Instant Neural Graphics Primitives with a Multiresolution Hash Encoding

THOMAS MÜLLER, ALEX EVANS, CHRISTOPH SCHIED, ALEXANDER KELLER, NVIDIA
ACM Transactions on Graphics (SIGGRAPH 2022)
Published on 16 Jan 2022

Presented by: Amir Alimohammadi

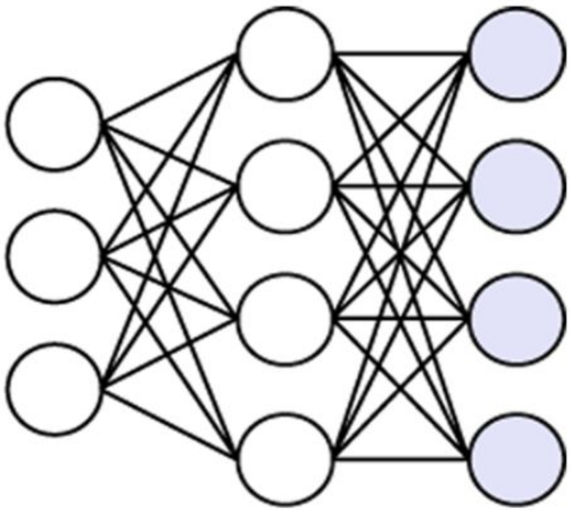
What is a Neural Graphics Primitive?

An object represented by queries to a neural network.

Position X

Extra parameters

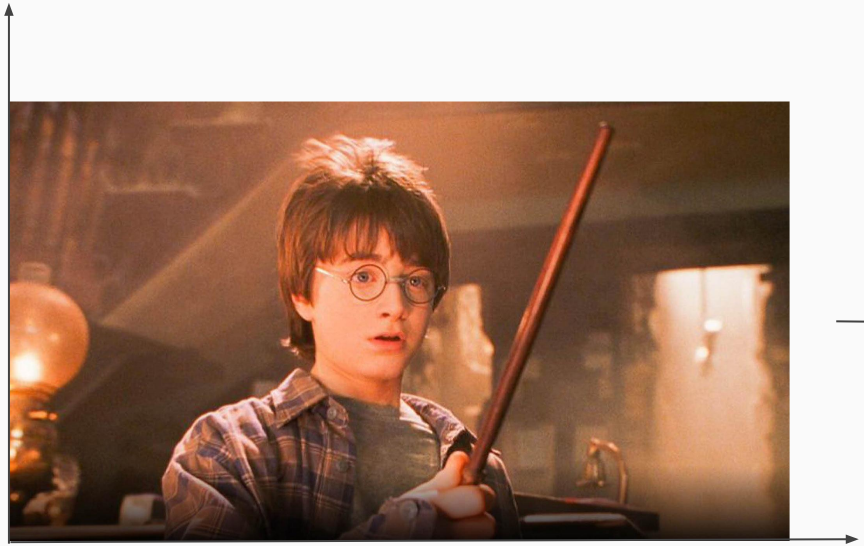
- View direction
- Albedo
- Displacement



Attributes

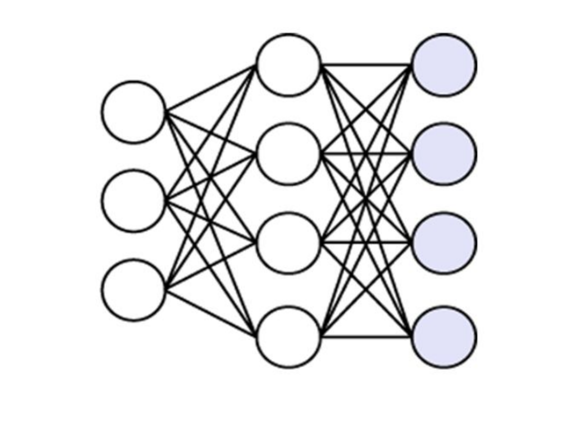
- Color
- Density
- Signed distance

Example: Image

2D Position X →  → RGB color

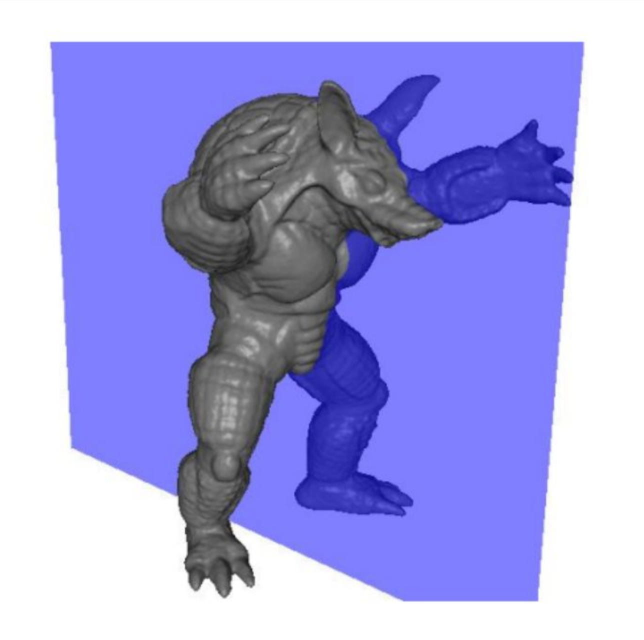
What is the pixel color at X?

Example: Image

2D Position X →  → RGB color

What is the pixel color at X?

Example: Signed Distance Function

3D Position X →  → Distance to surface

>0 outside

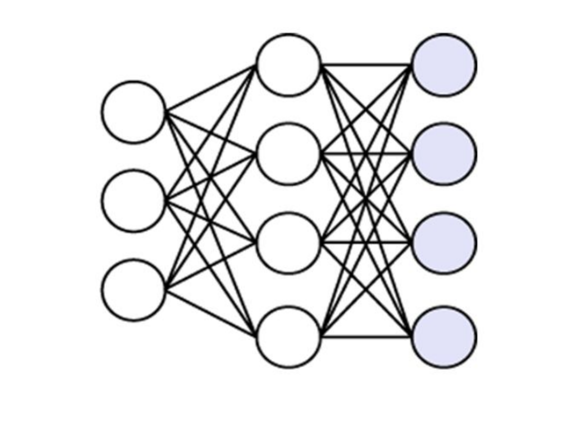
=0 on the surface

<0 inside

How far from surface is X?

Example: Signed Distance Function

3D Position X



Distance to surface

>0 outside

=0 on the surface

<0 inside

How far from surface is X?

