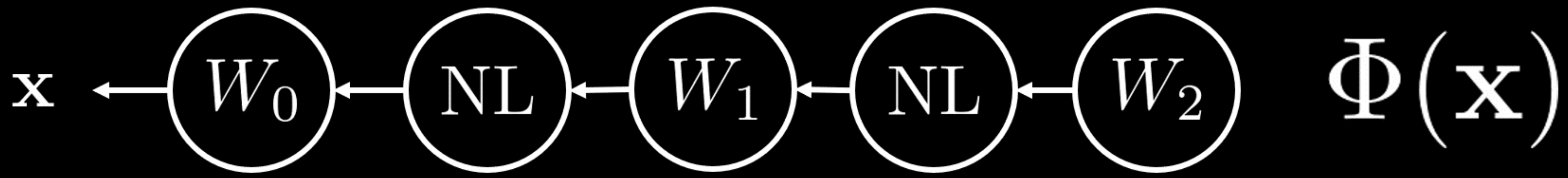


Monte Carlo

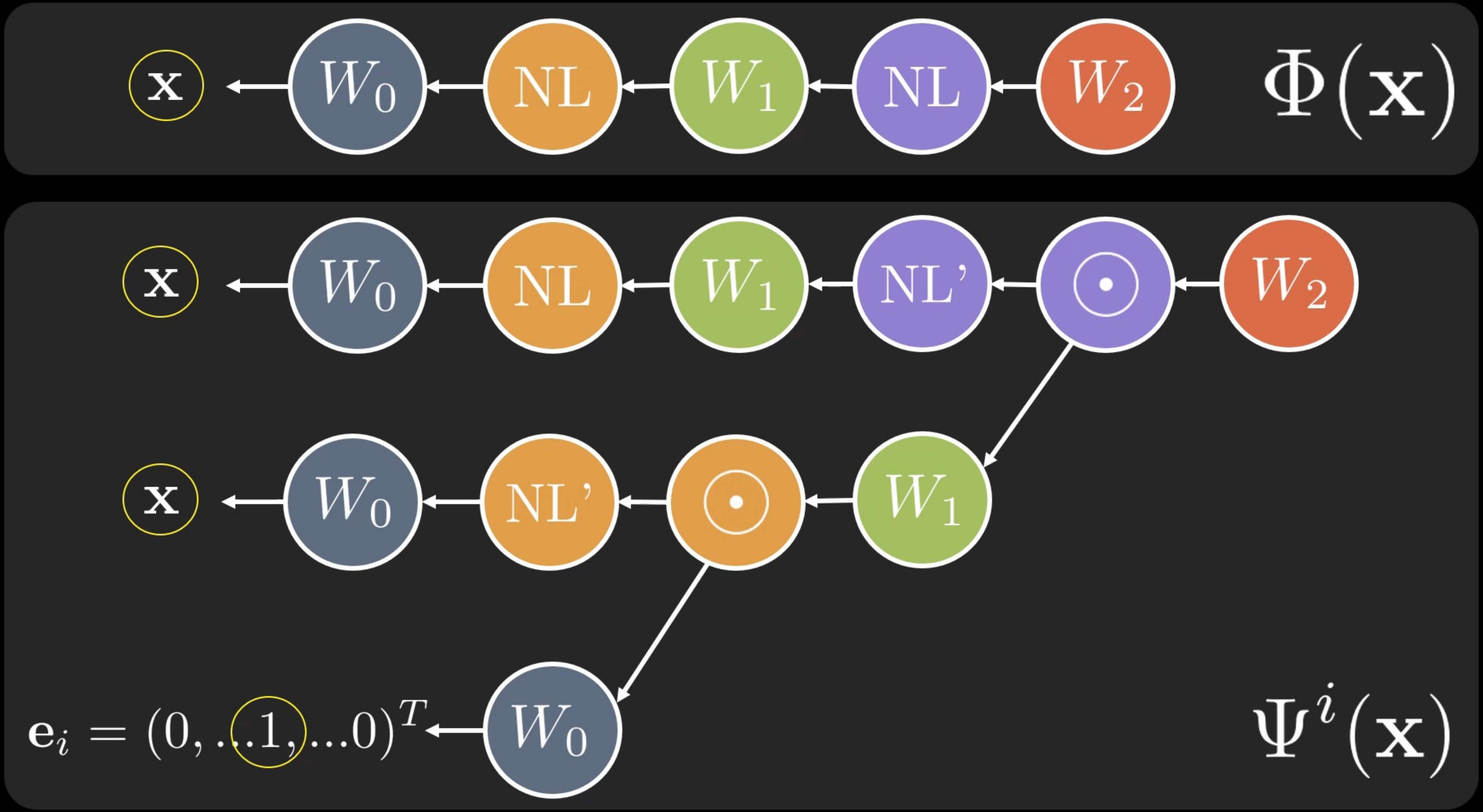


Automatic integration

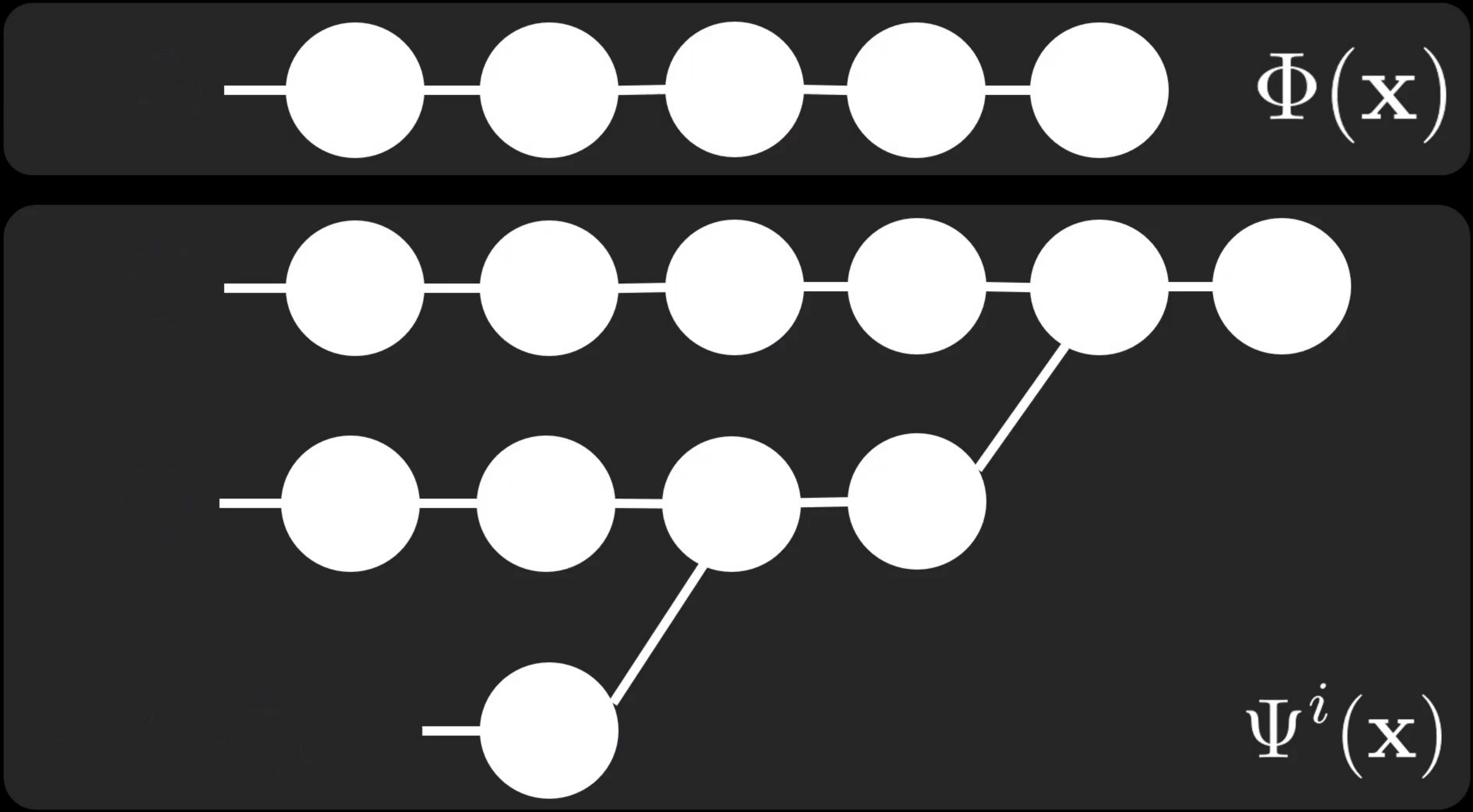
$$\mathbf{x} \leftarrow W_0 \leftarrow \mathrm{NL} \leftarrow W_1 \leftarrow \mathrm{NL} \leftarrow W_2 \qquad \Phi(\mathbf{x})$$



Integral Network



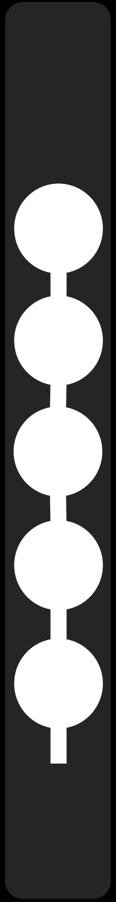
x
W_0
NL
W_1
NL
W_2
$\Phi(\mathbf{x})$
x
W_0
NL
W_1
NL'
$\odot$
W_2
x
W_0
NL'
$\odot$
W_1
$\mathbf{e}_i = (0, ...1, ...0)^T$
W_0
$\Psi^i(\mathbf{x})$


$\Phi(\mathbf{x})$
$\Psi^i(\mathbf{x})$

AutoInt steps

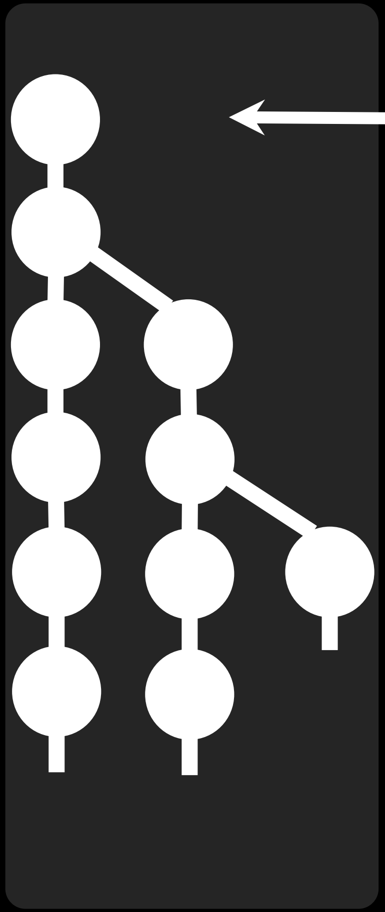
1 **Specify** integral network

$\Phi_\theta(\mathbf{x})$



2 **Instantiate** grad network

$\Psi_\theta^i(\mathbf{x})$

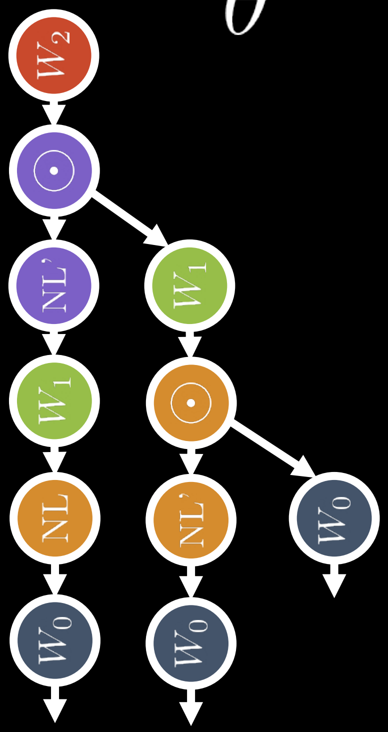


3 **Train** grad network

$\mathcal{L}\left(\Psi_\theta^i(\mathbf{x}), f(\mathbf{x})\right)$

4 **Reassemble** into the integral network

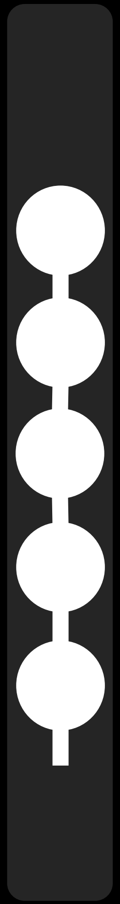
θ^*



AutoInt steps

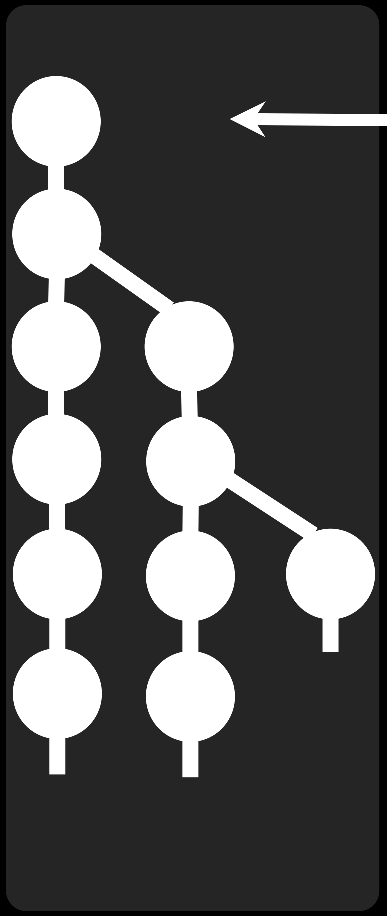
1 **Specify** integral network

$\Phi_\theta(\mathbf{x})$



2 **Instantiate** grad network

$\Psi_\theta^i(\mathbf{x})$

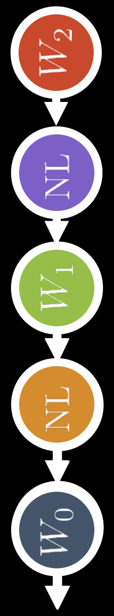


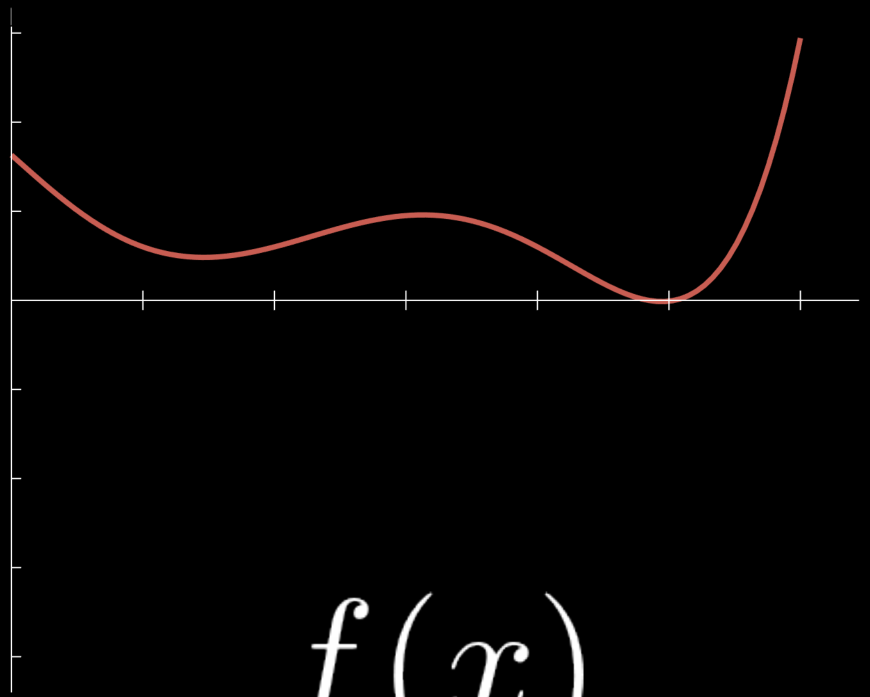
3 **Train** grad network

$$\mathcal{L}\left(\Psi_\theta^i(\mathbf{x}), f(\mathbf{x})\right)$$

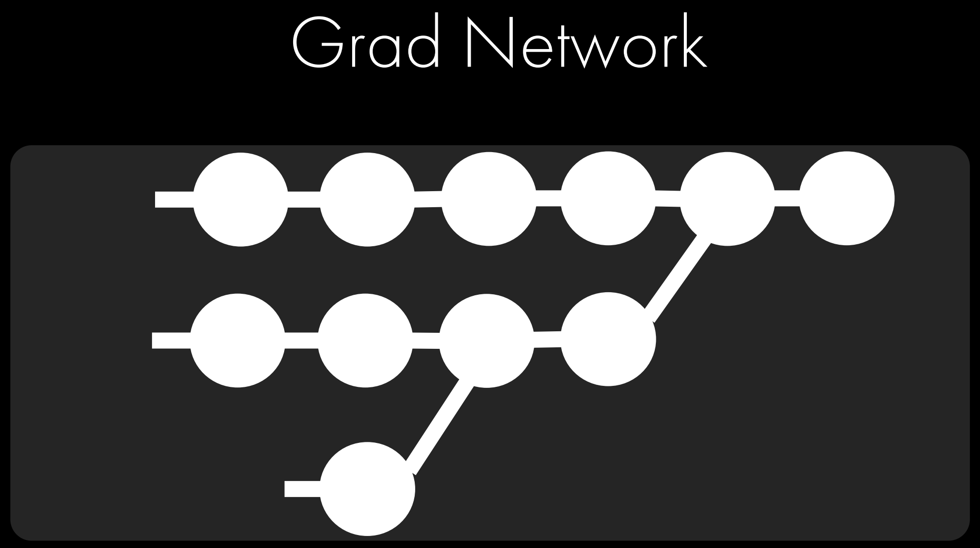
4 **Reassemble** into the integral network

