How Factorization Improves NeRF

D-NeRF and FastNeRF

May 18, 2022

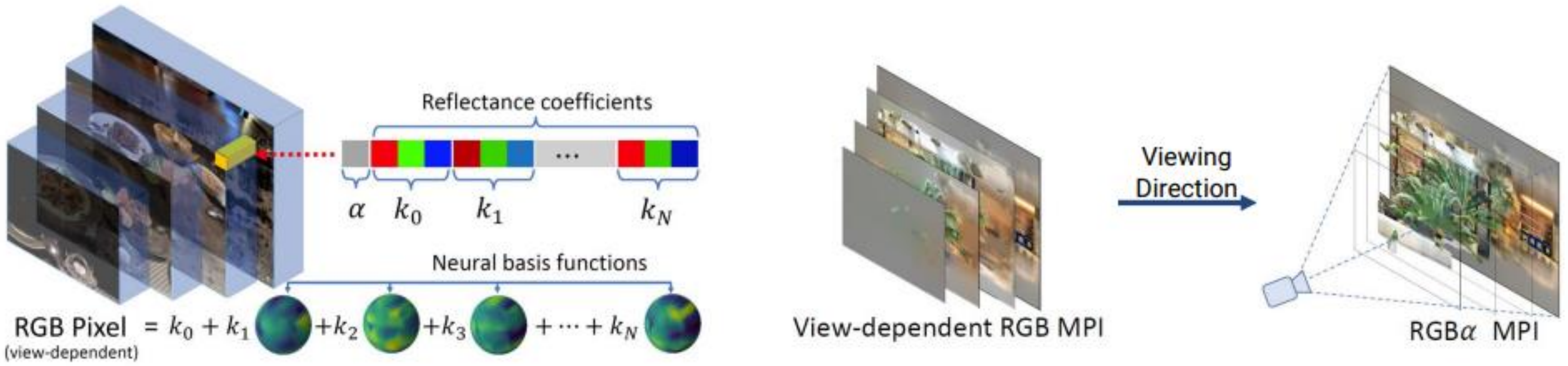
Seokhyeon Hong

Contents

Recap

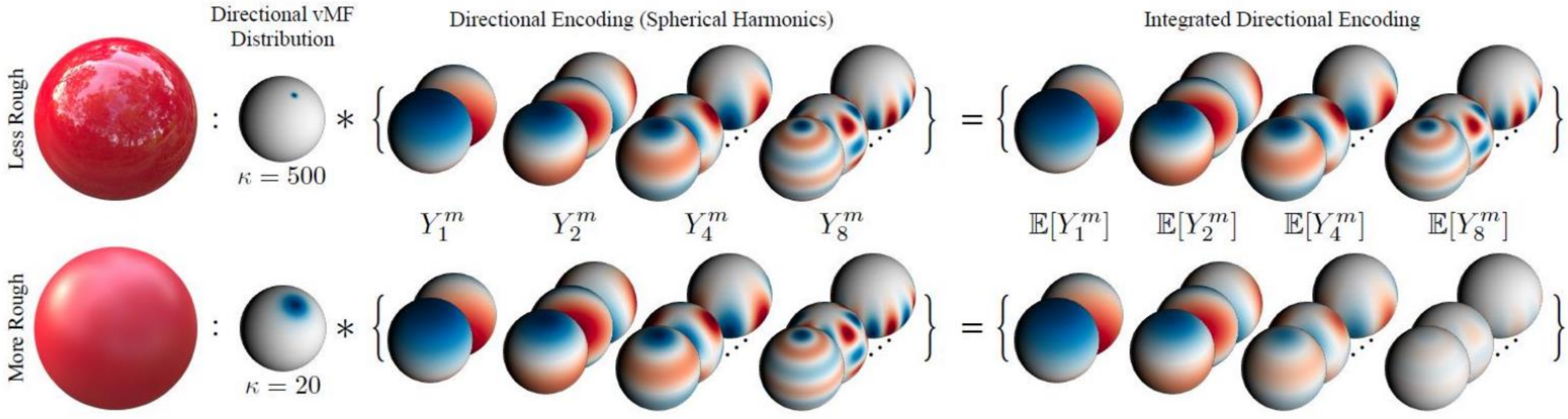
Recap

- **NeX**
  - Adding view-dependency in MPI
    - Reflectance coefficients and neural basis functions



Recap

- **Ref-NeRF**
  - Improved view-dependency in NeRF
    - Integrated directional encoding (IDE)



Recap

▪ Summary

- Enhanced view-dependency

D-NeRF: Neural Radiance Fields for Dynamic Scenes

[Pumarola et al. CVPR 2021]

D-NeRF

- **Limitations of NeRF**
  - Training time
  - Inference time
  - Scalability
  - Camera calibration
  - Bounded scenes
  - Static scenes

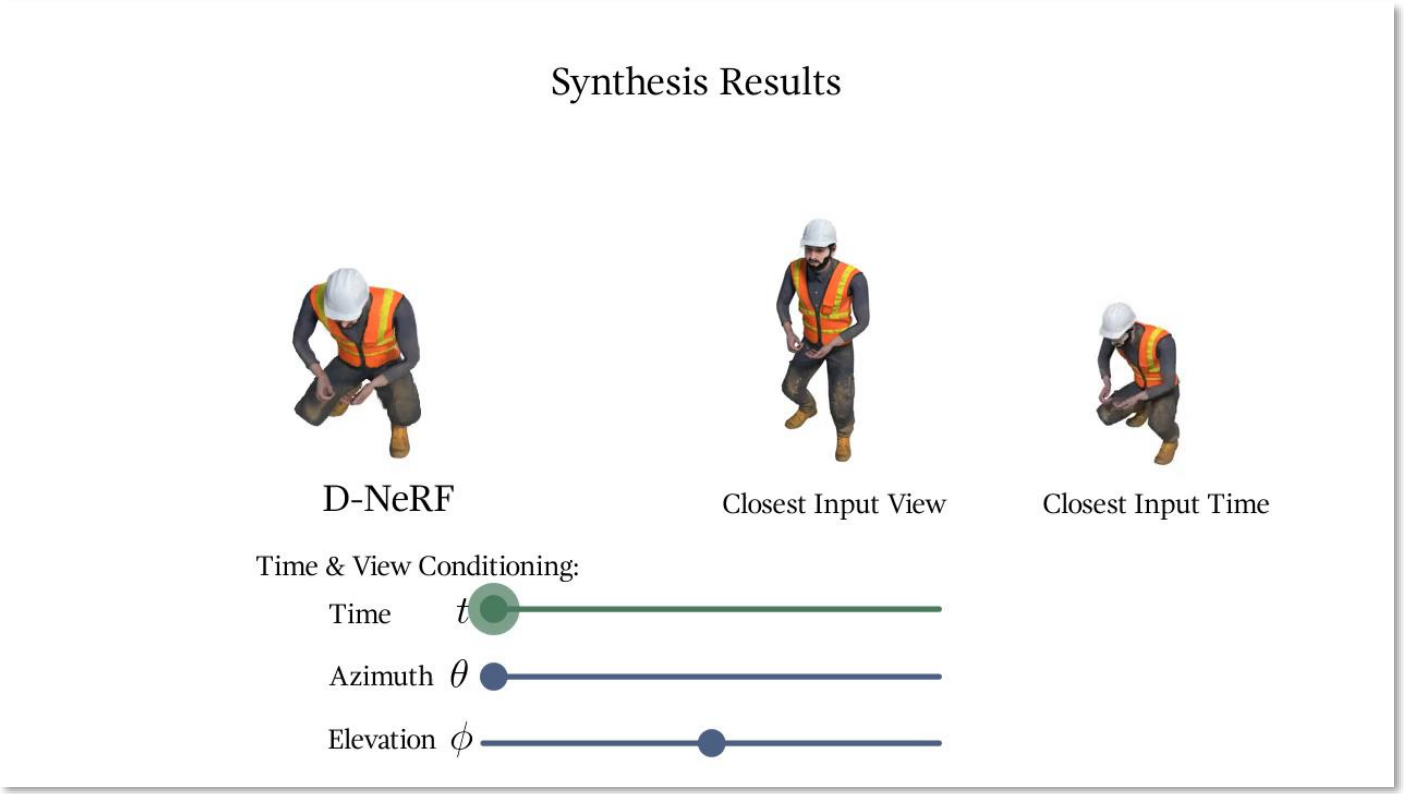
D-NeRF

- **Limitations of NeRF**
  - Training time
  - Inference time
  - Scalability
  - Camera calibration
  - Bounded scenes
  - Static scenes

