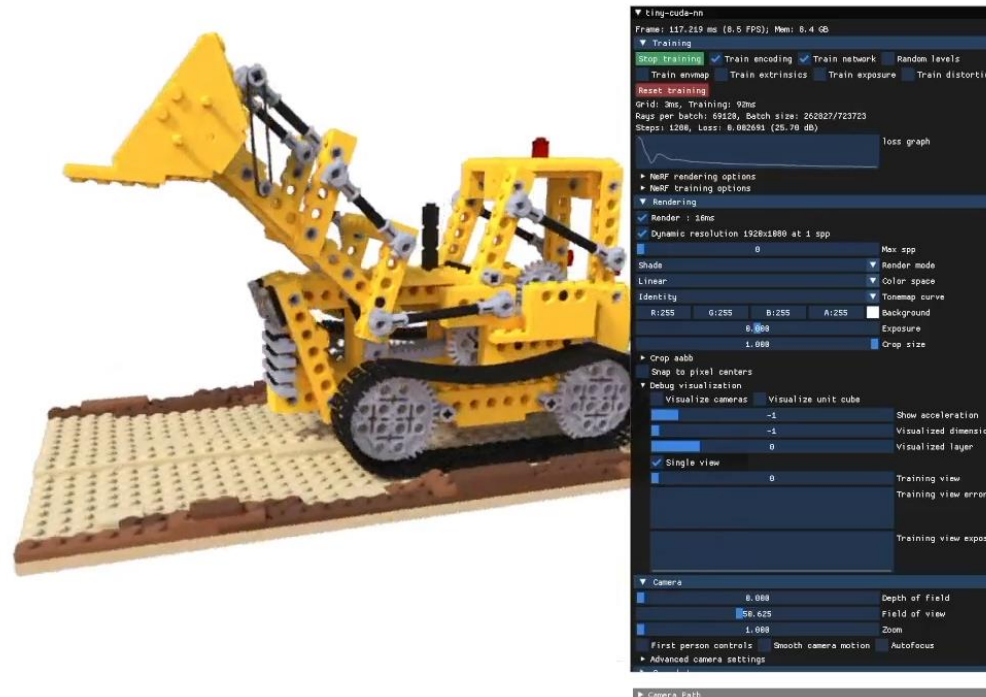Discussing
Instant Neural Graphics Primitives with a Multiresolution Hash Encoding
By Müller et.al, 2022
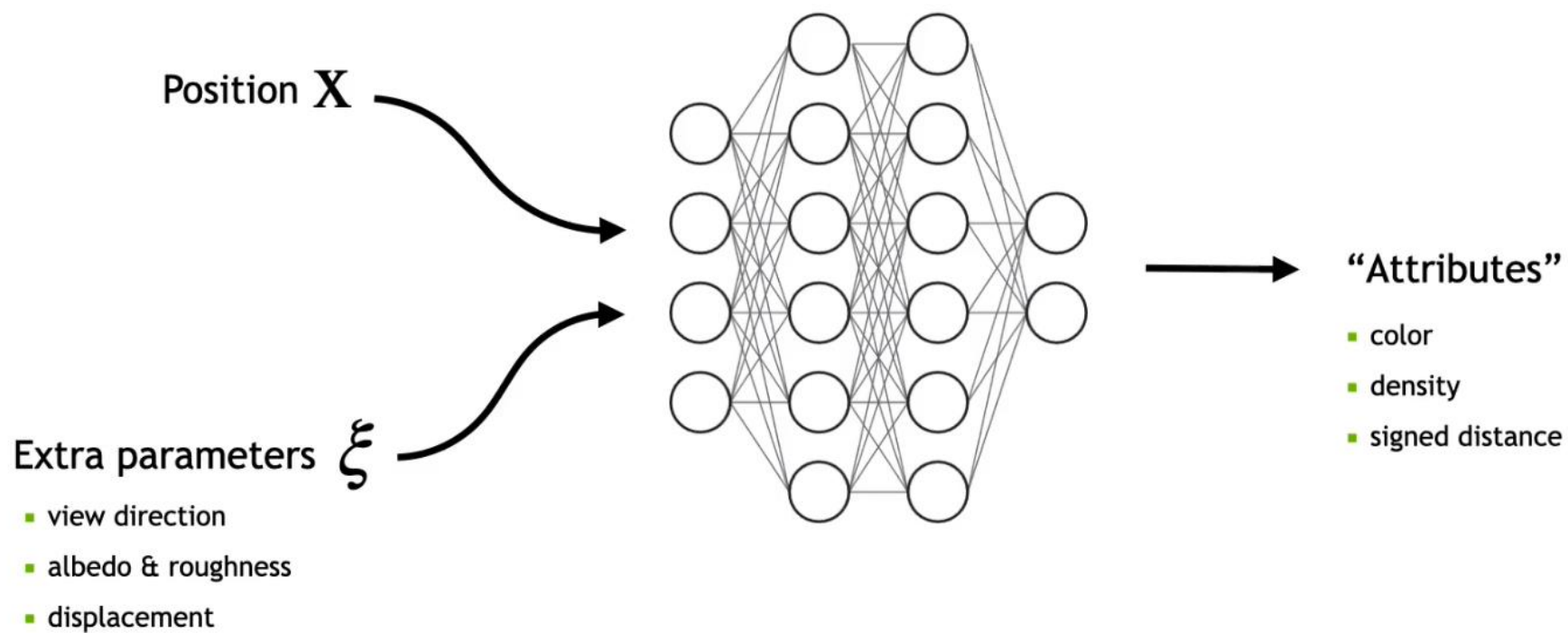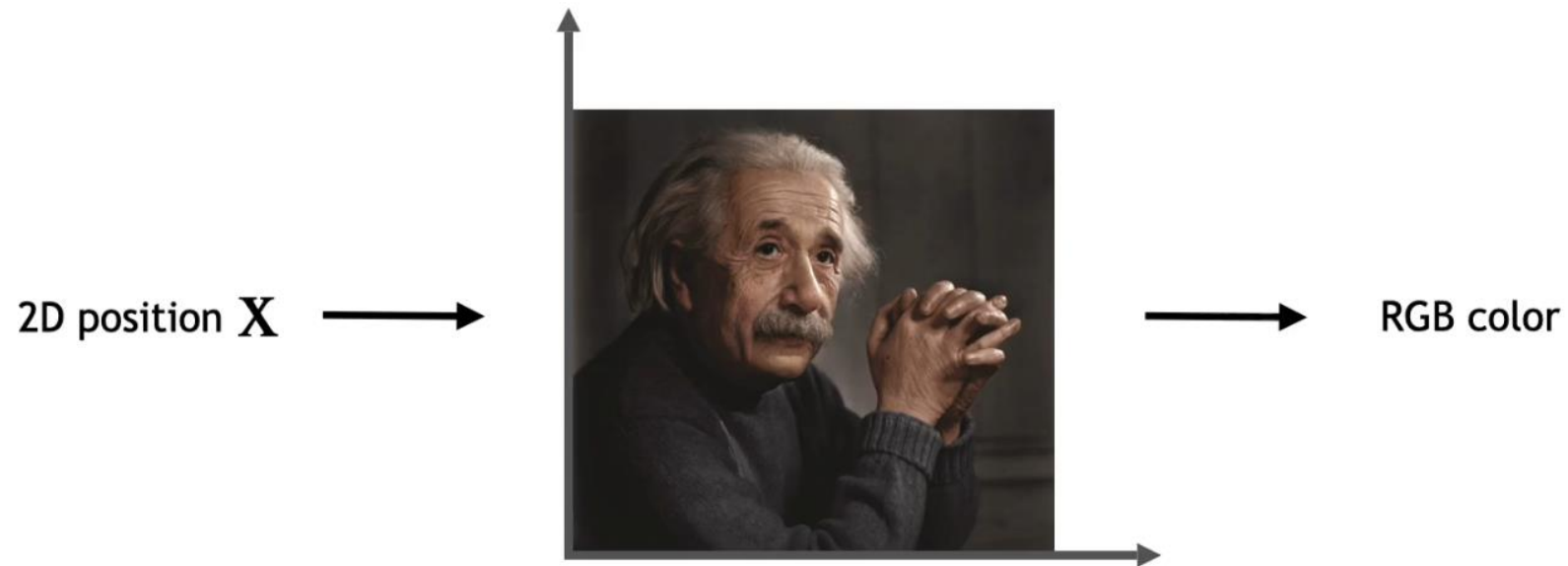