θ^*

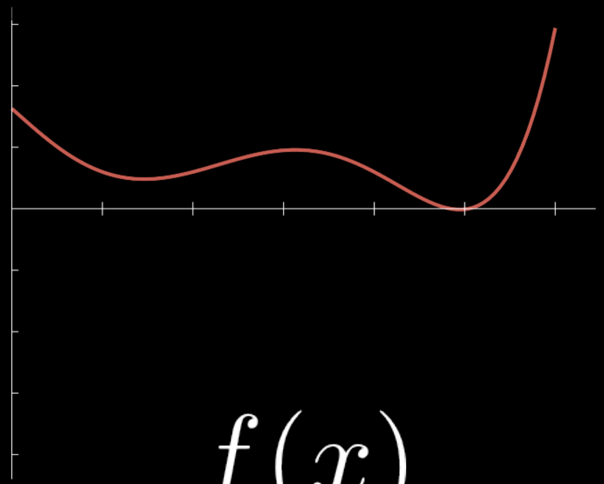




$$f(x)$$

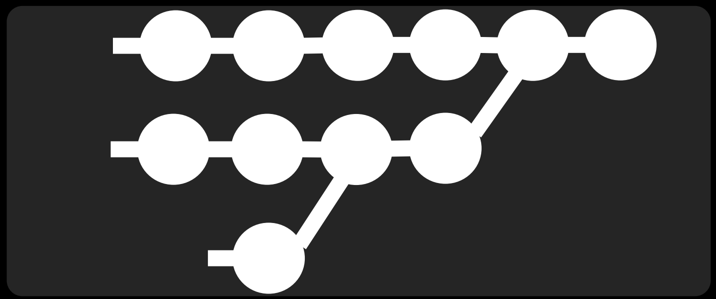


$$\Psi(x)$$



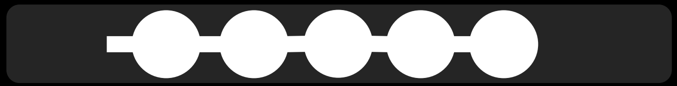
$$f(x)$$

Grad Network

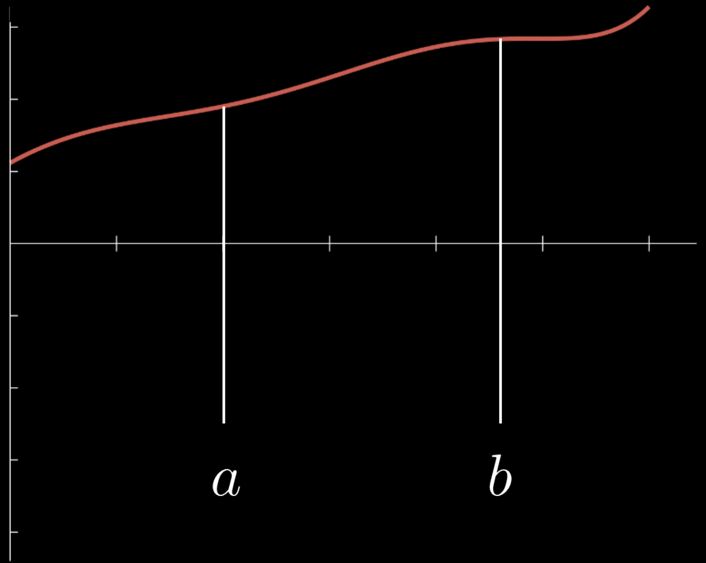


$$\Psi(x)$$

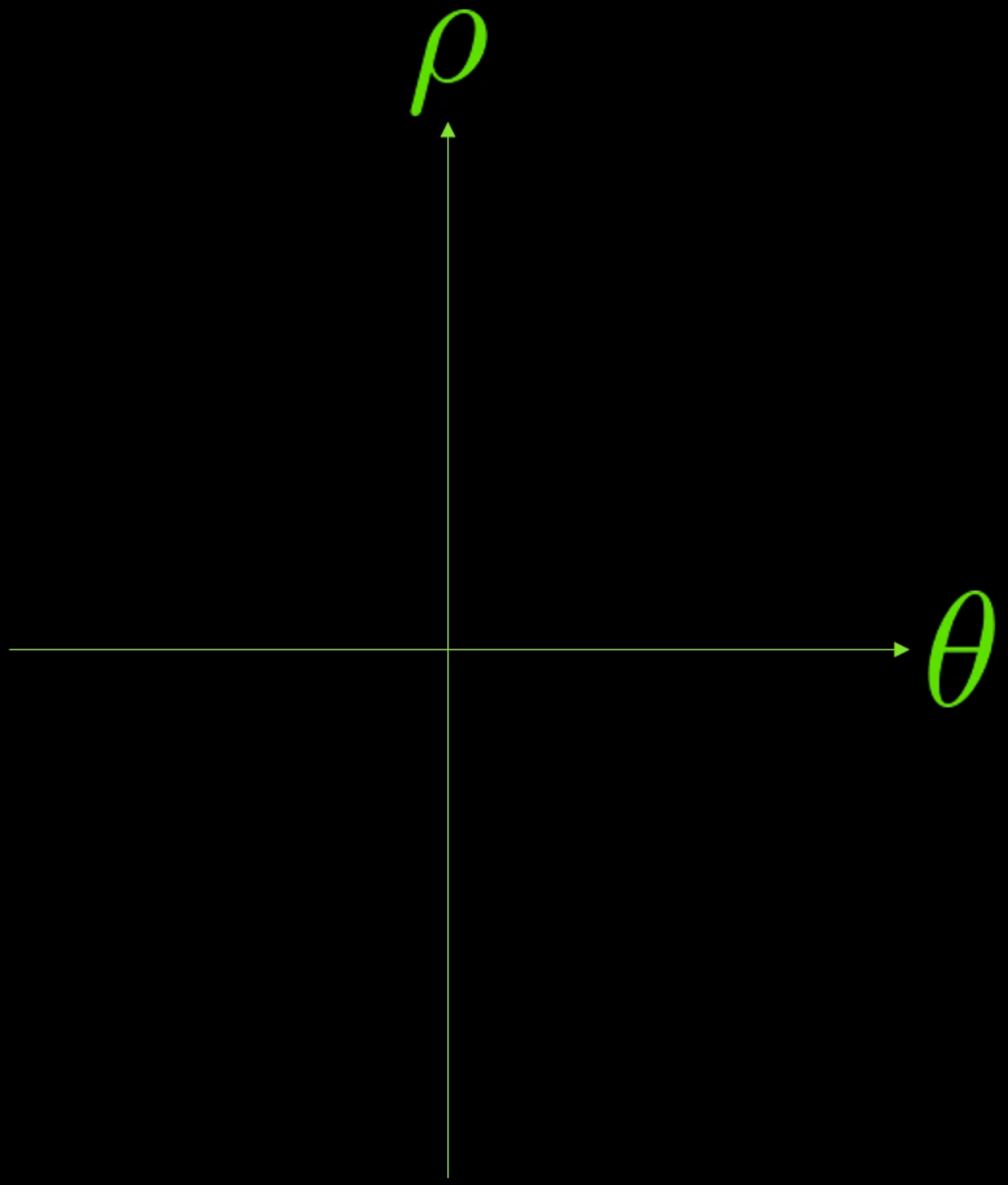
Integral Network



$$\Phi(x) = \int_0^x \Psi(x)\,\mathrm{d}x$$



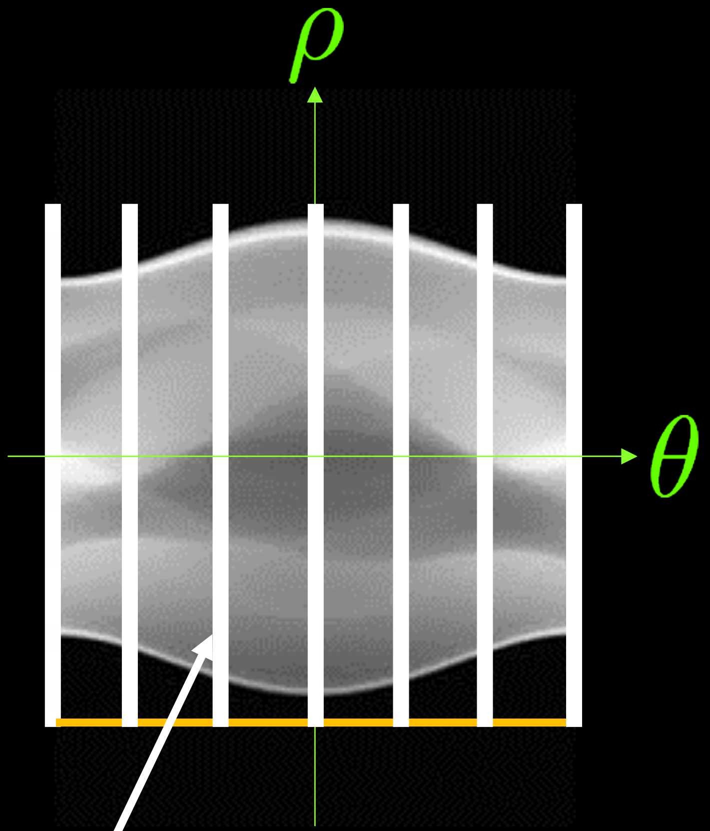
$$\int_a^b \Psi(x)\,\mathrm{d}x = \Phi(b) - \Phi(a)$$



ρ
θ

Training loss

$$\left\| \underbrace{\int_a^b \Psi(\mathbf{x})\, \mathrm{d}x}_{\approx \frac{b-a}{N}\sum_{i}^{N} \Psi(x_i)} \quad - \right.$$

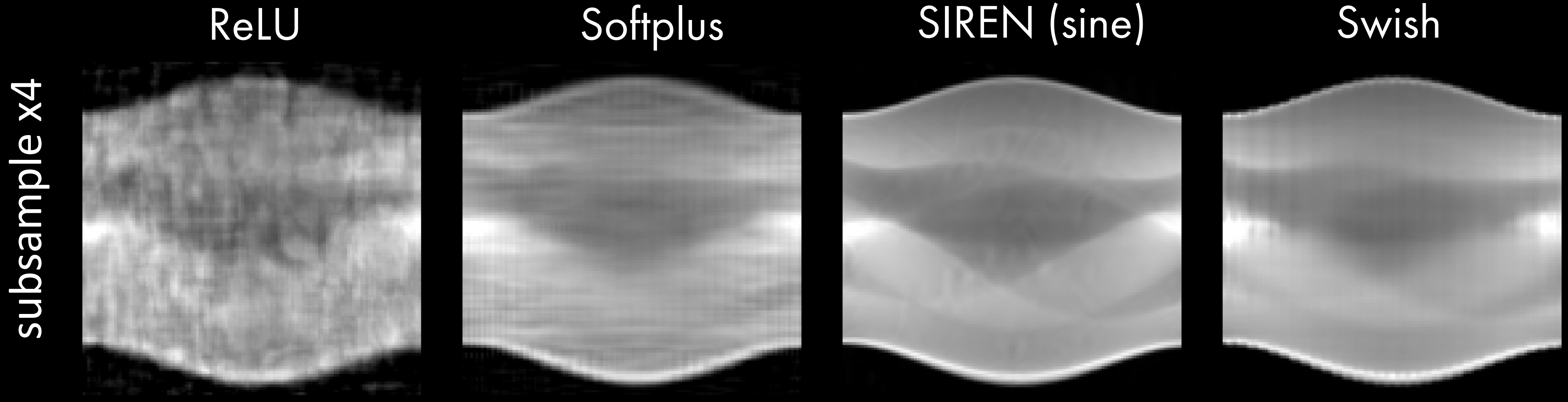


$$\left. \right\|_2^2$$

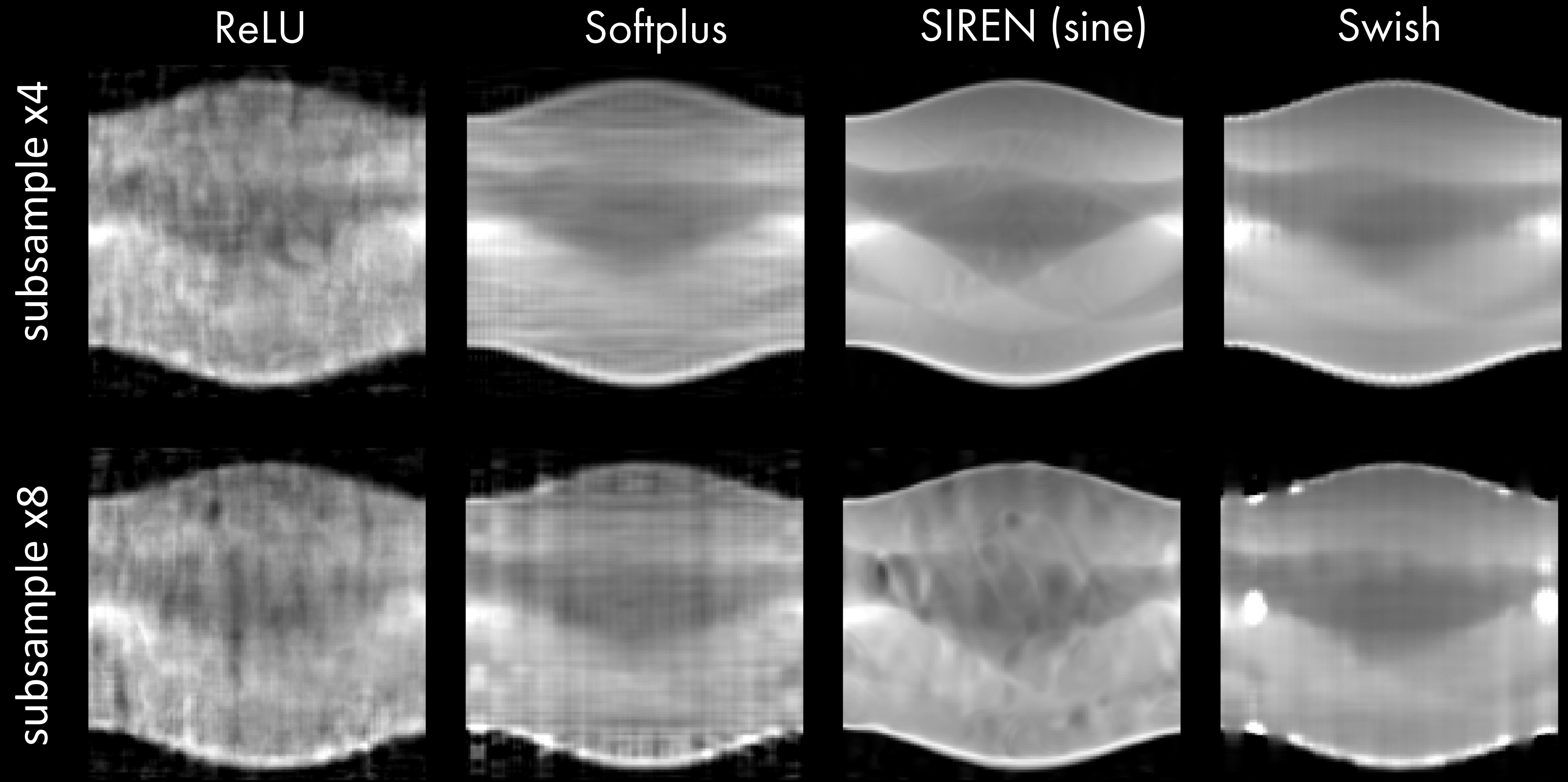
Monte Carlo approximation

$s(\rho, \theta)$

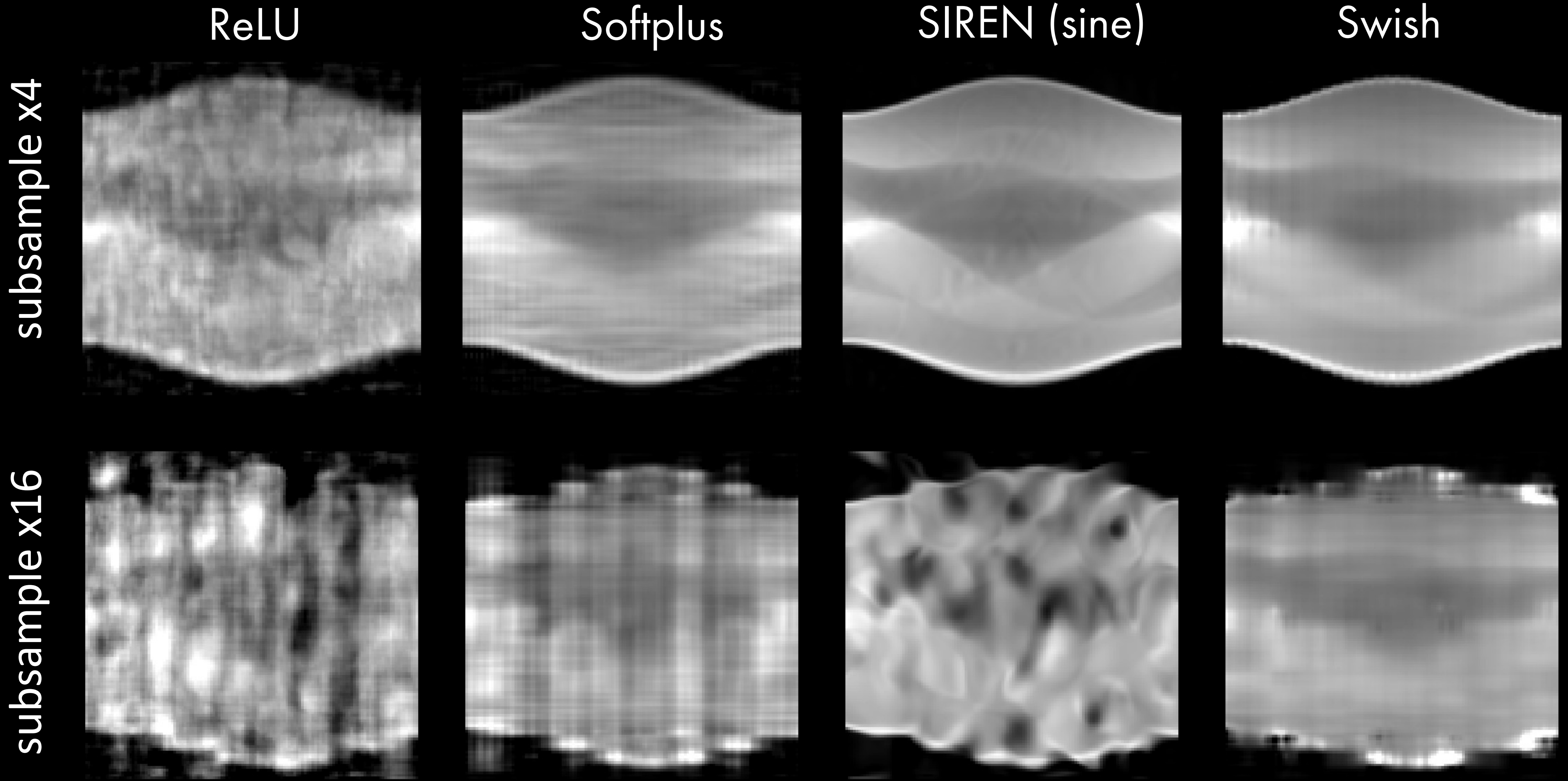
Integrating 2D Signals



Integrating 2D Signals



Integrating 2D Signals



Volume Rendering Equation (VRE)

$$\underbrace{\mathbf{C}(\mathbf{r})}_{\text{color of rendered ray}} = \int_{t_n}^{t_f} \underbrace{\sigma(\mathbf{r}(t))}_{\substack{\text{absorption}\\ \text{coefficient}}} \underbrace{e^{\left\{-\int_{t_n}^{t} \sigma(\mathbf{r}(s))\,\mathrm{d}s\right\}}}_{\text{transmittance}} \underbrace{\mathbf{c}(\mathbf{r}(t))}_{\substack{\text{emissive}\\ \text{radiance}}}\,\mathrm{d}t$$

Volume Rendering Equation (VRE)

$$\mathbf{C}(\mathbf{r}) = \boxed{\int_{t_n}^{t_f}} \sigma(\mathbf{r}(t)) e^{\left\{-\boxed{\int_{t_n}^{t}} \sigma(\mathbf{r}(s))\,\mathrm{d}s\right\}} \mathbf{c}(\mathbf{r}(t))\,\mathrm{d}t$$

