Instant Neural Graphics Primitives with a Multiresolution Hash Encoding

THOMAS, NVIDIA, Switzerland
ALEX EVANS, United Kingdom
CHRISTOPH, NVIDIA, USA
ALEXANDER, NVIDIA, Germany

Presented By Heguang Lin at 09/13/2023

Motivation

Need a fast, compact representations with high-frequency detail for Multi-layer perceptrons (MLPs)

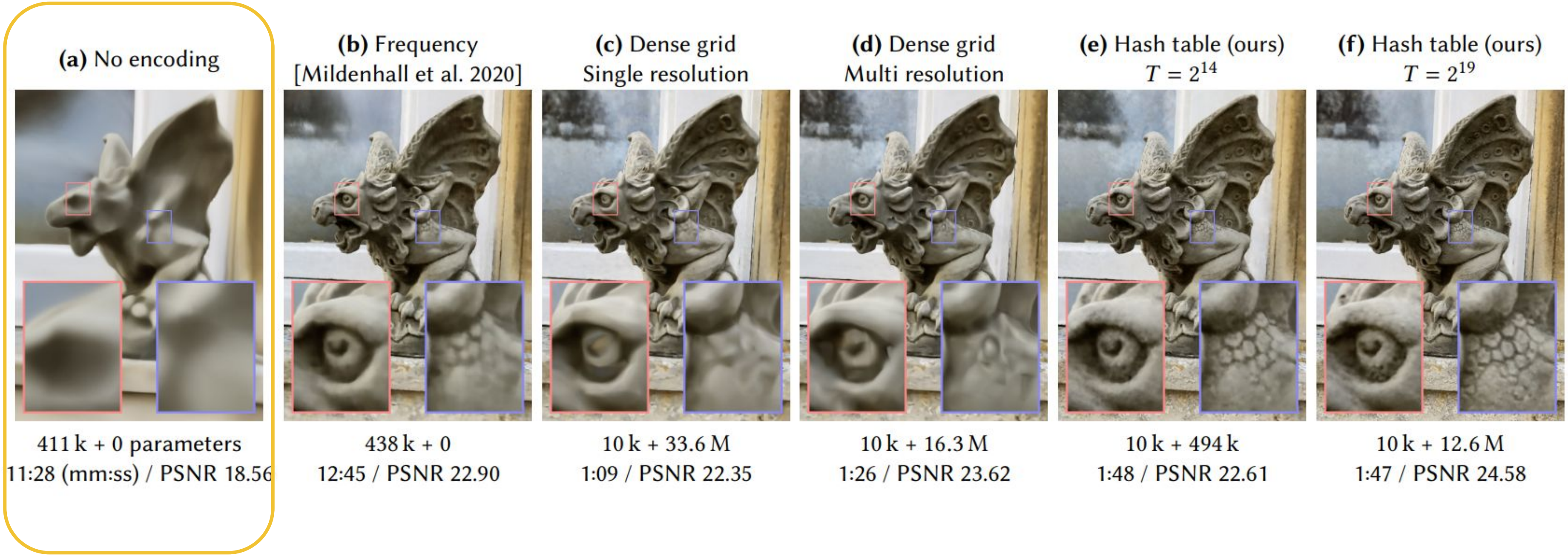
Existing Methods

- Most efficient encodings are trainable, task-specific structures.
- Rely on heuristics, may complicate training, and may limit performance on GPUs.

Proposed method:

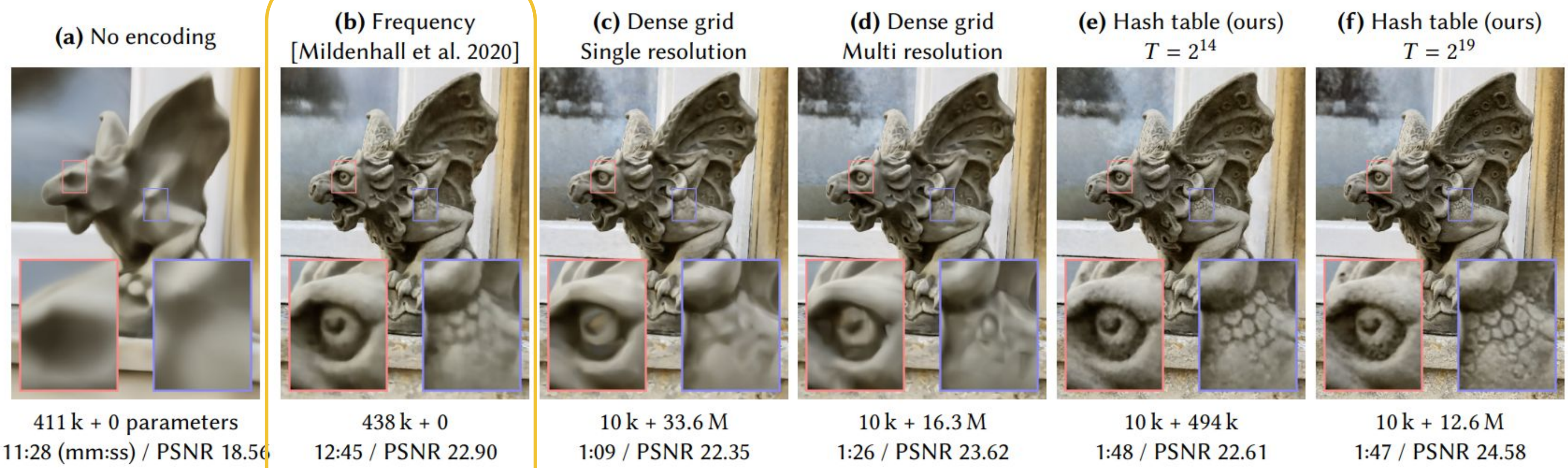
- Adaptive, efficient, and task-independent.
- Achieves top-tier quality across tasks after brief training.
- Adaptivity: No structural updates needed during training.
- Efficiency: Hash table lookups are O(1), and allows parallel querying for all resolutions.