Oct 2022
Moritz Hambach and Rishabh Raj

Outline

1) understand idea
2) understand details and implementation

WHAT'S A NEURAL GRAPHICS PRIMITIVE?
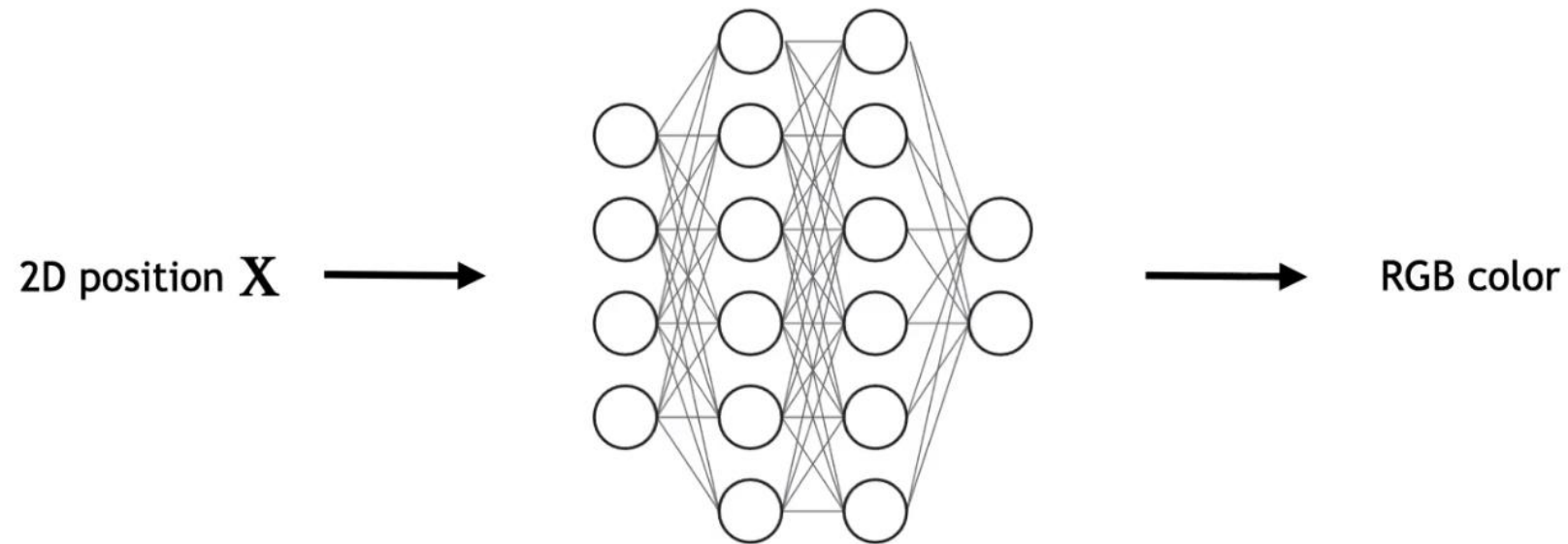An object represented by queries to a neural network!

