

Groupe de lecture :

efunc :

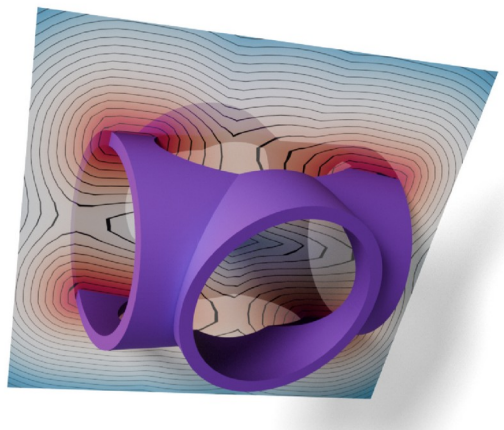
An Efficient Function Representation without Neural Networks

BIAO ZHANG, KAUST, Saudi Arabia
PETER WONKA, KAUST, Saudi Arabia

Buonomo Camille,
22/10/2025,

1. Function representation ...

Signed distance function :



Radiance field :

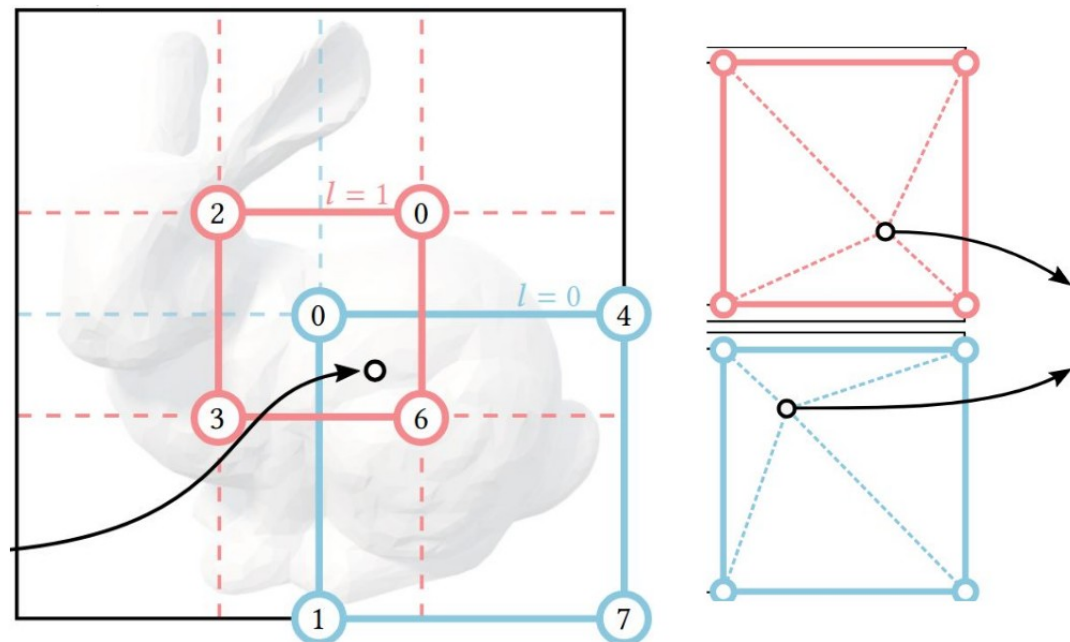
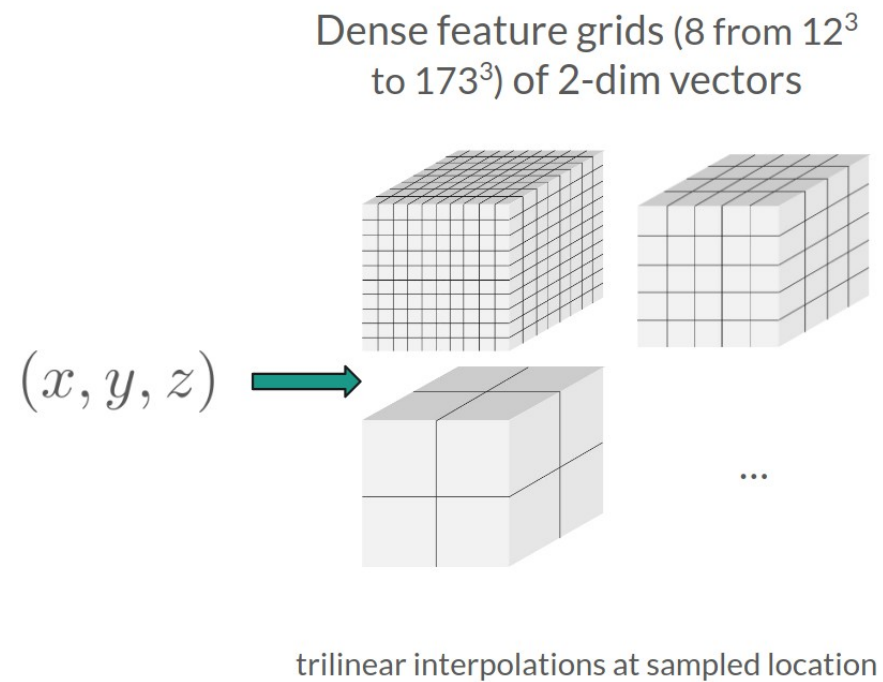