Existing Methods



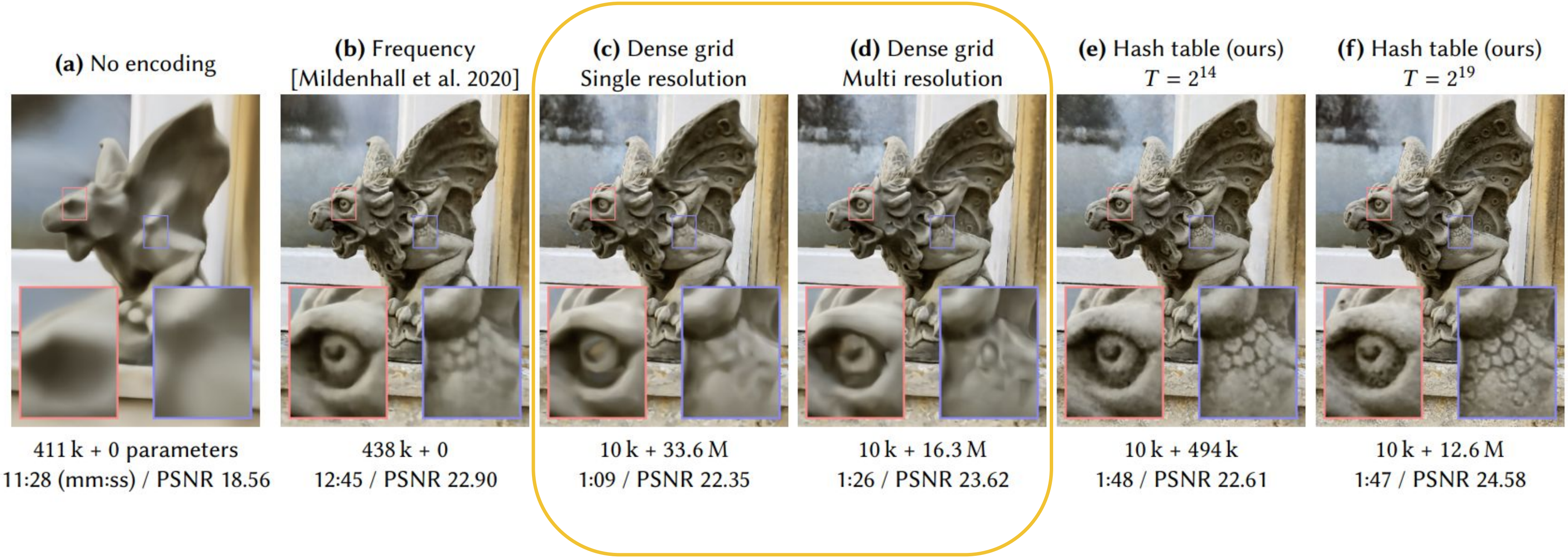
Without encoding, the reconstruction is blurry.

Existing Methods: Frequency Encoding



$$\mathrm{enc}(x) = \big(\sin(2^0 x), \sin(2^1 x), \ldots, \sin(2^{L-1} x),$$
$$\cos(2^0 x), \cos(2^1 x), \ldots, \cos(2^{L-1} x)\big).$$