Position $\mathbf{X}$
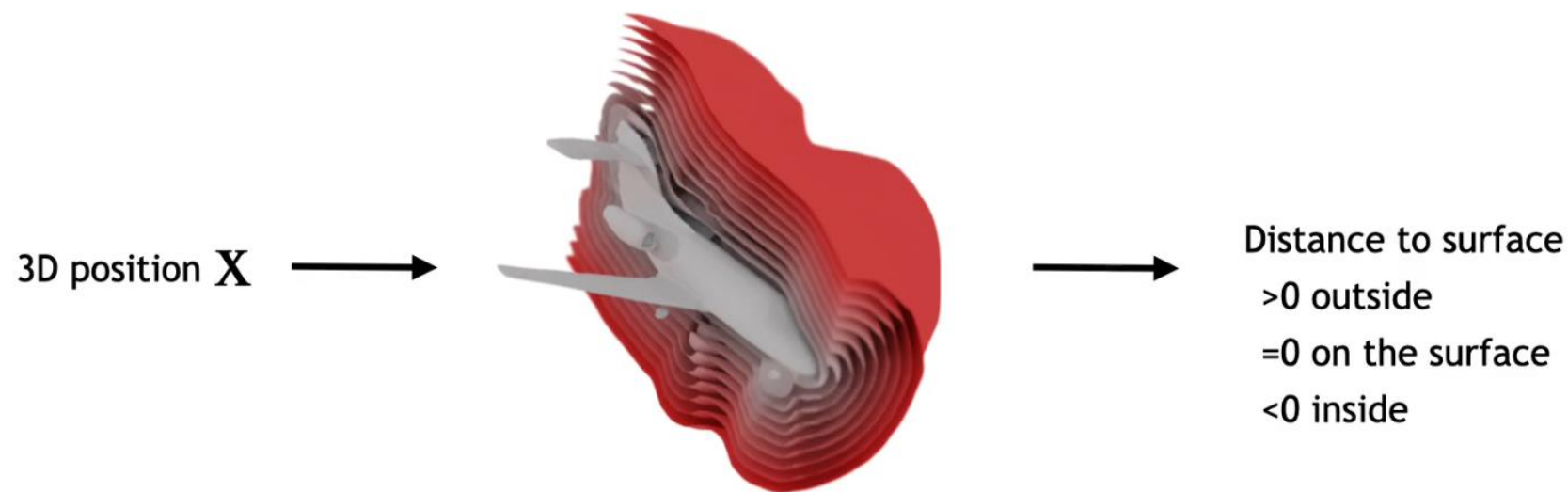
Extra parameters ξ
- view direction
- albedo & roughness
- displacement

"Attributes"
- color
- density
- signed distance

https://tom94.net/data/publications/mueller22instant/mueller22instant-gtc.mp4

EXAMPLE: IMAGE

2D position $\mathbf{X}$ → [image] → RGB color

"What's the pixel color at X?"

EXAMPLE: IMAGE

2D position X → → RGB color

"What's the pixel color at X?"

https://tom94.net/data/publications/mueller22instant/mueller22instant-gtc.mp4

EXAMPLE: SIGNED DISTANCE FUNCTION

3D position $\mathbf{X}$ $\longrightarrow$ [surface illustration] $\longrightarrow$ Distance to surface
  >0 outside
  =0 on the surface
  <0 inside

"How far away from surface is X?"

EXAMPLE: SIGNED DISTANCE FUNCTION

3D position $\mathbf{X}$ →

Distance to surface
>0 outside
=0 on the surface
<0 inside
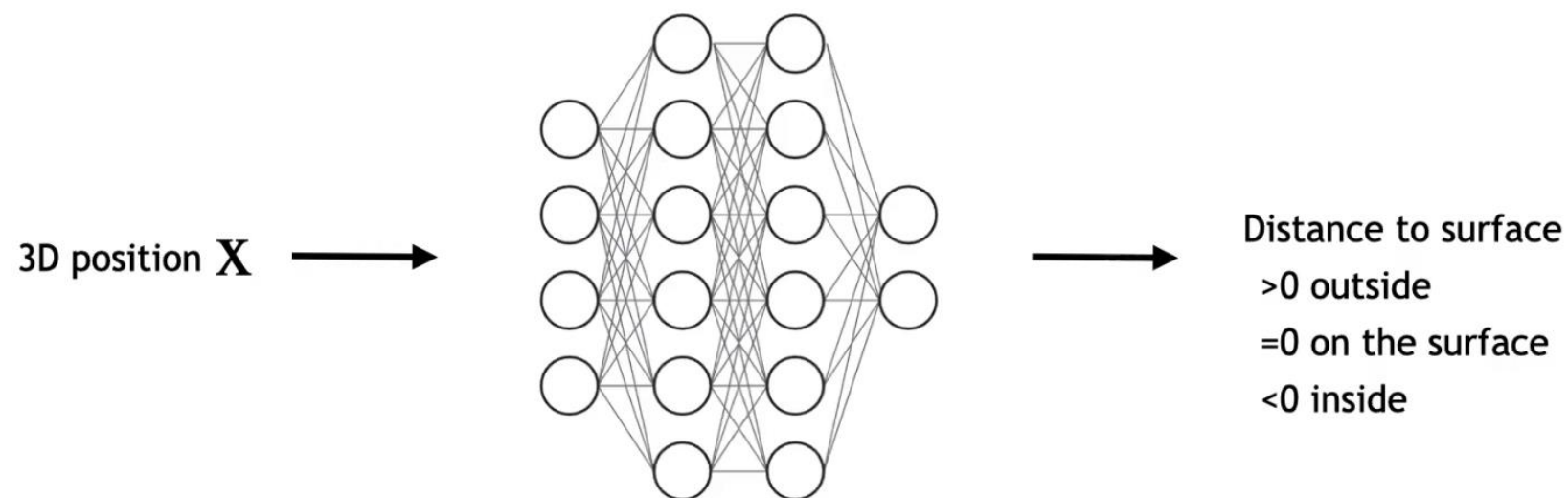
"How far away from surface is X?"