D-NeRF

▪ Purpose

- NeRF in a dynamic domain

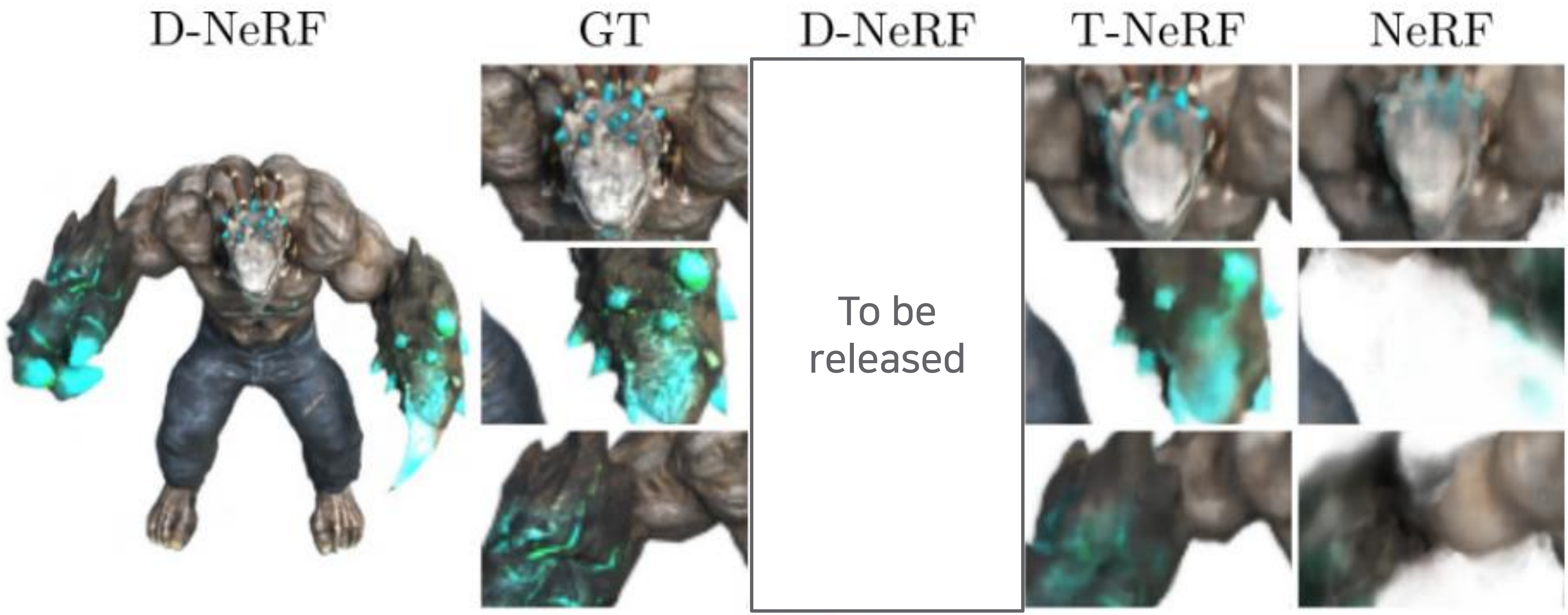


D-NeRF

▪ Purpose

- NeRF in a dynamic domain
  - Original NeRF: $(x, y, z, \theta, \phi) \rightarrow (r, g, b, \sigma)$
  - Naïve approach: $(x, y, z, \theta, \phi, t) \rightarrow (r, g, b, \sigma)$
    - Called as T-NeRF

