CSC2457 3D & Geometric Deep Learning

Scene Representation Networks: Continuous 3D-Structure-Aware Neural Scene Representations

Vincent Sitzmann, Michael Zollhöfer and Gordon Wetzstein

Feb 23rd

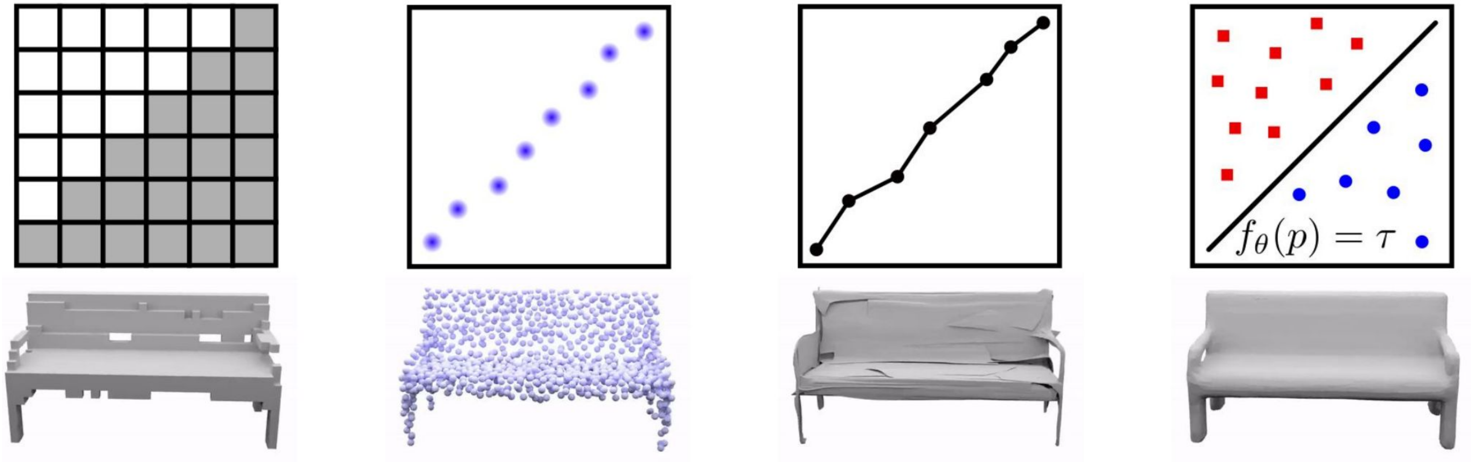
Presenter: Shayan Shekarforoush

Instructor: Animesh Garg

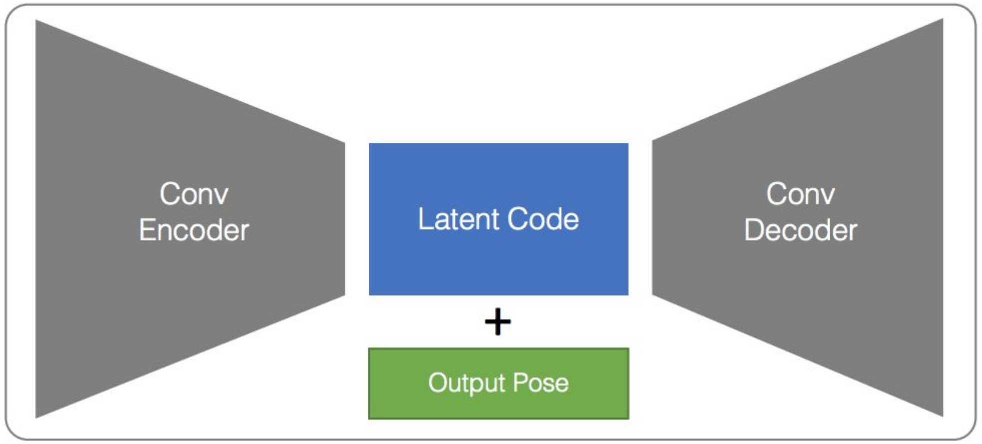
UNIVERSITY OF TORONTO

Learning Scene Representation

- With 3D Bias:

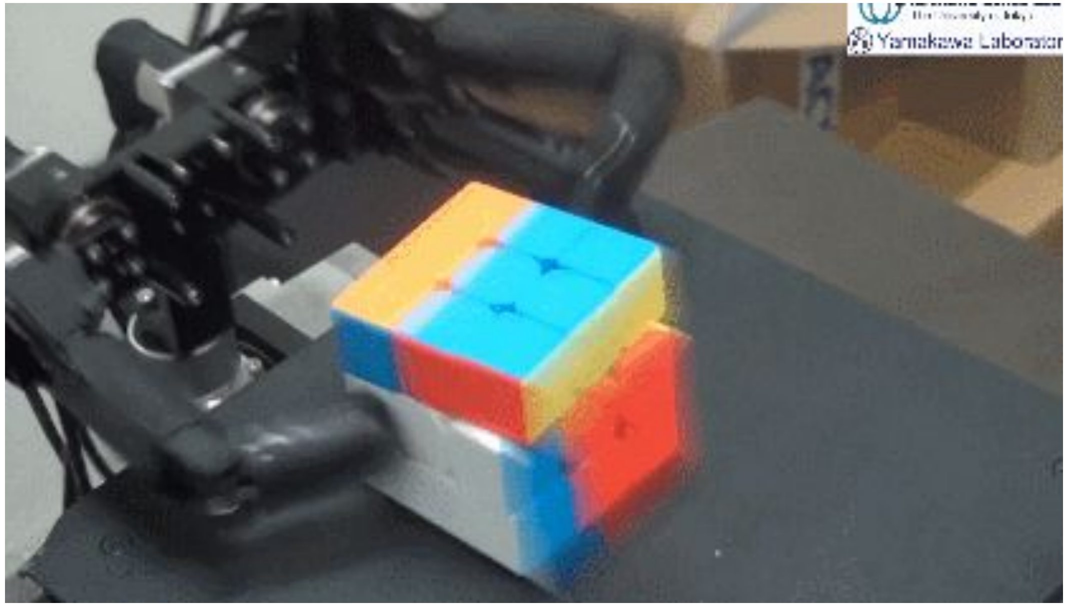


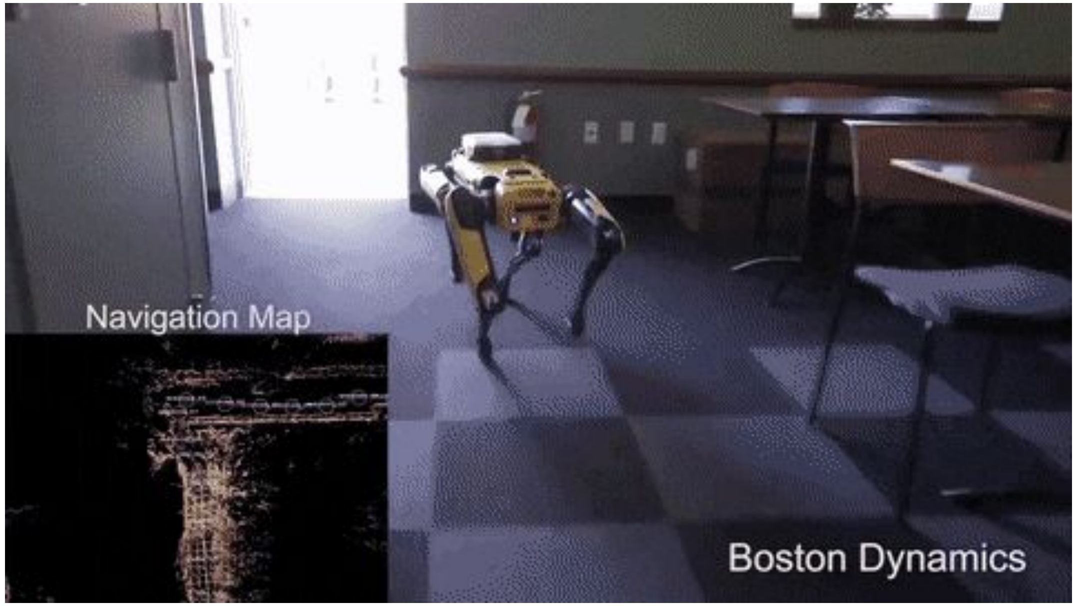
- Or not:



Applications

- Downstream tasks





Challenges

3D supervision

Challenges

3D supervision

Challenges

3D supervision

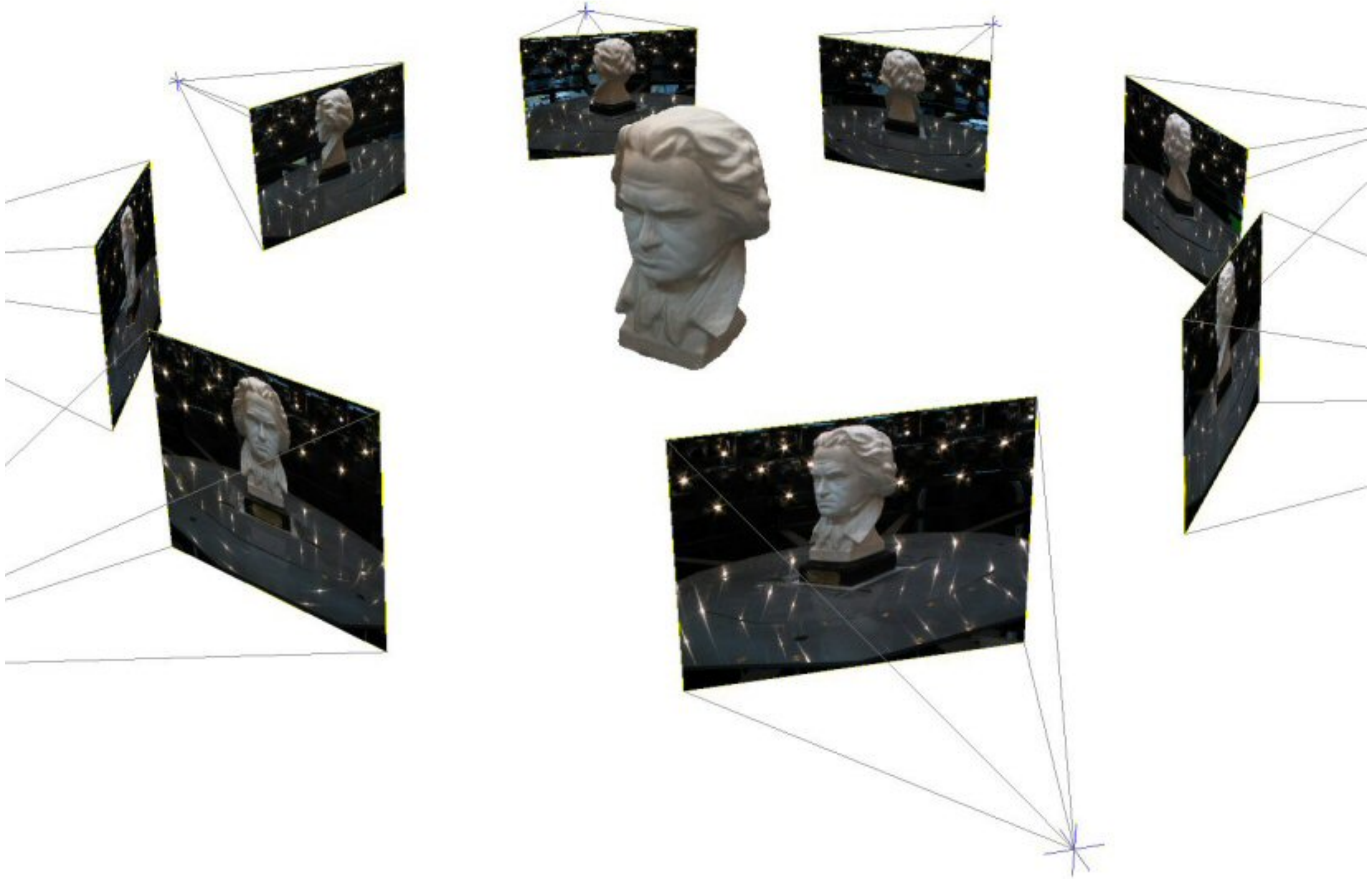
2D image + Camera pose

Challenges

Geometry

Geometry + Appearance

Challenges



Multi-view consistency

Challenges

Voxel resolutions

Challenges

Voxel resolutions

Point cloud sparsity

Contributions

- A continuous, 3D structure aware, neural scene representation encoding geometry and appearance a multi-view consistent manner.
  - Along with a Differentiable ray marching algorithm for rendering.
- End-to-end training without explicit 3D supervision.
- Generalizable to other geometry or appearance.
- Evaluation in:
  - Novel view synthesis.
  - Few-shot reconstruction.
  - ...

Problem Setting

Input data:

$\{$  $,$  $, \ldots \}$

2D image: $\mathcal{I}_i \in \mathbb{R}^{H \times W \times 3}$

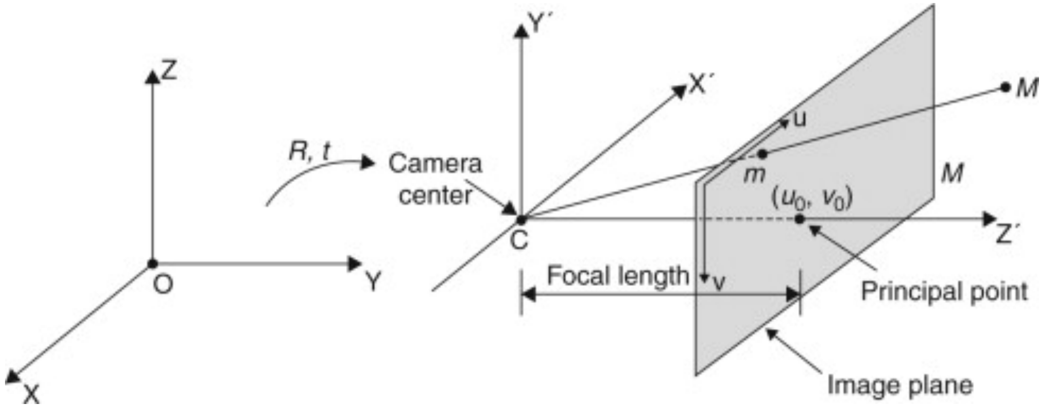
Problem Setting

Input data:

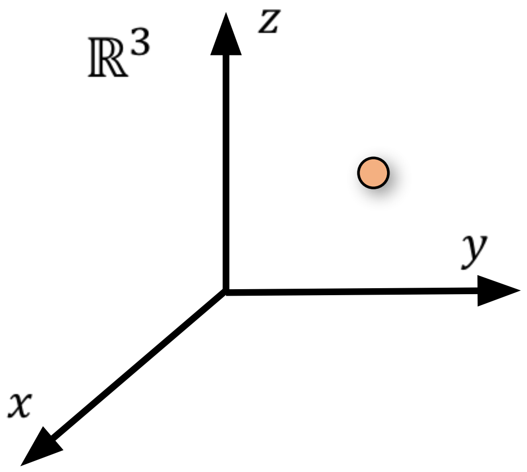
2D image: $\mathcal{I}_i \in \mathbb{R}^{H \times W \times 3}$

Extrinsic matrix: $\mathbf{E}_i = [\mathbf{R}|\mathbf{t}] \in \mathbb{R}^{3 \times 4}$