Image (compression) :

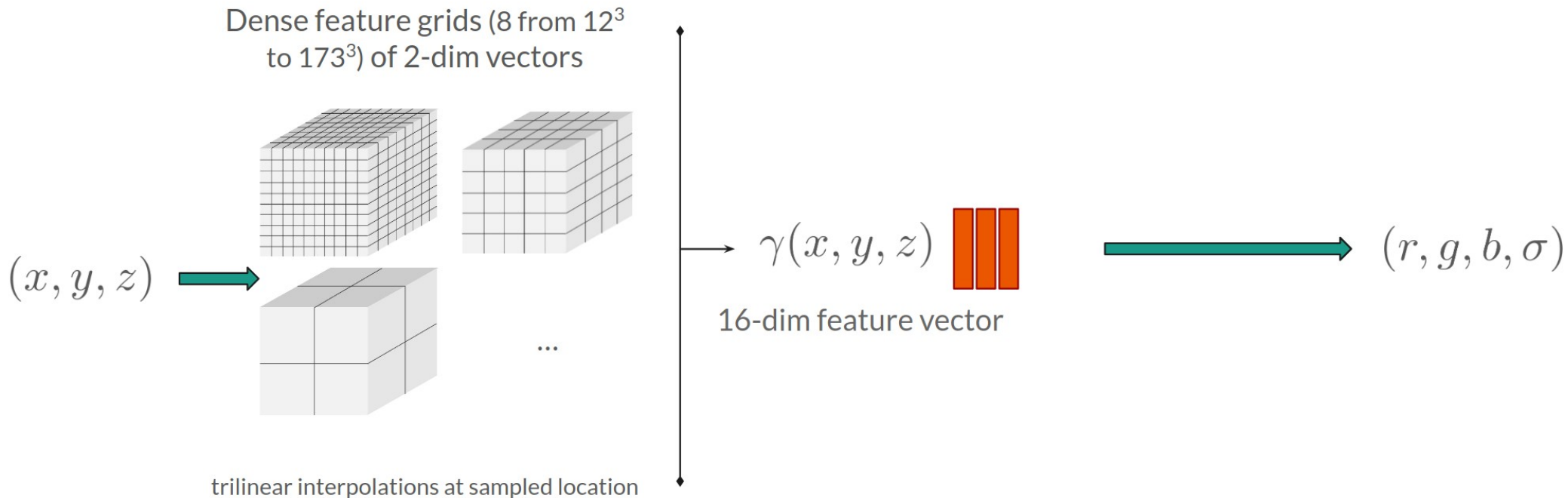


...