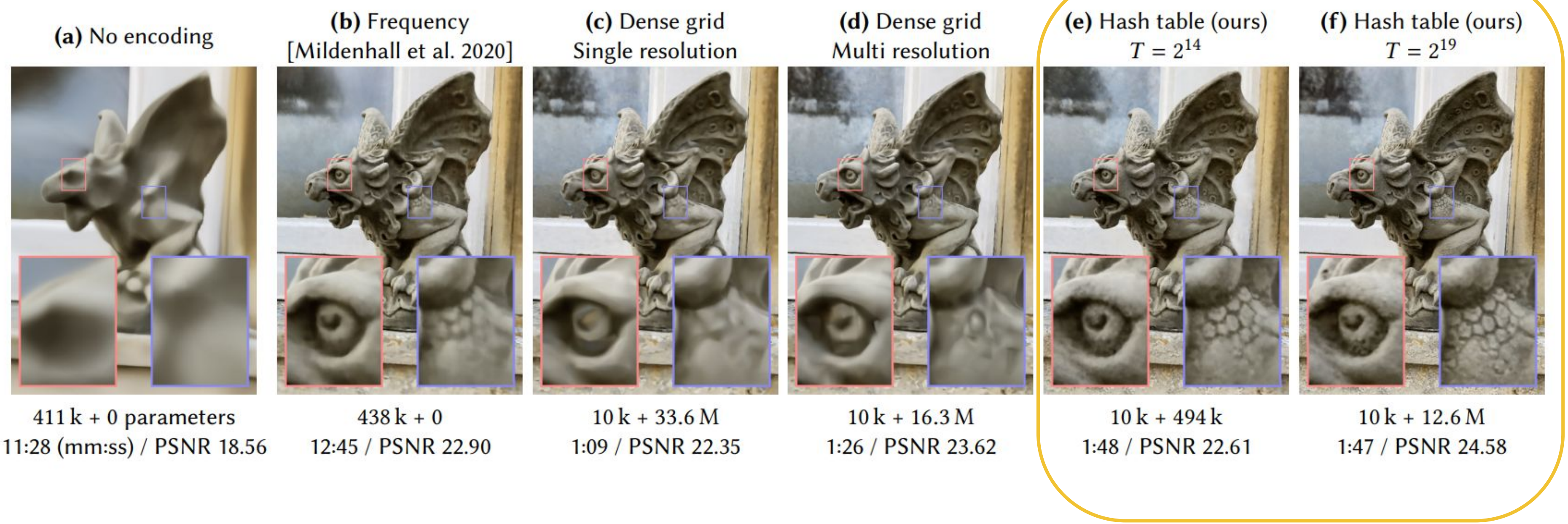
Existing Methods: Parametric Encoding



Parametric Encoding (Dense Grid):

- Computationally efficient: few feature vectors are updated per sample
- Memory waste: too many features in **empty** space

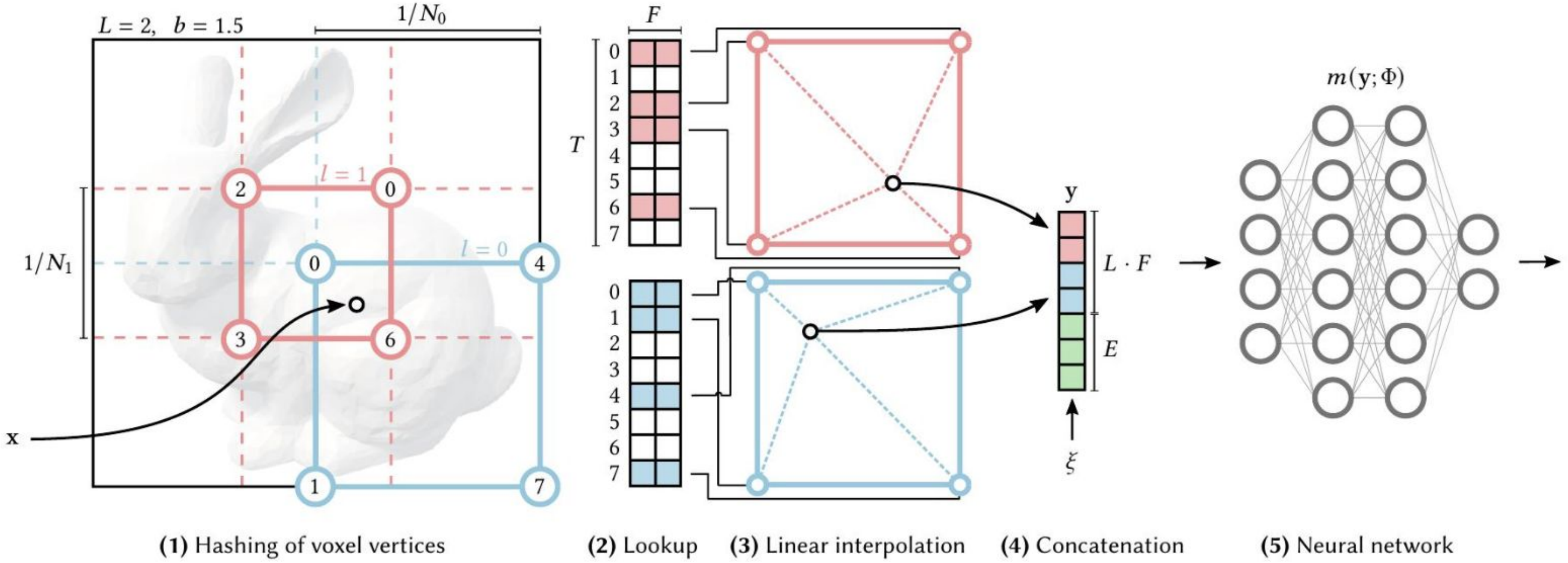
Multiresolution Hash Encoding



Parametric Encoding (Hash Table)