Example: Radiance & Density Field

Position X

View direction d

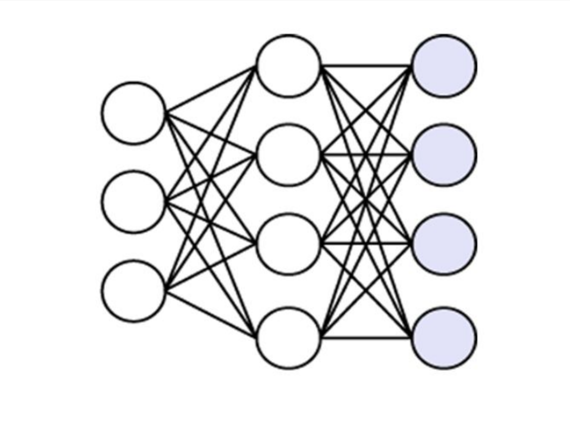
RGB color & Density

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

Example: Radiance & Density Field

Position X

View direction d



RGB color & Density

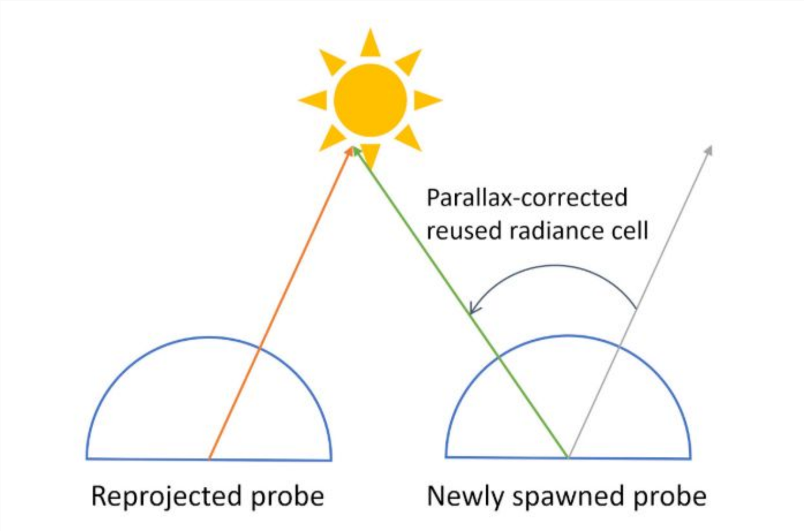
How much stuff is at X and what color does it have when viewed from d?

Example: Radiance Caching

Position X

Extra parameters

- View direction
- Surface normals
- Albedo



RGB color

Global illumination

What color does the object at X have when viewed from d?

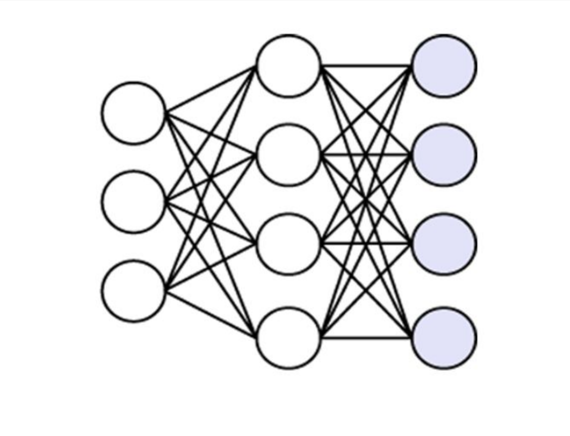
A Fast Scalable Two-Level Radiance Caching Scheme for Real-Time Global Illumination (Boissé et al.), arXiv'22.

Example: Radiance Caching

Position X

Extra parameters

- View direction
- Surface normals
- Albedo



RGB color

Global illumination

What color does the object at X have when viewed from d?

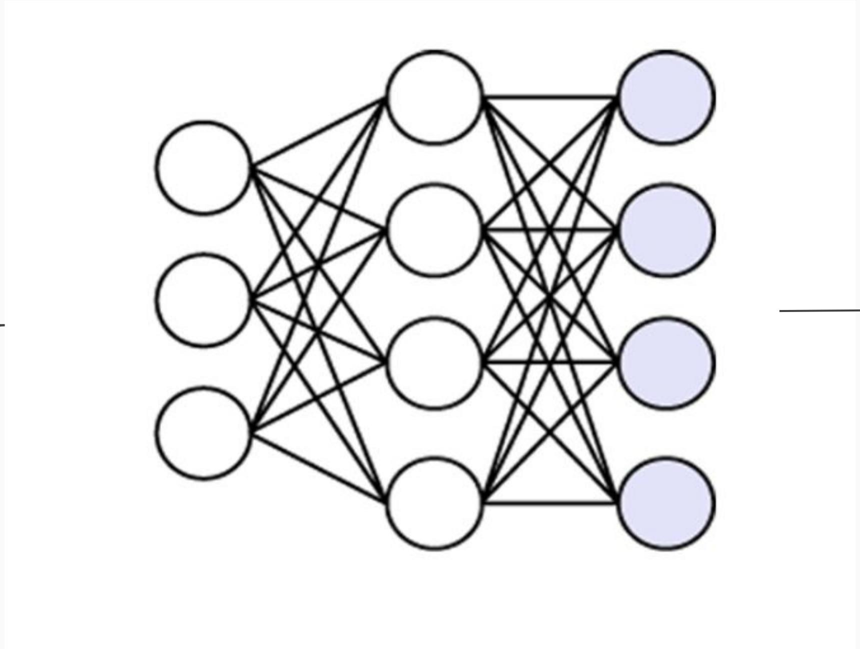
What is a Neural Graphics Primitive?

An object represented by queries to a neural network.