2. ... 'without' neural networks



2. ... 'without' neural networks



3. Generalized framework

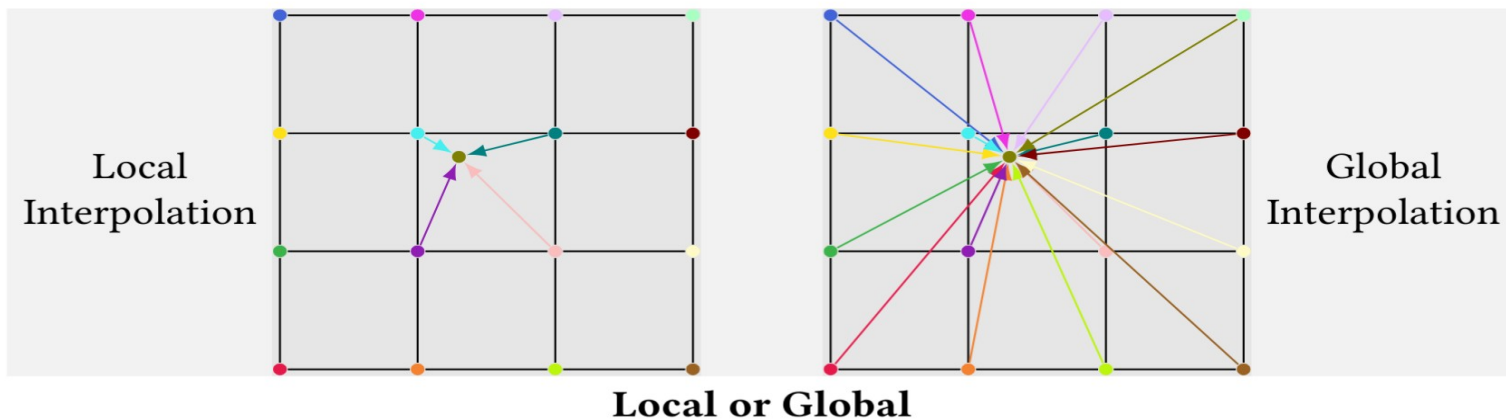
$$\mathcal{F}(\mathbf{q}) = \sum_i w(\mathbf{q}, \mathbf{k}_i) \cdot \mathbf{v}_i \quad (1)$$

k v w

- q : query
- W : weights
- K : key
- v : value

i, k, w, v ?

3. Generalized framework : Support (i) and Value (v)



$$\mathcal{F}(\mathbf{q}) = \sum_i w(\mathbf{q}, \mathbf{k}_i) \cdot \mathbf{f}(\mathbf{q}, \mathbf{k}_i; \phi_i) \quad (3)$$

3. Generalized framework : examples

Taylor series : $\mathcal{F}(q) = \sum_i \text{pow}(q, i) \cdot v_i;$

Fourier cosine series : $\mathcal{F}(q) = \sum_i \cos(2\pi i/P \cdot q) \cdot v_i;$

3. Generalized framework : Decoder

$O = \mathcal{D} \circ \mathcal{F}$	$\mathcal{D}$	$\mathcal{F}$
Type-I	Identity function $\mathcal{I}$	Eq. (1) or Eq. (3)
Type-II	Network MLP($\cdot$)	Identity function $\mathcal{I}$
Type-III	Network MLP($\cdot$)	Eq. (1)

4. efunc : Support

- Grid : efficient but expensive
- Local support : waste of information

→ Global support with gaussian kernel : $O^{\text{rbf}}(\mathbf{q}; \{\mathbf{k}_i, \beta_i, v_i\}_{i=1}^I) = \sum_i^I [\exp(-\beta_i \|\mathbf{q} - \mathbf{k}_i\|^2) \cdot v_i] .$

β_i : inverse radius of the affecting region of a key

$$\sum_i^I [\text{softmax}(-\beta_i \|\mathbf{q} - \mathbf{k}_i\|^2) \cdot v_i]$$

4. efunc : Value

- Vector are inefficient without a decoder → additional parameters and computations
- Trilinear interpolation is “not” twice differentiable

→ Use parametric function as Value

$$O(\mathbf{q}; \{\mathbf{k}_i, \beta_i, \phi_i\}_{i=1}^I) = \sum_i^I [\text{softmax}(-\beta_i \|\mathbf{q} - \mathbf{k}_i\|^2) f(\mathbf{q}, \mathbf{k}_i; \phi_i)]$$

Shift invariance : $f(\mathbf{q} - \mathbf{k}_i; \phi_i)$

4. efunc : Choice of f

Universality theorem :

- MLP : Too heavy and slow
- Polynomial function !!!