EXAMPLE: RADIANCE & DENSITY FIELD (NERF)

Position x

View direction d

RGB color & density
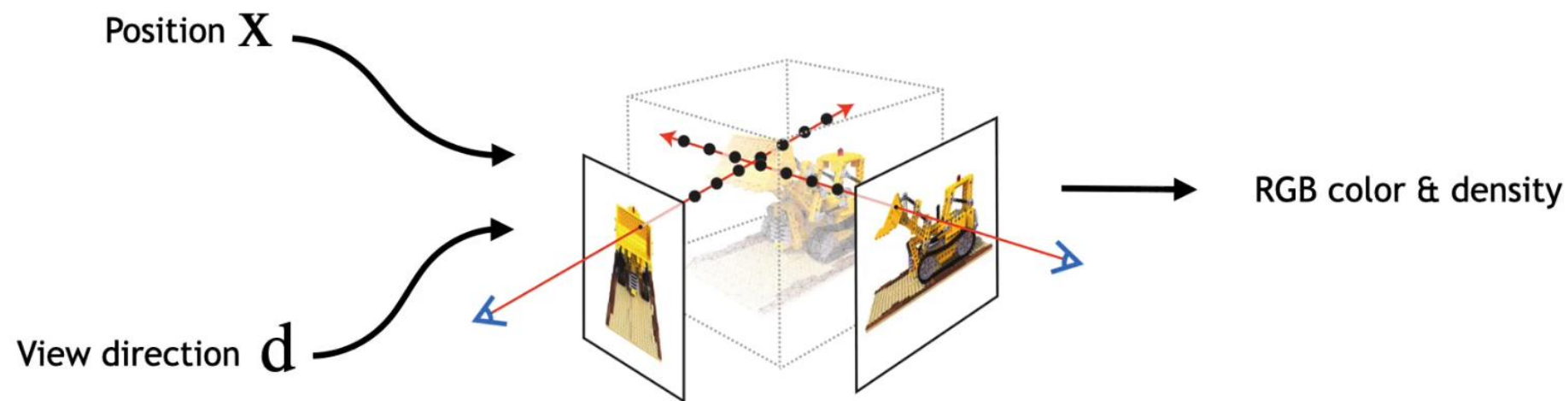
"How much stuff is at X and what color does it have when viewed from d?"

NERF: Mildenhall et.al. , 2020
https://tom94.net/data/publications/mueller22instant/mueller22instant-gtc.mp4
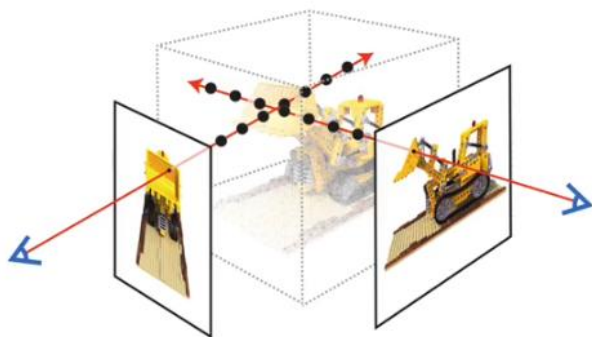
WHY NEURAL NETWORKS?
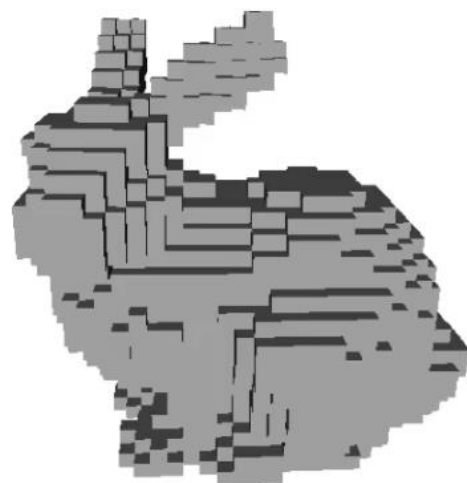
versus
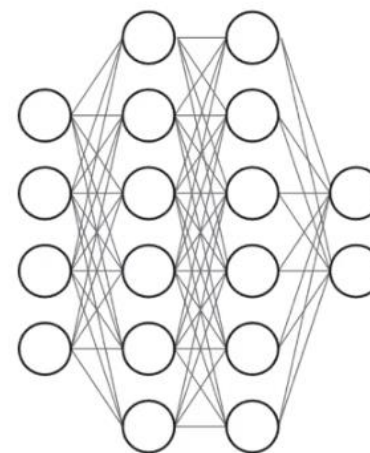
- Impractical memory usage
- Rigid; limited to 3D
- But really fast