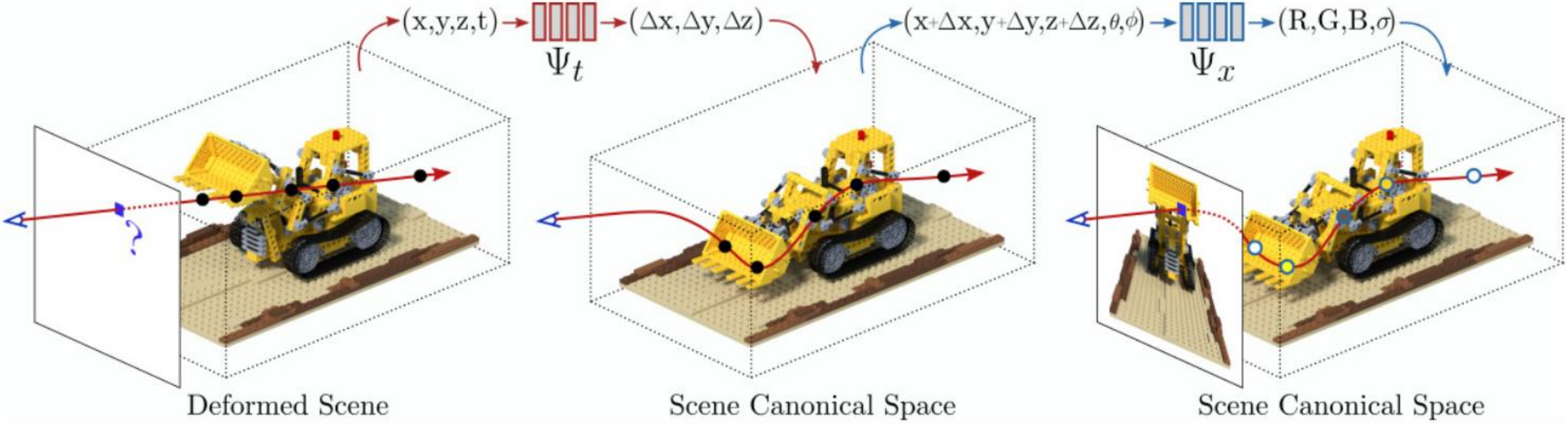
D-NeRF



D-NeRF

▪ Main Idea

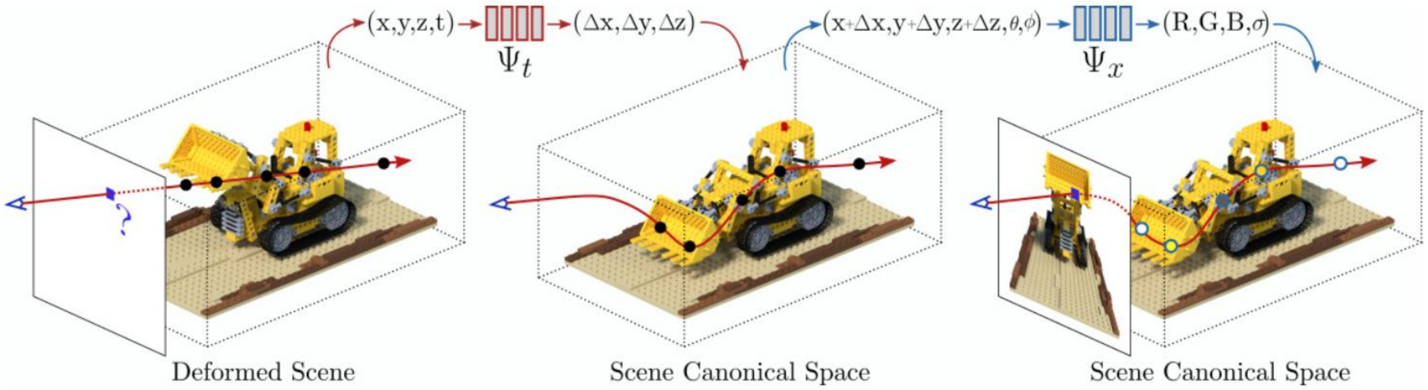
- Factorization



D-NeRF

▪ Main Idea

- Factorization
  - Deformation network Ψ_{def}
    - › Deformation field of a specific time instant with respect to the canonical space
  - Canonical network Ψ_{can}
    - › Color and density given a point and a direction



D-NeRF

- **Constraints**
  - Objects
    - Movable and deformable
    - NOT allowed to appear or disappear
  - Camera
    - Only a single camera is used

D-NeRF

▪ Deformation Network

- Formulation
  - $\Psi_{\mathrm{def}}(\mathbf{x}, t) = \begin{cases} \mathbf{\Delta x} & \text{if } t \neq 0 \\ 0 & \text{if } t = 0 \end{cases}$
  - Positional encoding $\gamma(p) =< \left(\sin(2^l \pi p), \cos(2^l \pi p)\right) >_0^L$
    - › $L = 10$ for $\mathbf{x}$
    - › $L = 4$ for $\mathbf{d}$ and t

D-NeRF

- **Canonical Network**
  - Formulation
    - $\Psi_{\text{can}}(\mathbf{x} + \Delta\mathbf{x}, \mathbf{d}) = (\mathbf{c}, \sigma)$

D-NeRF

Volume Rendering

- NeRF's volume rendering equation

$$C(p) = \int_{h_n}^{h_f} T(h,t)\sigma\big(\mathbf{x}(h)\big)\mathbf{c}(\mathbf{x}(h),\mathbf{d})dh\,,$$

$$\text{where } T(h,t) = \exp\left(-\int_{h_n}^{h}\sigma\big(\mathbf{x}(s)\big)ds\right) \text{ and } \mathbf{x}(h) = \mathbf{o} + h\mathbf{d}$$

D-NeRF

▪ Volume Rendering

- D-NeRF's volume rendering equation
  - Just the time parameter t is added

$$C(p,t) = \int_{h_n}^{h_f} T(h,t)\sigma\big(\mathbf{p}(h,t)\big)\mathbf{c}(\mathbf{p}(h,t),\mathbf{d})dh\,,$$

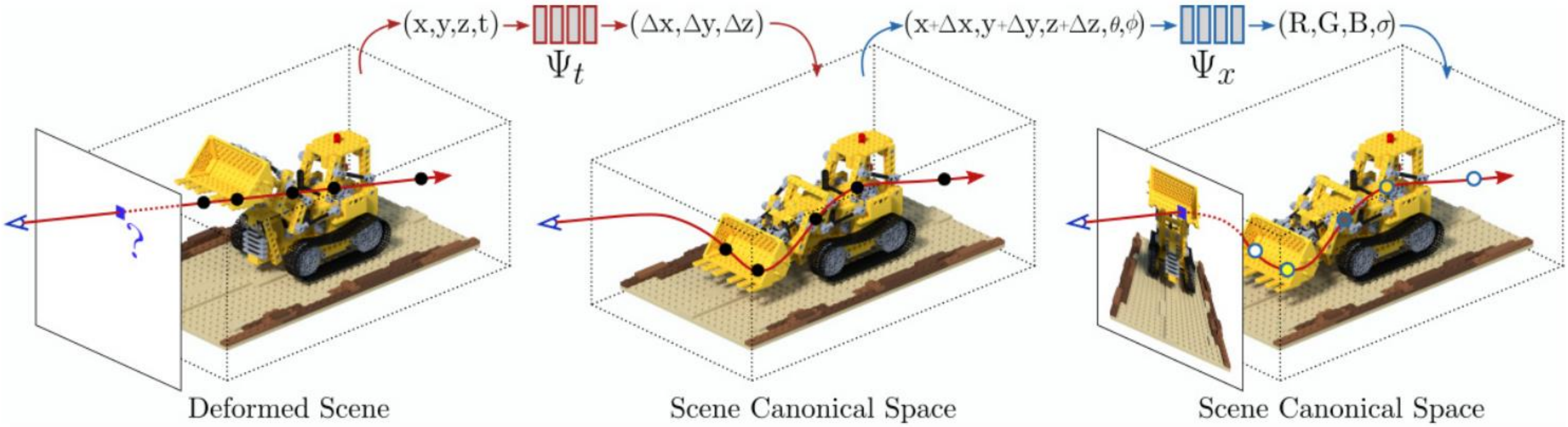
Deformation field

$$\text{where } T(h,t) = \exp\left(-\int_{h_n}^{h} \sigma\big(\mathbf{p}(s,t)\big)ds\right) \text{ and } \mathbf{p}(h,t) = \mathbf{x}(h) + \Psi_t(\mathbf{x}(h),t)$$

Ray point
in the canonical scene

D-NeRF

- **Volume Rendering**
  - Recap the overview



D-NeRF

- **Network**
  - MLP
    - Ψ_{def} and Ψ_{can} consist of 8-layer MLPs with ReLU activation
  - L2 loss
    - MSE between the rendered and real pixels

Synthesis Results

D-NeRF

Closest Input View

Closest Input Time

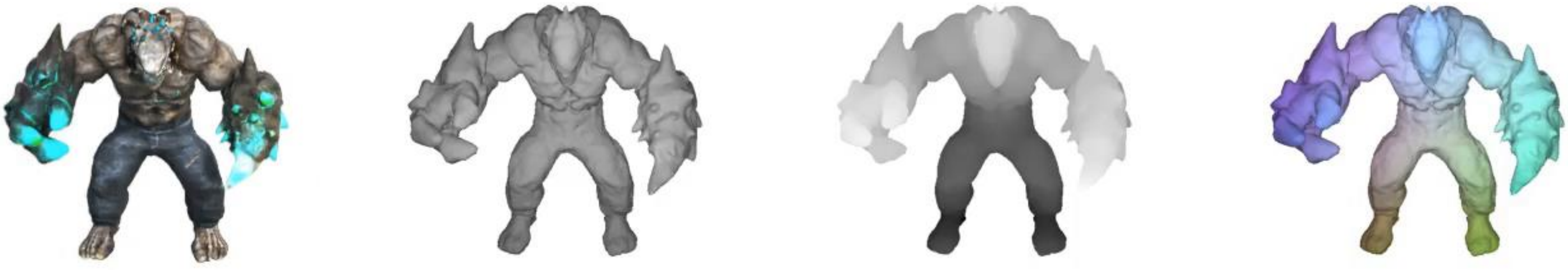
Time & View Conditioning:

Time t

Azimuth θ

Elevation ϕ

Visualization of the Learned Scene Representation



D-NeRF Radiance (as RGB)

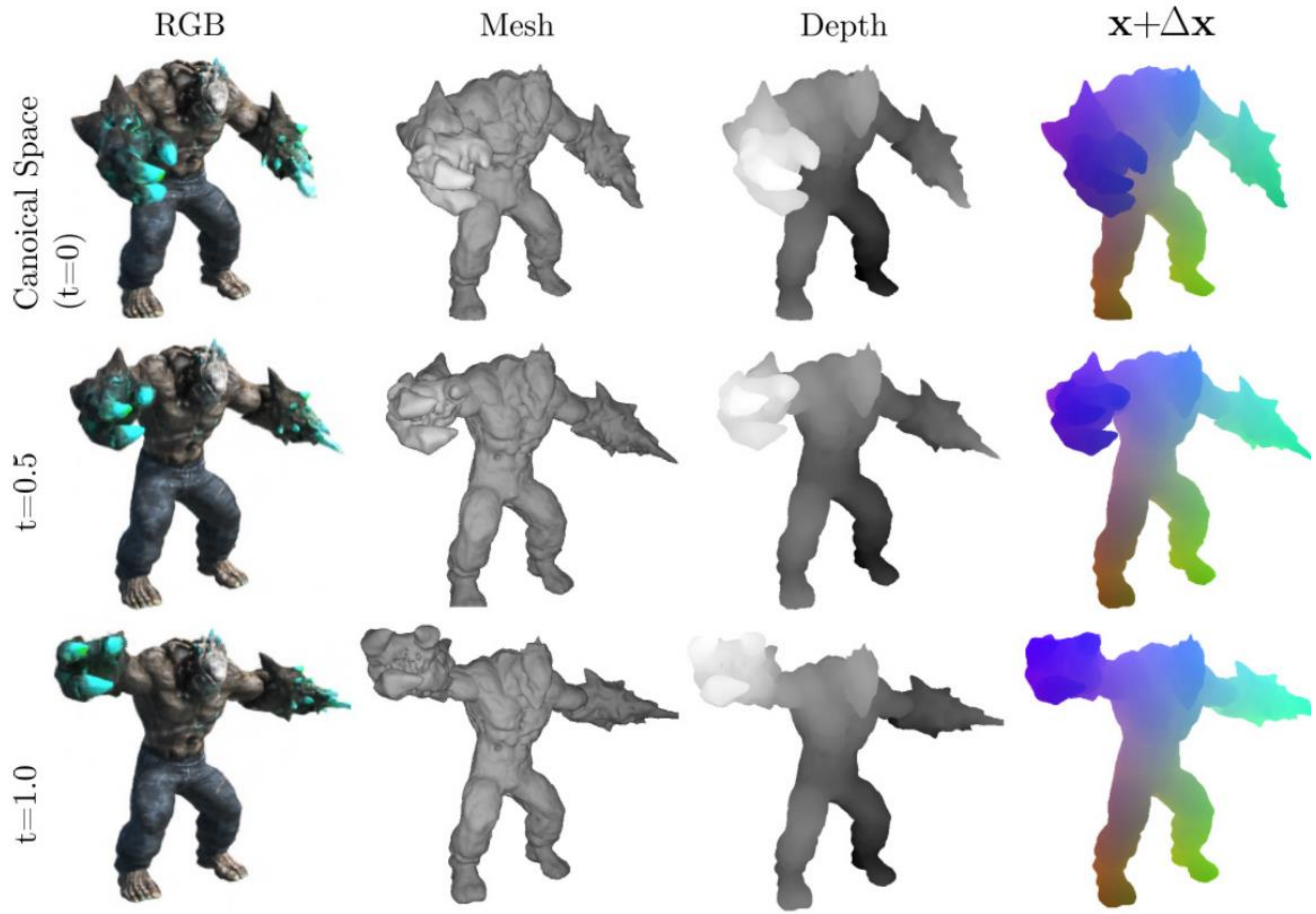
D-NeRF Volume Density (as Mesh) (as Depth)

D-NeRF Canonical Mapping (color-coded as $\mathbf{x} + \Delta\mathbf{x}$)