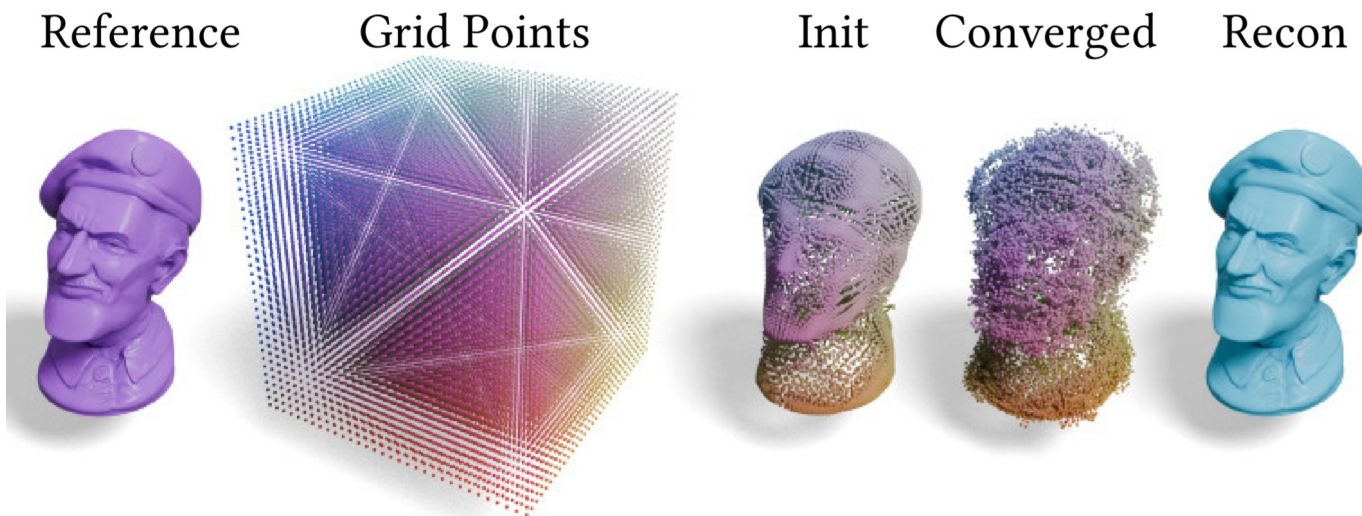
$$f(\mathbf{x}; \boldsymbol{\phi}) = \circ + \mathbf{x}^T \diamond + \frac{1}{2} \mathbf{x}^T \square \mathbf{x} + \dots$$

- 0 order : Value are vector
- K order : k time differentiable

5. Extension : offset

Idea : “Learn” the best keys, not a fixed grid

→ Learn an offset vector for each key $O^{\Delta}(\mathbf{q}; \{\mathbf{k}_i + \Delta_i, \beta'_i, \phi'_i\}_{i=1}^I)$