- Trades memory for compute
- Mutable; n-dim. inputs
- ~~Reputation for being slow~~
- Competitively fast!

https://tom94.net/data/publications/mueller22instant/mueller22instant-gtc.mp4

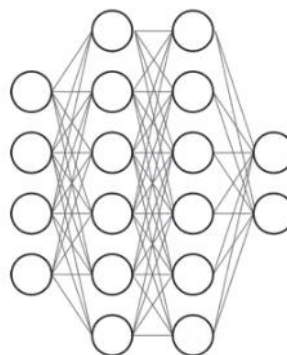# 3 PILLARS OF INSTANT NEURAL GRAPHICS

Multiplicative speed-ups >1000x!
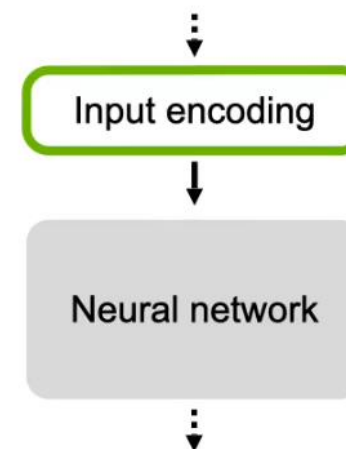
Rendering/training algorithm

- Task-specific GPU implementation
- **10-100x fewer steps than naïve tensor-based approach**

Small neural network

- Fully fused implementation
- **5-10x faster than TensorFlow**

"Good" input encoding

Input encoding
↓
Neural network

- Multiresolution hash encoding
- **Better speed-vs-quality tradeoff than prior work**
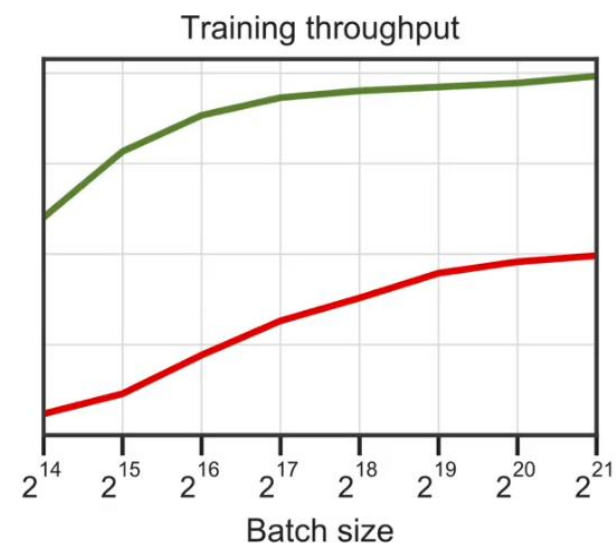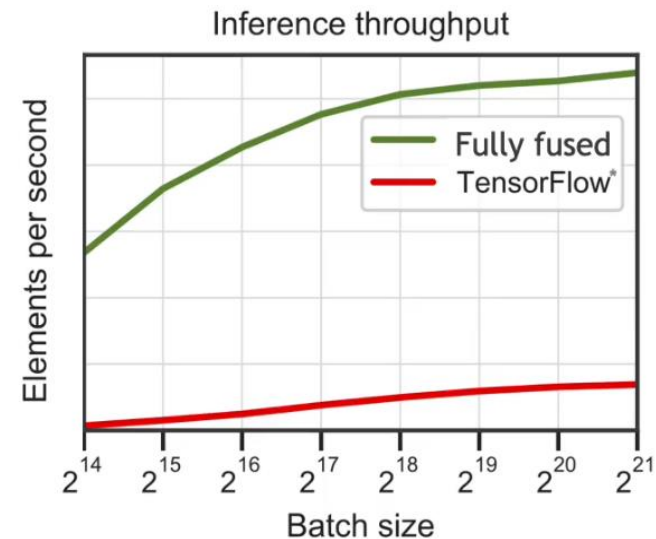- **Task agnostic**