- Decrease

Method - Overview



Hash Encoding: Multi-Resolution

Neural network $m(\mathbf{y}; \Phi)$ encodes inputs as $\mathbf{y} = \mathbf{enc}(\mathbf{x}; \theta)$ with trainable weights Φ and encoding parameters θ.

θ is split into L levels with up to T feature vectors of dimension F.

Grid resolutions range between $[N_{\min}, N_{\max}]$ and are computed as:

$$N_l := \left\lfloor N_{\min} \cdot b^l \right\rfloor,$$
$$b := \exp\left(\frac{\ln N_{\max} - \ln N_{\min}}{L - 1}\right).$$

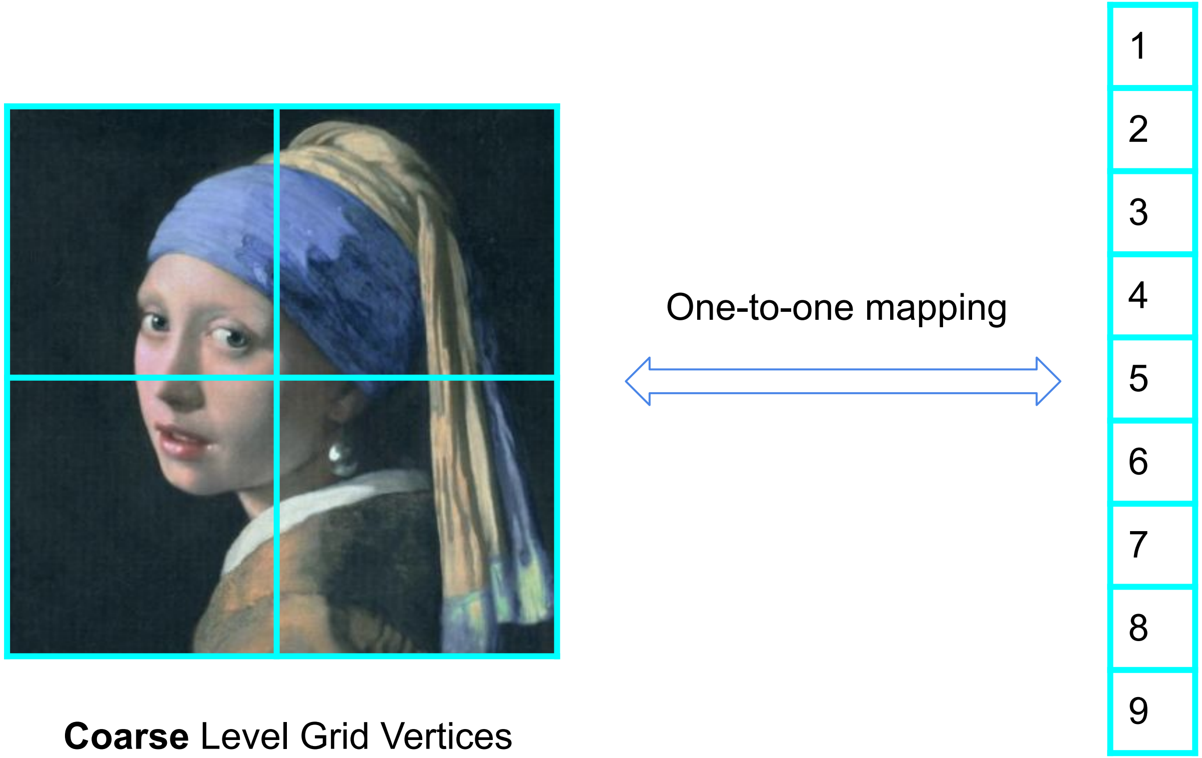
Hash Encoding: Hash Function

For each level l, input $\mathbf{x} \in \mathbb{R}^d$ is scaled to get $\lfloor \mathbf{x}_l \rfloor$ and $\lceil \mathbf{x}_l \rceil$, which is associated with a voxel in $\mathbb{Z}^d$. For coarse levels, mapping is **unique**. At finer levels, a hash function $h : \mathbb{Z}^d \to \mathbb{Z}_T$ is employed. The hash function is defined as:

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

π_i are some large prime numbers. For each of the L levels, interpolated vectors are merged with auxiliary inputs (view direction or textures) $\xi \in \mathbb{R}^E$ to form $\mathbf{y} \in \mathbb{R}^{LF+E}$.