Intrinsic matrix: $\mathbf{K}_i \in \mathbb{R}^{3 \times 3}$

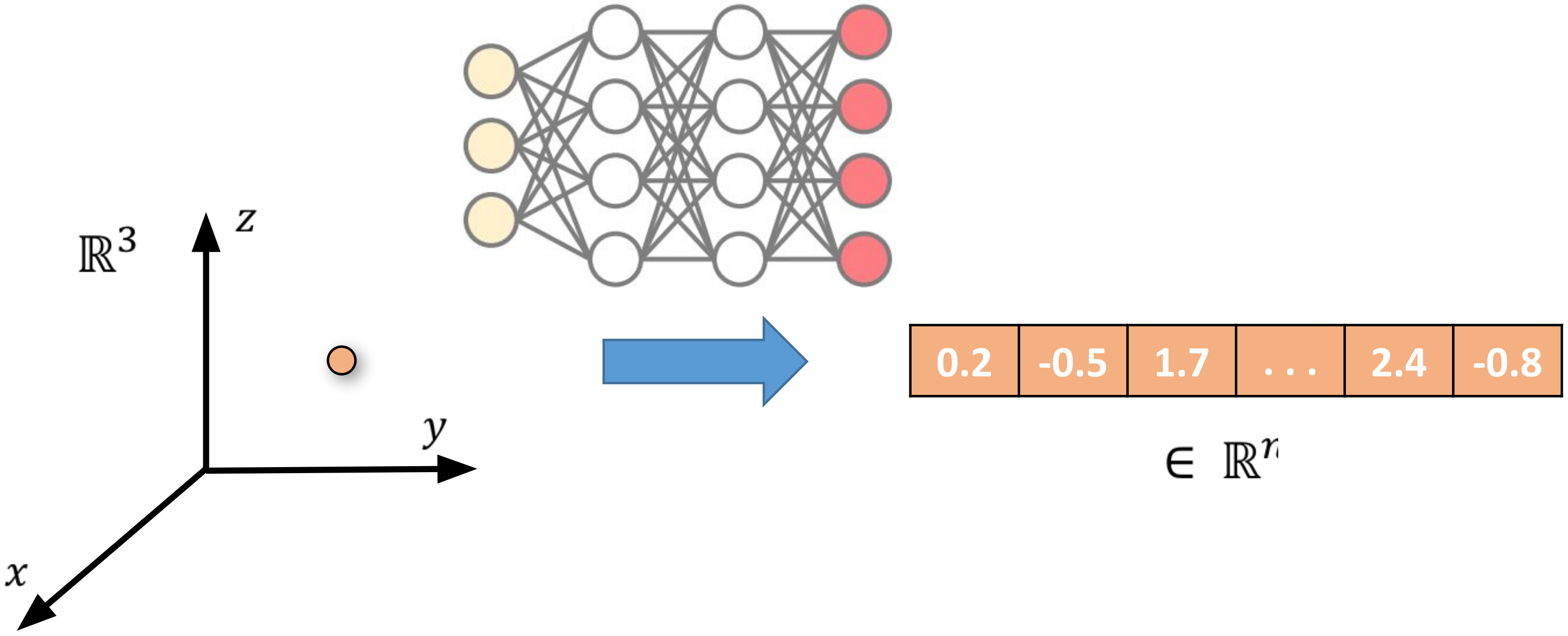


Implicit Scene Function

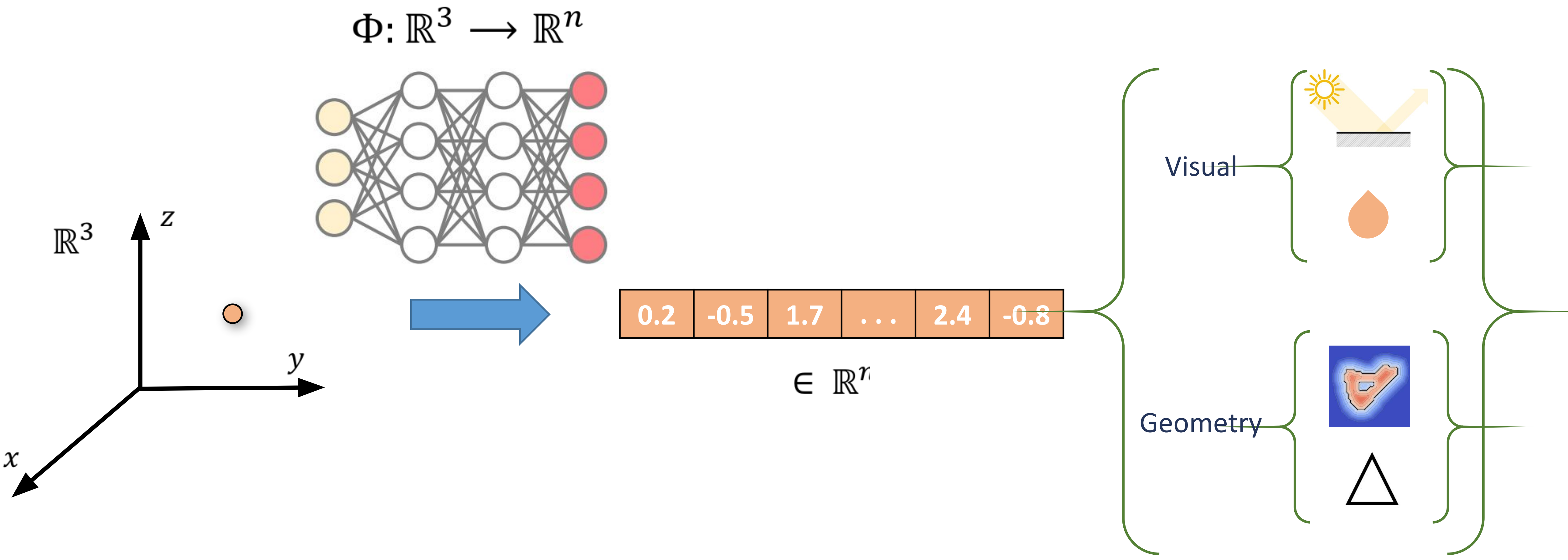


Implicit Scene Function

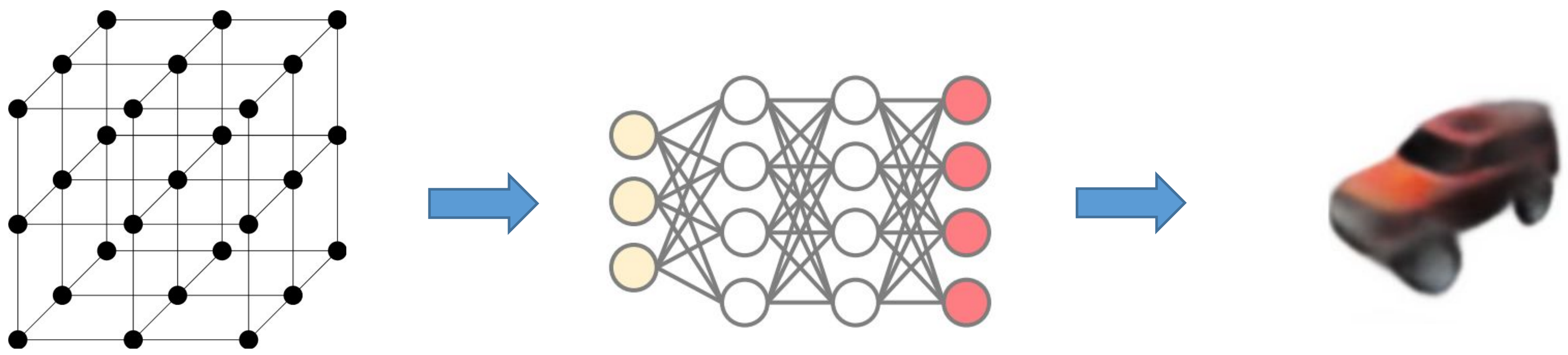
$$\Phi: \mathbb{R}^3 \longrightarrow \mathbb{R}^n$$



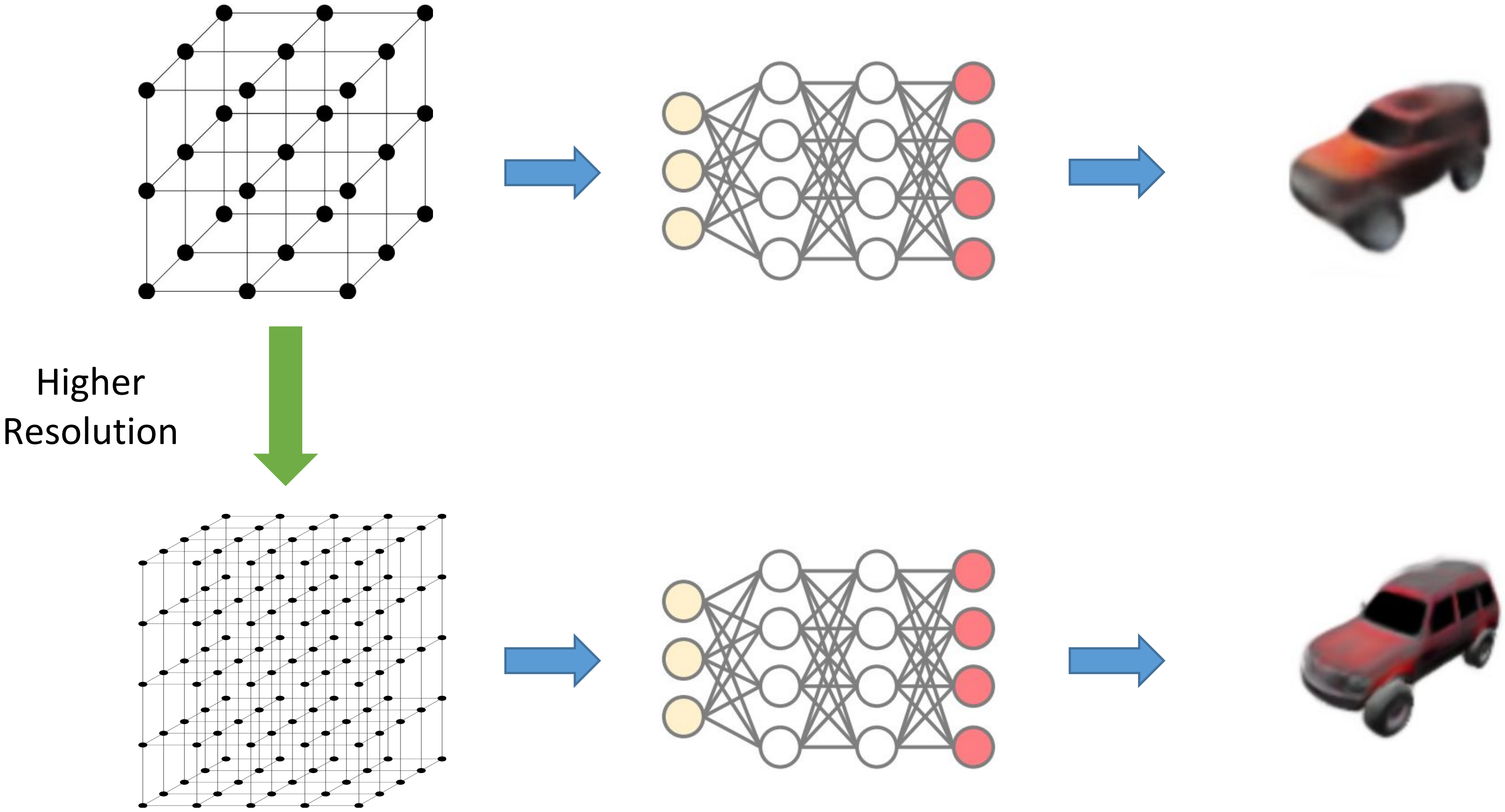
Implicit Scene Function



Implicit Scene Function

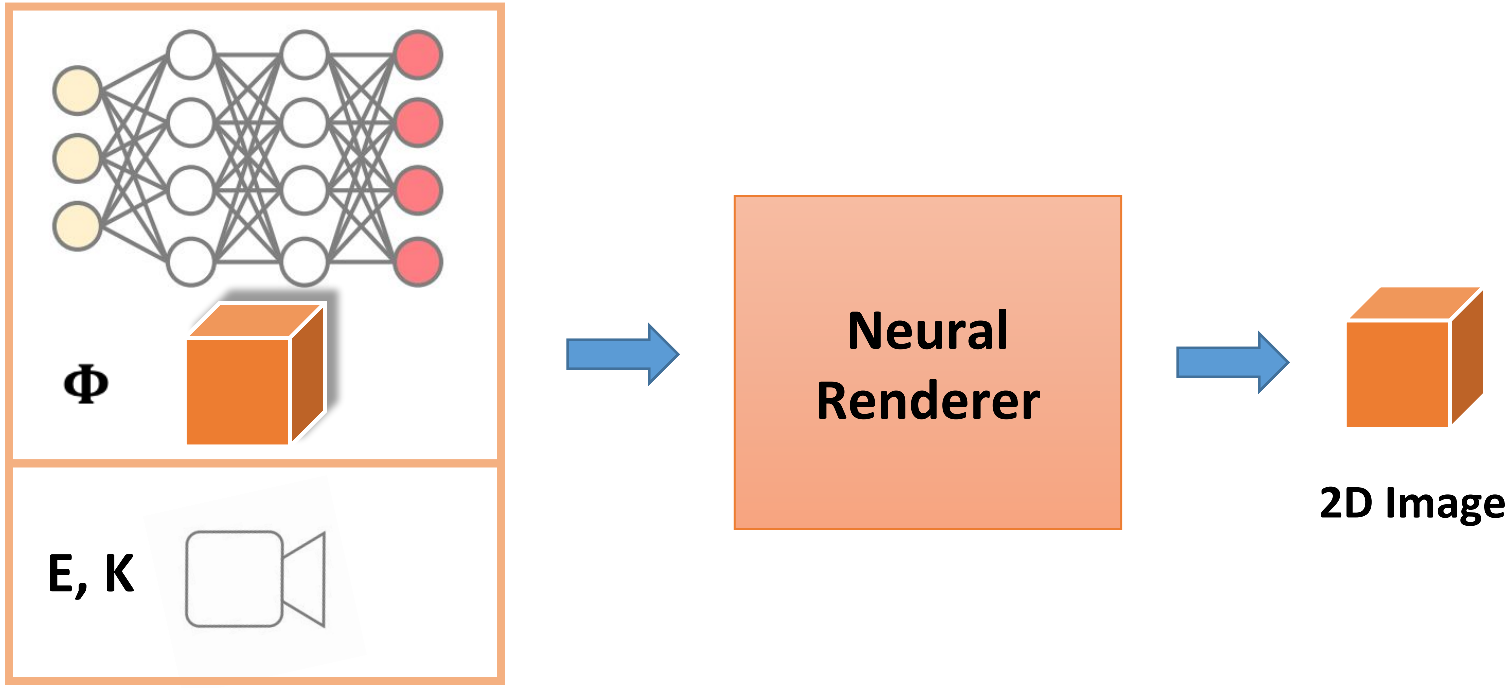


Implicit Scene Function



Neural Rendering

$$\Theta : \mathcal{X} \times \mathbb{R}^{3\times 4} \times \mathbb{R}^{3\times 3} \to \mathbb{R}^{H\times W\times 3}, \quad (\Phi, \mathbf{E}, \mathbf{K}) \mapsto \Theta(\Phi, \mathbf{E}, \mathbf{K}) = \mathcal{I}$$