Time t

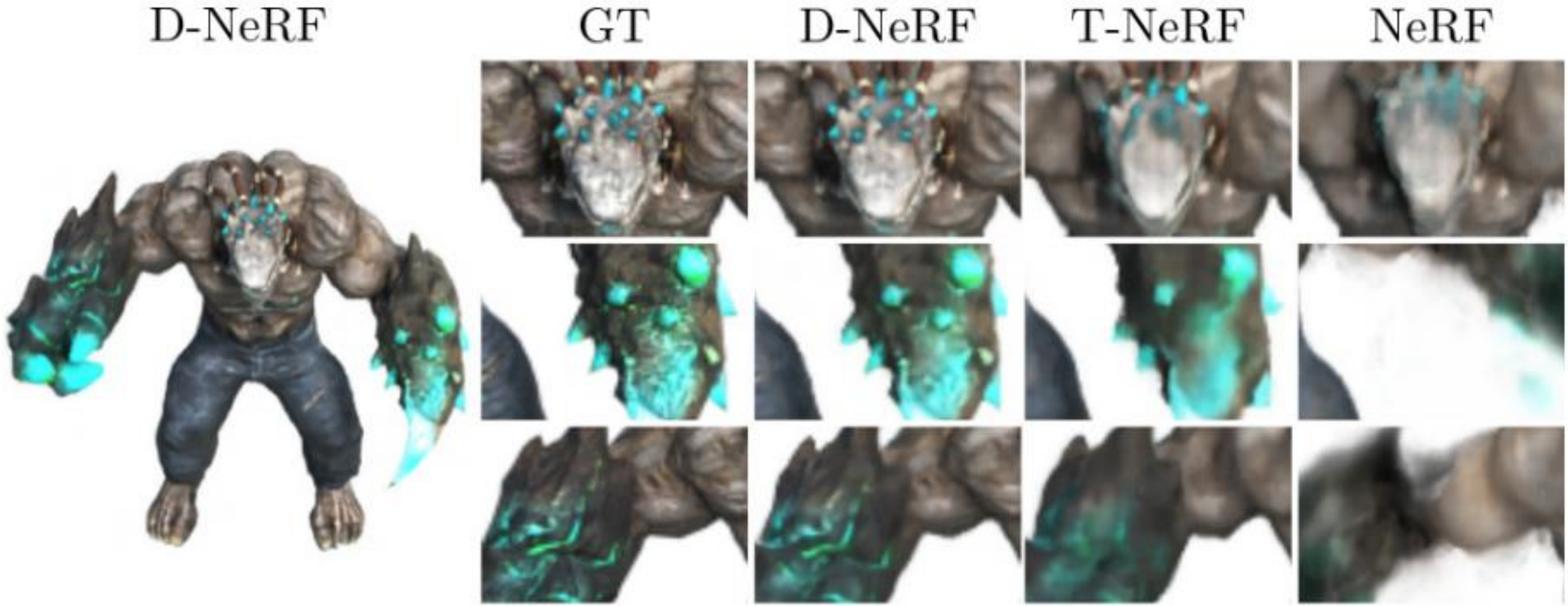
D-NeRF

- ## Results



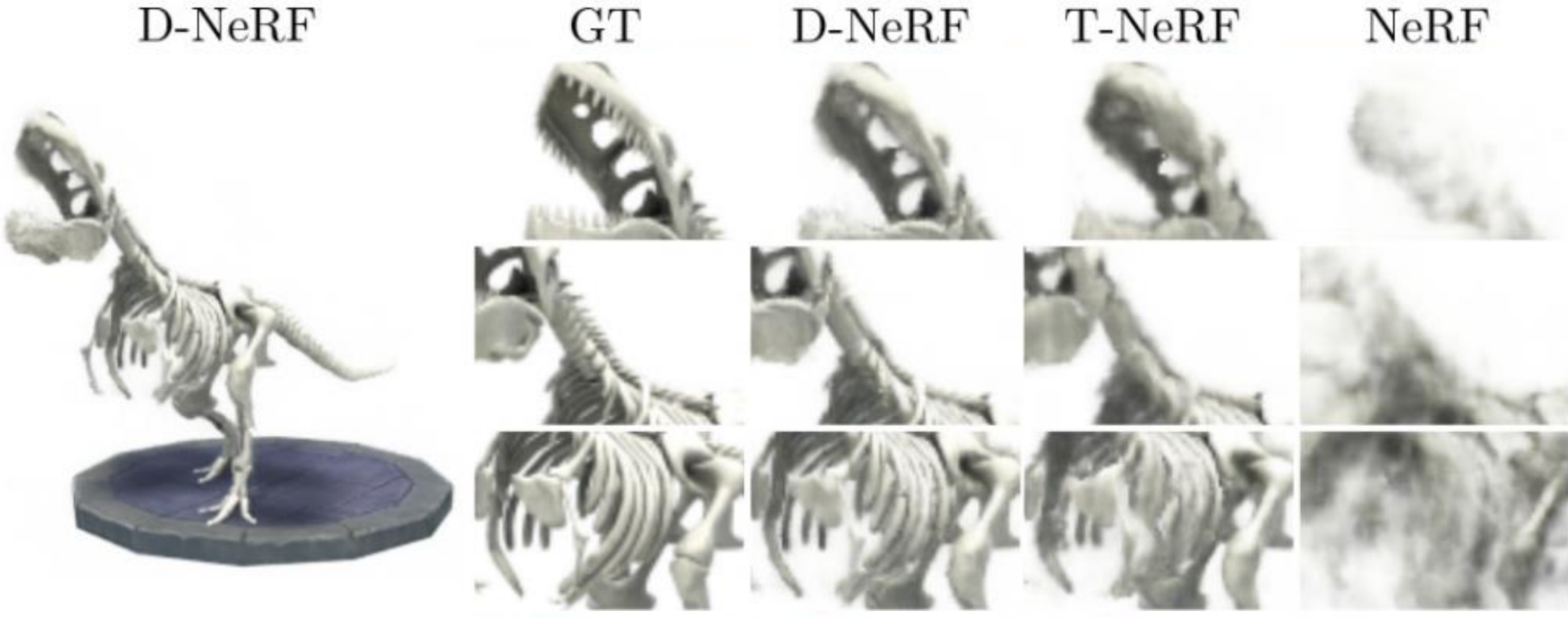
D-NeRF

- Results



D-NeRF

- **Results**



D-NeRF

- **Contributions**
  - Dynamic scenes
    - Time as well as novel camera configuration are considered
    - Only one view per each time instance

D-NeRF

- **Limitations**
  - Failure at poor camera poses
  - Missing large deformations
    - Higher frame rate can resolve this problem
  - Missing small details
  - Limited by a fixed sequence

FastNeRF: High-Fidelity Neural Rendering at 200FPS

[Garbin et al. ICCV 2021]

FastNeRF

- **Limitations of NeRF**
  - Training time
  - Inference time
  - Scalability
  - Camera calibration
  - Bounded scenes
  - Static scenes

FastNeRF

▪ Purpose

- Rendering NeRF in real-time

NeRF at 0.06FPS   NeRF at 31FPS   FastNeRF at 200FPS

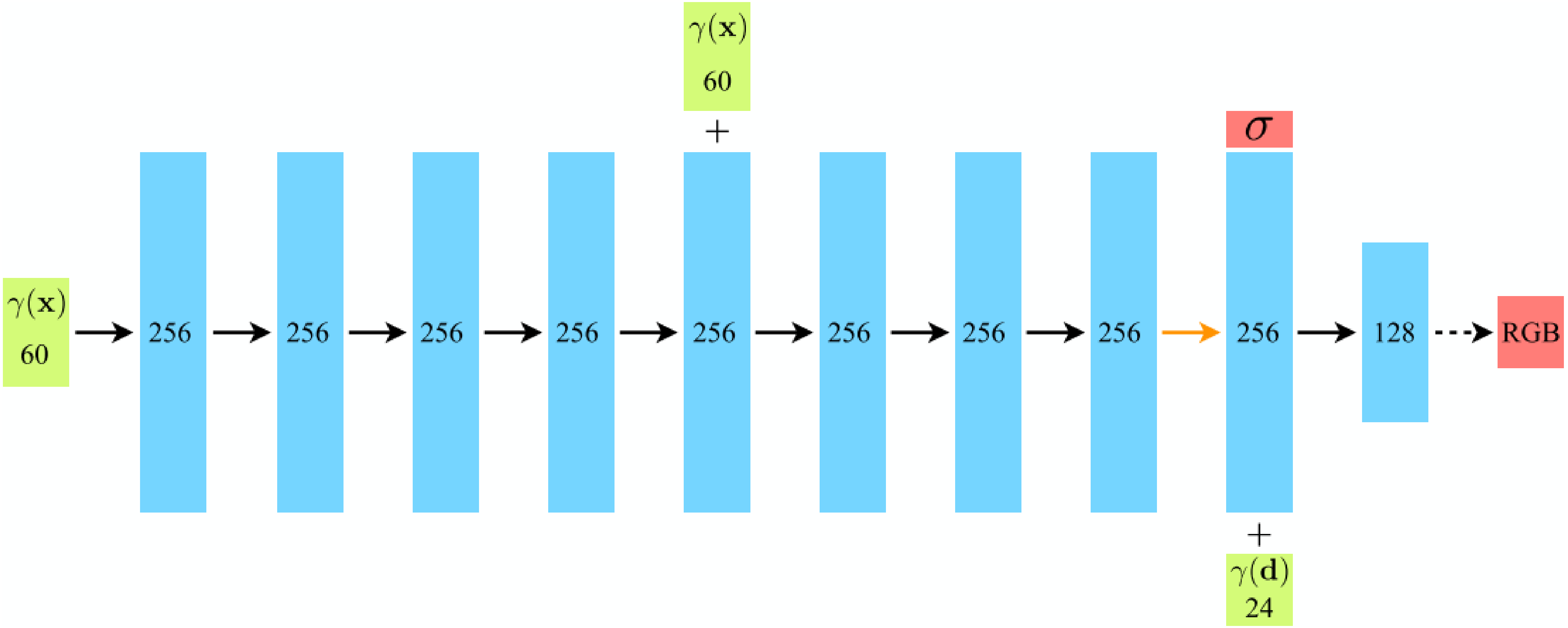
FastNeRF

- **Main Idea**
  - Caching
    - Trade-off between memory and time
    - Naïve approach
      - Store every pair of (x, y, z, θ, ϕ) and (r, g, b, σ)
      - $O(k^3 l^2)$ memory requirement (k: resolution for positions, l: resolution for directions)
      - 5600TB when $k = l = 1024$