Approximation of the VRE

sum over piecewise segments

$$\tilde{\mathbf{C}}(\mathbf{r}) = \sum_{i=1}^{N} \bar{\mathbf{c}}_i(t)\bar{T}_i(t)\bar{\sigma}_i(t)\delta_i$$

color of rendered ray

emissive radiance

transmittance

absorption coefficient

with $\delta_i = t_i - t_{i-1}$

Approximation of the VRE

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

average transmittance

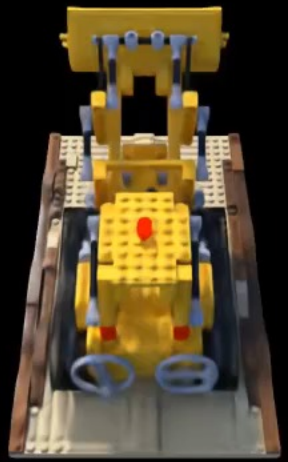
$$\tilde{\mathbf{C}}(\mathbf{r}) = \sum_{i=1}^{N} \bar{\mathbf{c}}_i(t)\bar{T}_i(t)\bar{\sigma}_i(t)\delta_i$$

$$\bar{\sigma}_i = \delta_i^{-1}\int_{t_{i-1}}^{t_i} \sigma(t)\mathrm{d}t$$

average absorption value

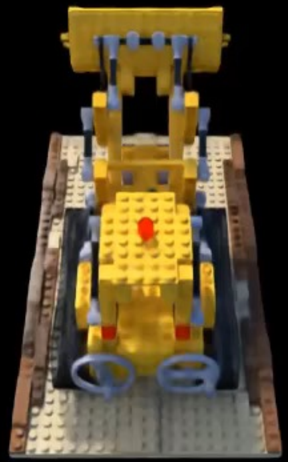
$$\bar{\mathbf{c}}_i = \delta_i^{-1}\int_{t_{i-1}}^{t_i} \mathbf{c}(t)\mathrm{d}t$$

average radiance value



4 sections

8 sections



16 sections

32 sections

64 sections

NeRF
[Mildenhall '20]
30 s/frame

Neural Volumes
[Lombardi '19]
0.3 s/frame

AutoInt
8 Sections
2.6 s/frame

AutoInt
32 Sections
9.3 s/frame

NeRF
[Mildenhall '20]
30 s/frame

Neural Volumes
[Lombardi '19]
0.3 s/frame

AutoInt
8 Sections
2.6 s/frame

AutoInt
32 Sections
9.3 s/frame

Statue

Globe

AutoInt
8 sections

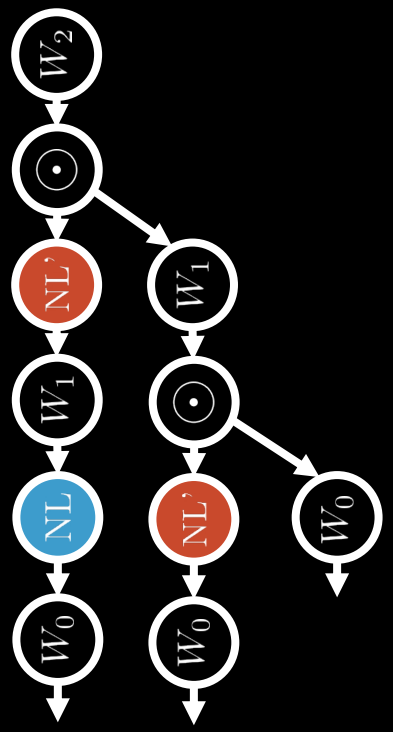
AutoInt
32 sections

Open questions

1

Expressiveness of grad network architectures deriving from standard MLPs

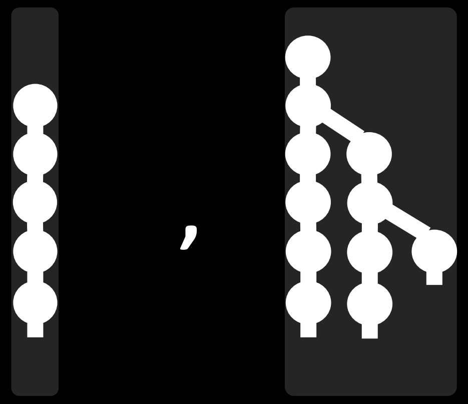
...network is a tree that contains both NL and NL'



2

Other pairs of (grad, integral) networks...

...that might be easier to train and more expressive



$\Phi(\mathbf{x})$, $\Psi^i(\mathbf{x})$

3

Training the integral network directly...

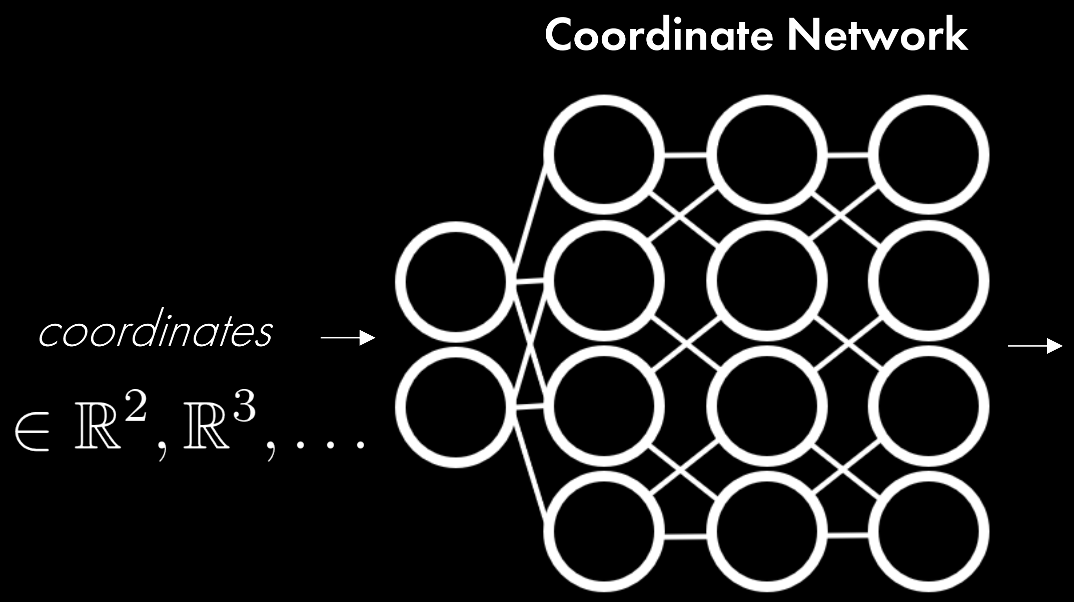
...for instance via the bounds:

$$F(a) - F(b)$$

seems to work when a and b are sampled densely

Outline

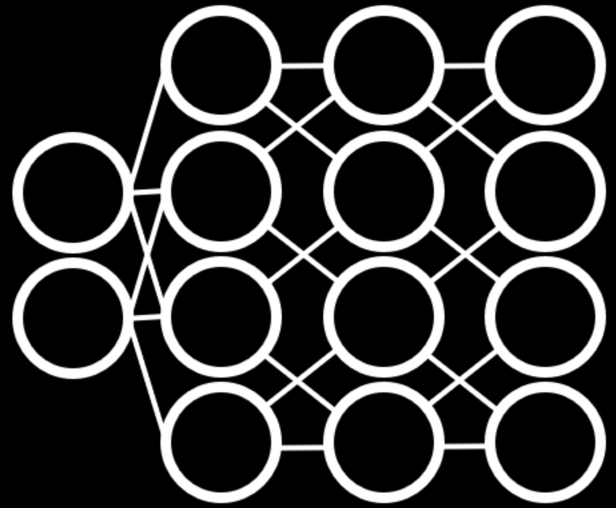
- Overview of neural fields
- Representing 3D shape
- Neural rendering with NeRF
- Case studies
    - SIREN
    - ACORN
    - AutoInt
    - BACON


Coordinate Network
coordinates
$\in \mathbb{R}^2, \mathbb{R}^3, \ldots$

Coordinate Network

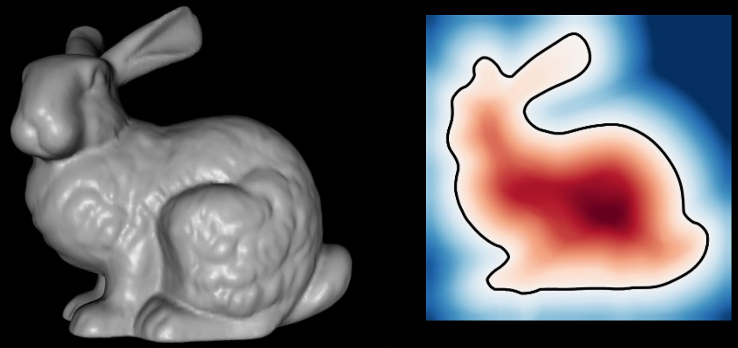
coordinates $\rightarrow$

$\in \mathbb{R}^2, \mathbb{R}^3, \ldots$



$\rightarrow$

pixel values



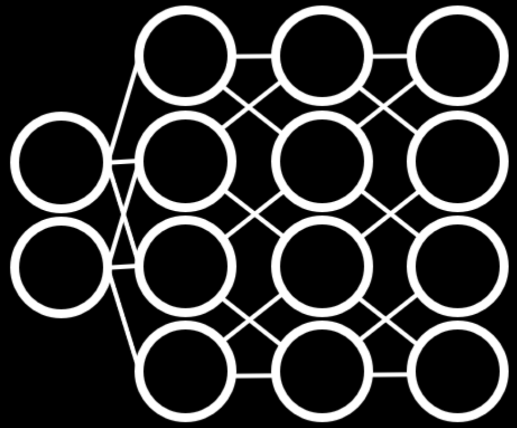
signed distance functions

neural radiance fields

coordinate network

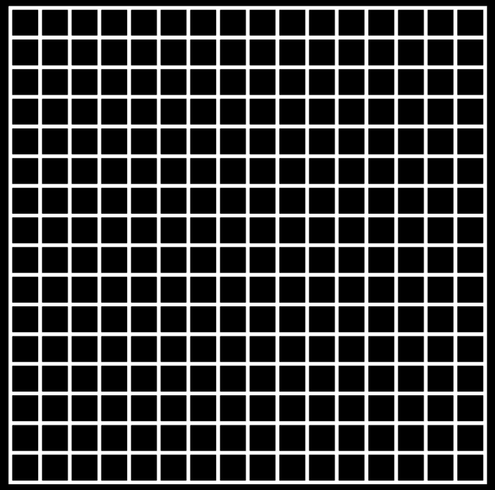
coordinates $\rightarrow$

$\in \mathbb{R}^2, \mathbb{R}^3, \ldots$

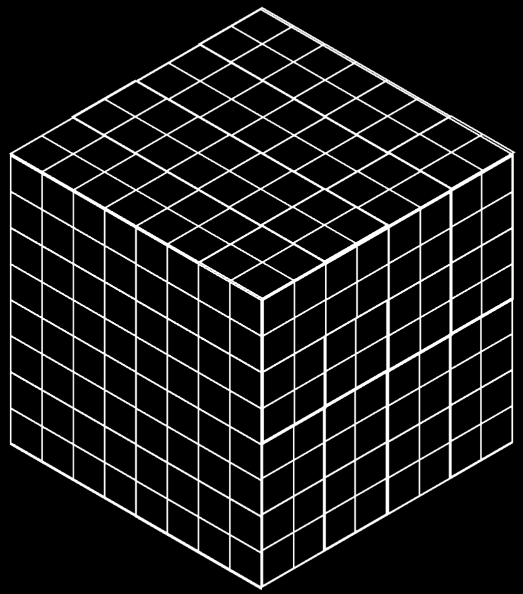


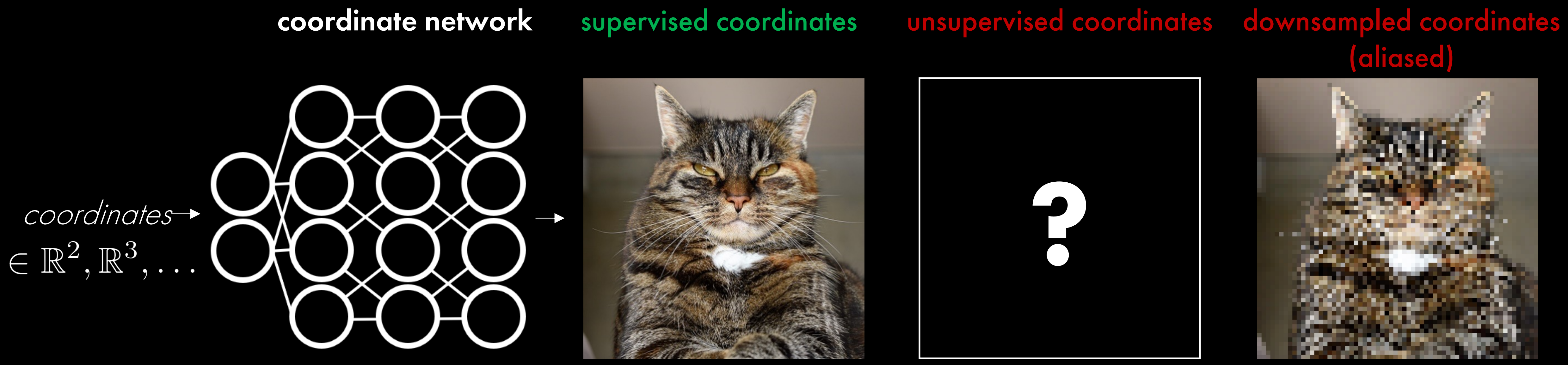
$\rightarrow$

array



voxel grid

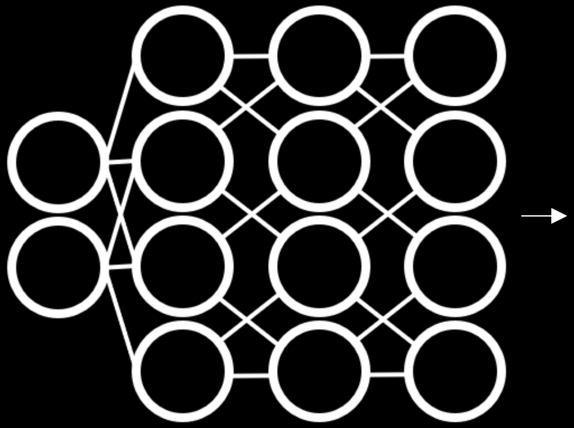



coordinate network
supervised coordinates
unsupervised coordinates
downsampled coordinates
(aliased)
coordinates
$\in \mathbb{R}^2, \mathbb{R}^3, \ldots$
?

Band-Limited Coordinate Network (BACON)

coordinates $\rightarrow$

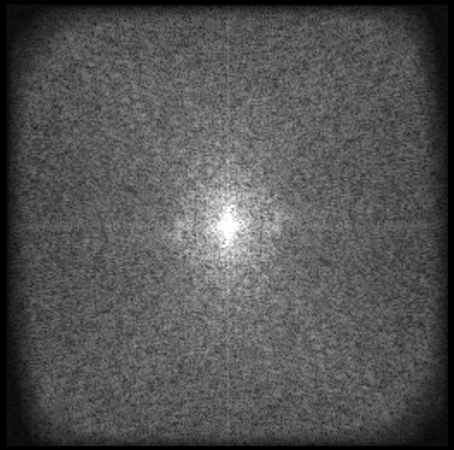
$\in \mathbb{R}^2, \mathbb{R}^3, \ldots$



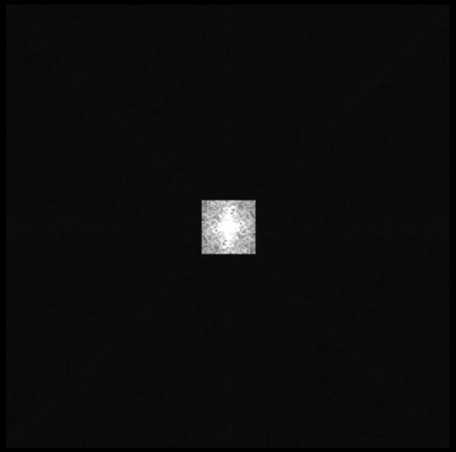
supervised coordinates

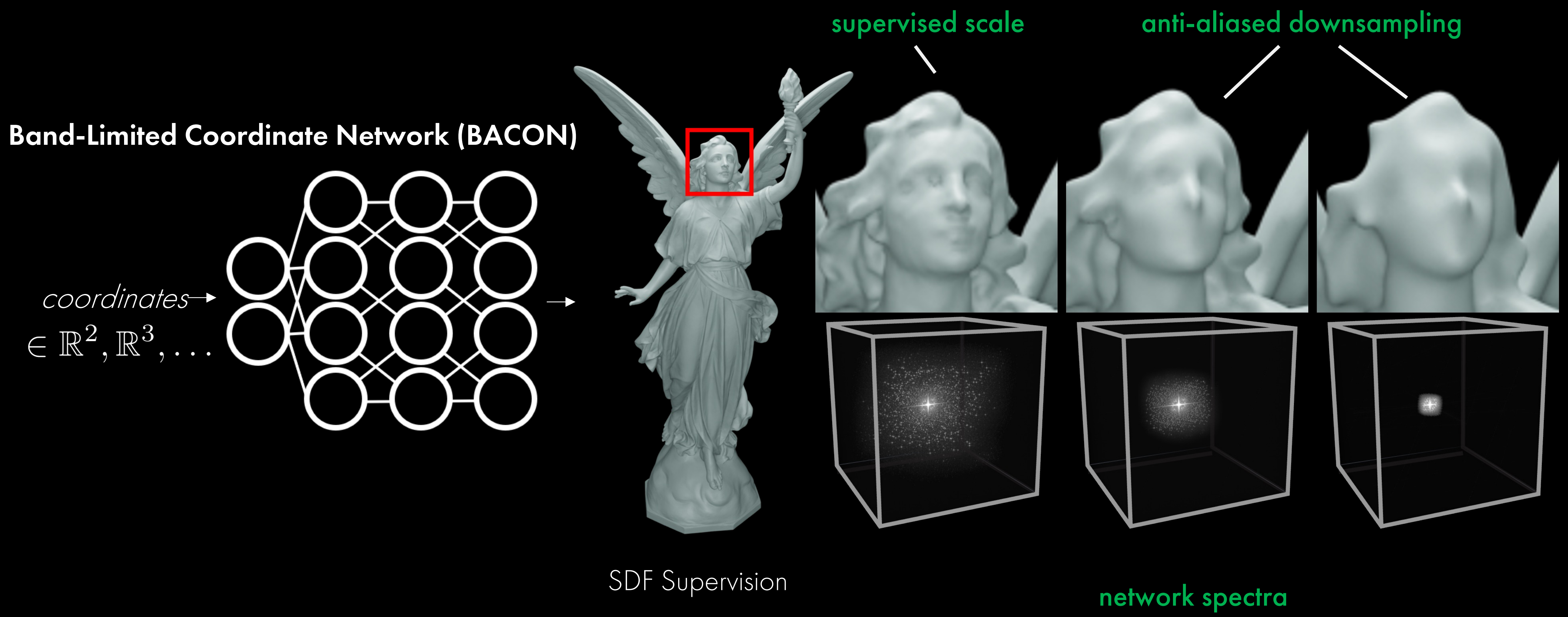
anti-aliased downsampling

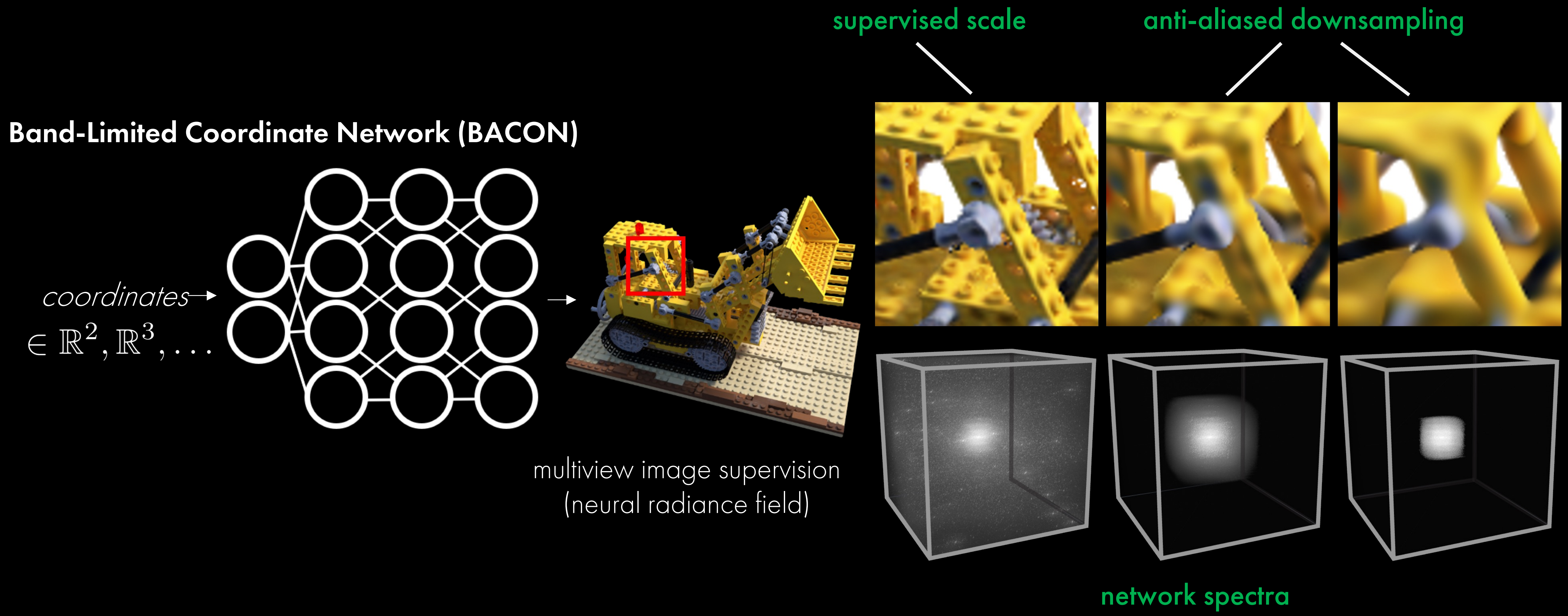
network spectrum



network spectrum

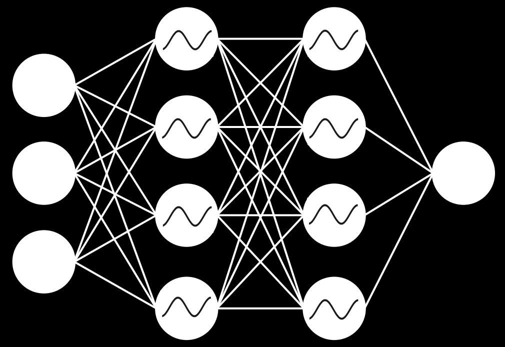



Band-Limited Coordinate Network (BACON)
coordinates
$\in \mathbb{R}^2, \mathbb{R}^3, \ldots$
SDF Supervision
supervised scale
anti-aliased downsampling
network spectra