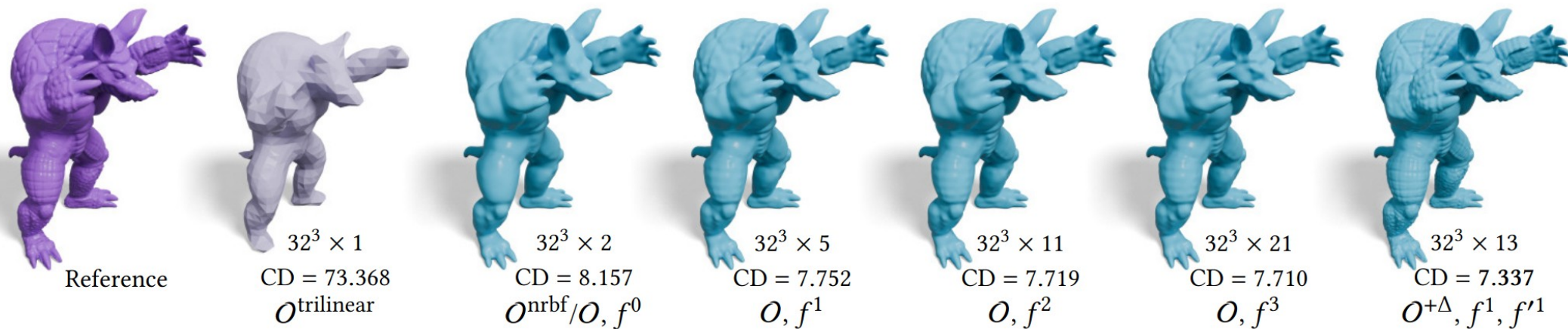
5. Extension : offset

Idea : “Learn” the best keys, not a fixed grid

→ Learn an offset vector for each key

$$\begin{aligned} & O^{+\Delta}(\mathbf{q}; \{\mathbf{k}_i, \Delta_i, \beta_i, \beta'_i, \phi_i, \phi'_i\}_{i=1}^I) \\ &= O(\mathbf{q}; \{\mathbf{k}_i, \beta_i, \phi_i\}_{i=1}^I \cup \{\mathbf{k}_i + \Delta_i, \beta'_i, \phi'_i\}_{i=1}^I) \end{aligned}$$

5. Extension : offset



6. Forward complexity

$$\mathcal{L}\left(\{(\mathbf{q}_j, o_j)\}_j^J; \boldsymbol{\theta}\right) = \frac{1}{J} \sum_j^J (O(\mathbf{q}_j) - o_j)^2$$

$$= \sum_i^I \left[\text{softmax}(-\beta_i \|\mathbf{q} - \mathbf{k}_i\|^2) f(\mathbf{q}, \mathbf{k}_i; \phi_i) \right]$$

- Batchsize = J
- Gridsize = I

→ Forward memory complexity $O(I*J)$

ALGORITHM 1: Forward

Data: $\{\mathbf{k}_i, \beta_i, \phi_i\}_i^I$ and $\mathbf{q}_j$

Result: $O(\mathbf{q}_j), e_j$

$e_j = 0;$

$m_j = 0;$

for $i \in [1, \dots, I]$ **do**

$l \leftarrow \exp(-\beta_i \|\mathbf{q}_j - \mathbf{k}_i\|^2);$

$e_j \leftarrow e_j + l;$

/* denominator */

$m_j \leftarrow m_j + l \cdot f(\mathbf{q}_j - \mathbf{k}_i; \phi_i);$

/* numerator */

end

$O(\mathbf{q}_j) = m_j / e_j;$

/* save e_j for backward */

→ $O(J)$

6. Backward complexity

$$\frac{\partial \mathcal{L}}{\partial \phi_i} = \sum_{j=1}^J \frac{\partial \mathcal{L}}{\partial O(\mathbf{q}_j)} \frac{\partial O(\mathbf{q}_j)}{\partial \phi_i}.$$

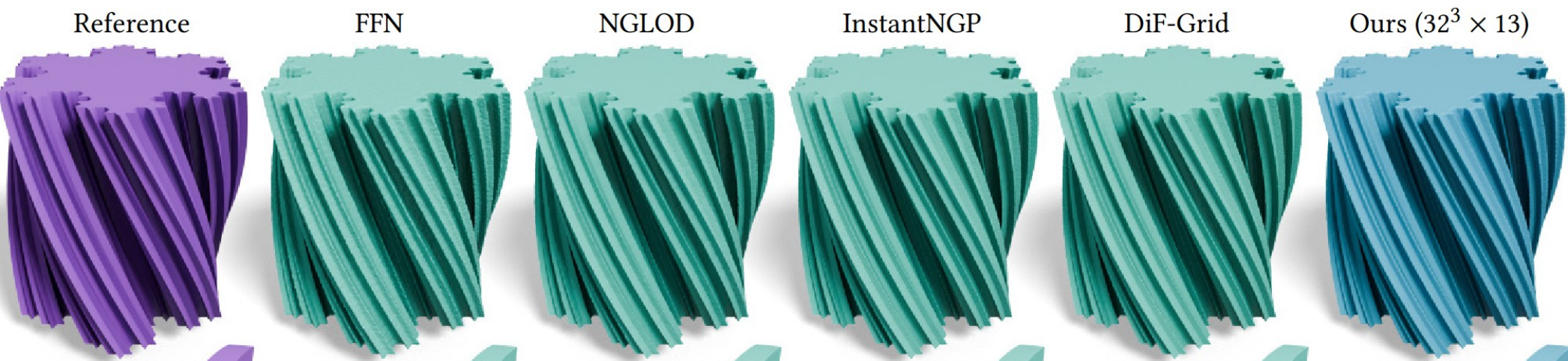
Auto-grad
(ou pas)