Position X

Extra parameters

- View direction
- Albedo
- Displacement



Attributes

- Color
- Density
- Signed distance

Abstract

Neural graphics primitives:

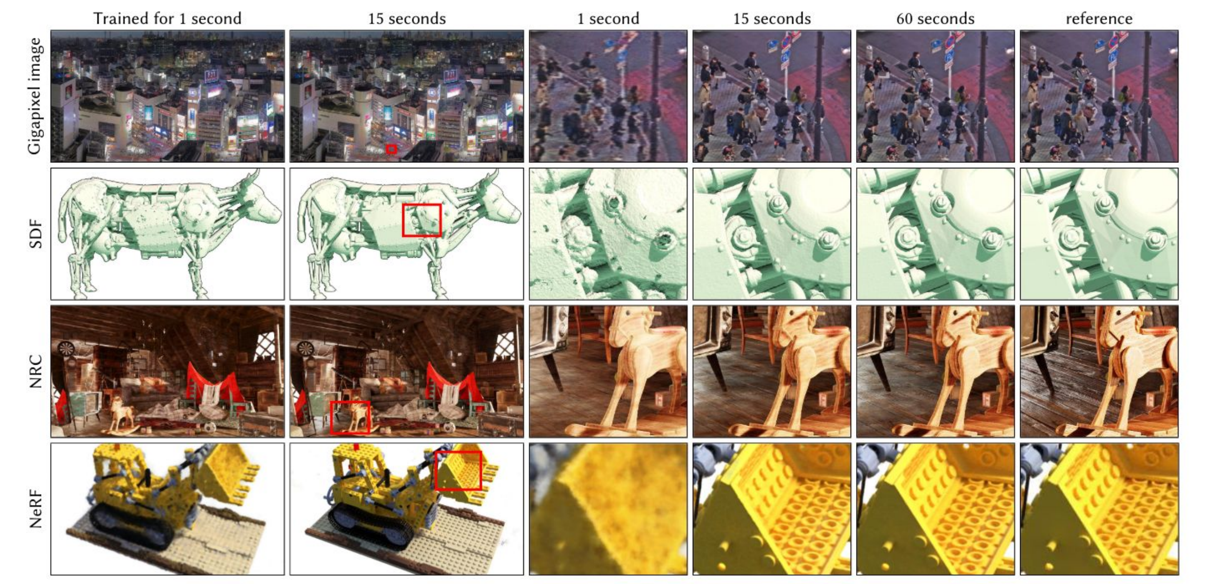
- Costly to train and evaluate.

Solution:

- New versatile input encoding.
- Allows use of a smaller network without quality loss.
- Reduces floating point and memory access operations.

Key Components:

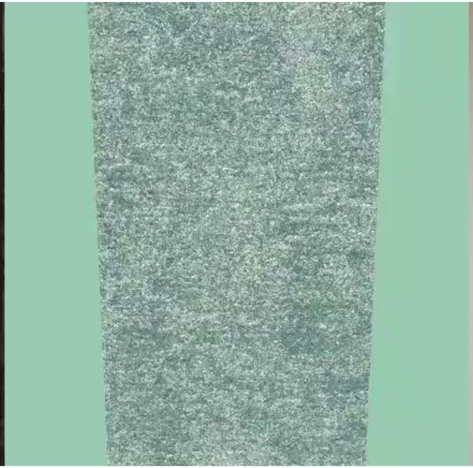
- Small neural network.
- Multiresolution hash table of trainable feature vectors.
- Feature vectors optimized via stochastic gradient descent.



Task-Agnostic Encoding

Neural gigapixel images

Neural SDF



NeRF

Neural volume

Representation

We desire representations that remain fast and compact while capturing high-frequency, local detail.

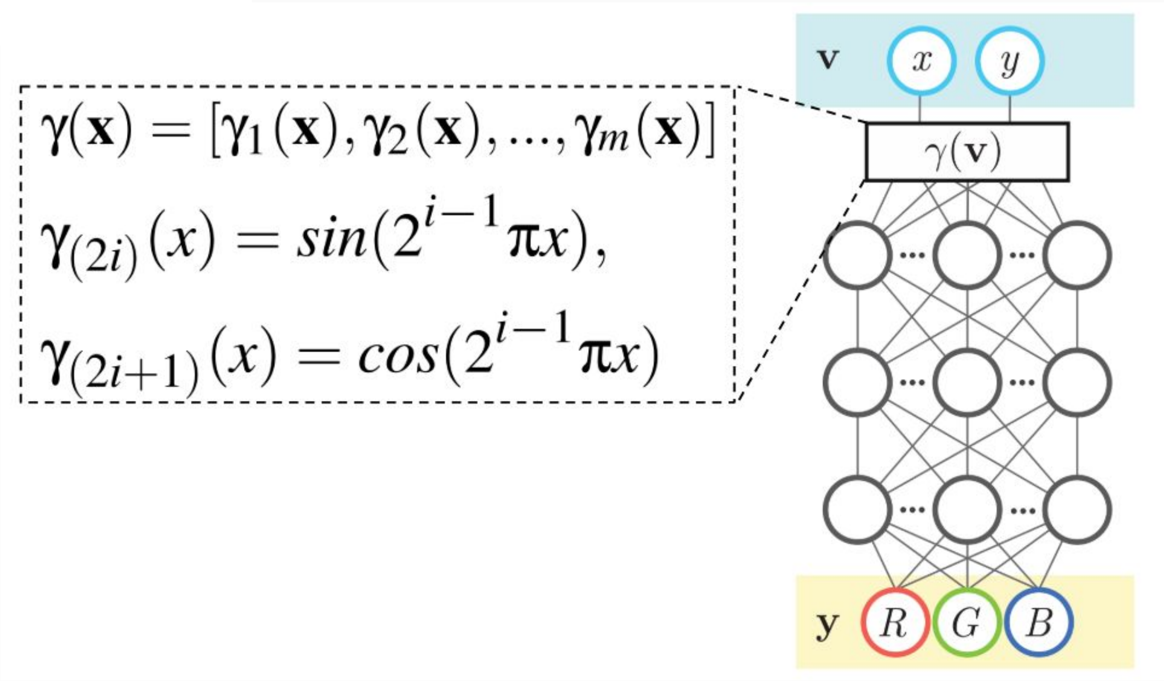
Therefore, multiresolutional hash encoding is a good answer as it provides us with two key features.

1- Adaptivity

2- Efficiency

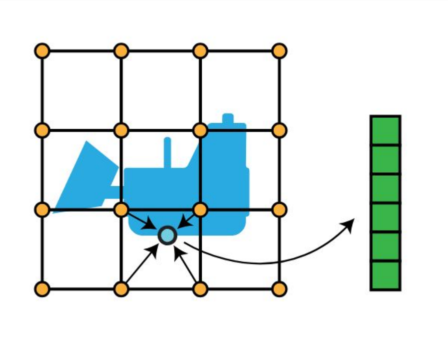
Positional Encodings

- Expand the dimensionality of the input (e.g. 2 → 2M dimensions).
- Via a collection of sinusoidal basis functions.