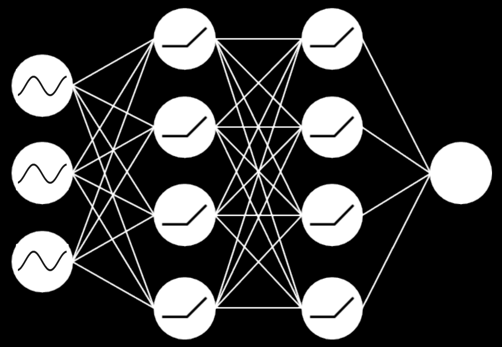
supervised scale
anti-aliased downsampling
Band-Limited Coordinate Network (BACON)
coordinates
$\in \mathbb{R}^2, \mathbb{R}^3, \ldots$
multiview image supervision
(neural radiance field)
network spectra

Sinusoidal Rep. Networks (SIREN) [Sitzmann et al. '20]



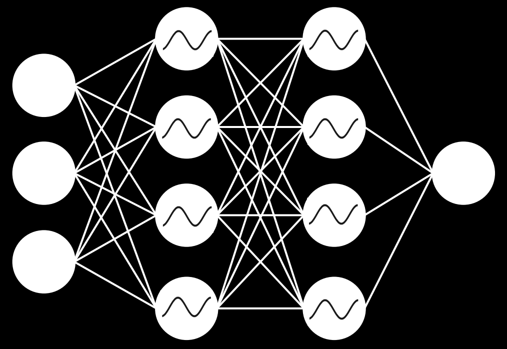
- single scale

Fourier Features [Tancik et al. '20]



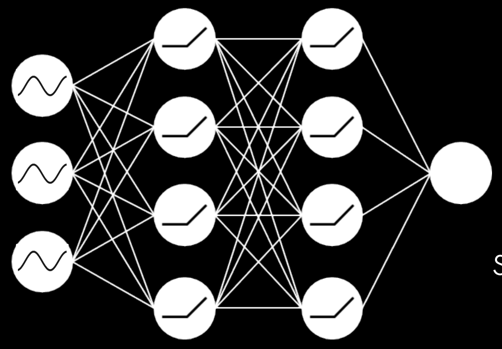
- single scale

Sinusoidal Rep. Networks (SIREN) [Sitzmann et al. '20]



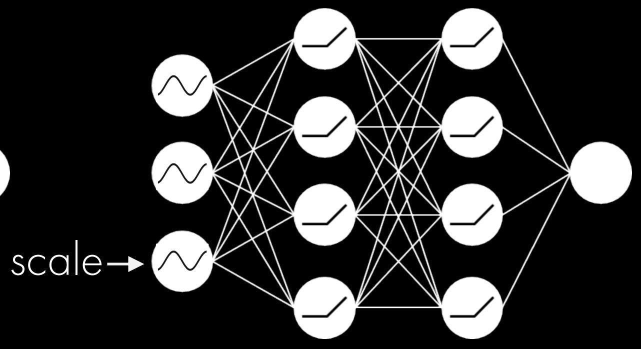
- single scale

Fourier Features [Tancik et al. '20]



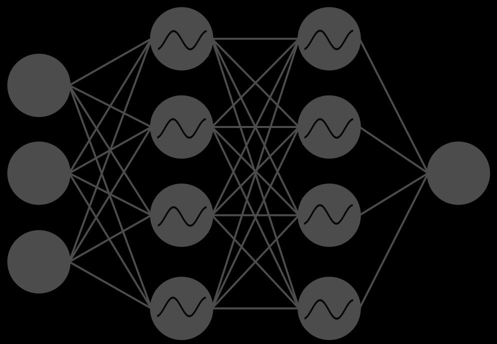
- single scale

Mip-NeRF [Barron et al. '21]



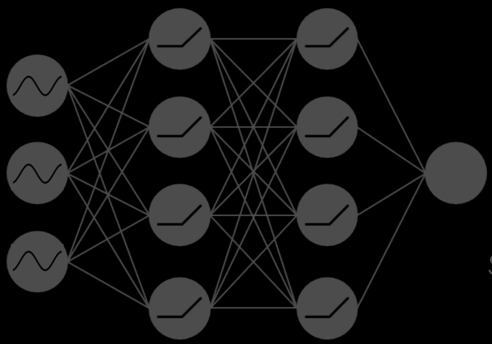
- multiscale
- multiscale supervision

Sinusoidal Rep. Networks
(SIREN) [Sitzmann et al. '20]



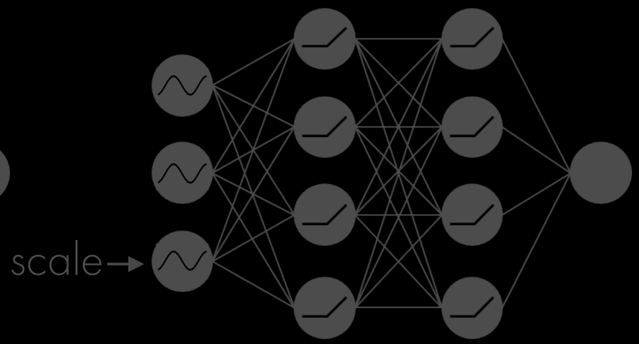
- single scale

Fourier Features
[Tancik et al. '20]



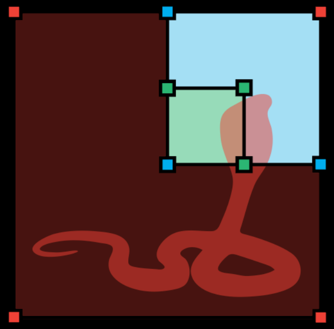
- single scale

Mip-NeRF
[Barron et al. '21]

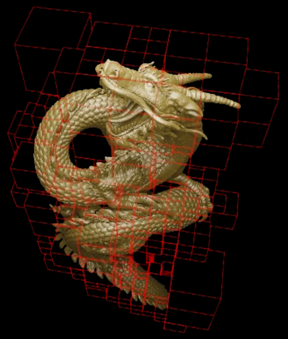


- multiscale
- multiscale supervision

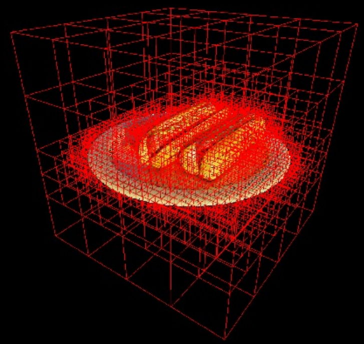
Neural Geometric LOD
[Takikawa et al. '21]



Adaptive Coordinate Networks
[Martel et al. '21]

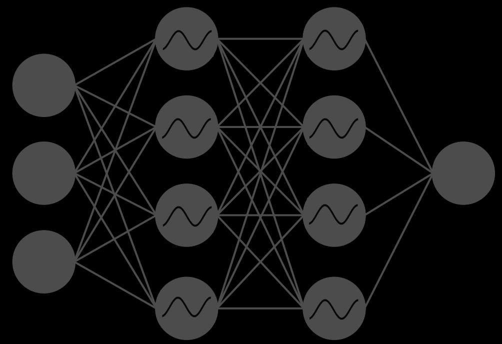


PlenOctrees
[Yu et al. '21]



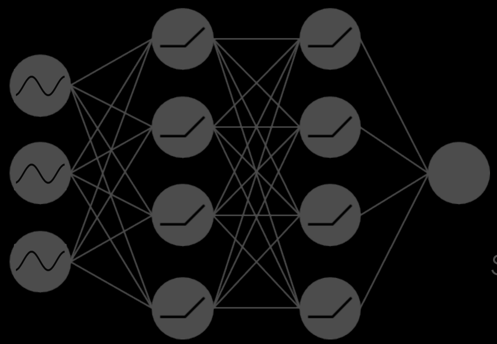
- black box behavior
- multiresolution outputs not bandlimited

Sinusoidal Rep. Networks
(SIREN) [Sitzmann et al. '20]



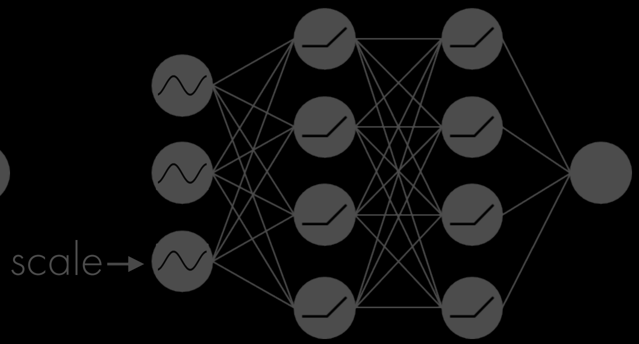
single scale

Fourier Features
[Tancik et al. '20]



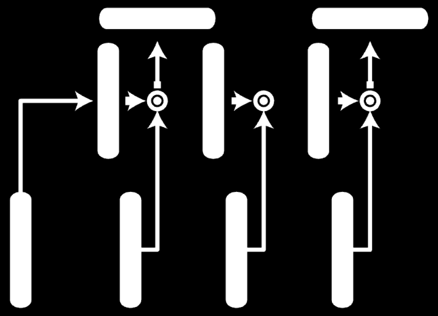
single scale

Mip-NeRF
[Barron et al. '21]



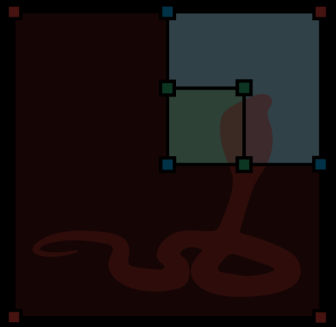
- multiscale
- multiscale supervision

BACON
(proposed)

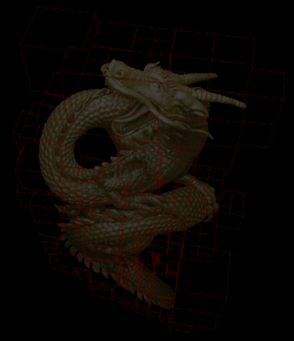


- multiscale
- single-scale supervision

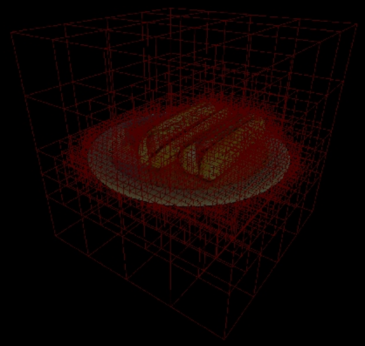
Neural Geometric LOD
[Takikawa et al. '21]



Adaptive Coordinate Networks
[Martel et al. '21]

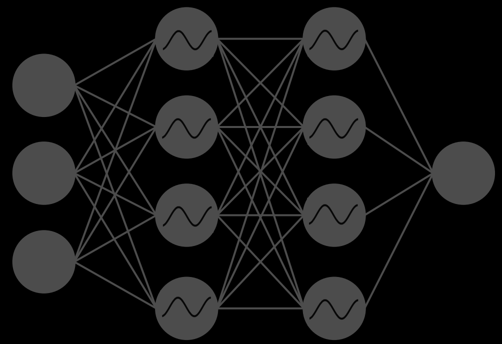


PlenOctrees
[Yu et al. '21]



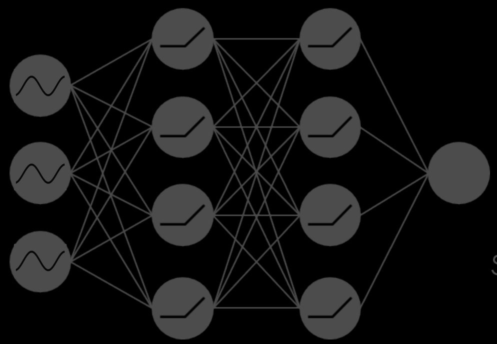
- black box behavior
- multiresolution outputs not bandlimited

Sinusoidal Rep. Networks
(SIREN) [Sitzmann et al. ‘20]



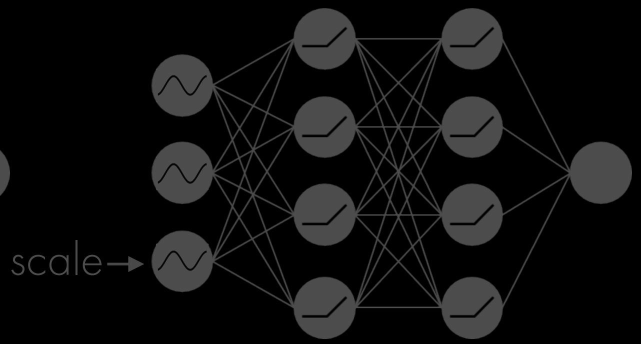
single scale

Fourier Features
[Tancik et al. ‘20]



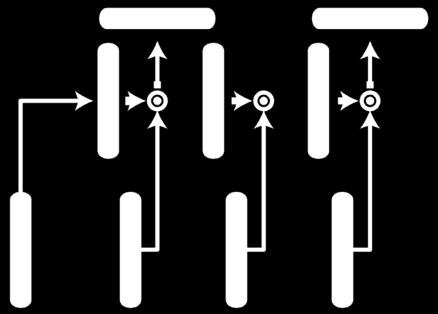
single scale

Mip-NeRF
[Barron et al. ‘21]



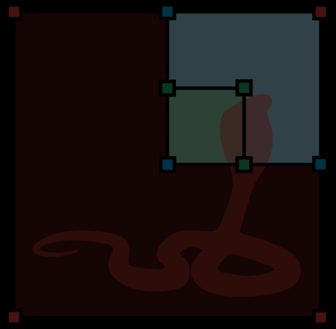
- multiscale
- multiscale supervision

BACON
(proposed)

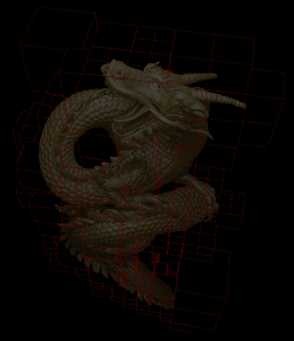


- multiscale
- single-scale supervision

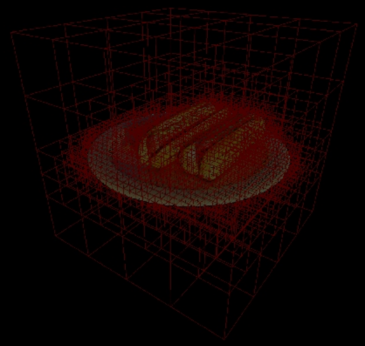
Neural Geometric LOD
[Takikawa et al. ‘21]



Adaptive Coordinate Networks
[Martel et al. ‘21]



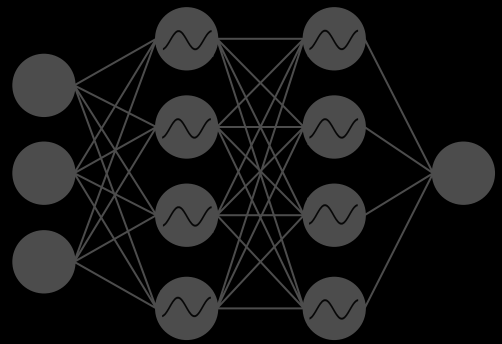
PlenOctrees
[Yu et al. ‘21]



Multiplicative Filter Networks
[Fathony et al. ‘21]

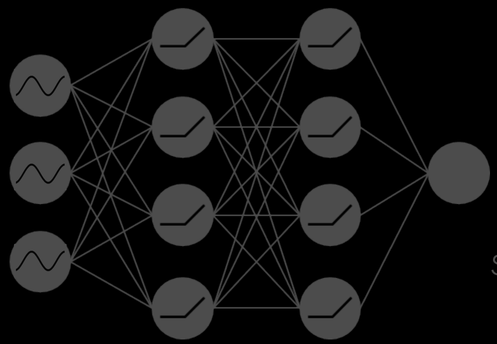
- black box behavior
- multiresolution outputs not bandlimited

Sinusoidal Rep. Networks
(SIREN) [Sitzmann et al. '20]



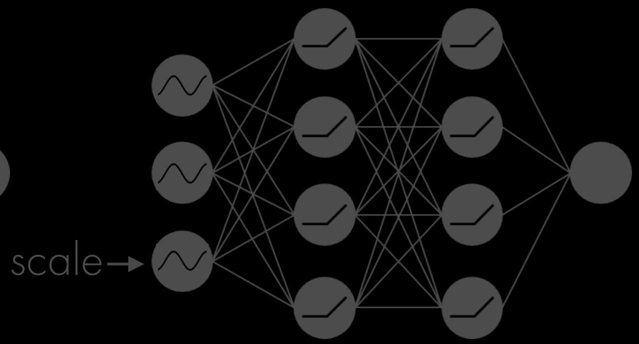
single scale

Fourier Features
[Tancik et al. '20]