FastNeRF

▪ Main Idea

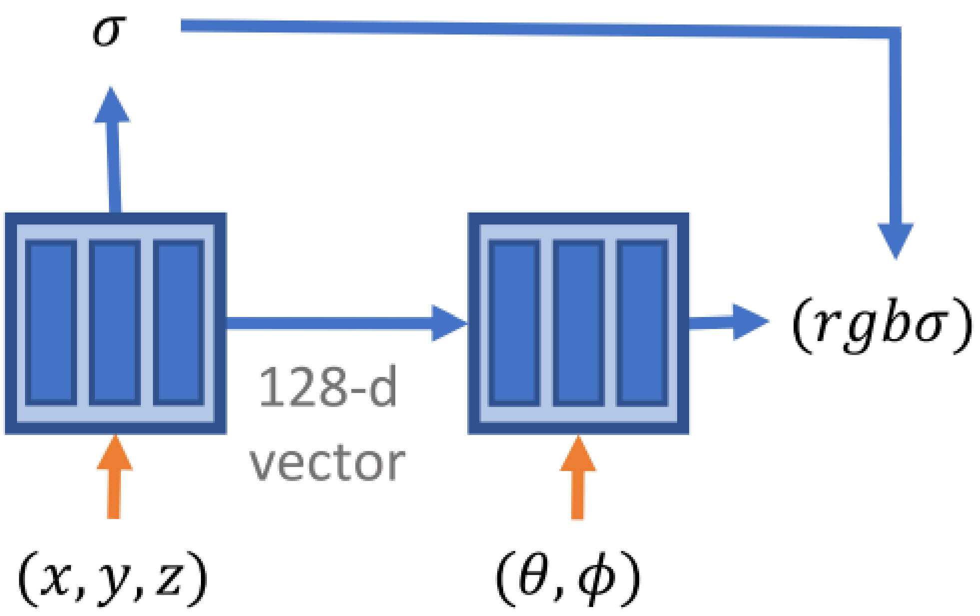
- NeRF



FastNeRF

▪ Main Idea

- NeRF
  - Factorization
    - › Density: position-dependent
    - › Color: position- and direction-dependent



FastNeRF

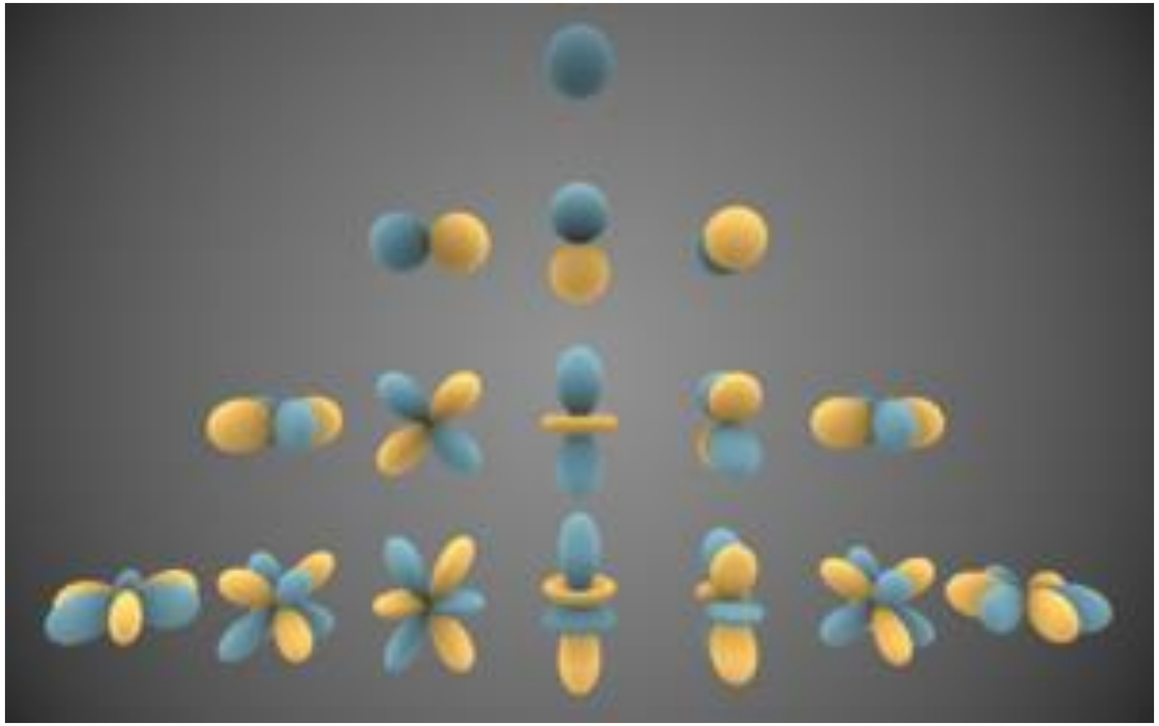
Main Idea

- Factorization
  - From what we have learned...
  - Rendering equation

$$L_o(\mathbf{p},\mathbf{d}) = \int_{\Omega} f_r(\mathbf{p},\mathbf{d},\boldsymbol{\omega}_i) L_i(\mathbf{p},\boldsymbol{\omega}_\mathbf{i})(\boldsymbol{\omega}_\mathbf{i}\cdot\mathbf{n})\,d\boldsymbol{\omega}_\mathbf{i}$$

FastNeRF

- **Main Idea**
  - Factorization
    - Spherical harmonics for approximation of rendering equation