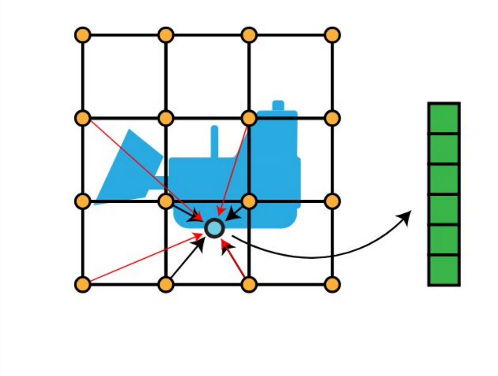


Representations

Dense grids



Multi-resolution grids



Effect of Representation

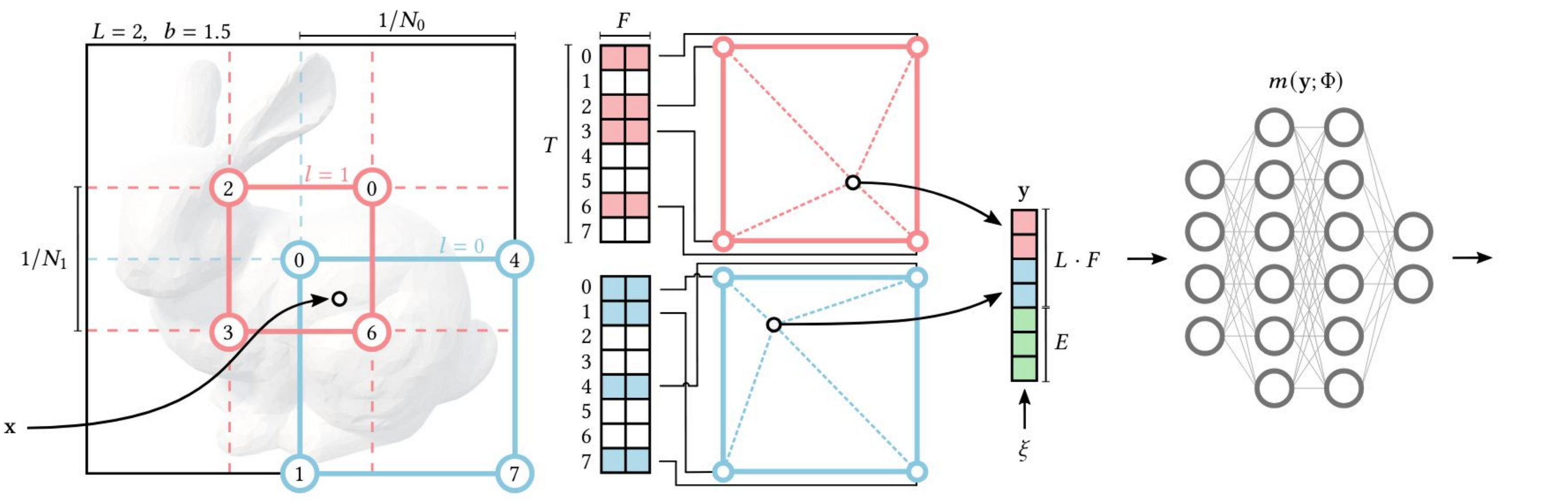
(a) No encoding	(b) Frequency [Mildenhall et al. 2020]	(c) Dense grid Single resolution	(d) Dense grid Multi resolution	(e) Hash table (ours) $T = 2^{14}$	(f) Hash table (ours) $T = 2^{19}$

411 k + 0 parameters	438 k + 0	10 k + 33.6 M	10 k + 16.3 M	10 k + 494 k	10 k + 12.6 M
11:28 (mm:ss) / PSNR 18.56	12:45 / PSNR 22.90	1:09 / PSNR 22.35	1:26 / PSNR 23.62	1:48 / PSNR 22.61	1:47 / PSNR 24.58

Method

Objective:

Given a neural network $m(y;\Phi)$, an encoding $y = \mathrm{enc}(x;\theta)$ that improves the approximation quality and training speed is desired.



(1) Hashing of voxel vertices (2) Lookup (3) Linear interpolation (4) Concatenation (5) Neural network

More Detail about the Network

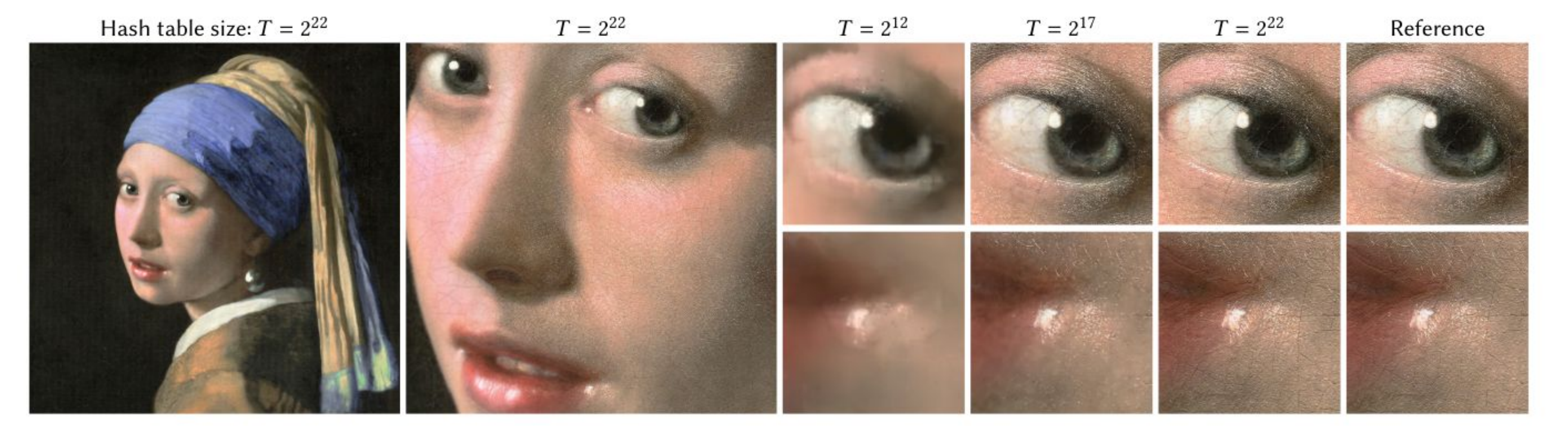
- Spatial Hash Function:

$$h(x) = \left(\bigoplus_{i=1}^{d} x_i \pi_i \right) \mod T$$

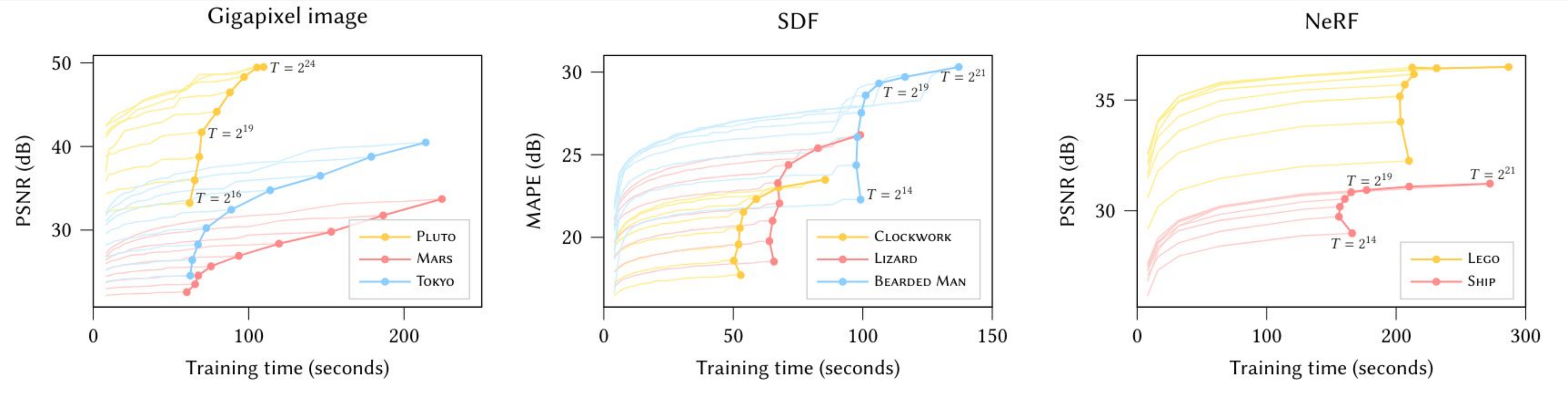
- Interpolation:
  - Feature vectors at each corner are d-linearly interpolated according to the relative position of x within its hypercube.
  - Interpolation Weight:

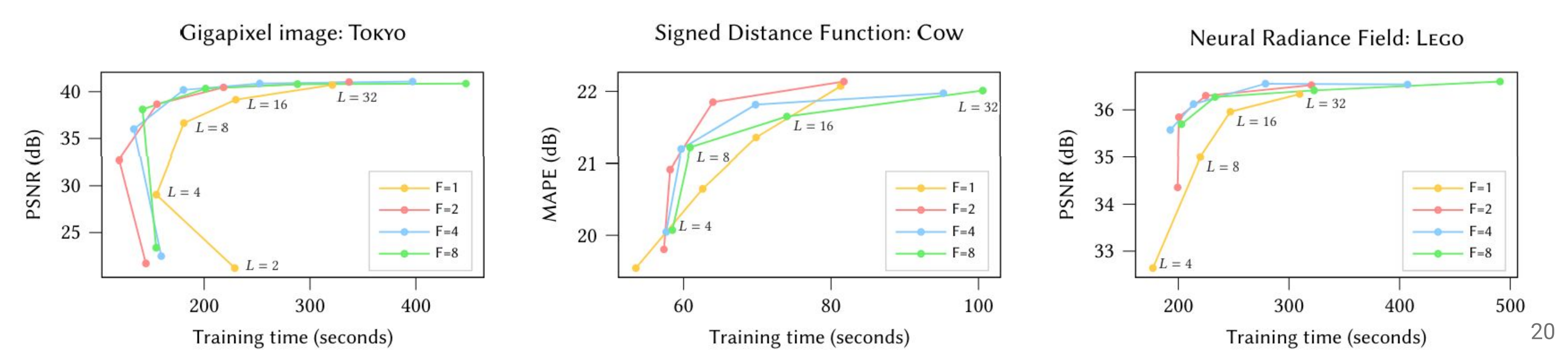
$$w_l := x_l - \lfloor x_l \rfloor$$

- Input to the MLP:
  - Interpolated feature vectors of each level are concatenated with auxiliary inputs $\xi \in \mathbb{R}^E$.



Performance vs. Quality





Implicit hash collision resolution

Different resolution levels complement each other.

a. Coarser Levels:
   i. Injective, no collisions.
   ii. Represent low-resolution scenes.
b. Fine Levels:
   i. Capture small features.
   ii. Fine grid resolution.
   iii. Suffer from many collisions, disparate points hash to the same table entry.

Gradient Averaging:

A. When training samples collide, their gradients are averaged.
B. Importance to the final reconstruction varies among collided samples.

Importance Weighting:

A. Importance of samples differs based on factors like visibility and density.
B. Samples on visible surfaces contribute strongly to reconstruction.
C. Samples in empty space have smaller weights.

Advantages of Multiresolution Hash Encoding Techniques

Online adaptivity:

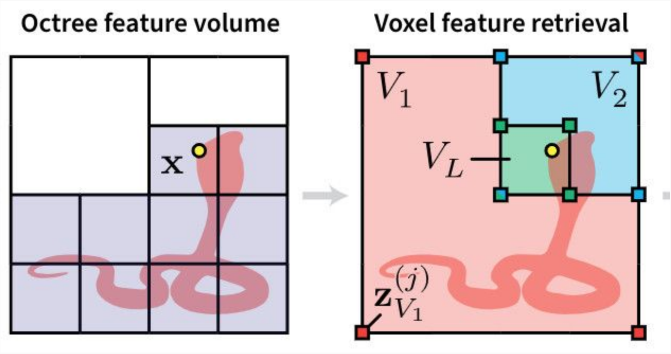
- Multiresolution hash encoding adapts to the training data distribution.
- Gains benefits of tree-based encodings without needing task-specific maintenance.

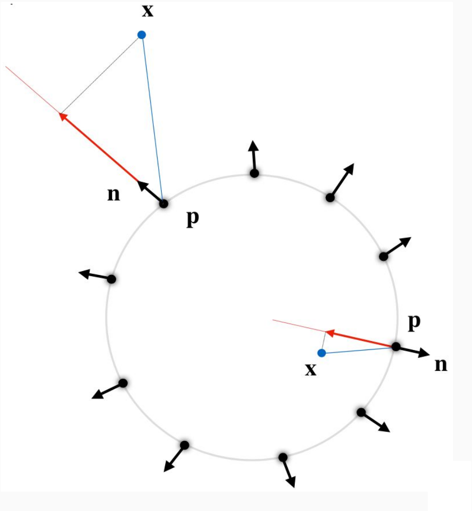
d-linear interpolation:

- Interpolating queried hash table entries brings continuity.
- Prevents blocky appearance.