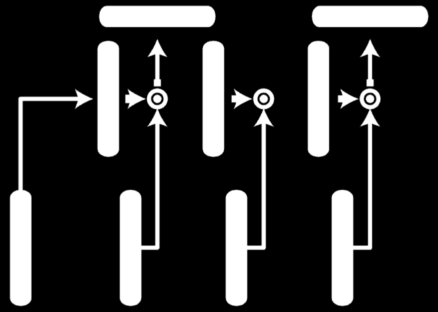
single scale

Mip-NeRF
[Barron et al. '21]



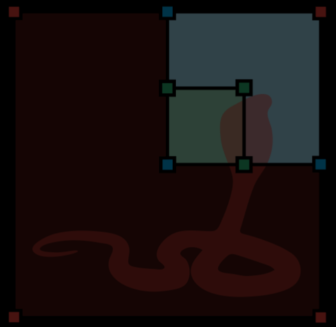
- multiscale
- multiscale supervision

BACON
(proposed)

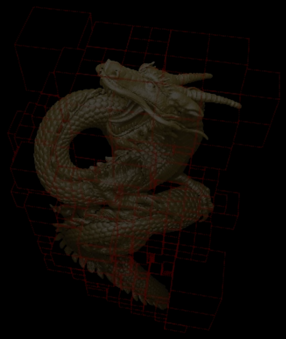


- multiscale
- single-scale supervision

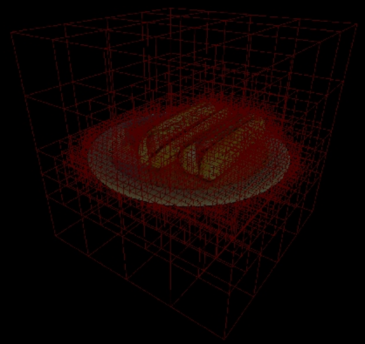
Neural Geometric LOD
[Takikawa et al. '21]



Adaptive Coordinate Networks
[Martel et al. '21]



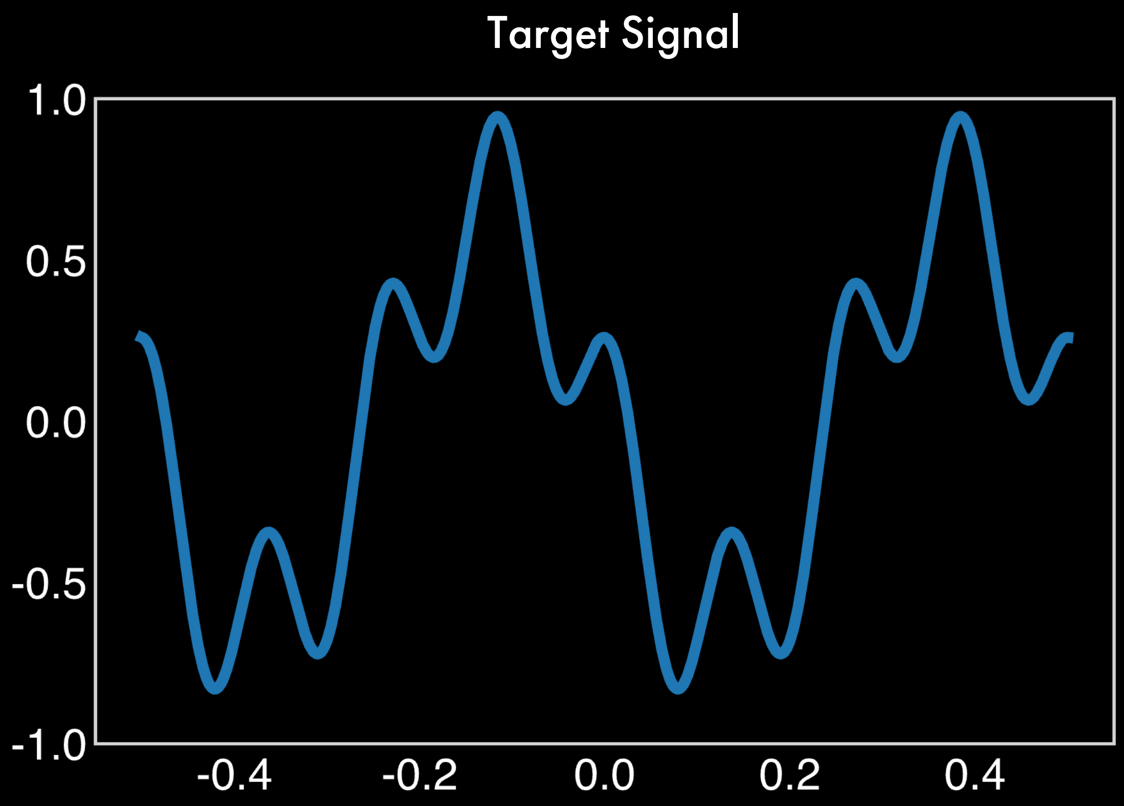
PlenOctrees
[Yu et al. '21]

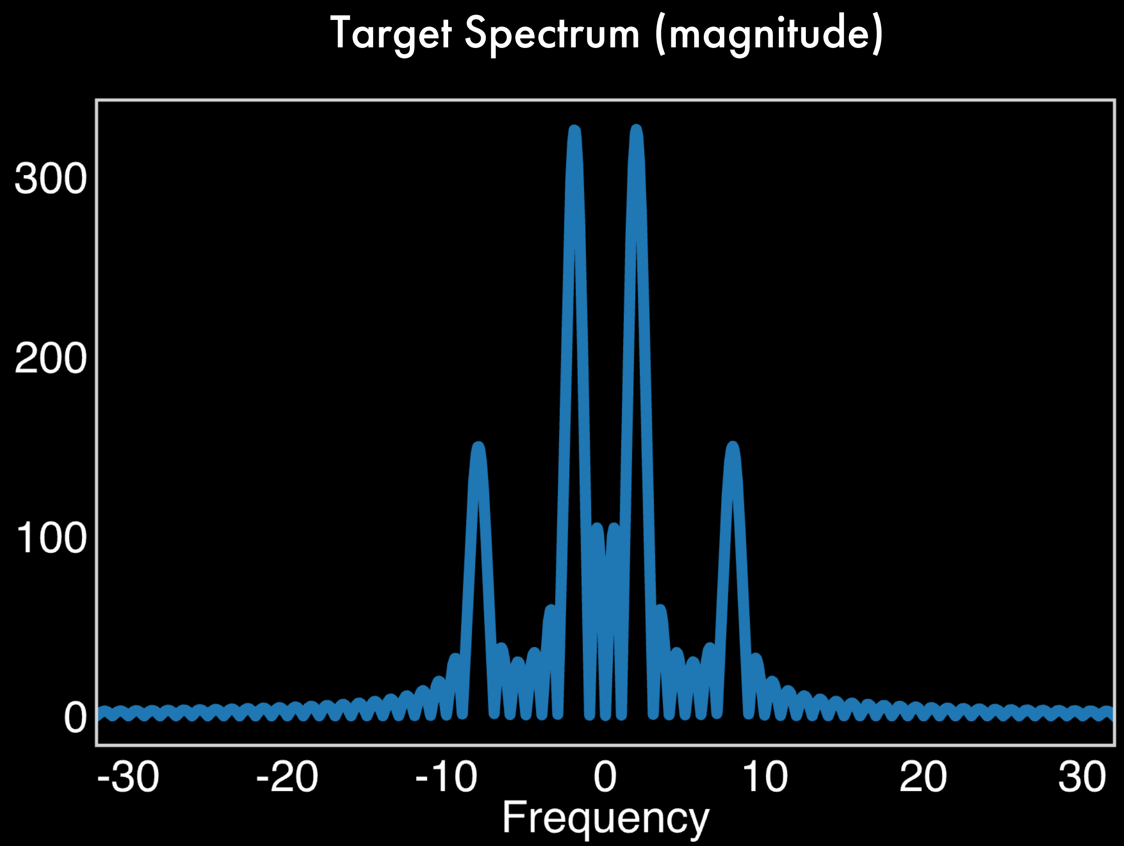


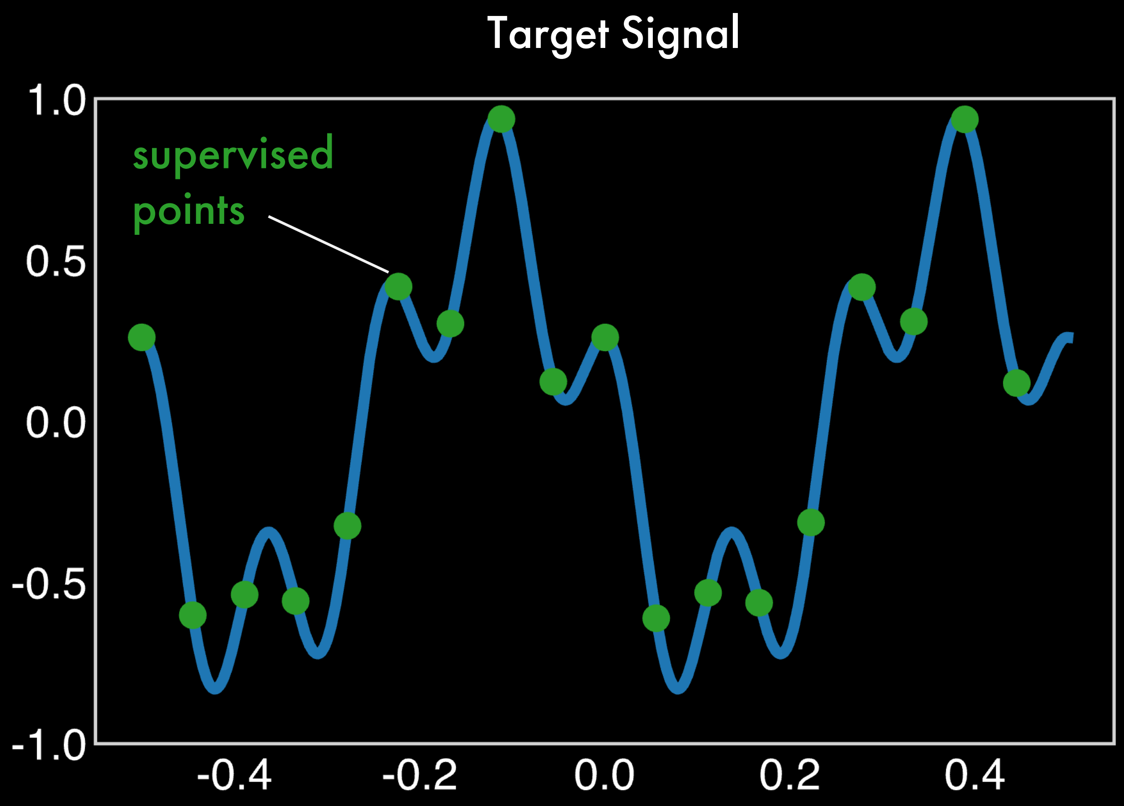
Multiplicative Filter Networks
[Fathony et al. '21]

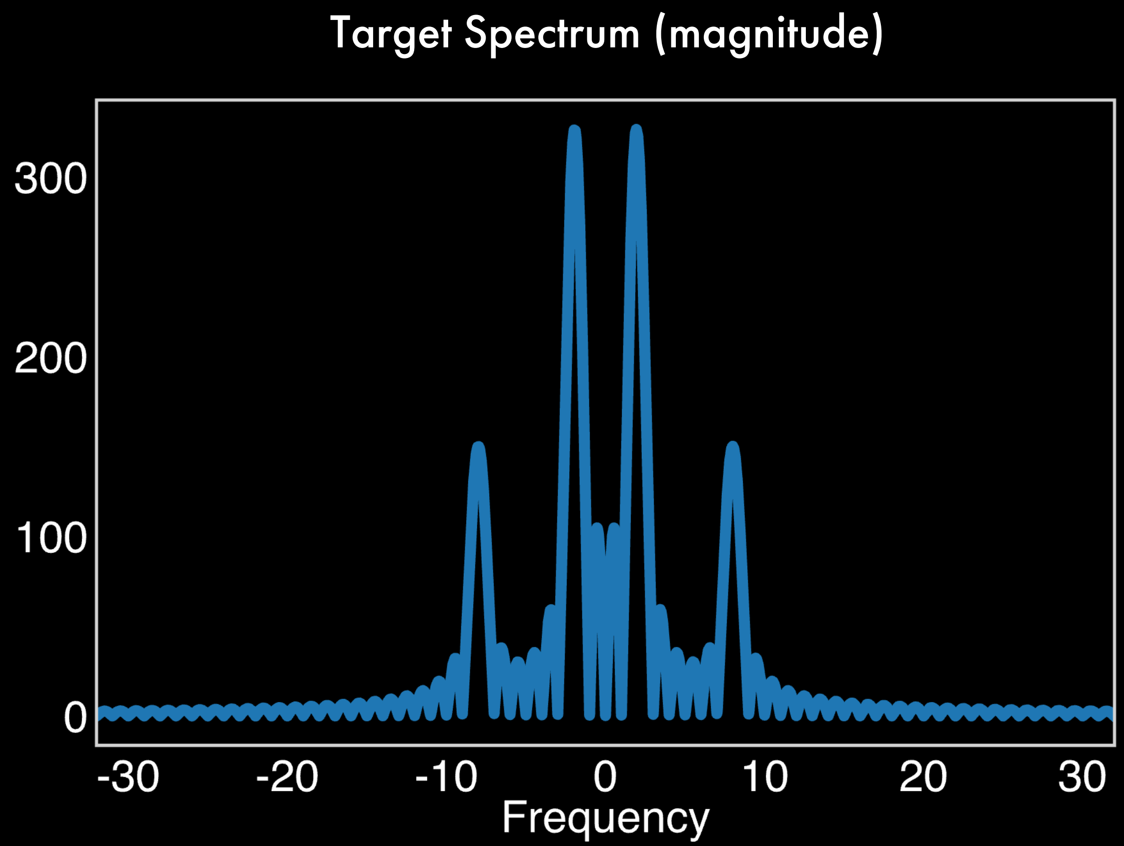
- analytical Fourier spectra
- adjustable bandwidth
- initialization for deep networks