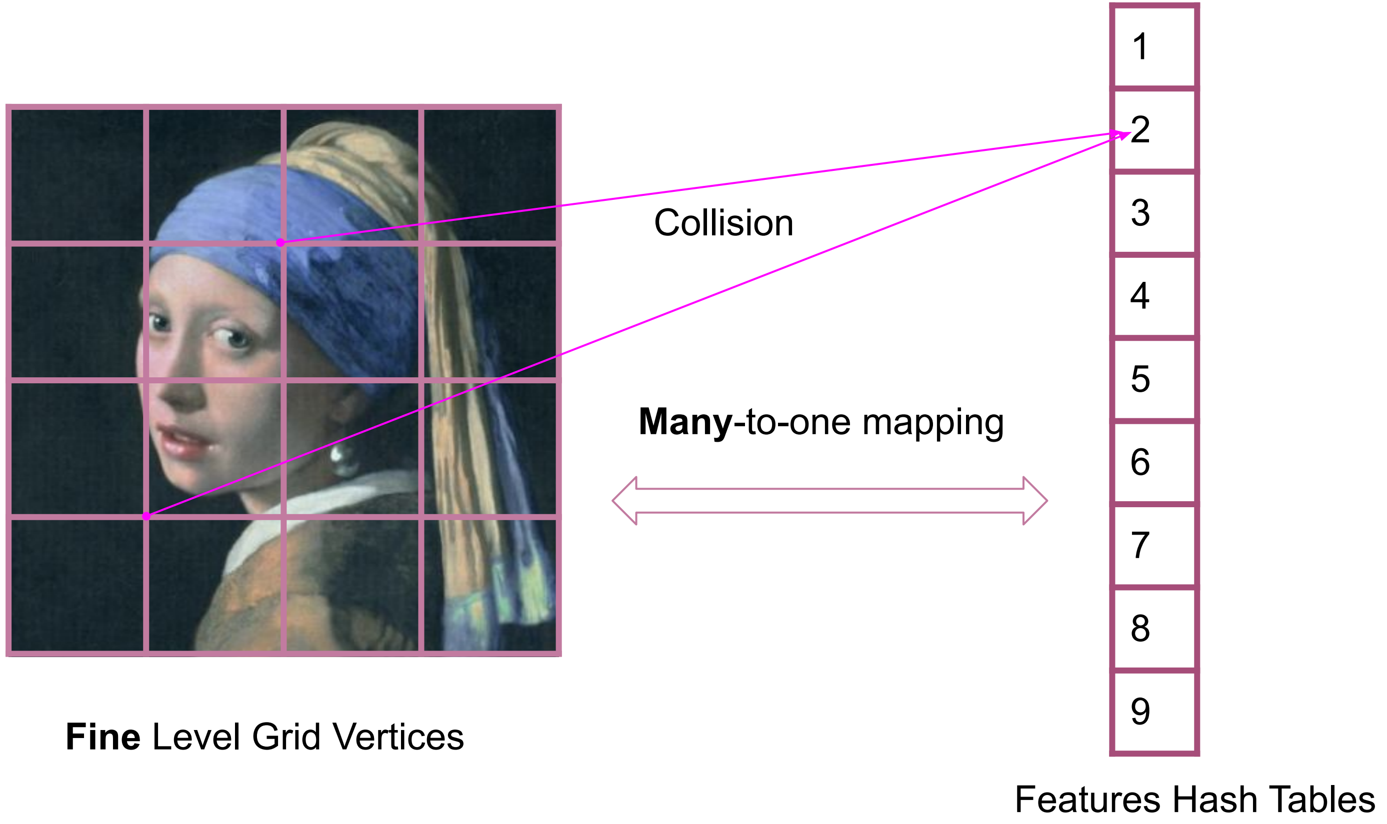
Hash Encoding: Collision Resistance



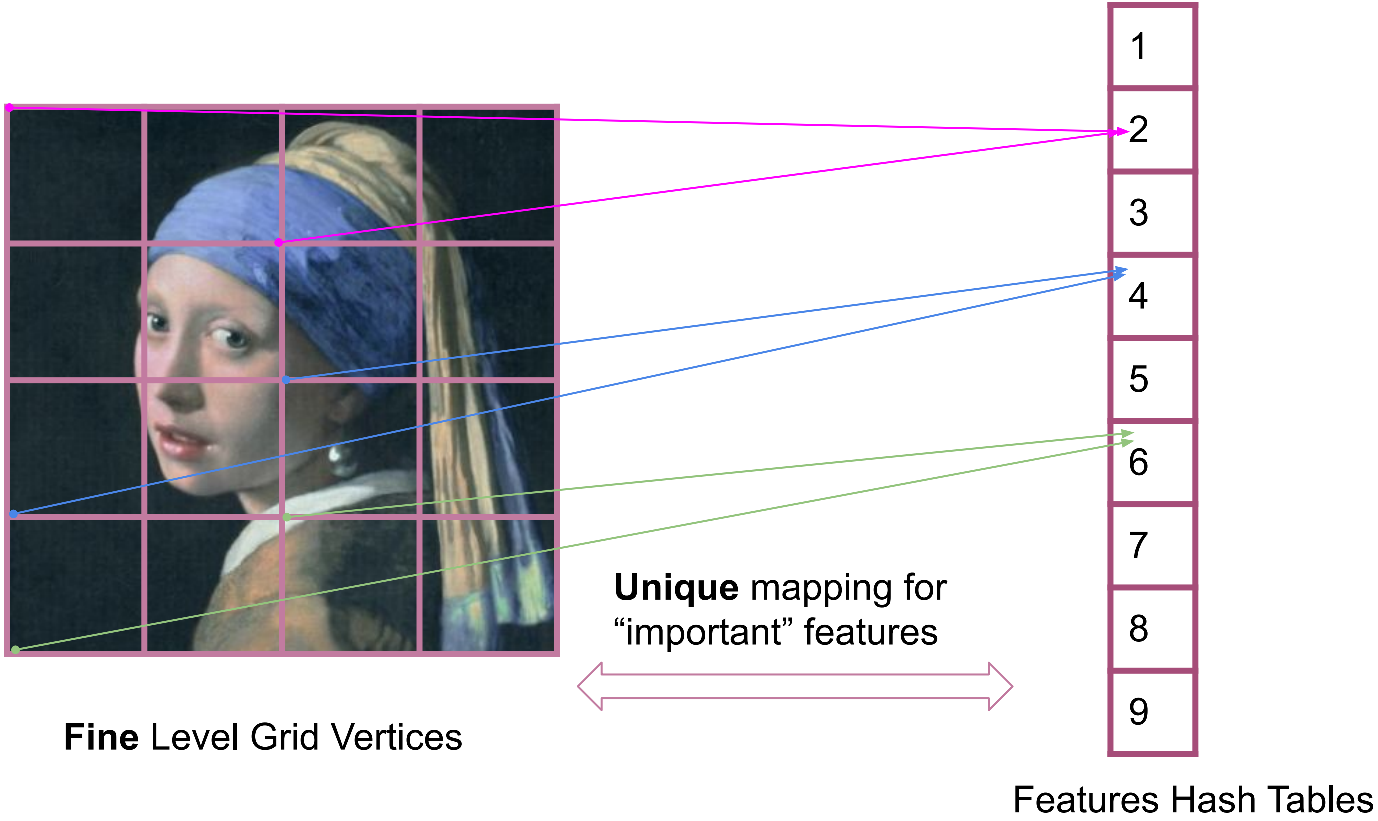
Coarse Level Grid Vertices