Higher-Order Smoothness:

- In some applications higher-order smoothness is beneficial.
- For example, in signed distance functions, the gradient $\partial m(\mathrm{enc}(x;\theta);\Phi)/\partial x$ (surface normal) should ideally be continuous.





Neural Geometric Level of Detail (Takikawa et al.), CVPR’21

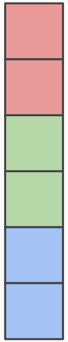
Concatenation vs. reduction

Concatenation at the end of the encoding process is favored over reduction for two main reasons:

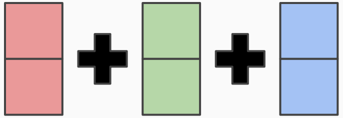
- Enables independent, fully parallel processing of each resolution.
- Avoids potential loss of useful information caused by dimensionality reduction.

While concatenation is preferred by default due to its advantages, reduction might be favorable in certain scenarios where the computational cost is a significant concern compared to the encoding process itself.

In instant-ngp, concatenation with a fixed dimensionality (F = 2) consistently yielded superior results.



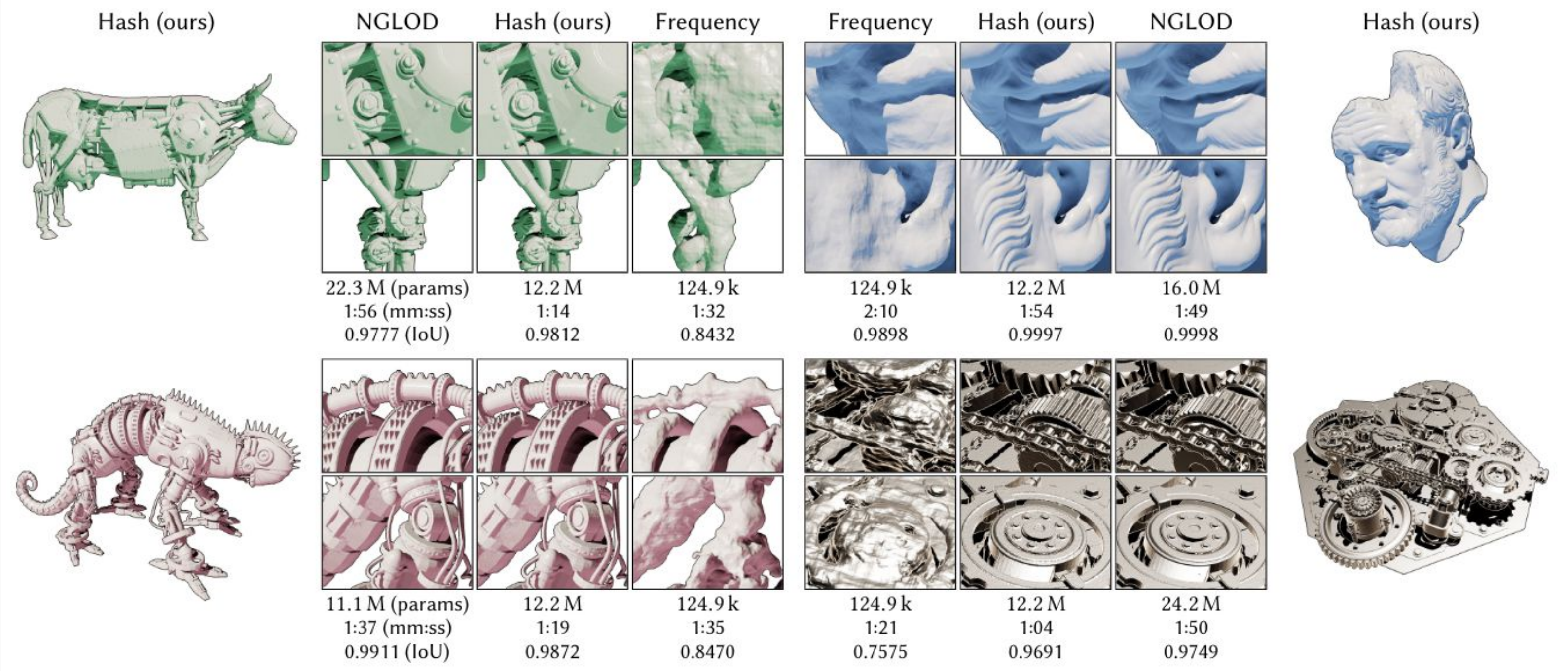
Concatenation



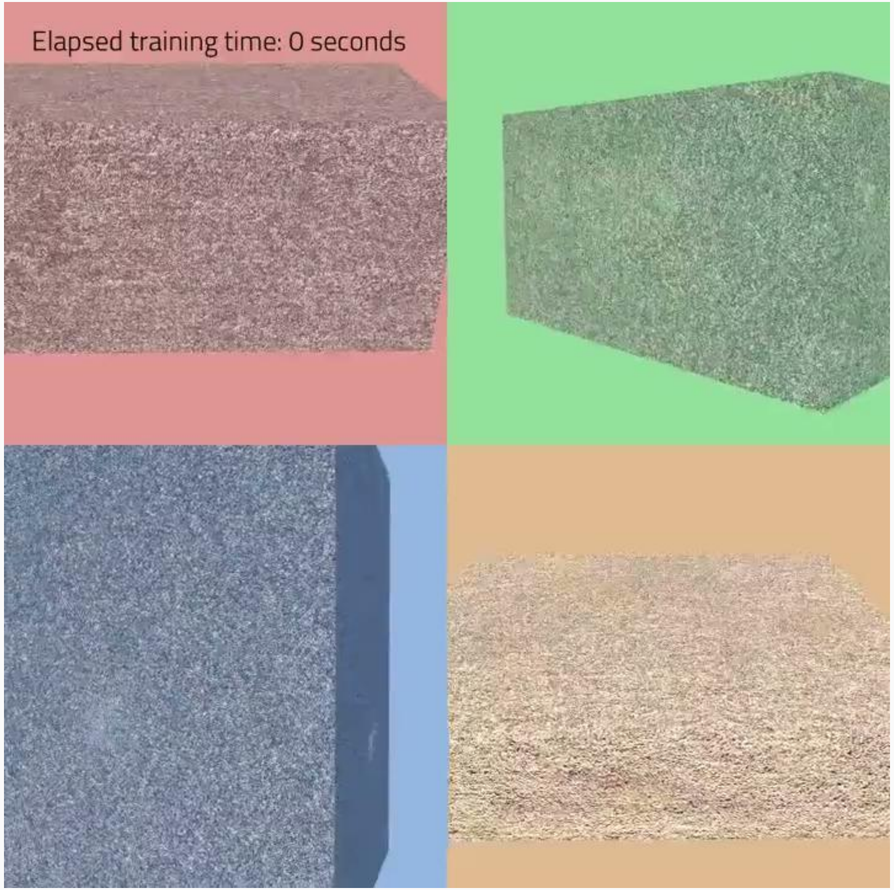
Join

Comparison on Neural Signed Distance Function

Neural signed distance functions trained for 11,000 steps. The frequency encoding [Mildenhall et al. 2020] struggles with sharp details, NGLOD [Takikawa et al. 2021] achieves high visual quality within an octree, and while our hash encoding offers similar IoU quality and universal scene evaluation, it suffers from surface roughness due to hash collisions.

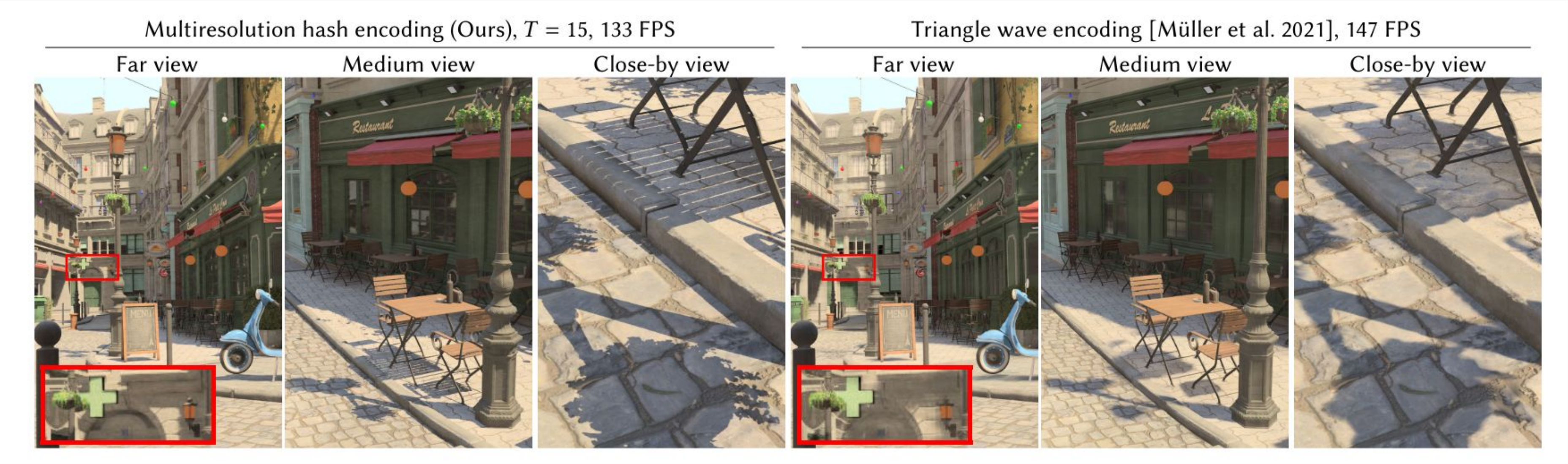


Comparison on Neural Signed Distance Function



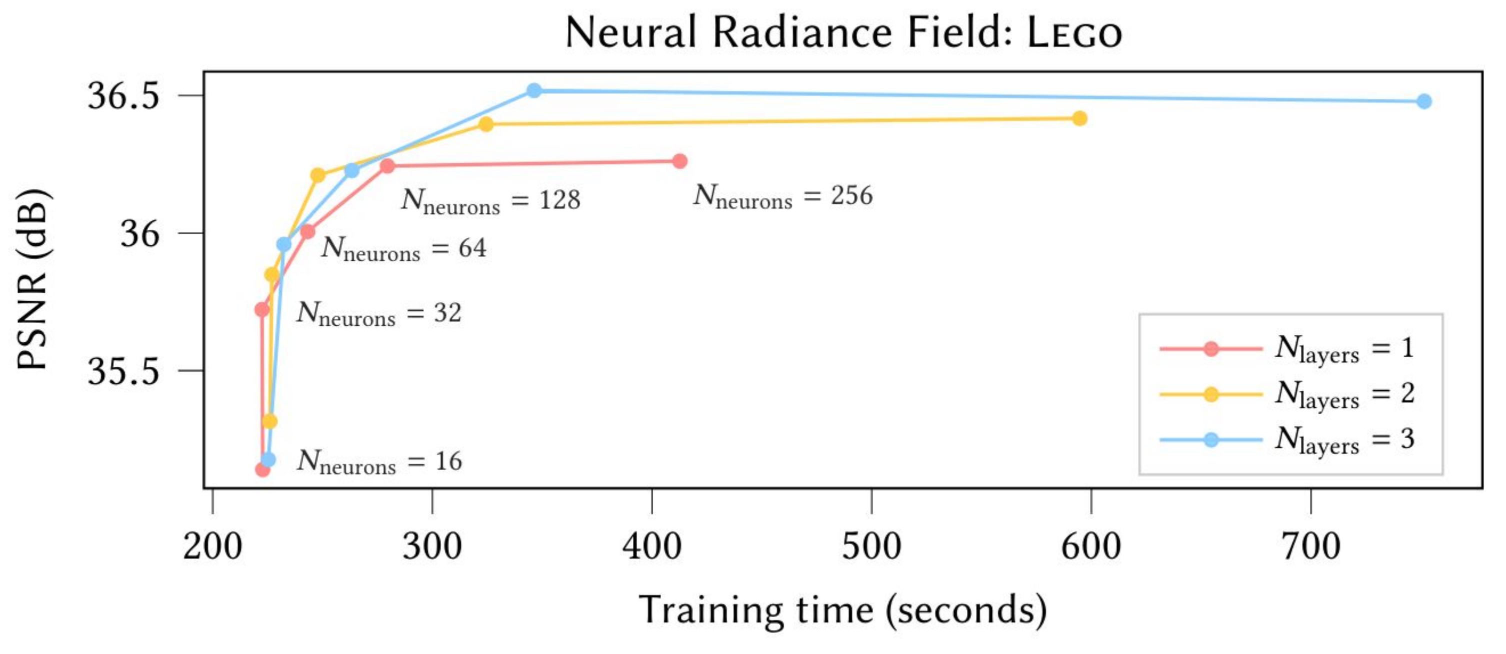
Neural Radiance Caching

Neural radiance caching significantly improves in quality with multiresolution hash encoding, demonstrating online adaptivity and maintaining high performance at 133 frames per second for 1920×1080px resolution.