- black box behavior
- multiresolution outputs not bandlimited

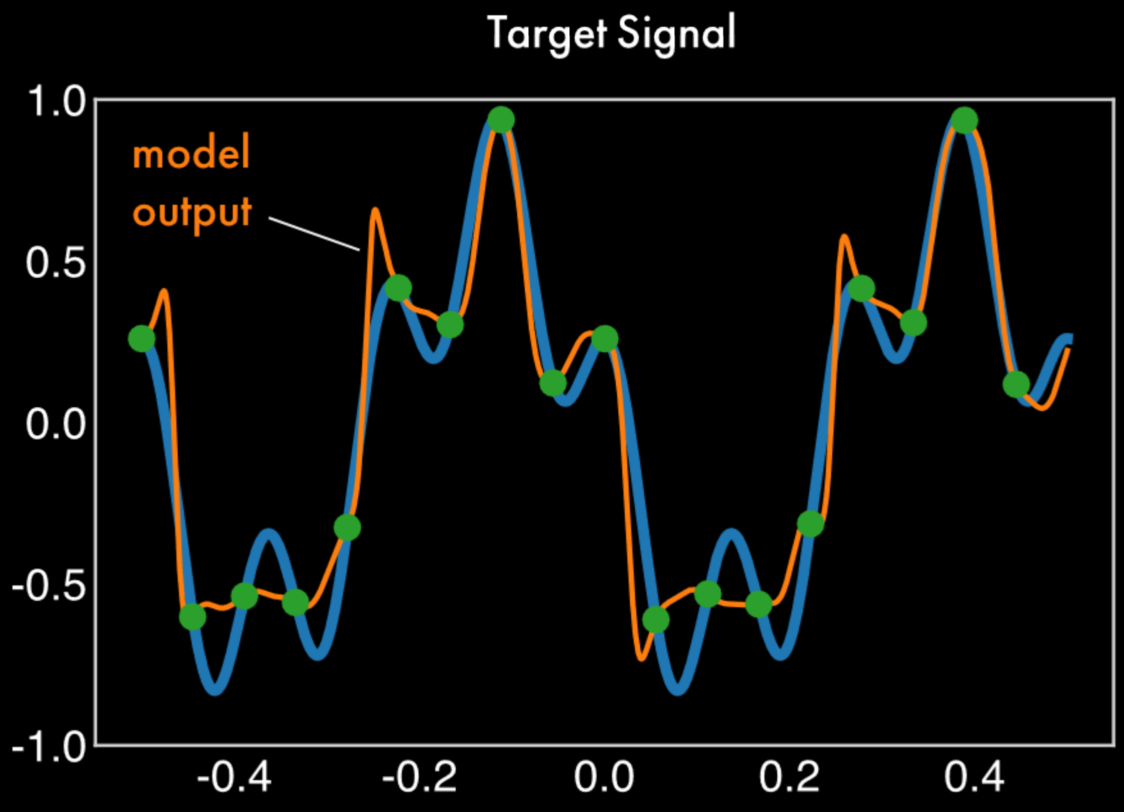

Target Signal
1.0
0.5
0.0
-0.5
-1.0
-0.4
-0.2
0.0
0.2
0.4

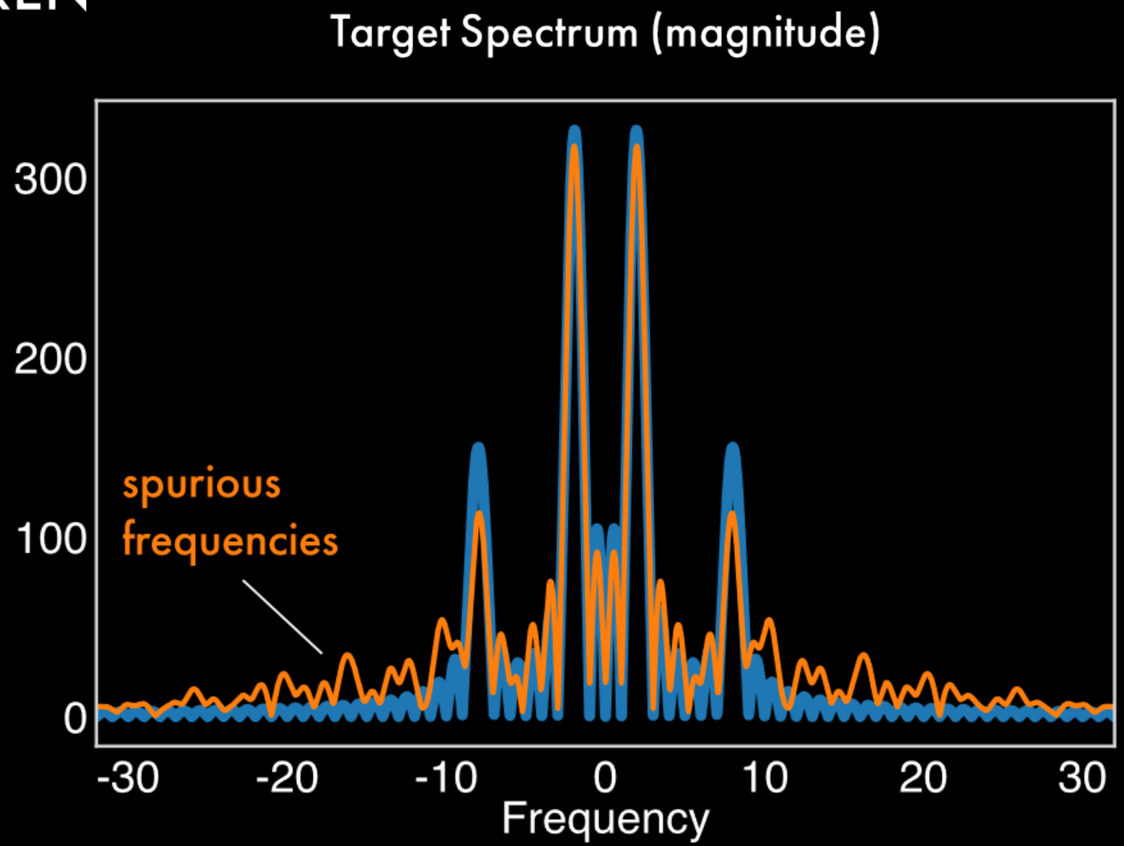

Target Spectrum (magnitude)
300
200
100
0
-30
-20
-10
0
10
20
30
Frequency


Target Signal
supervised
points
1.0
0.5
0.0
-0.5
-1.0
-0.4
-0.2
0.0
0.2
0.4

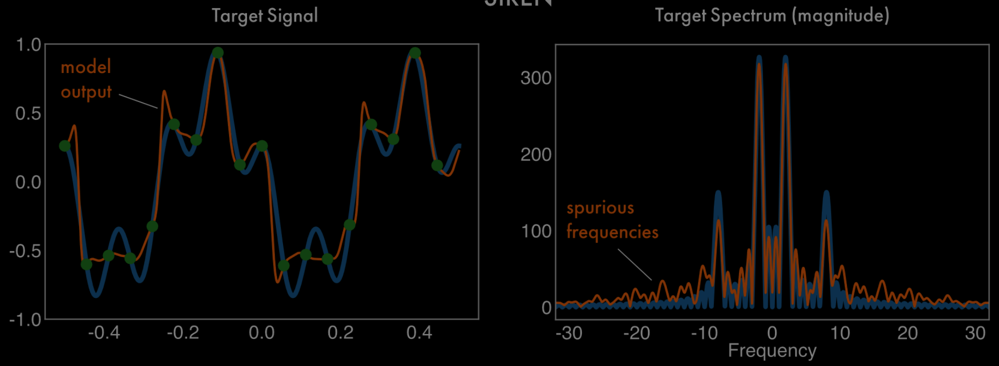

Target Spectrum (magnitude)
300
200
100
0
-30
-20
-10
0
10
20
30
Frequency

SIREN


Target Signal
model output
1.0
0.5
0.0
-0.5
-1.0
-0.4
-0.2
0.0
0.2
0.4

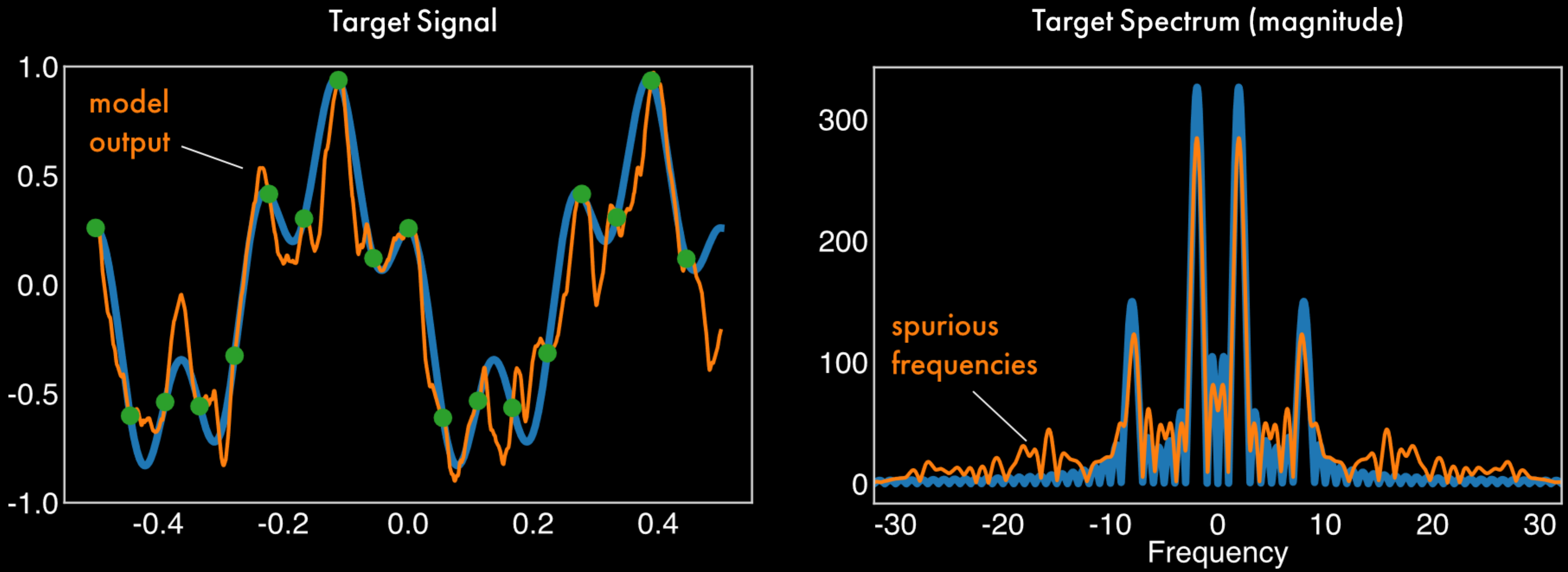

Target Spectrum (magnitude)
spurious frequencies
300
200
100
0
-30
-20
-10
0
10
20
30
Frequency

SIREN

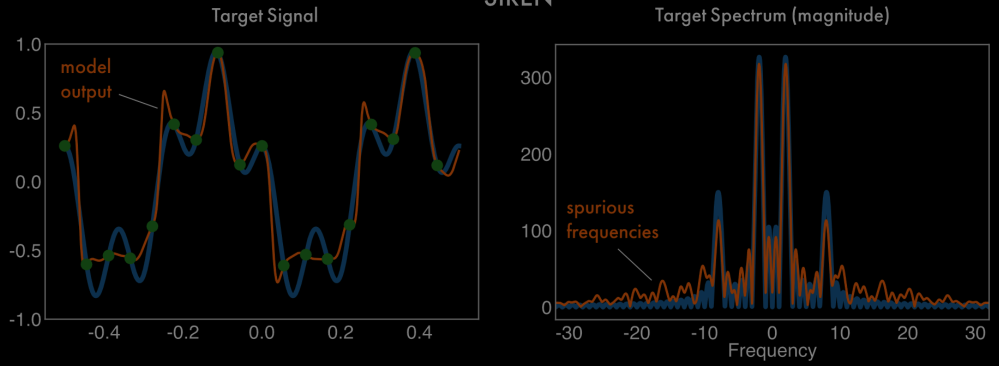


Target Signal
model
output
1.0
0.5
0.0
-0.5
-1.0
-0.4
-0.2
0.0
0.2
0.4
Target Spectrum (magnitude)
spurious
frequencies
300
200
100
0
-30
-20
-10
0
10
20
30
Frequency