Neural Rendering

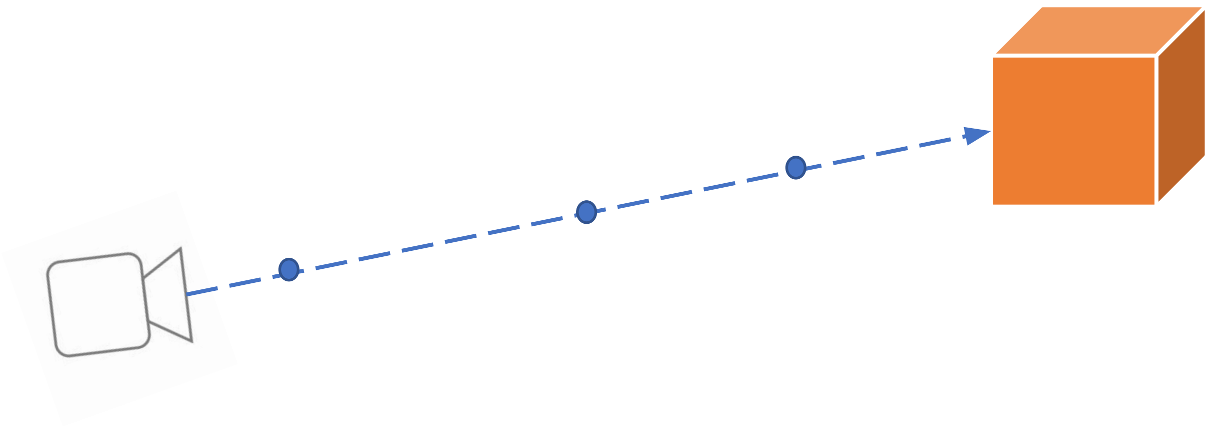
$$\Theta : \mathcal{X} \times \mathbb{R}^{3\times 4} \times \mathbb{R}^{3\times 3} \to \mathbb{R}^{H\times W\times 3}, \quad (\Phi, \mathbf{E}, \mathbf{K}) \mapsto \Theta(\Phi, \mathbf{E}, \mathbf{K}) = \mathcal{I}$$

- Ray Marching
- Pixel Generator

Ray Marching

Parametrize ray marching out of pixel (u, v):

$$\mathbf{r}_{u,v}(d) = \mathbf{R}^T(\mathbf{K}^{-1}\begin{pmatrix} u \\ v \\ d \end{pmatrix} - \mathbf{t})$$



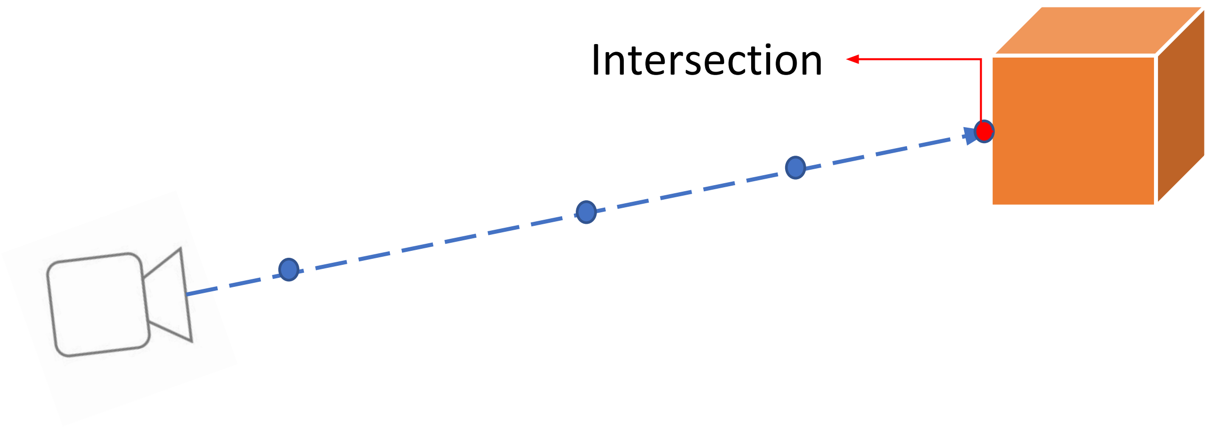
Ray Marching

Parametrize ray marching out of pixel (u, v): $\mathbf{r}_{u,v}(d) = \mathbf{R}^T(\mathbf{K}^{-1}\begin{pmatrix} u \\ v \\ d \end{pmatrix} - \mathbf{t})$

Intersection as optimization:

Surface

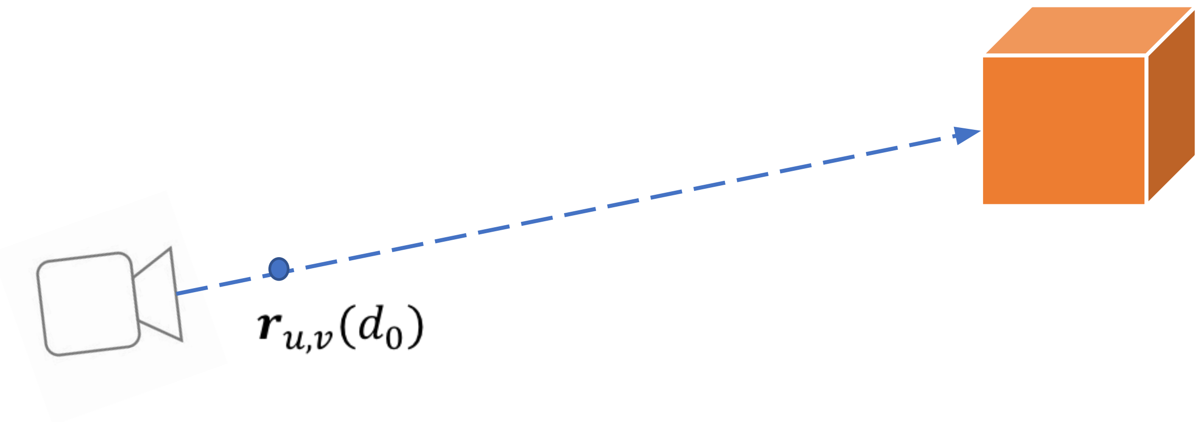
$$\begin{aligned} \arg\min \quad & d \\ \text{s.t.} \quad & \mathbf{r}_{u,v}(d) \in \boxed{\Omega}, \quad d > 0 \end{aligned}$$



Ray Marching

Parametrize ray marching out of pixel (u, v):

$$\mathbf{r}_{u,v}(d) = \mathbf{R}^T\left(\mathbf{K}^{-1}\begin{pmatrix} u \\ v \\ d \end{pmatrix} - \mathbf{t}\right)$$



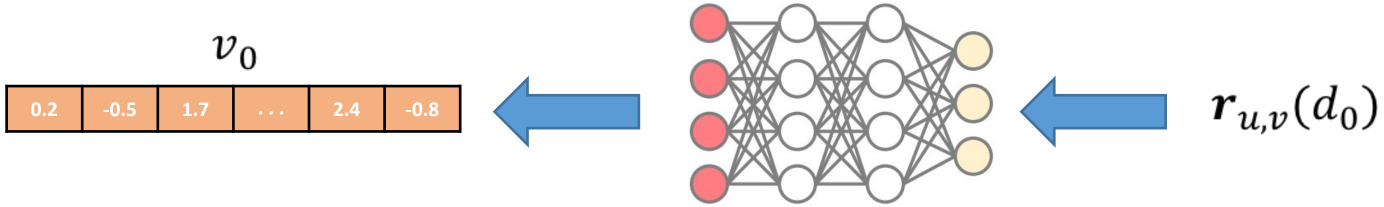
Algorithm 1 Differentiable Ray-Marching

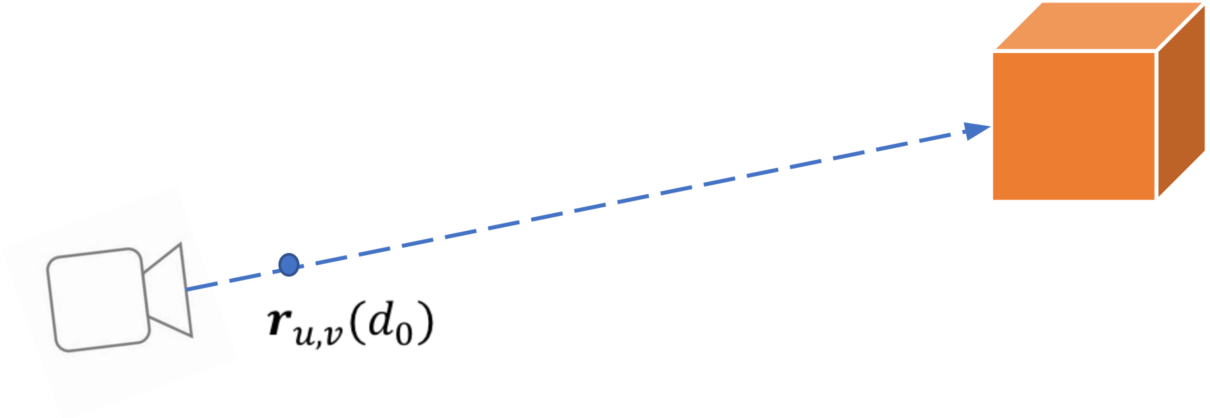
function FINDINTERSECTION($\Phi, \mathbf{K}, \mathbf{E}, (u, v)$)
  $\boxed{d_0 \leftarrow 0.05}$
  $(\mathbf{h}_0, \mathbf{c}_0) \leftarrow (\mathbf{0}, \mathbf{0})$
  for $i \leftarrow 0$ to max_iter **do**
    $\mathbf{x}_i \leftarrow \mathbf{r}_{u,v}(d_i)$
    $\mathbf{v}_i \leftarrow \Phi(\mathbf{x}_i)$
    $(\delta, \mathbf{h}_{i+1}, \mathbf{c}_{i+1}) \leftarrow LSTM(\mathbf{v}, \mathbf{h}_i, \mathbf{c}_i)$
    $d_{i+1} \leftarrow d_i + \delta$
  return $\mathbf{r}_{u,v}(d_{max\_iter})$

Ray Marching

Parametrize ray marching out of pixel (u, v):

$$\mathbf{r}_{u,v}(d) = \mathbf{R}^T\left(\mathbf{K}^{-1}\begin{pmatrix} u \\ v \\ d \end{pmatrix} - \mathbf{t}\right)$$