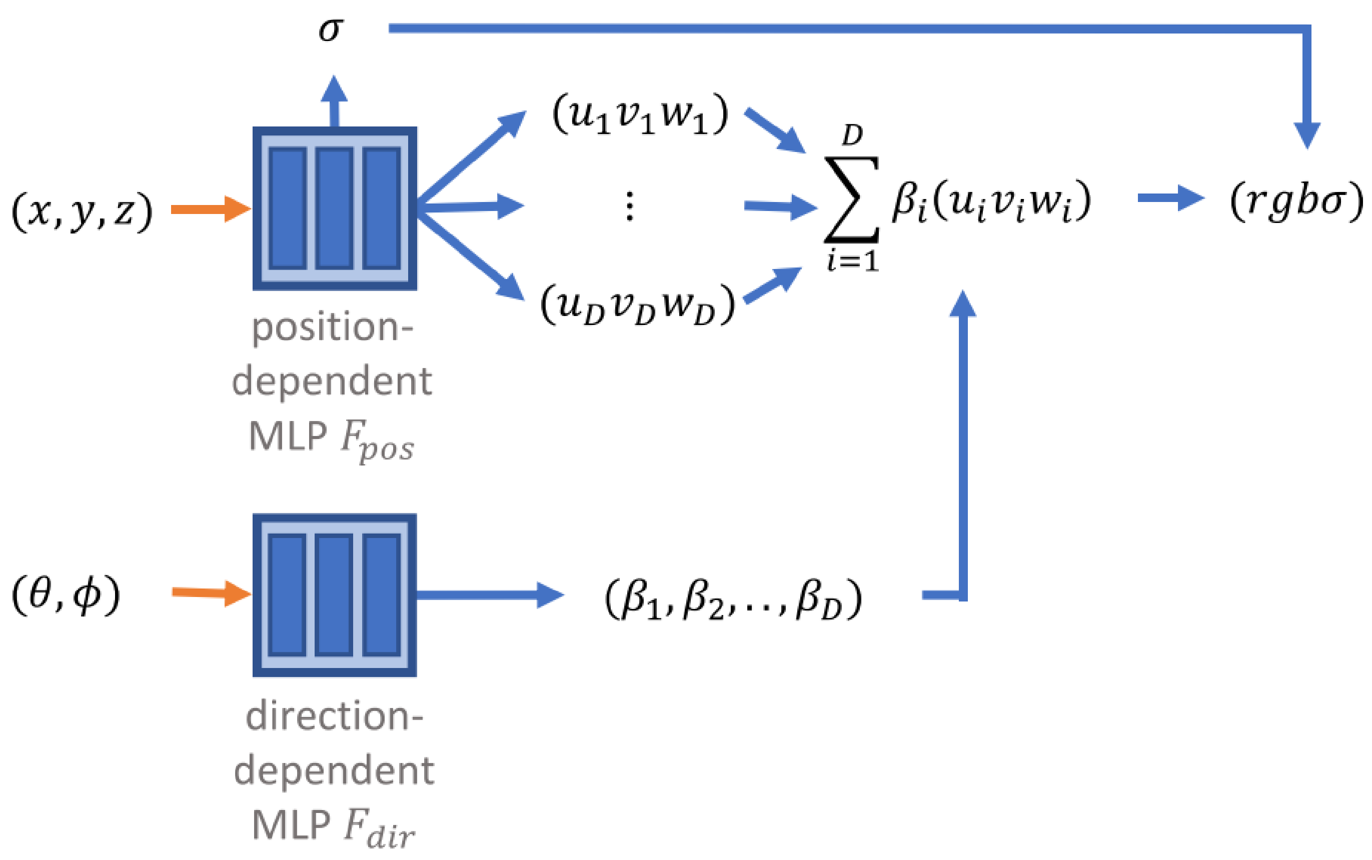
FastNeRF

- **Main Idea**
  - Factorization
    - Spherical harmonics for approximation of rendering equation
    - Dot product!

FastNeRF

▪ Main Idea

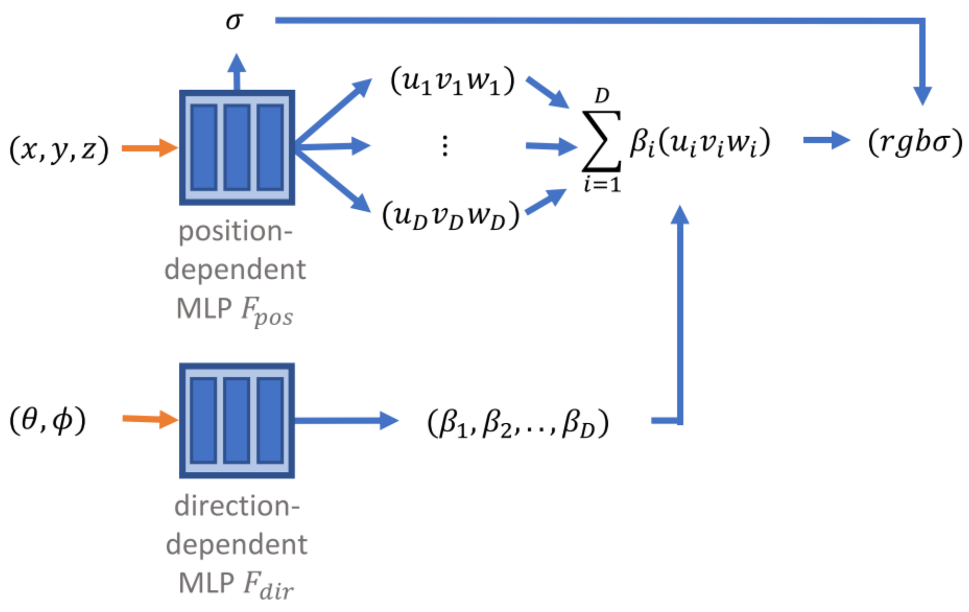
- Factorization



FastNeRF

▪ Network Architecture

- Outputs
  - Position-dependent network F_{pos}
    - Density
    - D-dimensional deep radiance map
  - Direction-dependent network F_{dir}
    - D-dimensional weights for the deep radiance map



FastNeRF

▪ Network Architecture

- $F_{\text{pos}}(\mathbf{p})$
  - $F_{\text{pos}}(\mathbf{p}) = (\sigma, \mathbf{u}, \mathbf{v}, \mathbf{w})$ where $\mathbf{u}, \mathbf{v}, \mathbf{w} \in \mathbb{R}^D$
  - 8-layer with 384 hidden units
- $F_{\text{dir}}(\mathbf{d})$
  - $F_{\text{dir}}(\mathbf{d}) = \boldsymbol{\beta}$ where $\boldsymbol{\beta} \in \mathbb{R}^D$
  - 4-layer with 256 hidden units
- Output
  - $\mathbf{c} = (r, g, b) = \sum_{i=1}^{D} \beta_i(\mathbf{u}_\mathbf{i}, \mathbf{v}_\mathbf{i}, \mathbf{w}_\mathbf{i}) = \boldsymbol{\beta}^T \cdot (\mathbf{u}, \mathbf{v}, \mathbf{w})$

FastNeRF

- **Caching**
  - Naïve approach
    - $O(k^3 l^2)$
      - k: resolution for positions
      - l: resolution for directions
      - 5600TB when $k = l = 1024$

FastNeRF

Caching

- FastNeRF
  - $O(k^3(1+3D)+l^2D)$
    - k: resolution for positions
    - l: resolution for directions
    - D: dimension of deep radiance maps
    - 54GB when $k=l=1024, D=8$

FastNeRF

- **Caching**
  - Is the size reasonable?
    - Smaller cache is enough in most cases
      - $k = 512, l = 256$
    - Original NeRF actually spends more memory for inference
      - 192 forward passes through an 8-layer 256 hidden unit MLP per pixel
      - Therefore, tremendous memory will be spent when NeRF is parallelized for similar performance

Neural radiance fields (NeRF)