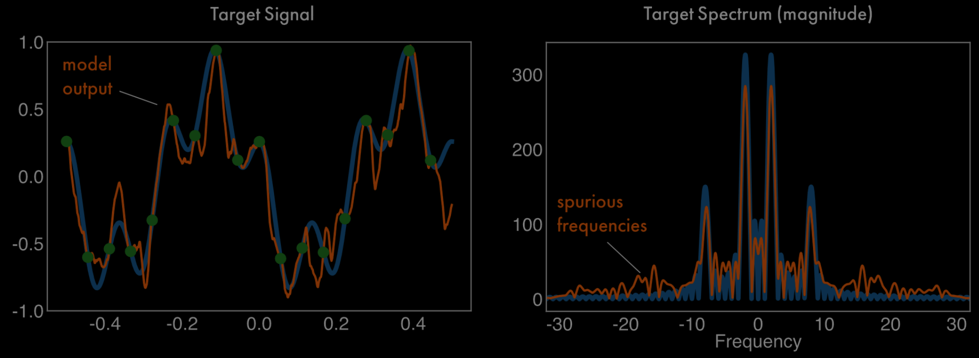
Fourier Features



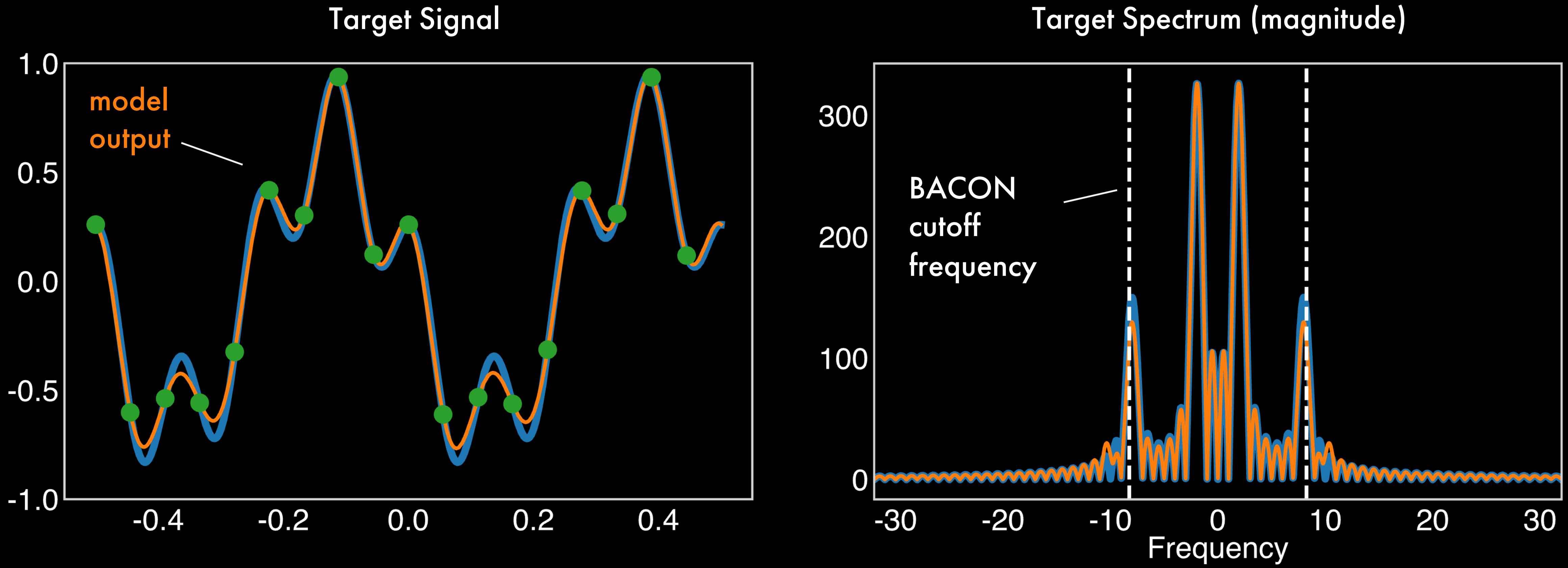
SIREN



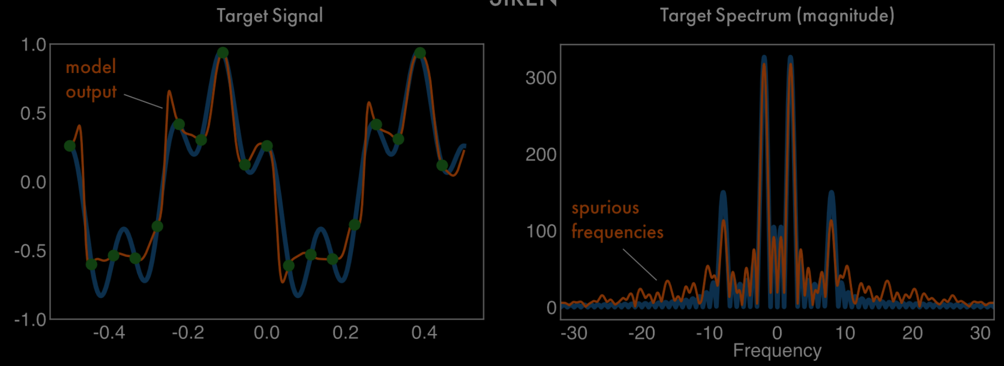
Fourier Features



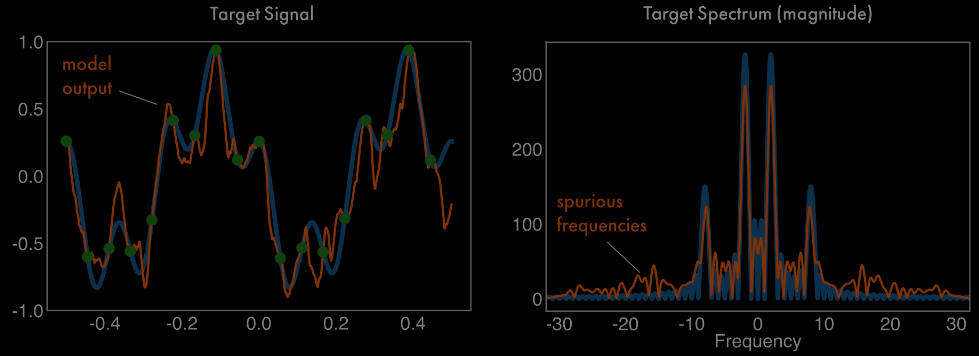
BACON



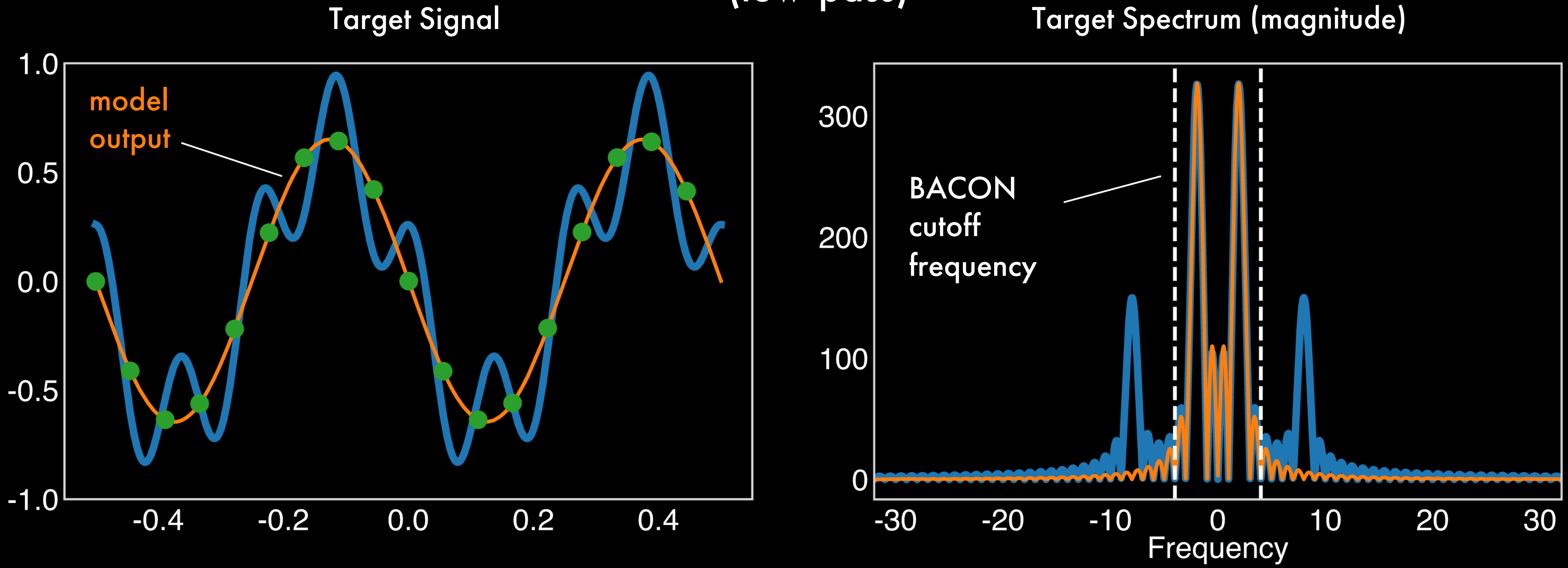
SIREN

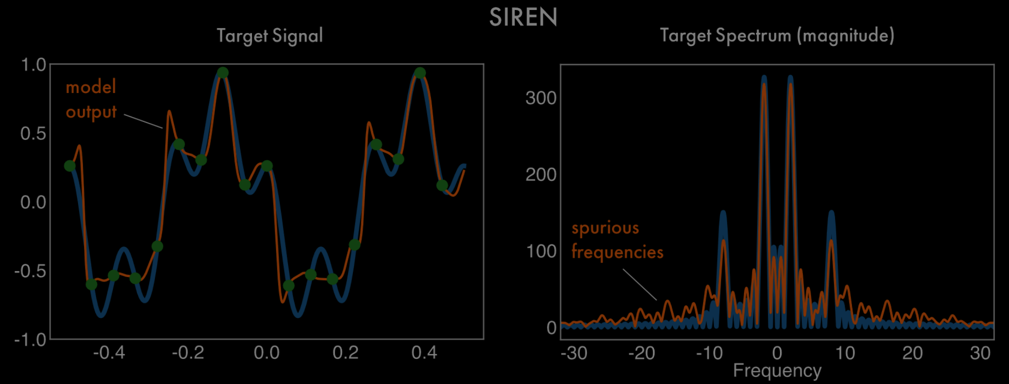


Fourier Features

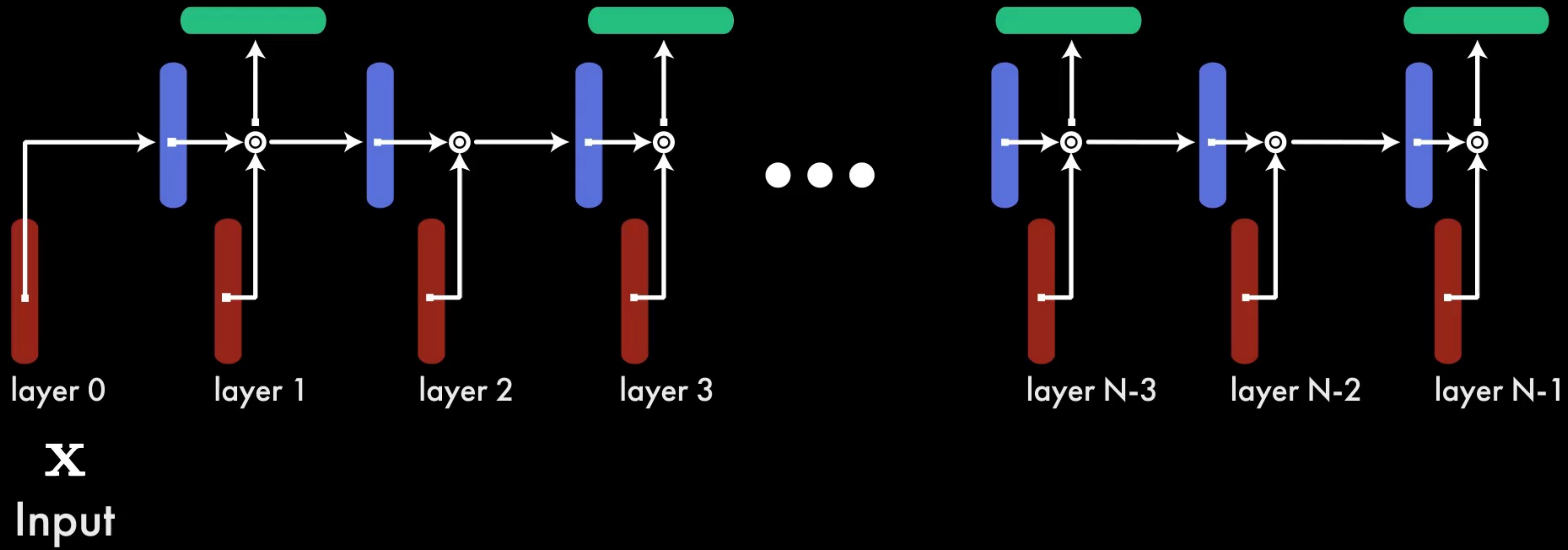


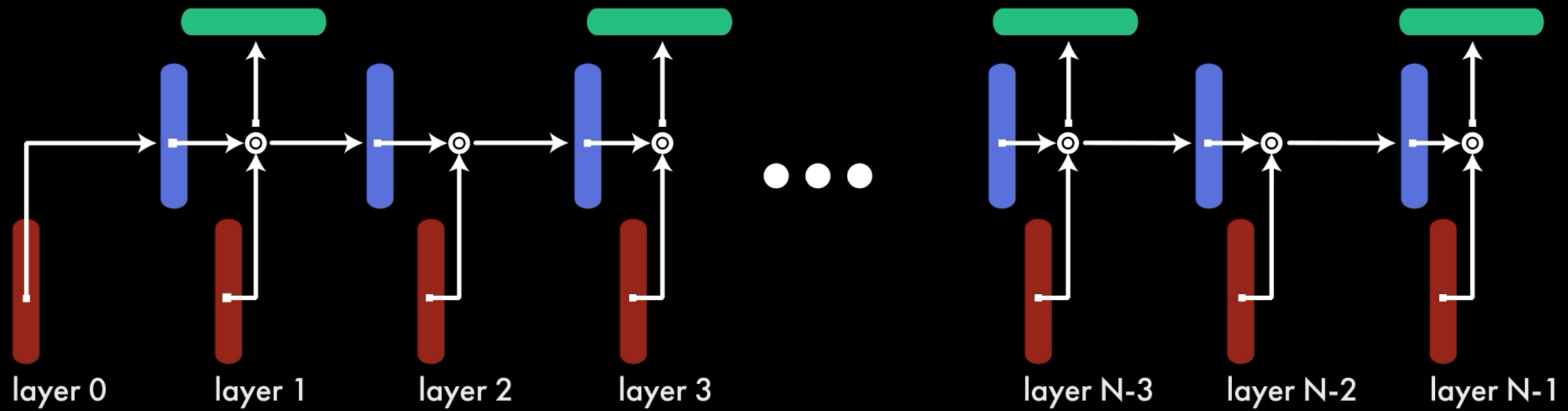
BACON
(low-pass)



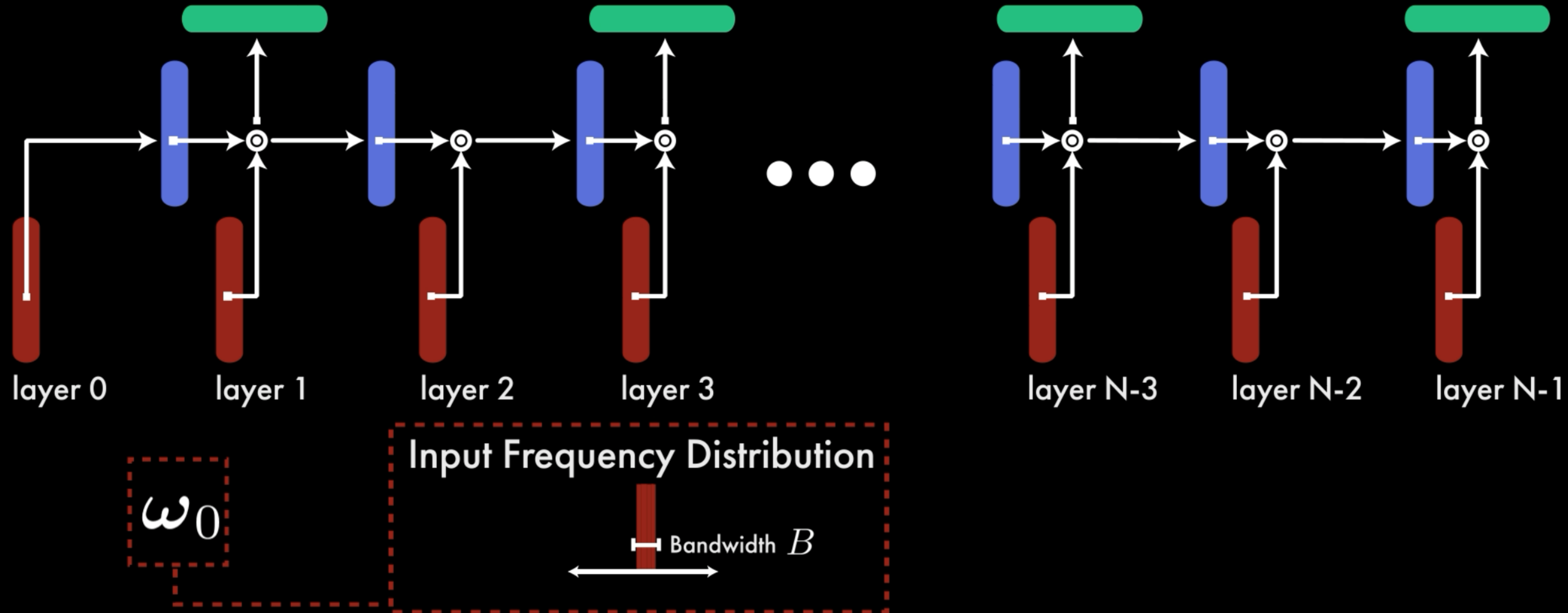


Architecture

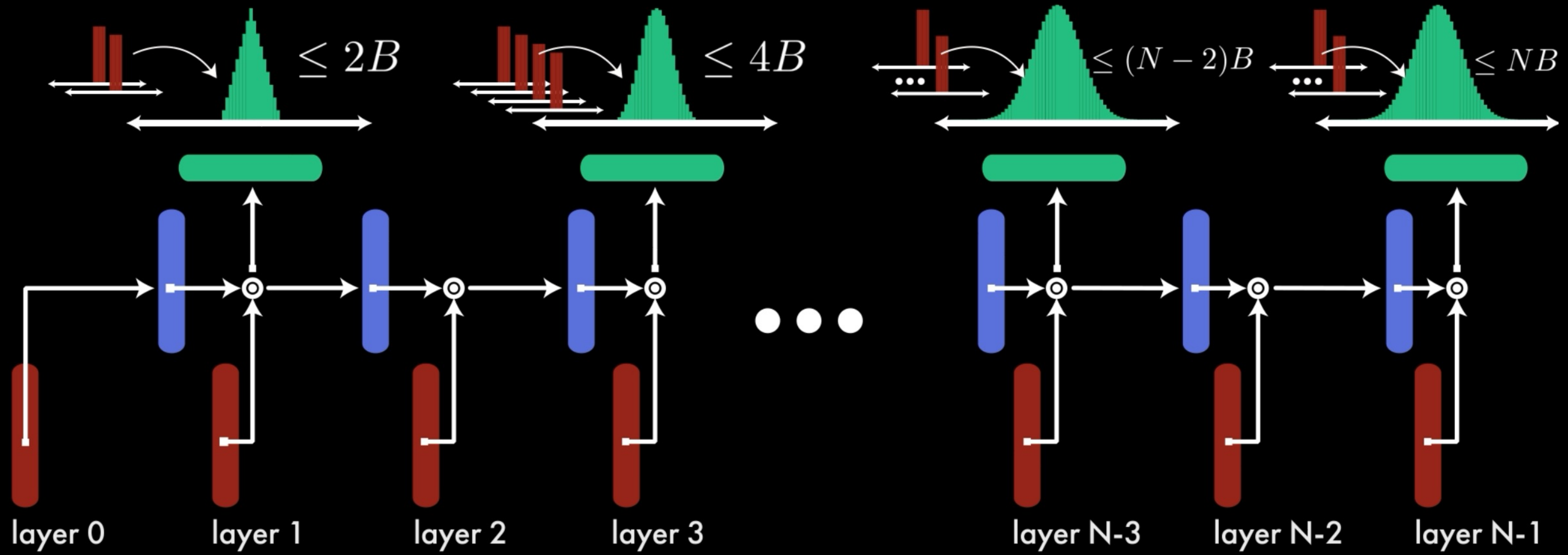

layer 0
layer 1
layer 2
layer 3
layer N-3
layer N-2
layer N-1
x
Input

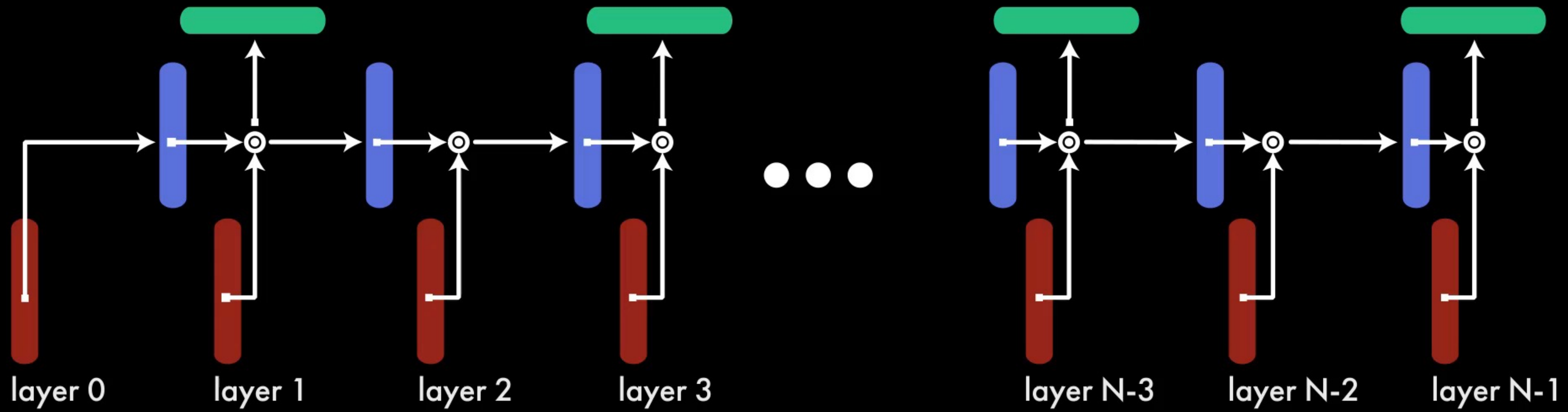


$$\sin(\boldsymbol{\omega}_0 \mathbf{x} + \boldsymbol{\phi}_0)$$


layer 0
layer 1
layer 2
layer 3
layer N-3
layer N-2
layer N-1
ω_0
Input Frequency Distribution
Bandwidth B

Output Bandwidth

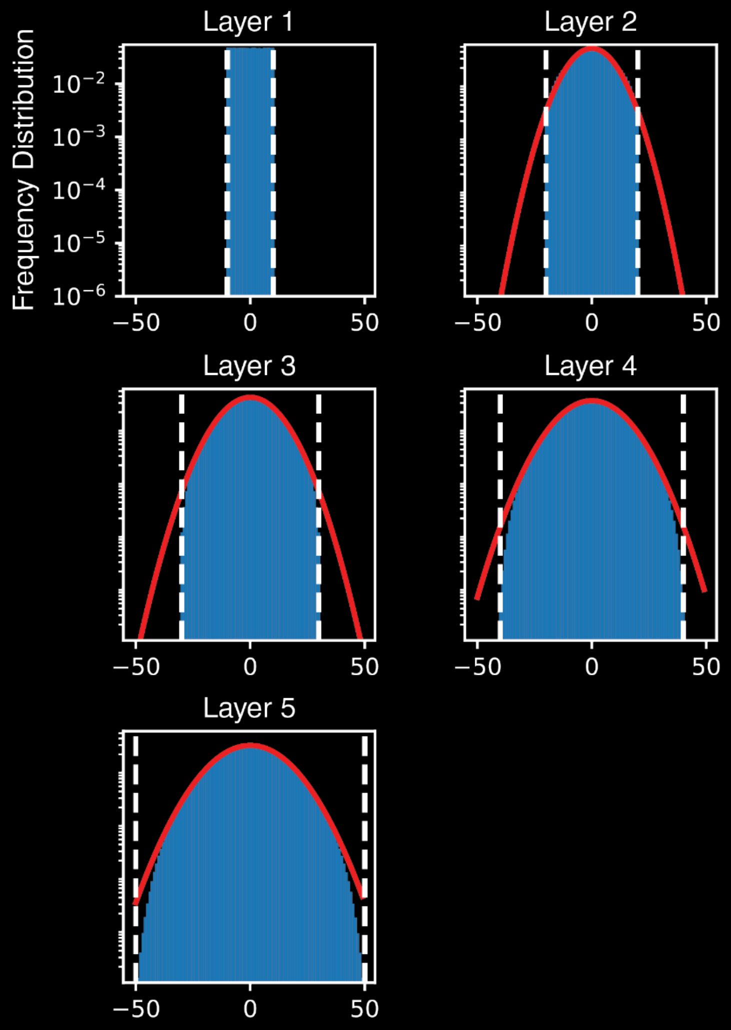



layer 0
layer 1
layer 2
layer 3
layer N-3
layer N-2
layer N-1

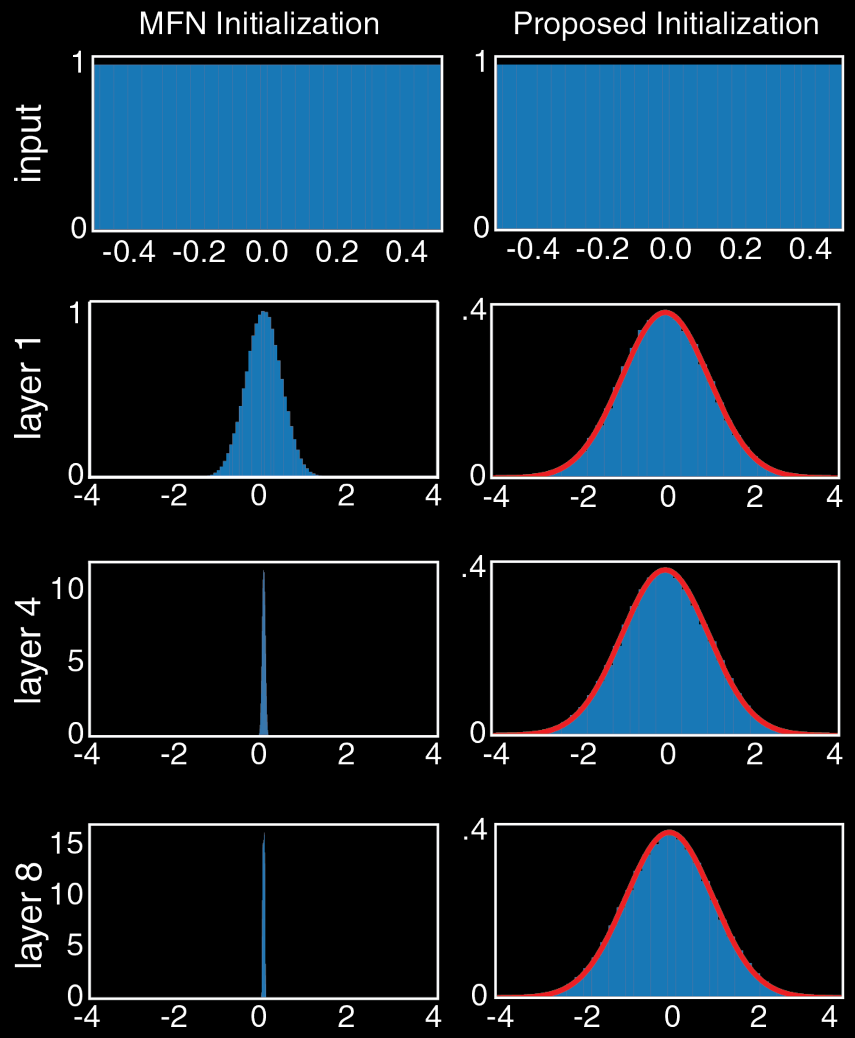
Parameterized Sines

$$N_{\mathrm{sine}}^{(N_{\mathrm{L}})} = \sum_{i=0}^{N_{\mathrm{L}}-1} 2^i d_{\mathrm{h}}^{i+1}$$