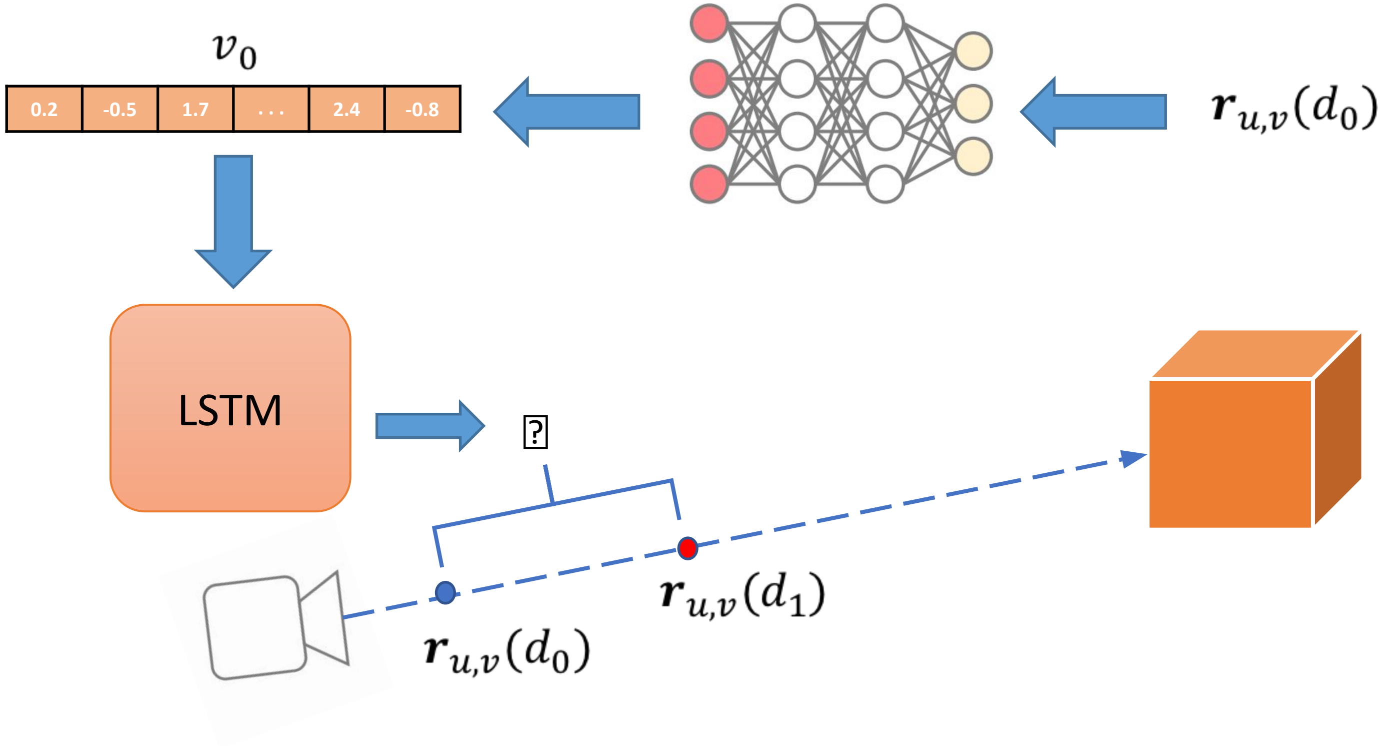
Algorithm 1 Differentiable Ray-Marching

function FINDINTERSECTION($\Phi, \mathbf{K}, \mathbf{E}, (u, v)$)
  $d_0 \leftarrow 0.05$
  $(\mathbf{h}_0, \mathbf{c}_0) \leftarrow (\mathbf{0}, \mathbf{0})$
  for $i \leftarrow 0$ to max_iter **do**
    $\mathbf{x}_i \leftarrow \mathbf{r}_{u,v}(d_i)$
    $\mathbf{v}_i \leftarrow \Phi(\mathbf{x}_i)$
    $(\delta, \mathbf{h}_{i+1}, \mathbf{c}_{i+1}) \leftarrow LSTM(\mathbf{v}, \mathbf{h}_i, \mathbf{c}_i)$
    $d_{i+1} \leftarrow d_i + \delta$
  return $\mathbf{r}_{u,v}(d_{max\_iter})$

Ray Marching

Parametrize ray marching out of pixel (u, v):

$$\mathbf{r}_{u,v}(d) = \mathbf{R}^T(\mathbf{K}^{-1}\begin{pmatrix} u \\ v \\ d \end{pmatrix} - \mathbf{t})$$



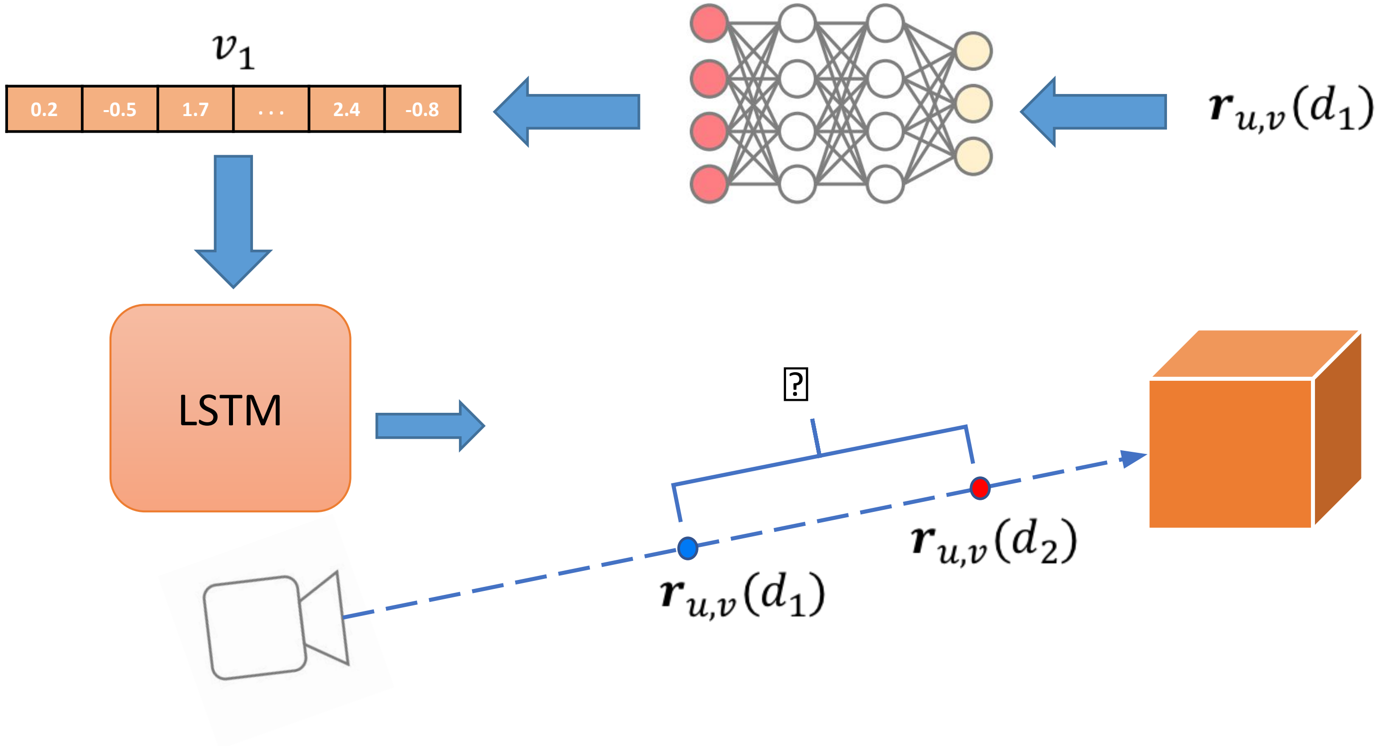
Algorithm 1 Differentiable Ray-Marching

```
function FINDINTERSECTION(Φ, K, E, (u, v))
    d_0 ← 0.05
    (h_0, c_0) ← (0, 0)
    for i ← 0 to max_iter do
        x_i ← r_{u,v}(d_i)
        v_i ← Φ(x_i)
        (δ, h_{i+1}, c_{i+1}) ← LSTM(v, h_i, c_i)
        d_{i+1} ← d_i + δ
    return r_{u,v}(d_{max_iter})
```

Ray Marching

Parametrize ray marching out of pixel (u, v):

$$\mathbf{r}_{u,v}(d) = \mathbf{R}^T \left(\mathbf{K}^{-1} \begin{pmatrix} u \\ v \\ d \end{pmatrix} - \mathbf{t} \right)$$



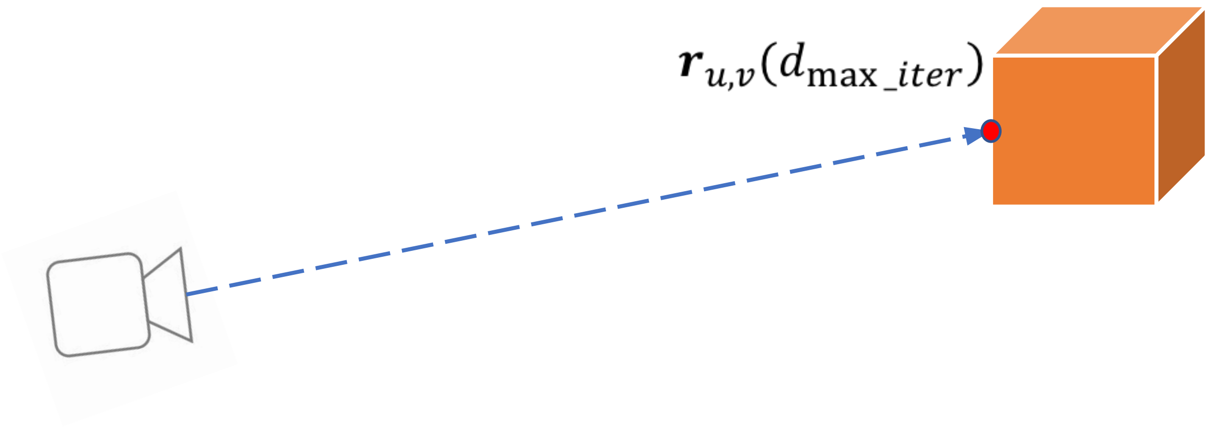
Algorithm 1 Differentiable Ray-Marching

function FINDINTERSECTION($\Phi, \mathbf{K}, \mathbf{E}, (u, v)$)
  $d_0 \leftarrow 0.05$
  $(\mathbf{h}_0, \mathbf{c}_0) \leftarrow (\mathbf{0}, \mathbf{0})$
  for $i \leftarrow 0$ to max_iter **do**
    $\mathbf{x}_i \leftarrow \mathbf{r}_{u,v}(d_i)$
    $\mathbf{v}_i \leftarrow \Phi(\mathbf{x}_i)$
    $(\delta, \mathbf{h}_{i+1}, \mathbf{c}_{i+1}) \leftarrow LSTM(\mathbf{v}, \mathbf{h}_i, \mathbf{c}_i)$
    $d_{i+1} \leftarrow d_i + \delta$
  return $\mathbf{r}_{u,v}(d_{max\_iter})$