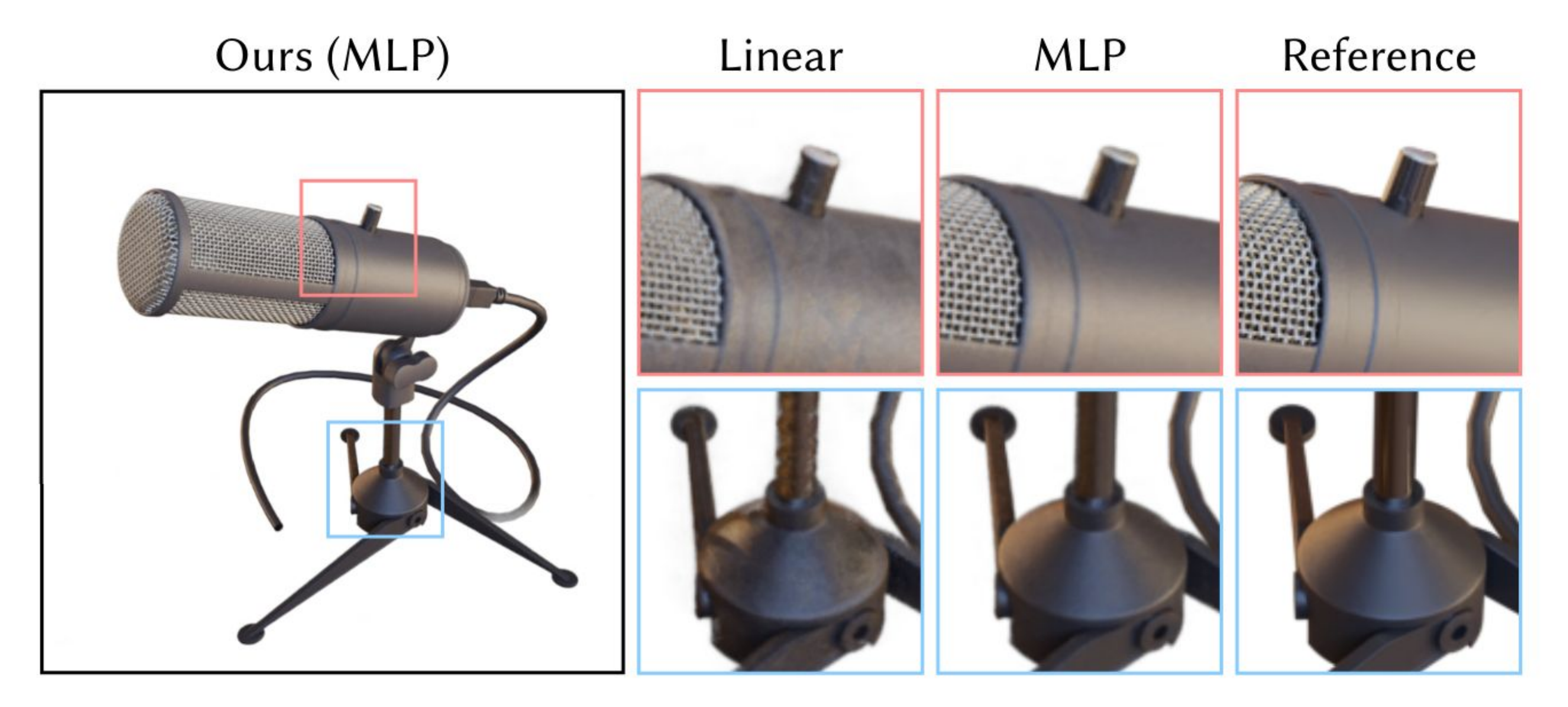
Impact of MLP Configuration

The effect of the MLP size on test error vs. training time (31 000 training steps) on the Lego scene.
Informed by this analysis, we choose N_{layers} = 2 and N_{neurons} = 64.



Impact of MLP Configuration

Using an MLP for learning a NeRF, compared to a linear transformation, significantly improves quality by resolving specular details and reducing background noise, with only a 15% increase in computational cost.



Comparison

Instant-ngp NeRF implementation with multiresolution hash encoding significantly outperforms traditional methods like NeRF, mip-NeRF, and NSVF in terms of training speed, achieving high PSNR values in just 1 second to 5 minutes. This rapid improvement, showing a 20–60× speedup, is largely attributed to the optimized hash encoding.

	Mic	Ficus	Chair	Hotdog	Materials	Drums	Ship	Lego	avg.
Ours: Hash (1 s)	26.09	21.30	21.55	21.63	22.07	17.76	20.38	18.83	21.202
Ours: Hash (5 s)	32.60	30.35	30.77	33.42	26.60	23.84	26.38	30.13	29.261
Ours: Hash (15 s)	34.76	32.26	32.95	35.56	28.25	25.23	28.56	33.68	31.407
Ours: Hash (1 min)	35.92	33.05	34.34	36.78	29.33	25.82	30.20	35.63	32.635
Ours: Hash (5 min)	36.22	33.51	35.00	37.40	29.78	26.02	31.10	36.39	33.176
mip-NeRF (~hours)	36.51	33.29	35.14	37.48	30.71	25.48	30.41	35.70	33.090
NSVF (~hours)	34.27	31.23	33.19	37.14	32.68	25.18	27.93	32.29	31.739
NeRF (~hours)	32.91	30.13	33.00	36.18	29.62	25.01	28.65	32.54	31.005
Ours: Frequency (5 min)	31.89	28.74	31.02	34.86	28.93	24.18	28.06	32.77	30.056
Ours: Frequency (1 min)	26.62	24.72	28.51	32.61	26.36	21.33	24.32	28.88	26.669

Thank you!

- Thank you for your attention!
- I appreciate your time and interest.
- If you have any questions, please feel free to ask.
- Contact information: alimohammadiamirhossein@gmail.com