NeRF@800x800 pixels - 0.06FPS

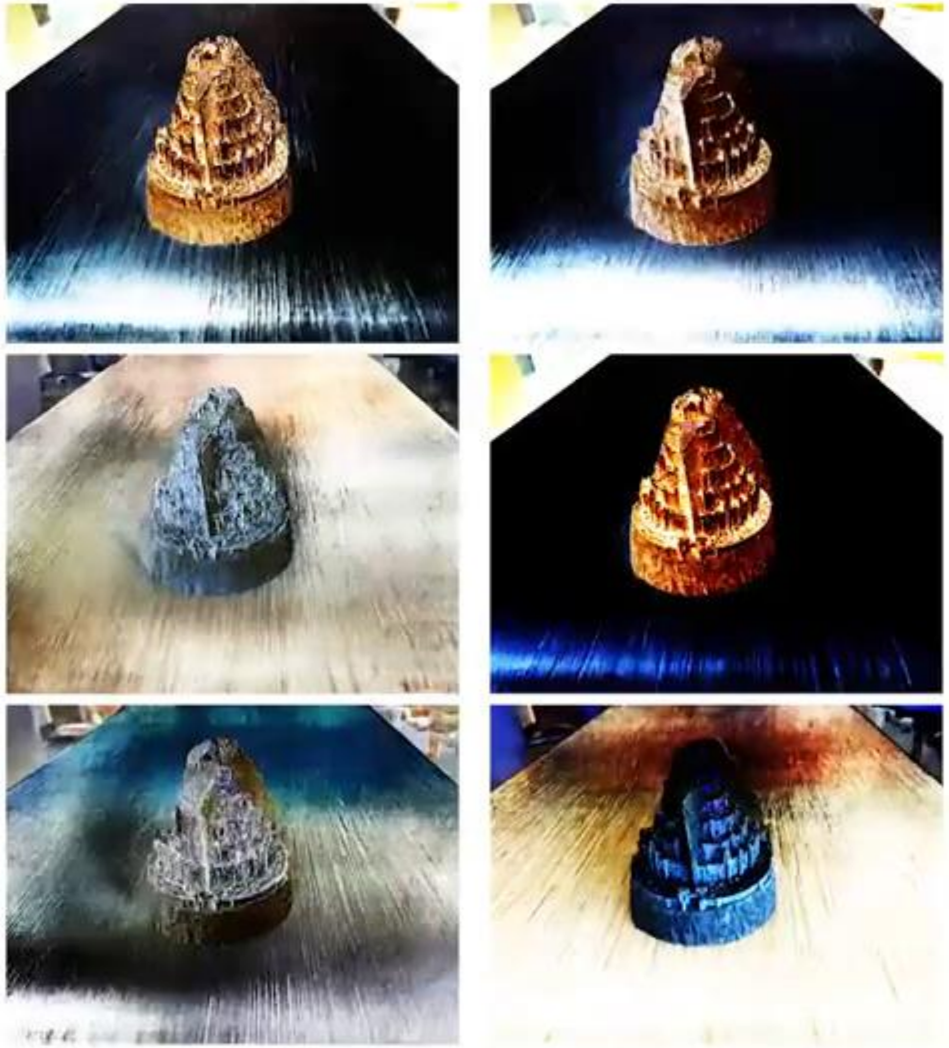
Comparison to NeRF

NeRF – 17.5**K** ms per frame

FastNeRF – 5.6 ms per frame

Deep radiance map

Output render



Deep radiance map components

Cache size

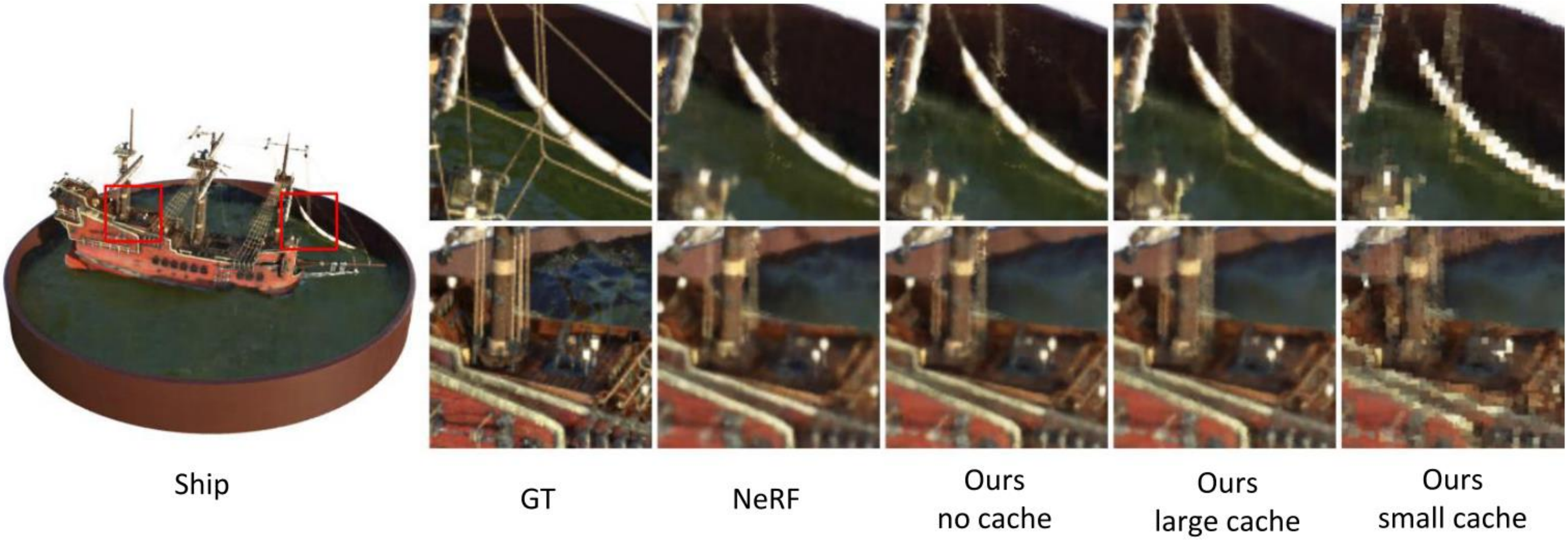
Cache 256^3

Cache 512^3

Cache 768^3

FastNeRF

- ## Results



Applications

FastNeRF

▪ Results

- Quantitative comparison

Scene	NeRF			Ours - No Cache			Ours - Cache			Speed
	PSNR↑	*SSIM*↑	*LPIPS*↓	*PSNR*↑	*SSIM*↑	*LPIPS*↓	*PSNR*↑	*SSIM*↑	*LPIPS*↓	
Nerf Synthetic	29.54dB	0.94	0.05	29.155dB	0.936	0.053	29.97dB	0.941	0.053	4.2ms
LLFF	27.72dB	0.88	0.07	27.958dB	0.888	0.063	26.035dB	0.856	0.085	1.4ms

FastNeRF

Results

- Quantitative comparison

Scene	NeRF	Ours - No Cache	256^3	384^3	512^3	768^3	1024^3	Speedup over NeRF
Chair	17.5**K**	28.2**K**	0.8	1.1	1.4	2.0	2.7	6468× - 21828×
Lego	17.5**K**	28.2**K**	1.5	2.1	2.8	4.2	5.6	3118× - 11639×
Horns*	3.8**K**	6.2**K**	0.5	0.7	0.9	1.2	-	3183× - 7640×
Leaves*	3.9**K**	6.3**K**	0.6	0.8	1.0	1.5	-	2626× - 6566×

FastNeRF

Results

- Ablation study on
  - Resolution
  - Value of D

Factors	**No Cache**		256^3		384^3		512^3		768^3	
	PSNR↑	*Memory*	*PSNR*↑	*Memory*	*PSNR*↑	*Memory*	*PSNR*↑	*Memory*	*PSNR*↑	*Memory*
4	27.11dB	-	24.81dB	0.34GB	26.29dB	0.61GB	26.94dB	1.09GB	27.54dB	2.51GB
6	27.12dB	-	24.82dB	0.5GB	26.34dB	0.93GB	27.0dB	1.67GB	27.58dB	4.1GB
8	27.24dB	-	24.89dB	0.71GB	26.42dB	1.41GB	27.1dB	2.7GB	27.72dB	7.15GB
16	27.68dB	-	25.07dB	1.2GB	26.77dB	2.08GB	27.55dB	3.72GB	28.3dB	9.16GB

FastNeRF

▪ Contributions

- (More than) Real-time rendering
  - No forward passes are called by caching
  - Resolution rarely matters
- Reasonable memory requirement
  - Much less than the original NeRF parallelized for similar runtime performance

FastNeRF

▪ Limitations

- Others than the rendering time were not solved
  - Training time, camera calibration, scalability, ...
  - Convergence with other methods can resolve this problem
- Quality cannot outperform the baseline
- Sparse caching decreases the rendering quality
  - More memory is enforced for higher quality
- Reasonable but still burdensome memory requirement

Q&A

Seokhyeon Hong

ghd3079@kaist.ac.kr