Features Hash Tables

Hash Encoding: Collision Resistance



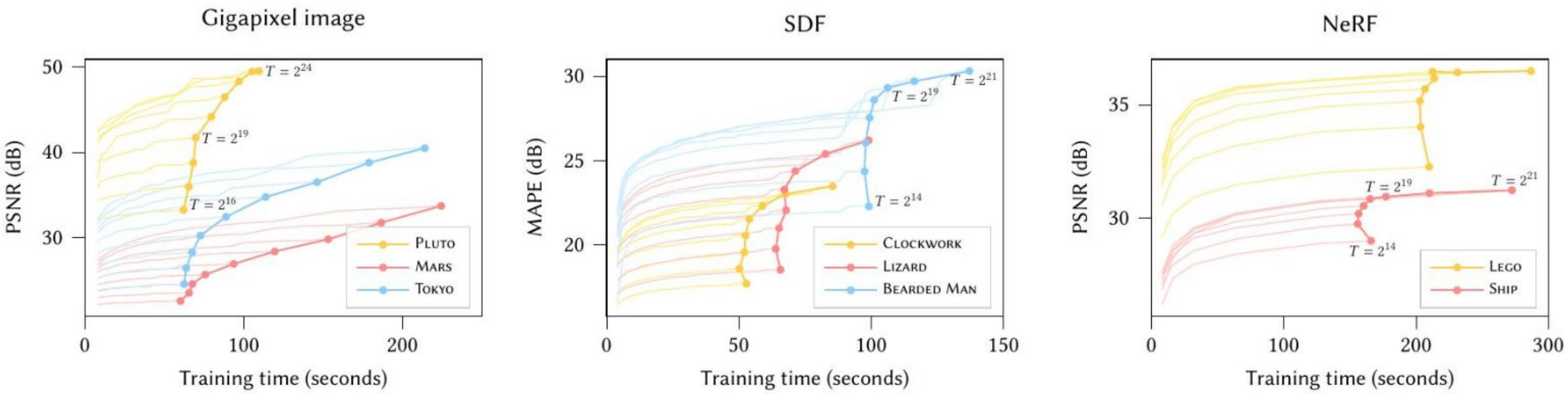
Hash Encoding: Learnable Indices?