$$\frac{\partial O(\mathbf{q})}{\partial \phi_i} = \text{softmax}(-\beta_i \|\mathbf{q} - \mathbf{k}_i\|^2) \frac{\partial f(\mathbf{q} - \mathbf{k}_i; \phi_i)}{\partial \phi_i}.$$

Accumulation of the gradient by iterating over j : $O(I)$ memory usage

Idem for $\beta_i, \Delta_i \dots$

7. Results

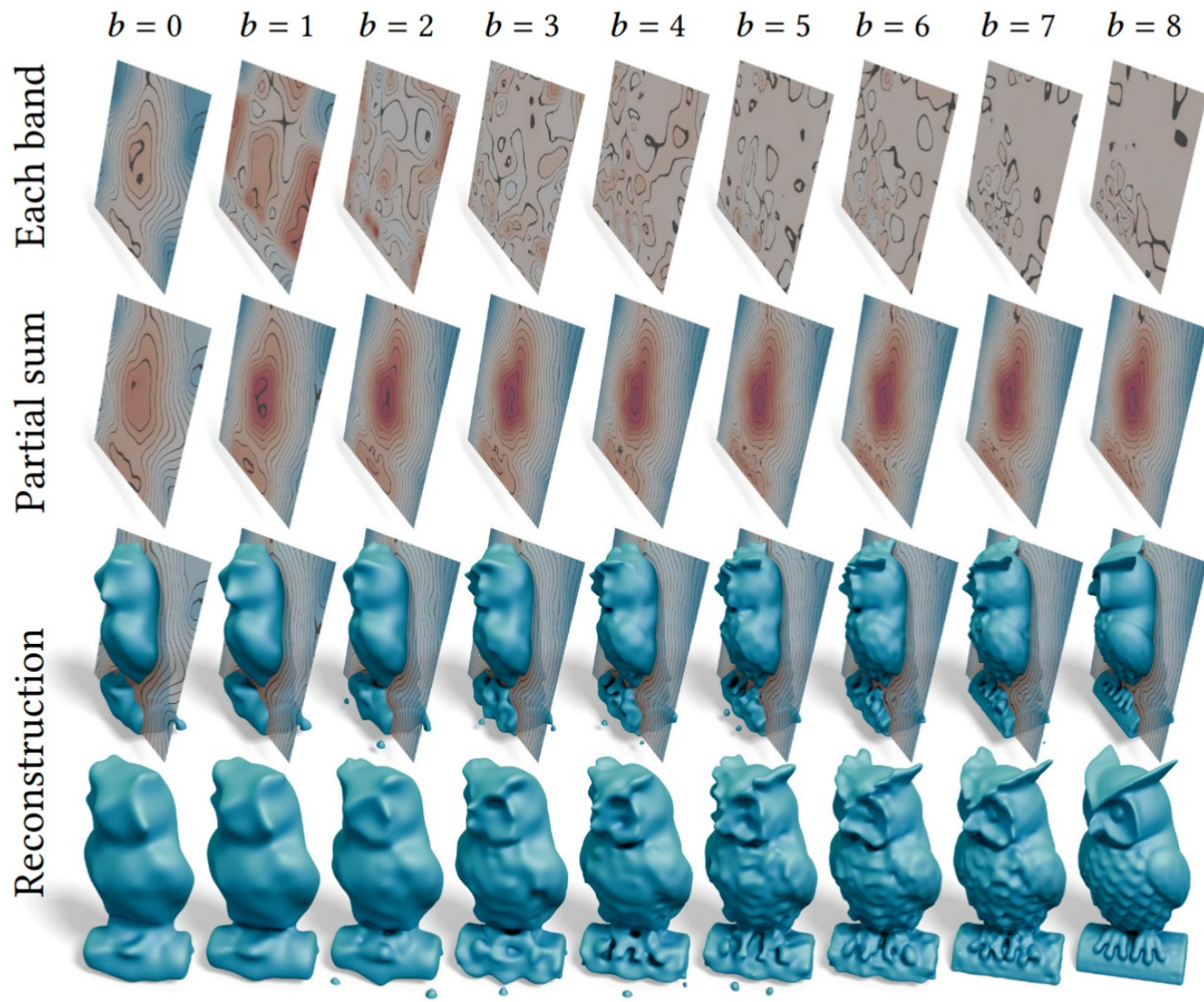


7. Results

	Type-I $\mathcal{F}(\mathbf{q})$	Type-II MLP($\mathbf{q}$)	Type-III MLP($\mathcal{F}(\mathbf{q})$)			Ours			
	Trilinear (256^3)	FFN	NGLOD	I-NGP	DiF-Grid	$4^3 \times 13$	$8^3 \times 13$	$16^3 \times 13$	$32^3 \times 13$
Data Structures	Grid	-	Octree	Hash	Grid	Grid	Grid	Grid	Grid
# Parameters	16.8m	0.46m	10.1m	14.2m	5.3m	832	6656	53k	0.43m
Non-Neural	✓	✗	✗	✗	✗	✓	✓	✓	✓
Armadillo (3.71MB)	11.964	7.625	7.420	7.354	7.352	14.253	8.850	7.622	7.336
Happy (3.76MB)	10.531	6.942	6.858	6.817	6.808	19.640	9.349	7.175	6.807
Dragon (9.65MB)	9.514	5.856	5.722	5.635	5.610	29.952	10.405	6.228	5.656
Lucy (3.71MB)	9.564	5.987	5.835	5.734	5.702	18.144	9.603	6.336	5.717
Fandisk (0.45MB)	12.847	8.581	8.493	8.470	8.461	14.398	8.990	8.505	8.410
Bed (0.06MB)	12.281	7.315	7.364	7.227	7.213	31.442	11.084	8.190	7.232
Building (13.67MB)	17.456	12.715	12.710	12.676	12.717	31.286	15.497	12.804	12.685