Rendering / training algorithm

- Depends on task

- NeRF:
  - query rays mostly near surfaces
  - step size grows linear with distance
  - stop marching once almost all light is absorbed
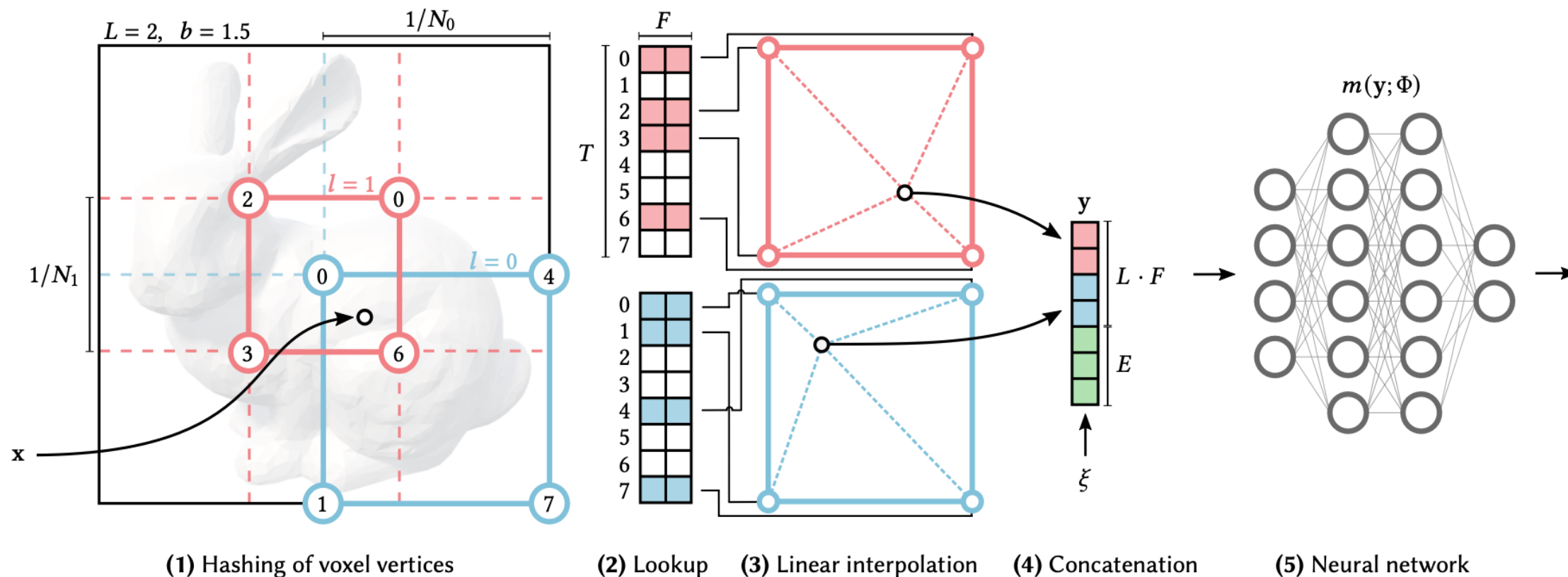 - keep coarse occupancy grid to allow skipping empty space
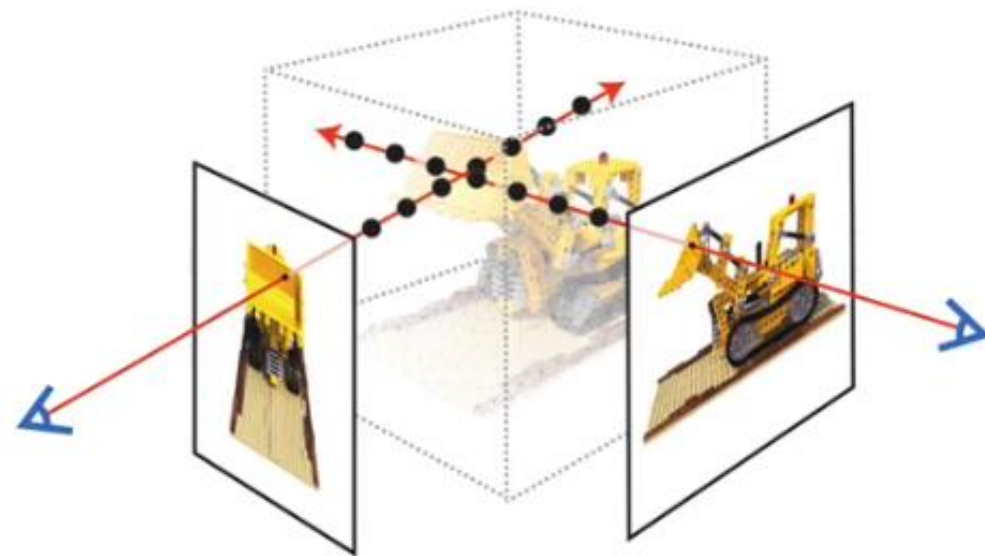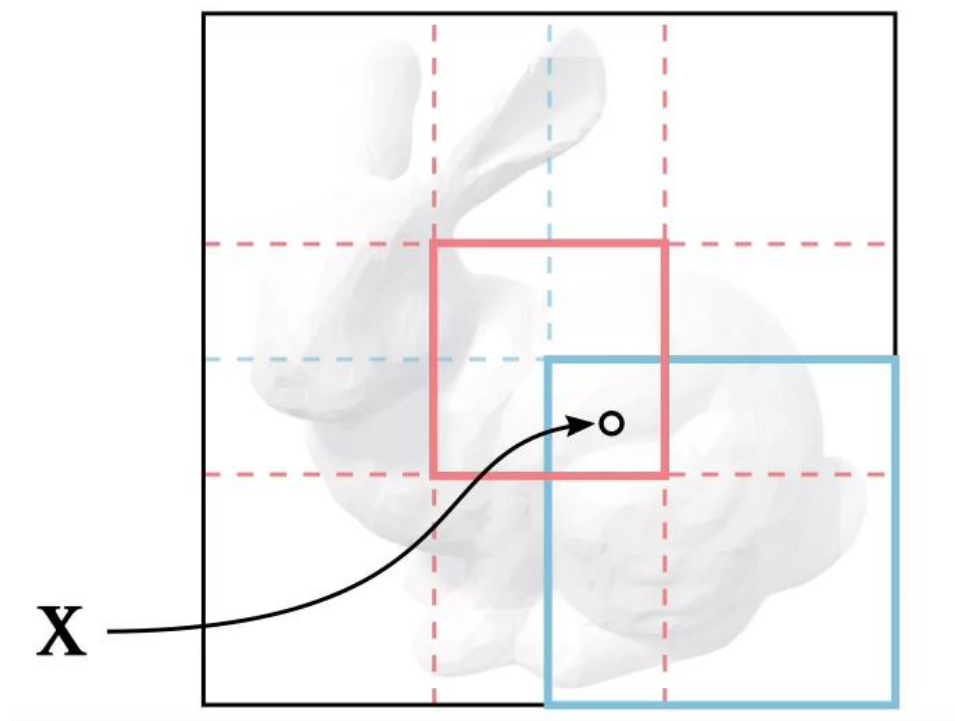
Small + fully fused Neural Networks