Distribution of Frequencies

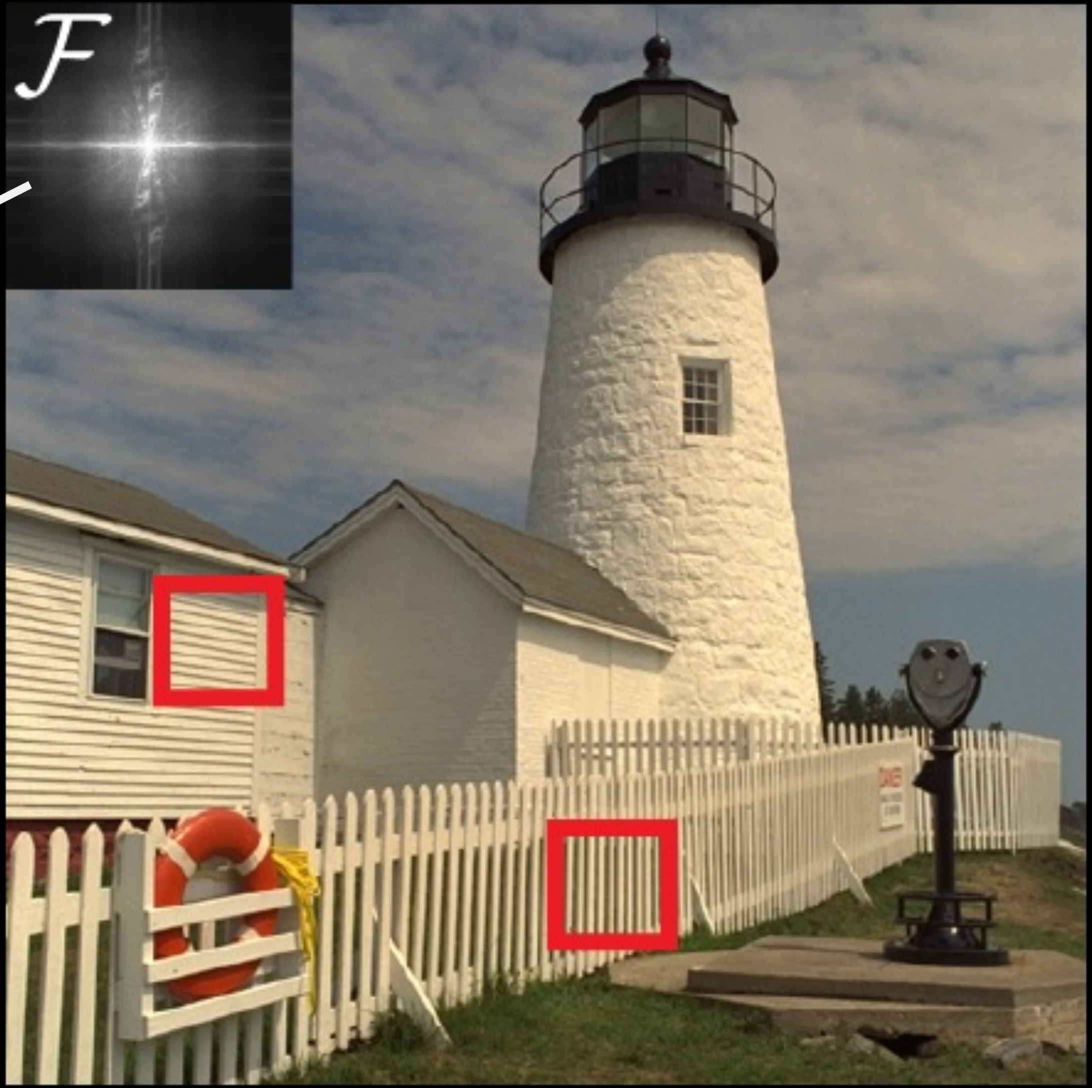


Initialization Scheme

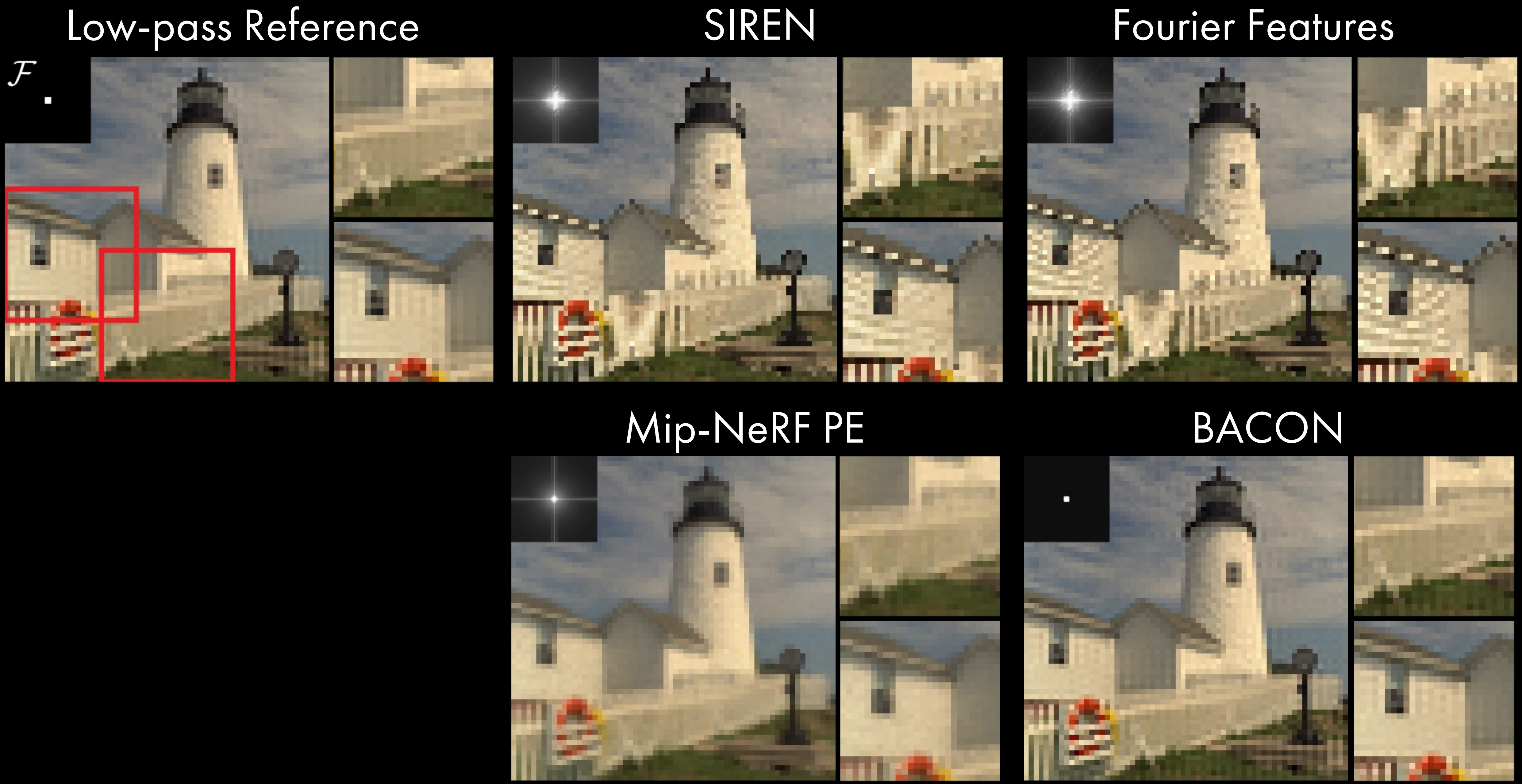


Results

Fourier
Spectrum



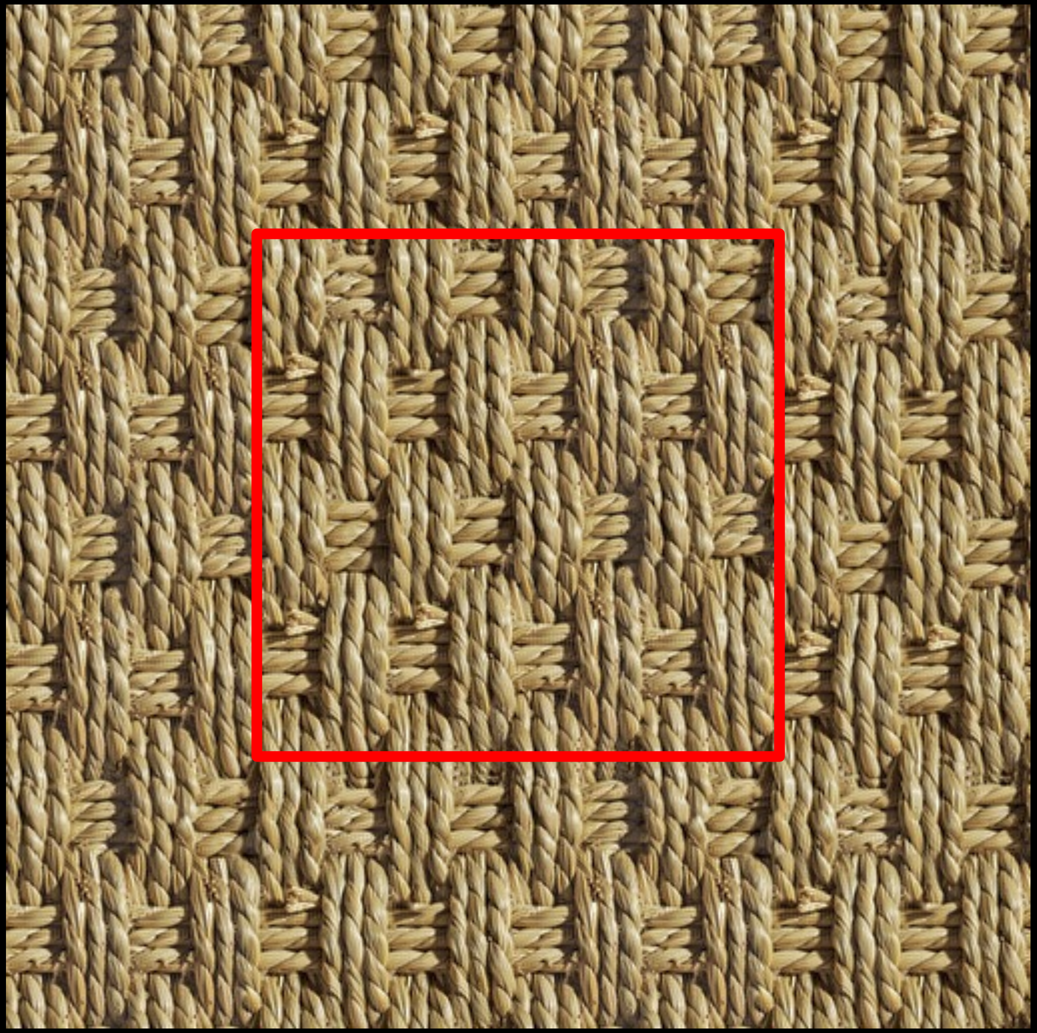
4x Downsampling



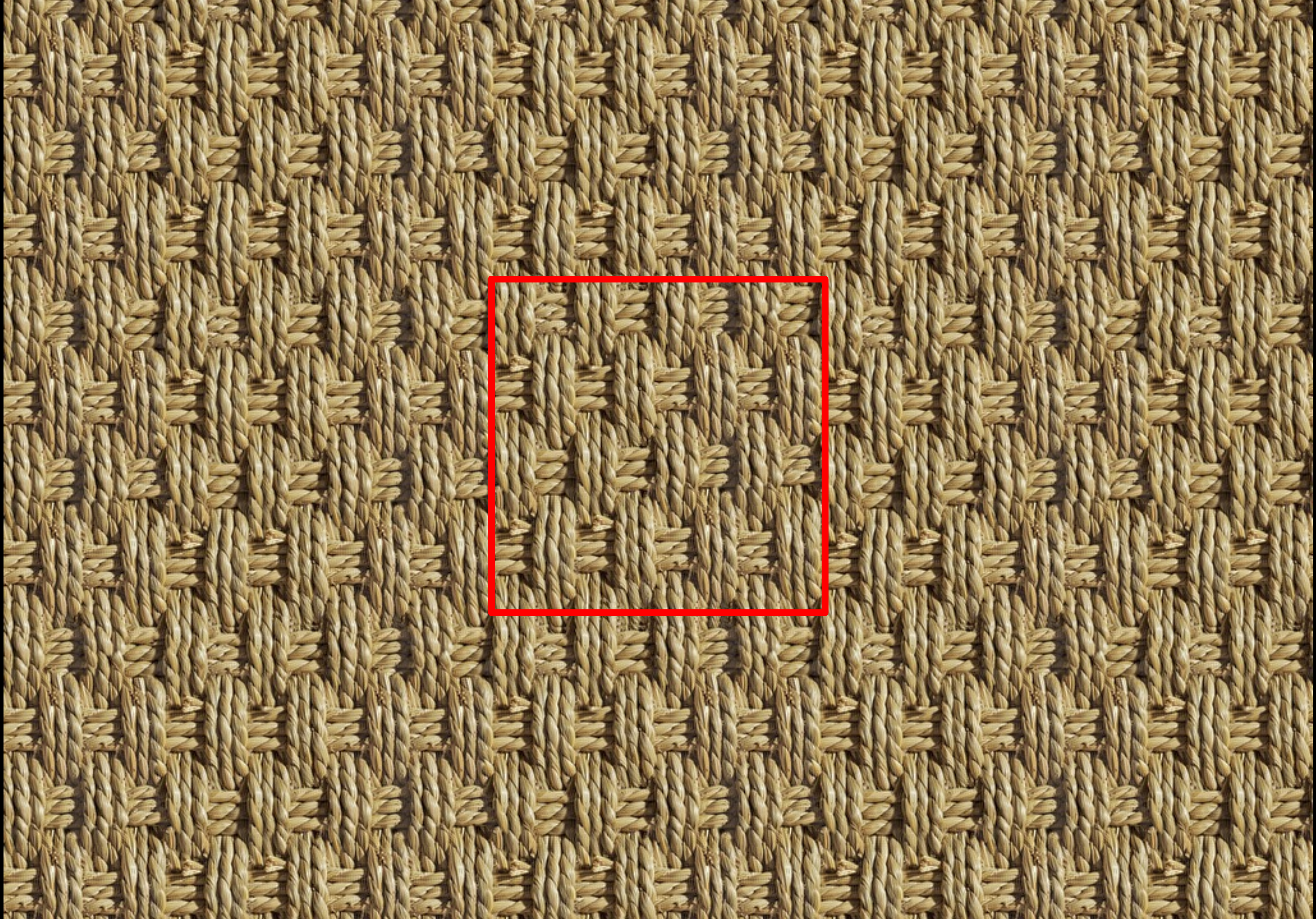
4x Upsampling

supervised region

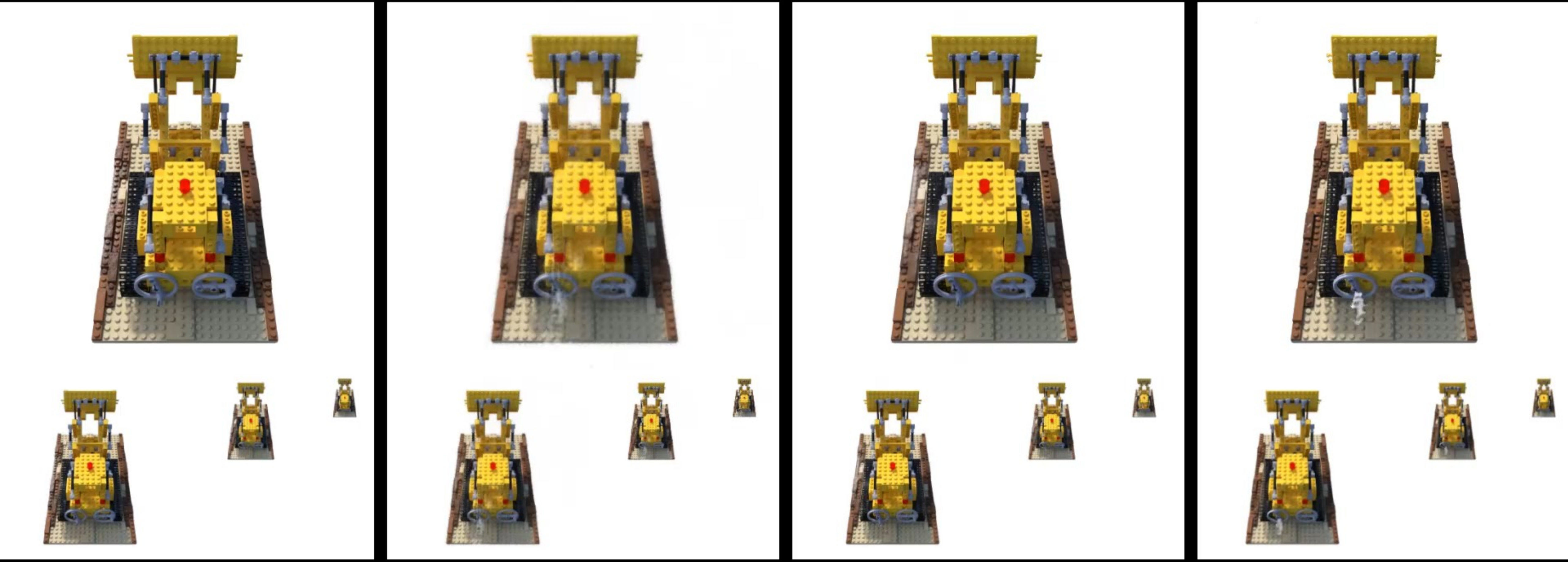
supervised region

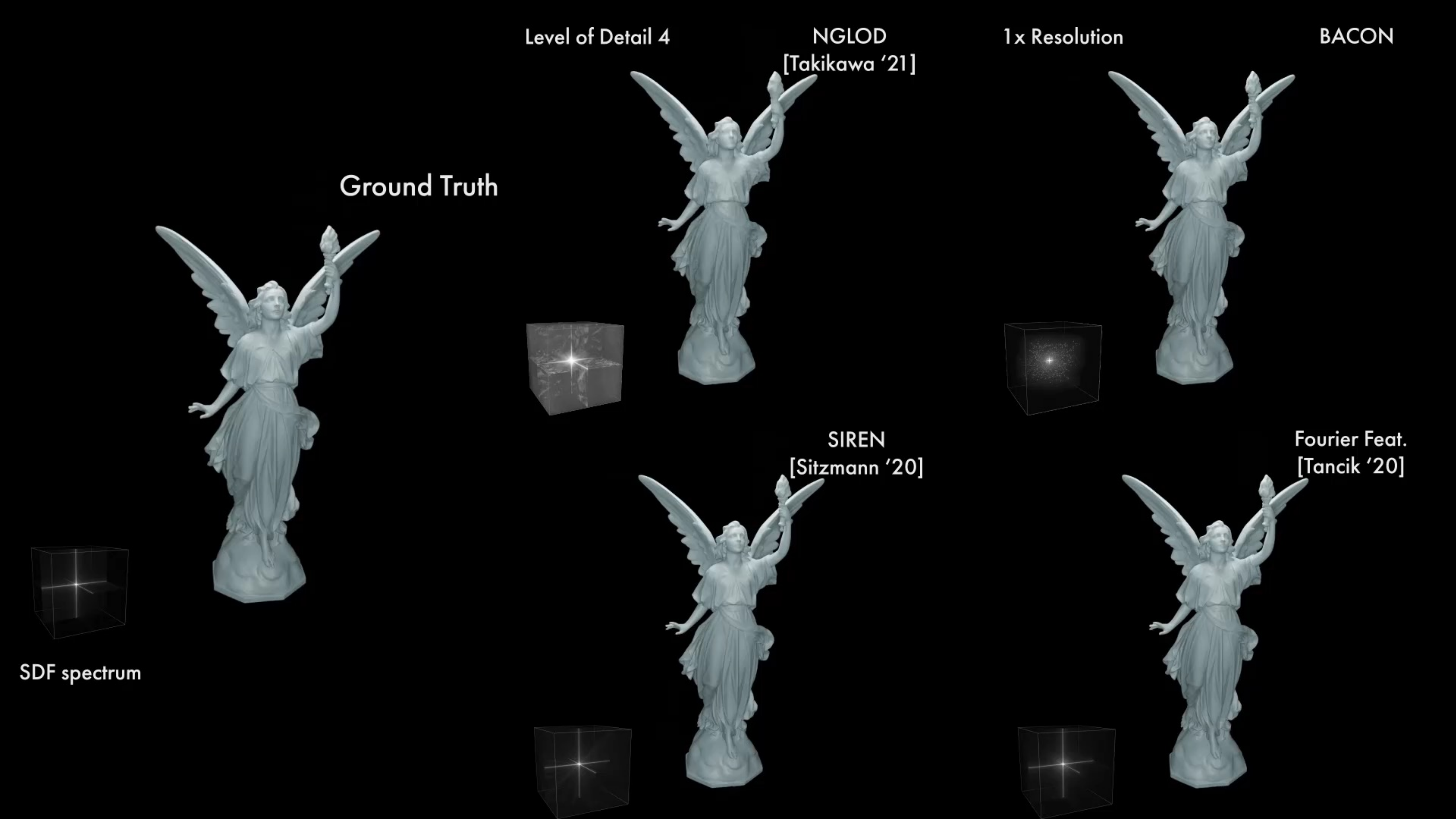


-1.0 1.0



Ground Truth	NeRF [Mildenhall et al. '20]	Mip-NeRF [Barron et al. '21]	BACON




Level of Detail 4
NGLOD
[Takikawa '21]
1x Resolution
BACON
Ground Truth
SIREN
[Sitzmann '20]
Fourier Feat.
[Tancik '20]
SDF spectrum

Neural Fields

- Exciting and rapidly evolving research area!
- Many hard problems being solved, but still more work to be done
  - Robust generalization
  - Compositionality
  - Compact, efficient, & scalable 3D reconstruction
- How to integrate with computational imaging problems?

Next time…

Guest lecture (on Zoom!)

Mark Sheinin (CMU)

Imaging, Fast and Slow:
Computational Imaging for Sensing High-speed Phenomena

(two-time CVPR Best Paper Award winner!)