Experiments

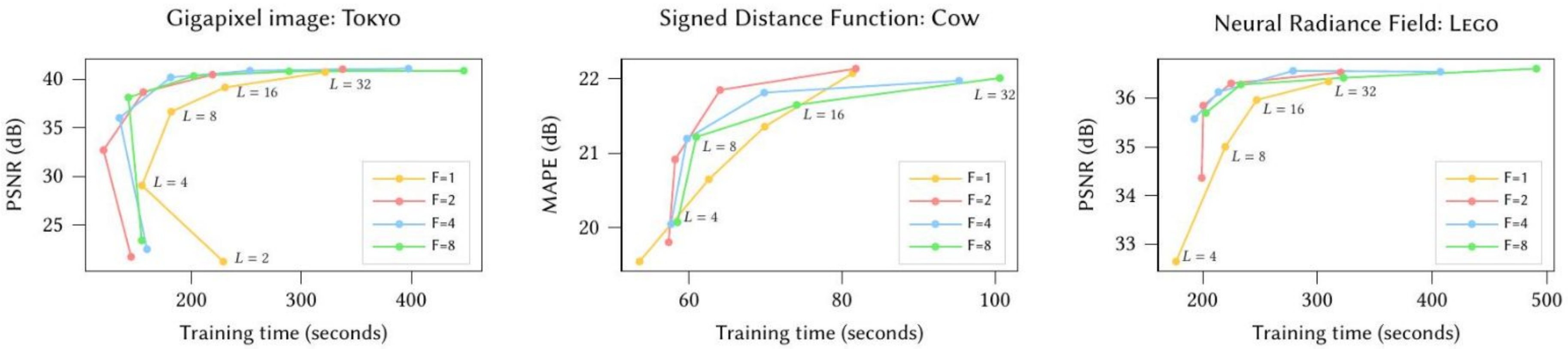
(1) **Gigapixel image:** the MLP learns the mapping from 2D coordinates to RGB colors of a high-resolution image.
(2) **Neural signed distance functions (SDF):** the MLP learns the mapping from 3D coordinates to the distance to a surface.
(3) **Neural radiance caching (NRC):** the MLP learns the 5D light field of a given scene from a Monte Carlo path tracer.
(4) **Neural radiance and density fields (NeRF):** the MLP learns the 3D density and 5D light field of a given scene from image observations and corresponding perspective transforms.