- MLP with only M=64 Neurons, L=2..4 hidden layers, no biases, ReLU activation

- Run on a single CUDA kernel (avoid memory traffic)

- compute is $O(M2)$, traffic is $O(M)$, but can dominate for small M

Multi-resolution hash encoding

(1) Hashing of voxel vertices   (2) Lookup   (3) Linear interpolation   (4) Concatenation   (5) Neural network
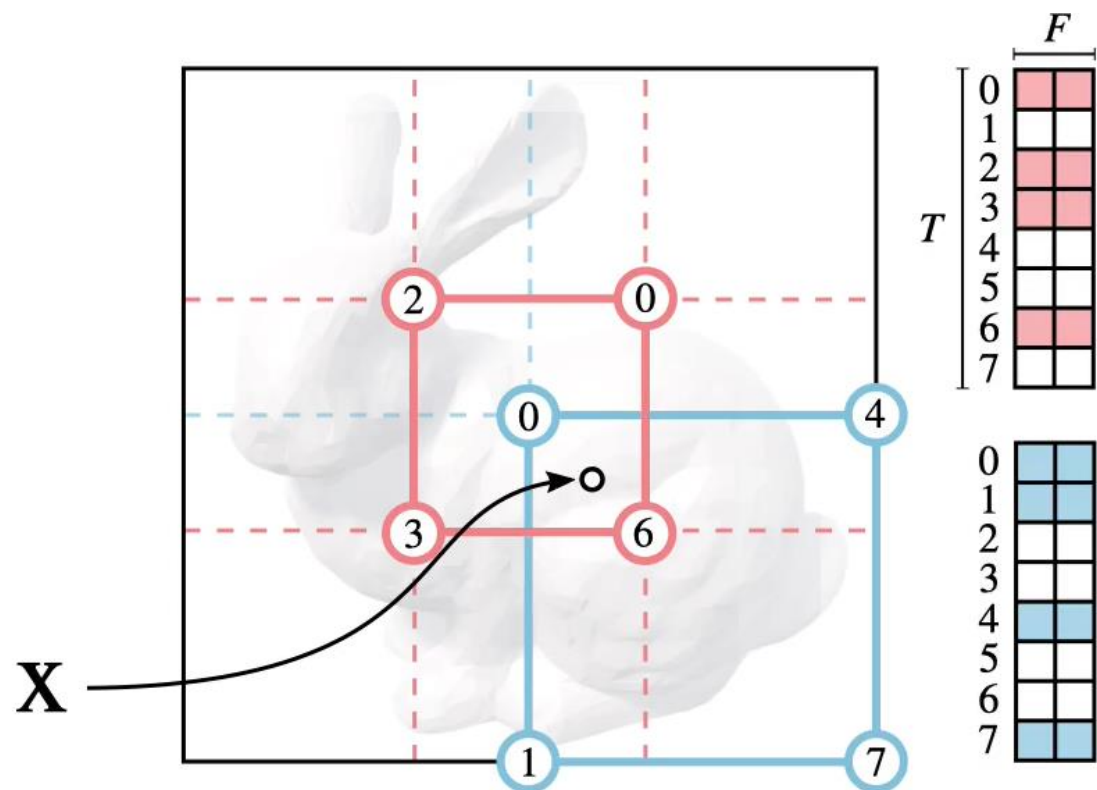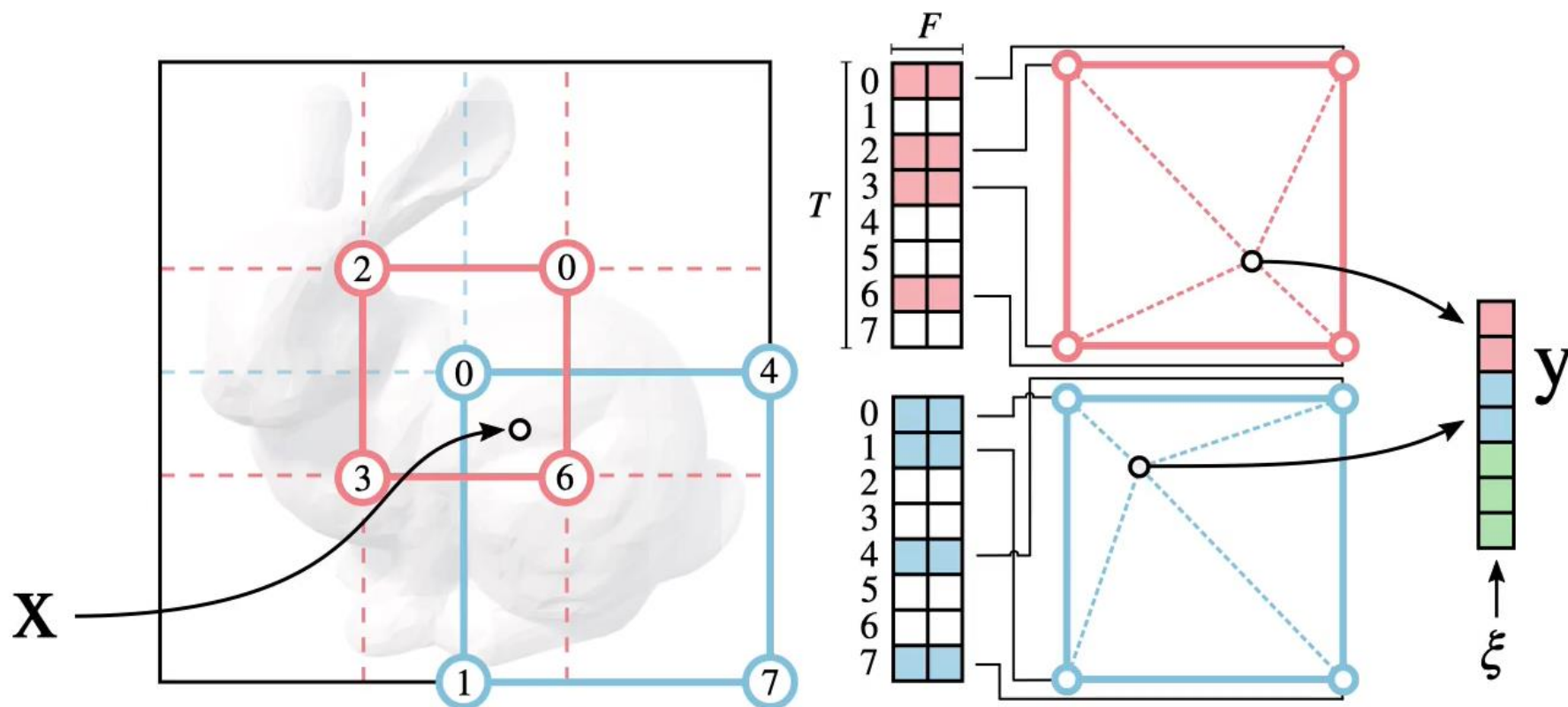
Multi-resolution hash encoding