Container (1.72MB)	20.118	14.069	13.712	14.411	15.151	22.704	16.194	14.292	13.976
Snowflake (0.22MB)	19.572	15.420	15.343	15.268	15.283	38.916	23.407	15.776	15.223
Mean	13.761	9.390	9.273	9.288	9.366	24.526	12.598	9.659	9.227

7. Function Analysis

$$\mathcal{S}(\mathbf{q}) = \sum_{b=0}^B \cos(b \cdot \pi \cdot \mathbf{q}) \cdot O_b(\mathbf{q}) \leftarrow \sum_i^I [\text{softmax}(-\beta_i \|\mathbf{q} - \mathbf{k}_i\|^2) \cdot v_i]$$



Gradients of Queries. The gradient of $O(\mathbf{q})$ with respect to $\mathbf{q}$ can be derived as,

$$\frac{\partial O(\mathbf{q})}{\partial \mathbf{q}} = \sum_i^I \left[\text{softmax} \left(-\beta_i \|\mathbf{q} - \mathbf{k}_i\|^2 \right) g(\mathbf{q} - \mathbf{k}_i) \right] \quad (13)$$

where

$$g(\mathbf{q} - \mathbf{k}_i) = \frac{\partial f(\mathbf{q} - \mathbf{k}_i; \phi_i)}{\partial \mathbf{q}} + 2 \cdot \beta_i \cdot (\mathbf{q} - \mathbf{k}_i) \cdot (O(\mathbf{q}) - f(\mathbf{q} - \mathbf{k}_i; \phi_i)).$$

$$\Delta_i = \frac{\sum_n^N \left[\exp \left(-100 \|\mathbf{k}_i - \mathbf{s}_n\|^2 \right) \cdot \mathbf{s}_n \right]}{\sum_n^N \exp \left(-100 \|\mathbf{k}_i - \mathbf{s}_n\|^2 \right)} - \mathbf{k}_i,$$