Hyperparameters



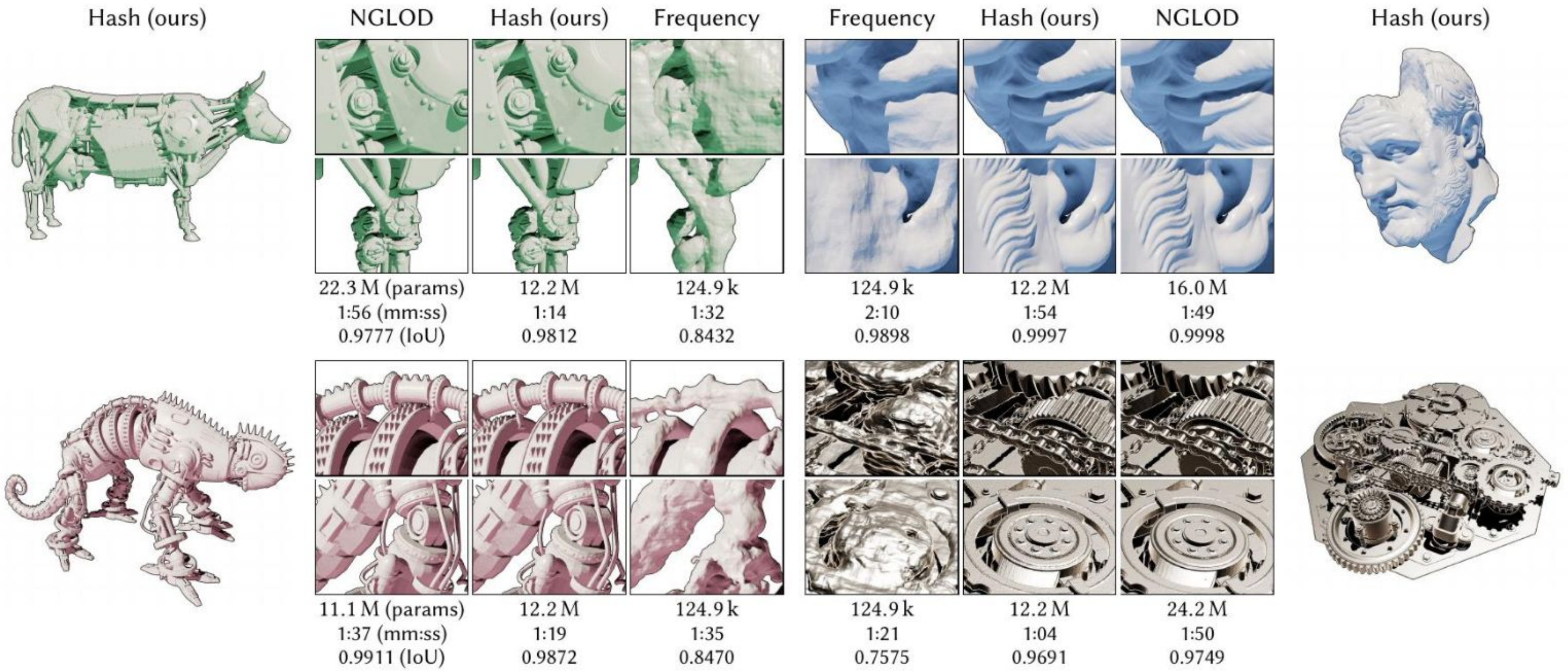
Varies Hash Table size **T**

Hyperparameters



Varies Feature Dimensionality **F**

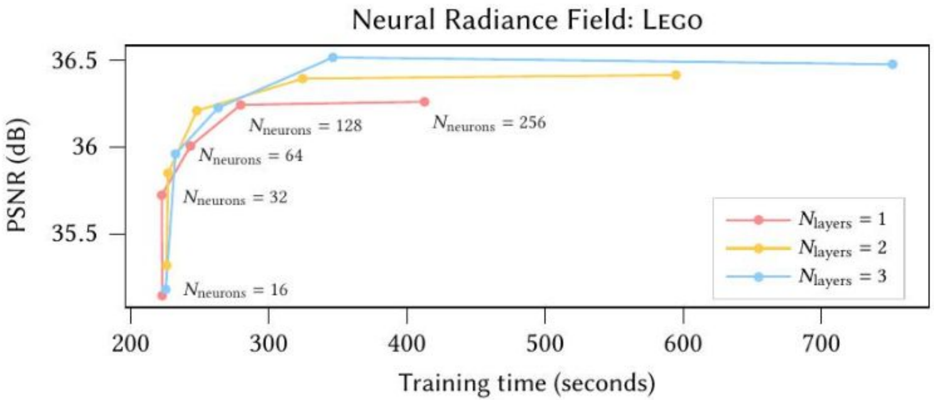
Neural SDF


Hash (ours)
NGLOD
Hash (ours)
Frequency
Frequency
Hash (ours)
NGLOD
Hash (ours)
22.3 M (params)
1:56 (mm:ss)
0.9777 (IoU)
12.2 M
1:14
0.9812
124.9 k
1:32
0.8432
124.9 k
2:10
0.9898
12.2 M
1:54
0.9997
16.0 M
1:49
0.9998
11.1 M (params)
1:37 (mm:ss)
0.9911 (IoU)
12.2 M
1:19
0.9872
124.9 k
1:35
0.8470
124.9 k
1:21
0.7575
12.2 M
1:04
0.9691
24.2 M
1:50
0.9749

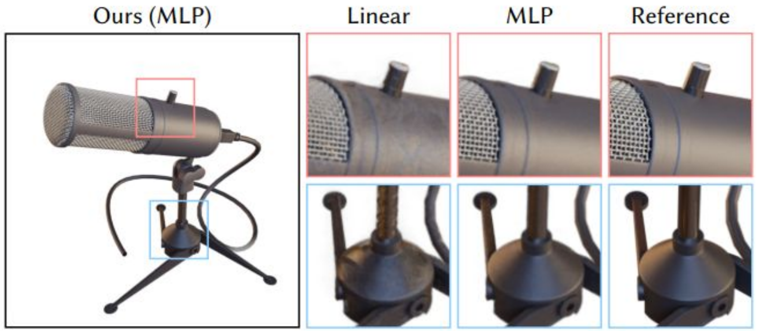
NeRF

	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

Size of the MLPs



Hash Encoding enables a small MLP



Linear model v.s. MLP

References

Instant-NGP: https://nvlabs.github.io/instant-ngp/assets/mueller2022instant.pdf

Instant-NGP snapshot: https://docs.nerf.studio/en/latest/nerfology/methods/instant_ngp.html

NSVF: https://arxiv.org/abs/2007.11571

Neural Radiance Caching:
https://research.nvidia.com/publication/2021-06_real-time-neural-radiance-caching-path-tracing

Thank You!