- Multi-resolution
  - Automatic level of detail
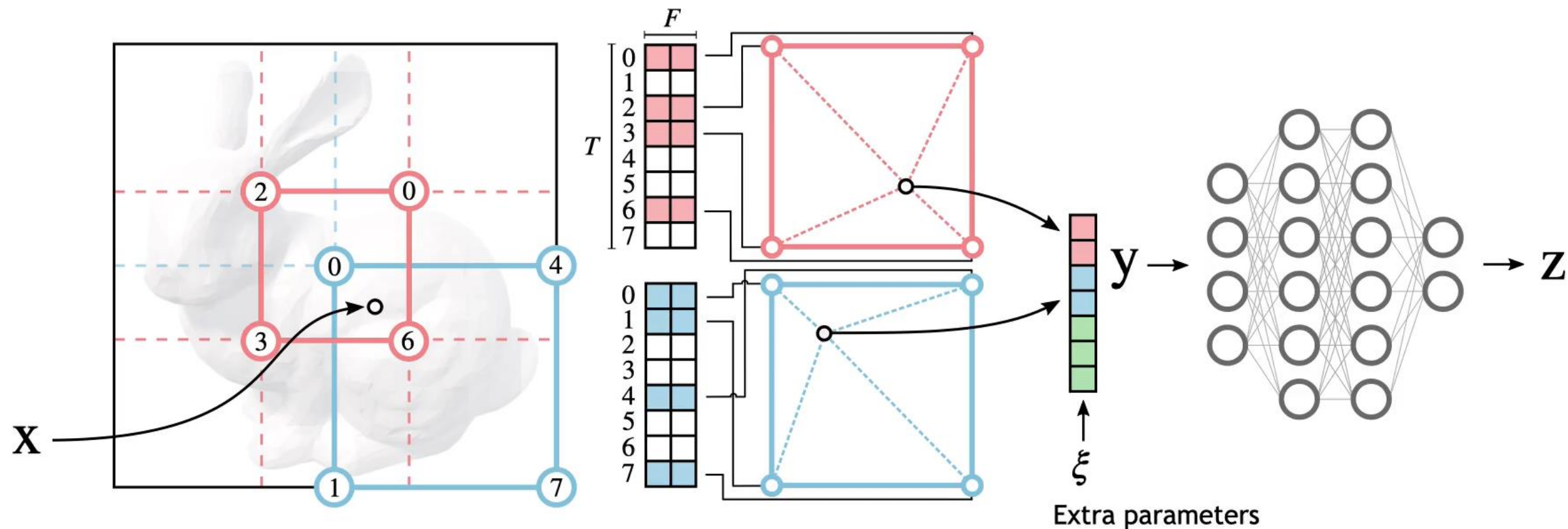
Multi-resolution hash encoding

- Multi-resolution
  - Automatic level of detail
- Hashing
  - Task agnostic
  - Table size T controls quality vs memory

Multi-resolution hash encoding

- Multi-resolution
  - Automatic level of detail
- Hashing
  - Task agnostic
  - Table size T controls quality vs memory
- Linear interpolation to enable continuity and differentiability

Multi-resolution hash encoding

Demo!

Questions?