Ray Marching

Parametrize ray marching out of pixel (u, v):

$$\mathbf{r}_{u,v}(d) = \mathbf{R}^T\left(\mathbf{K}^{-1}\begin{pmatrix} u \\ v \\ d \end{pmatrix} - \mathbf{t}\right)$$

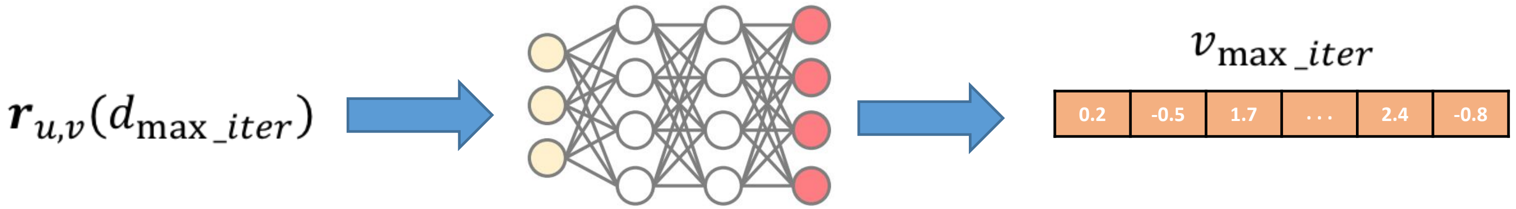


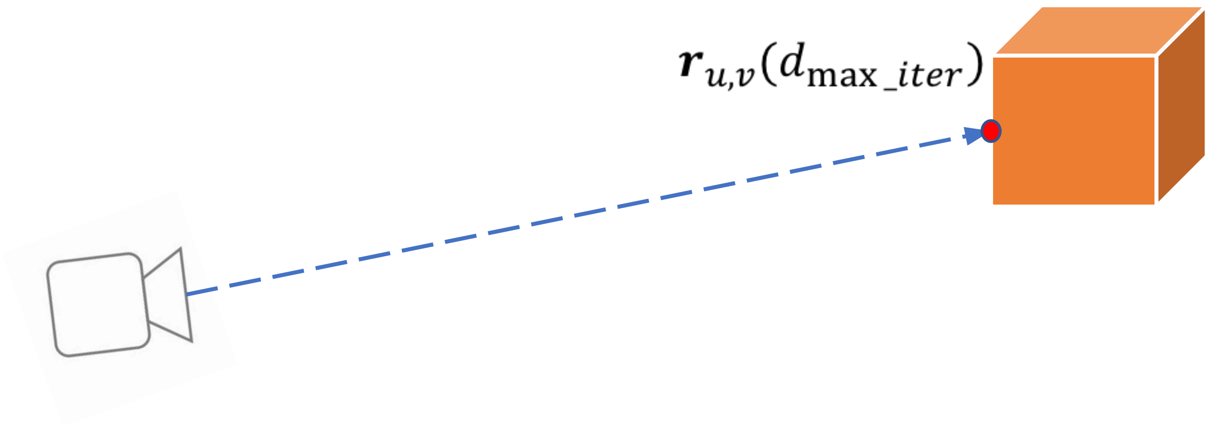
Algorithm 1 Differentiable Ray-Marching

```
function FINDINTERSECTION(Φ, K, E, (u, v))
    d_0 ← 0.05
    (h_0, c_0) ← (0, 0)
    for i ← 0 to max_iter do
        x_i ← r_{u,v}(d_i)
        v_i ← Φ(x_i)
        (δ, h_{i+1}, c_{i+1}) ← LSTM(v, h_i, c_i)
        d_{i+1} ← d_i + δ
    return r_{u,v}(d_{max_iter})
```

Pixel Generator

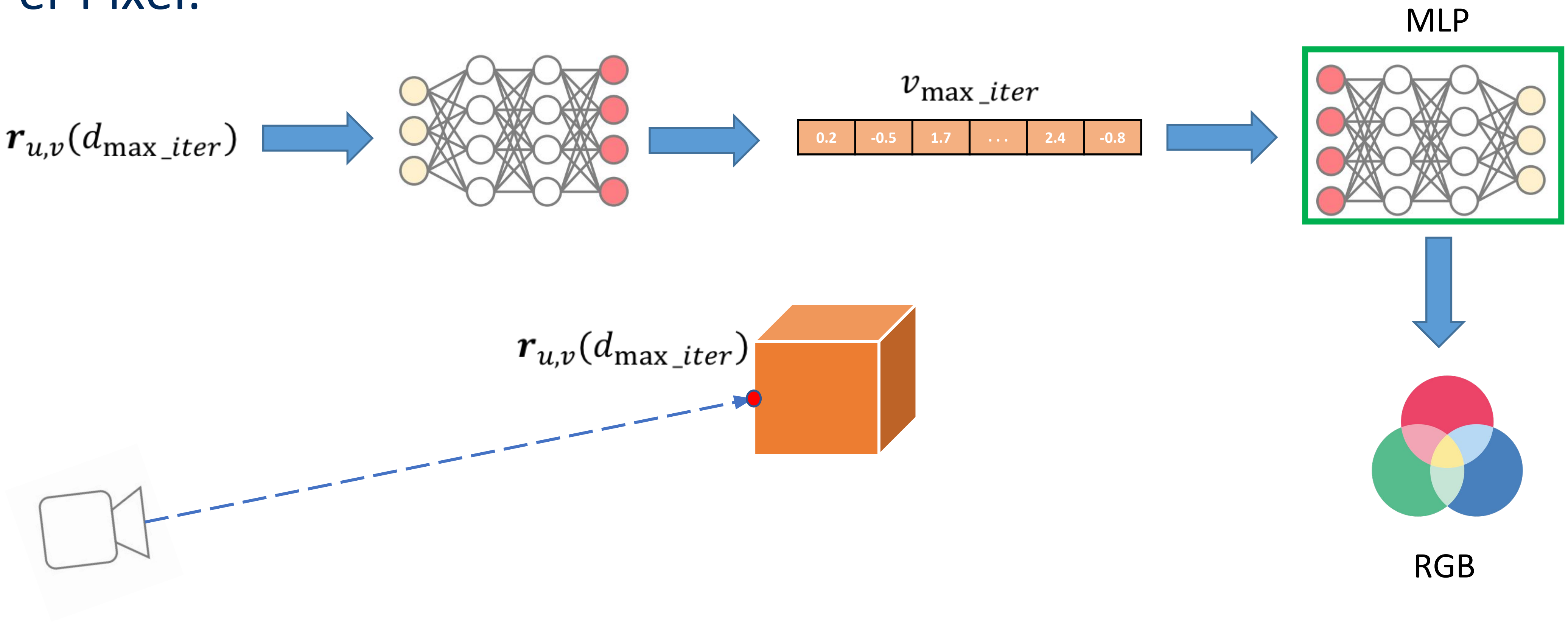
Per Pixel:



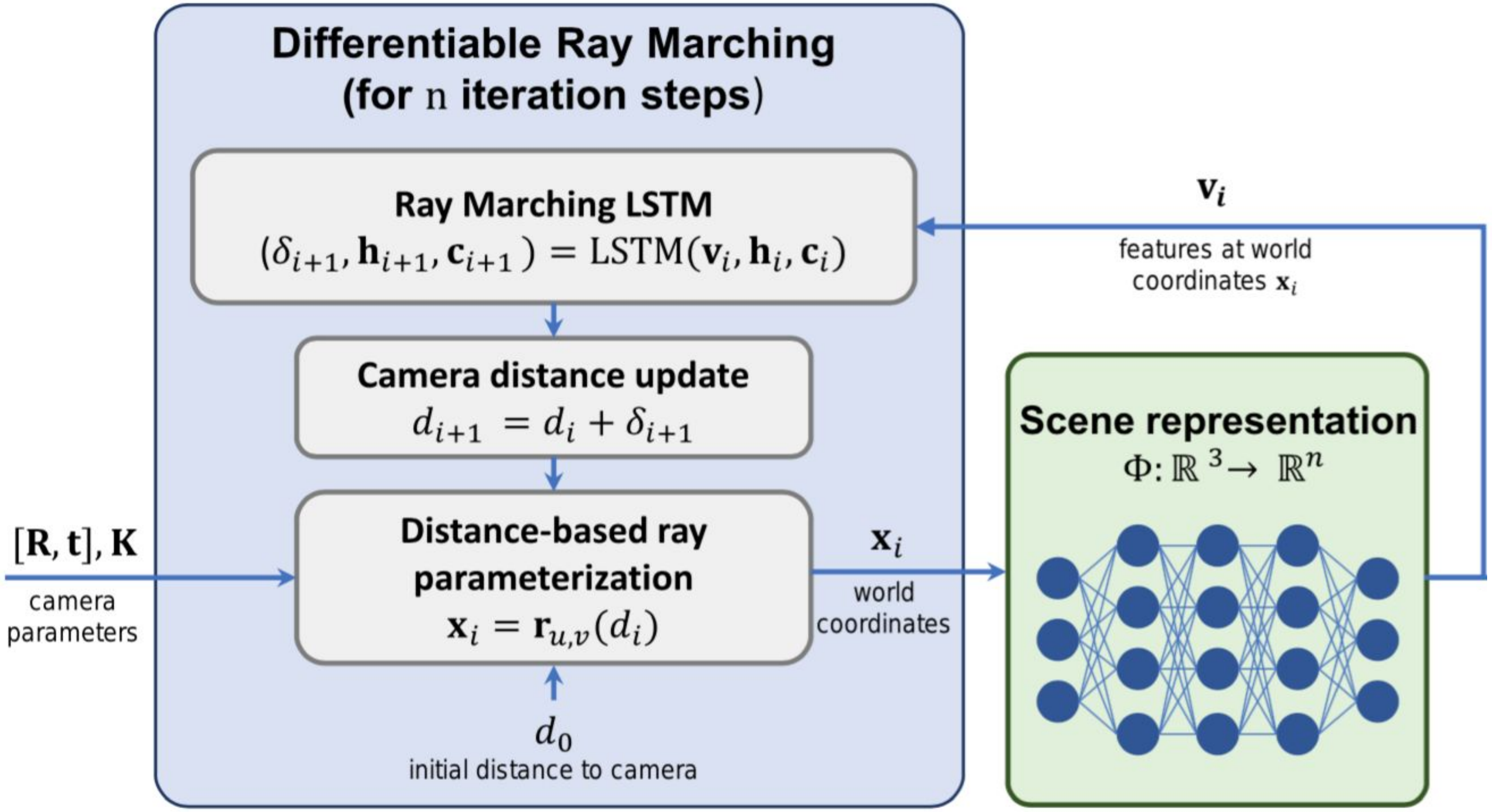


Pixel Generator

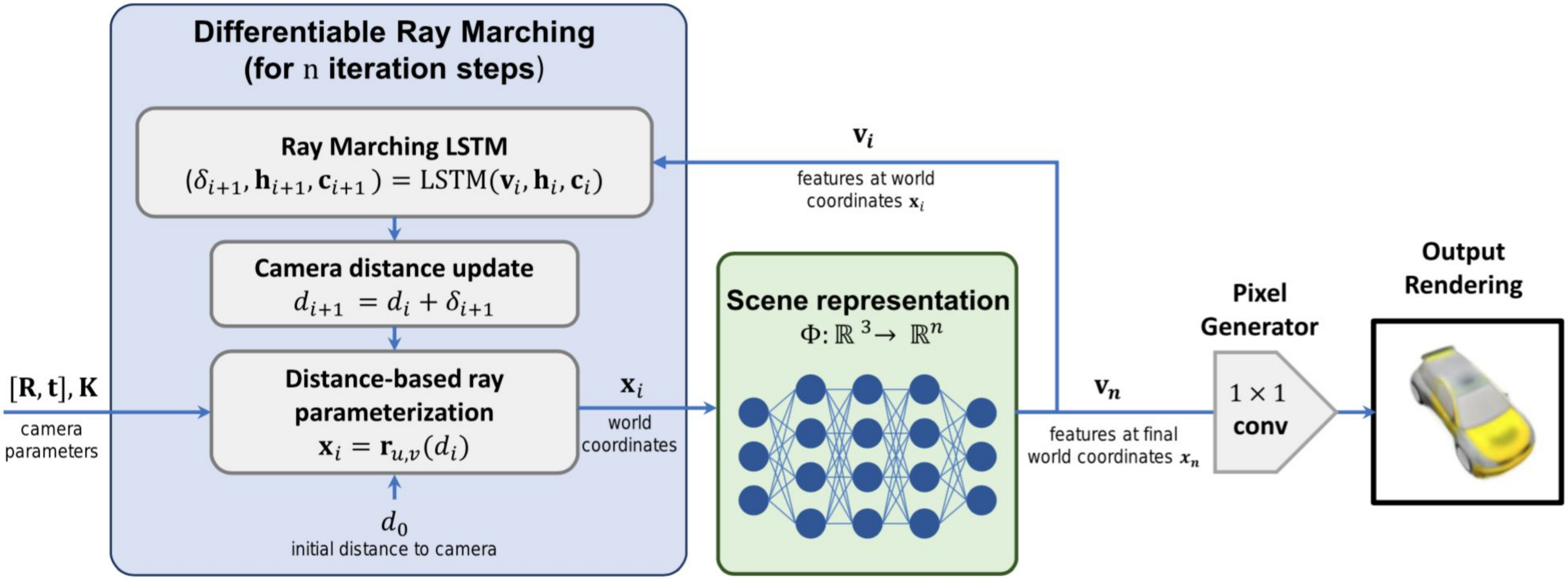
Per Pixel:



General Framework



General Framework



Generalization over Scenes

Generalization over Scenes

Φ_1

Φ_M

Φ_2

Φ_M

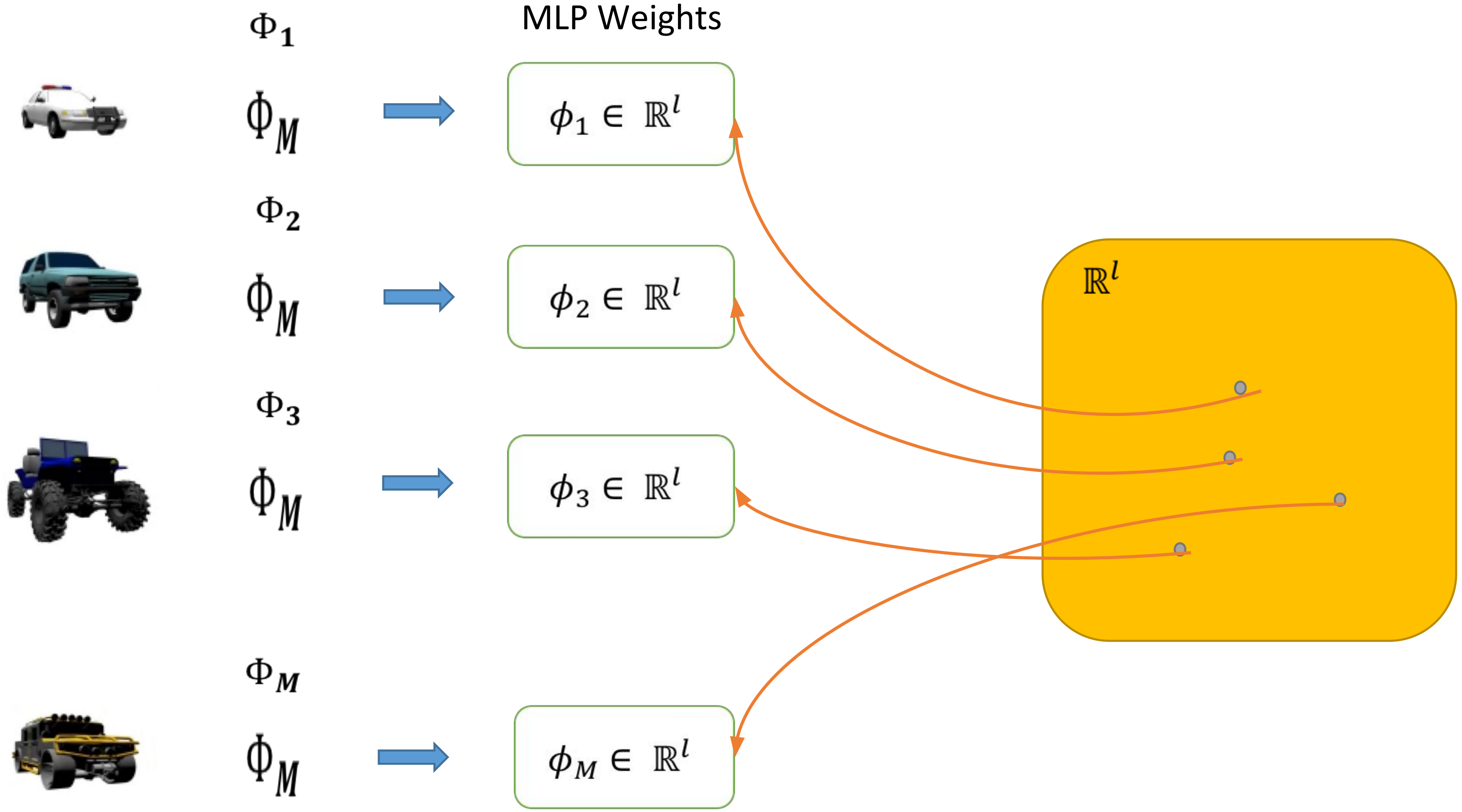
Φ_3

Φ_M

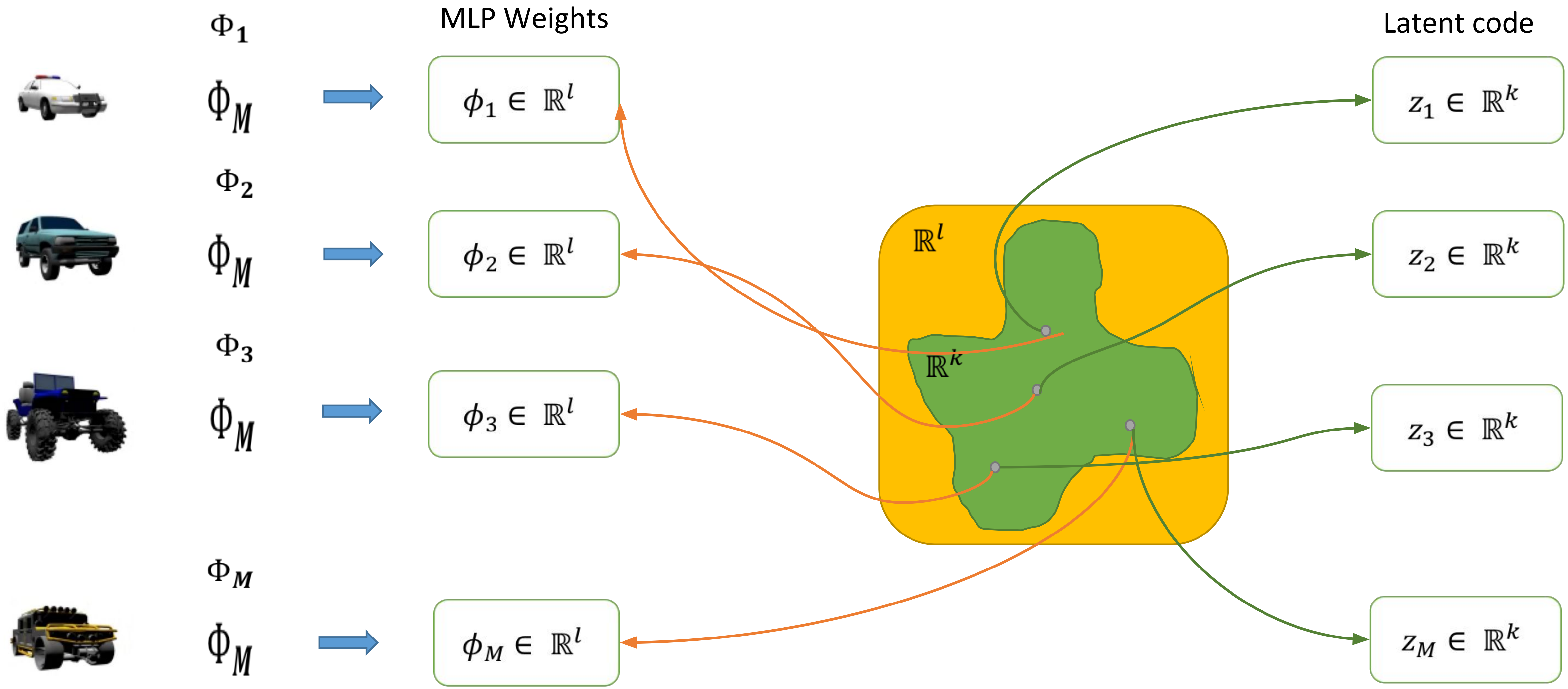
Φ_M

Φ_M

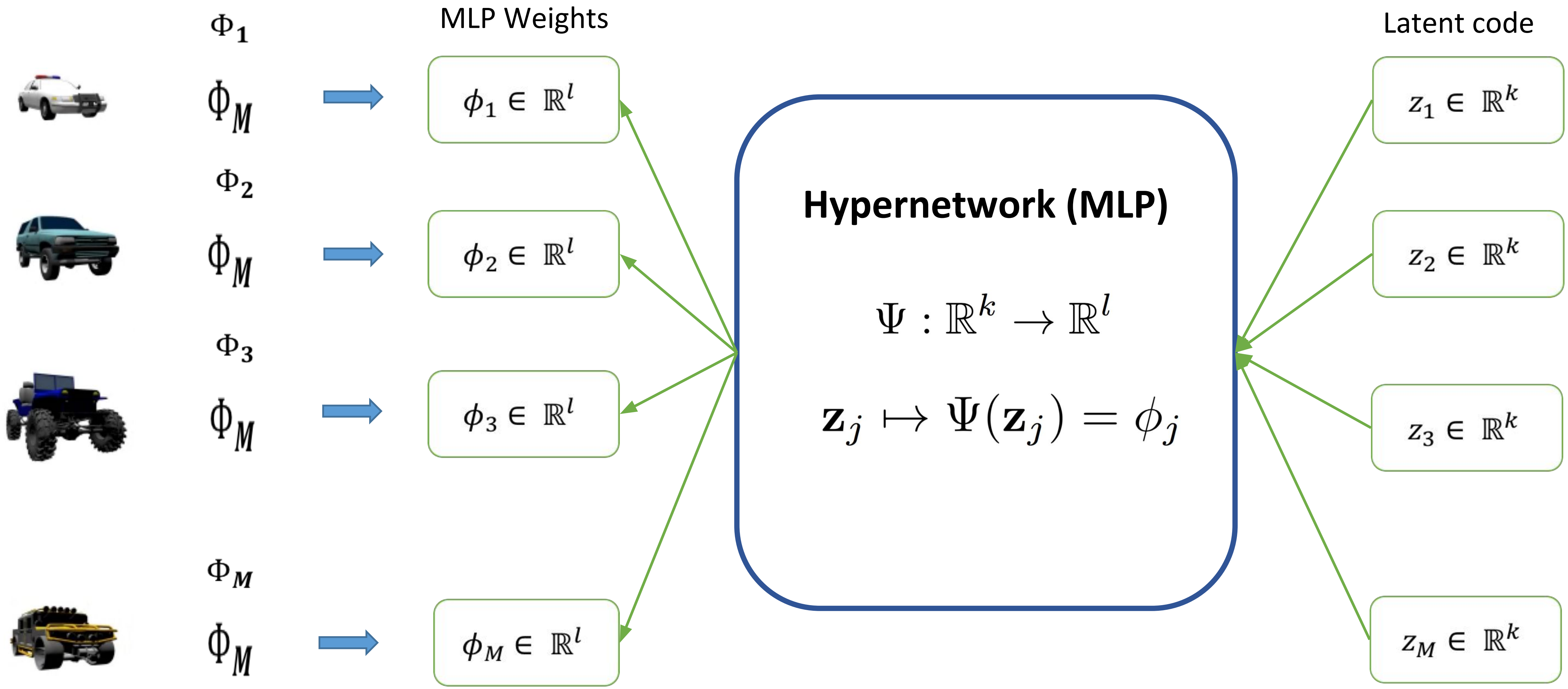
Generalization over Scenes



Generalization over Scenes



Generalization over Scenes



Optimization

Joint optimization:

$$\underset{\boxed{\theta,\psi,\{\mathbf{z}_j\}_{j=1}^{M}}}{\arg\min}$$

$\boldsymbol{\theta}$: Neural Renderer

ψ: Hypernetwork

z: Scene latent code

Optimization

Joint optimization using SGD:

$$\underset{\{\theta,\psi,\{\mathbf{z}_j\}_{j=1}^{M}\}}{\arg\min} \boxed{\sum_{j=1}^{M}}\boxed{\sum_{i=1}^{N}}$$

Instances

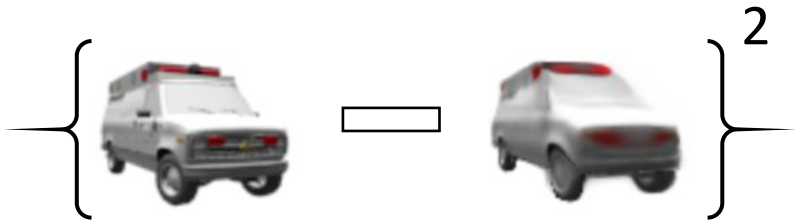
Viewpoints

, …

Optimization

Joint optimization using SGD:

$$\underset{\{\theta,\psi,\{\mathbf{z}_j\}_{j=1}^{M}\}}{\arg\min} \sum_{j=1}^{M}\sum_{i=1}^{N} \underbrace{\|\Theta_\theta(\Phi_{\Psi(\mathbf{z_j})}, \mathbf{E}_i^j, \mathbf{K}_i^j) - \mathcal{I}_i^j\|_2^2}_{\mathcal{L}_{\text{img}}} +$$

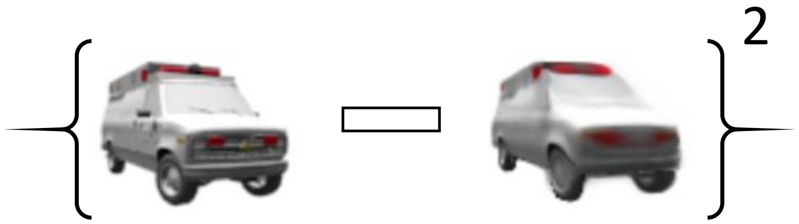


L2 reconstruction loss

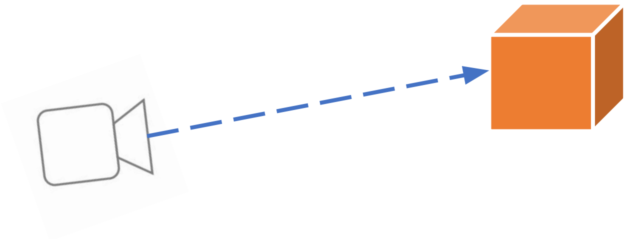
Optimization

Joint optimization using SGD:

$$\underset{\{\theta,\psi,\{\mathbf{z}_j\}_{j=1}^{M}\}}{\arg\min} \sum_{j=1}^{M}\sum_{i=1}^{N} \underbrace{\|\Theta_\theta(\Phi_{\Psi(\mathbf{z_j})}, \mathbf{E}_i^j, \mathbf{K}_i^j) - \mathcal{I}_i^j\|_2^2}_{\mathcal{L}_{\text{img}}} + \underbrace{\lambda_{dep}\|\min(\mathbf{d}^j_{i,final}, \mathbf{0})\|_2^2}_{\mathcal{L}_{\text{depth}}} +$$



L2 reconstruction loss

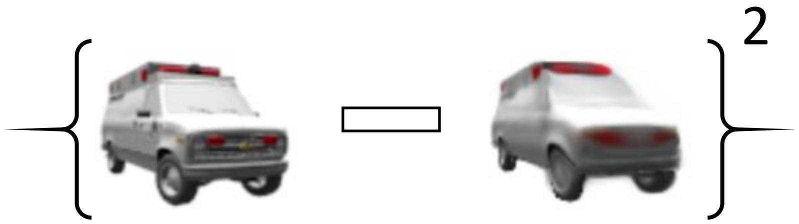


Positive depth

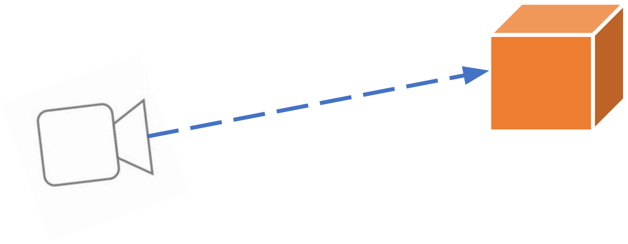
Optimization

Joint optimization using SGD:

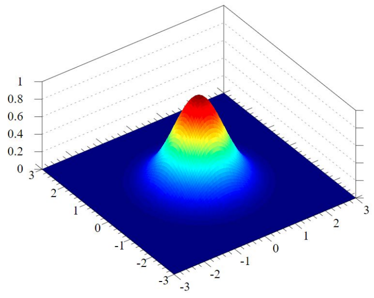
$$\arg\min_{\{\theta,\psi,\{\mathbf{z}_j\}_{j=1}^M\}} \sum_{j=1}^{M}\sum_{i=1}^{N} \underbrace{\|\Theta_\theta(\Phi_{\Psi(\mathbf{z_j})}, \mathbf{E}_i^j, \mathbf{K}_i^j) - \mathcal{I}_i^j\|_2^2}_{\mathcal{L}_{\text{img}}} + \underbrace{\lambda_{dep}\|\min(\mathbf{d}^j_{i,final}, \mathbf{0})\|_2^2}_{\mathcal{L}_{\text{depth}}} + \underbrace{\lambda_{lat}\|\mathbf{z}_j\|_2^2}_{\mathcal{L}_{\text{latent}}}.$$



L2 reconstruction loss



Positive depth



Gaussian Prior

Optimization

- Few-shot ($N = 1, 2$):

$$\hat{\mathbf{z}} = \arg\min_{\mathbf{z}} \sum_{i=1}^{N} \|\boxed{\Theta_\theta}(\boxed{\Phi_\Psi}_{(\mathbf{z})}, \mathbf{E}_i, \mathbf{K}_i) - \mathcal{I}_i\|_2^2 + \lambda_{dep}\| \min(\mathbf{d}_{i,final}, \mathbf{0})\|_2^2 + \lambda_{lat}\|\mathbf{z}\|_2^2$$

Trained beforehand

1st shot 2nd shot

Shepard Metzler

- 7 element objects
- Novel view synthesis on:
  - Training set
  - Few-shot on 100 test objects

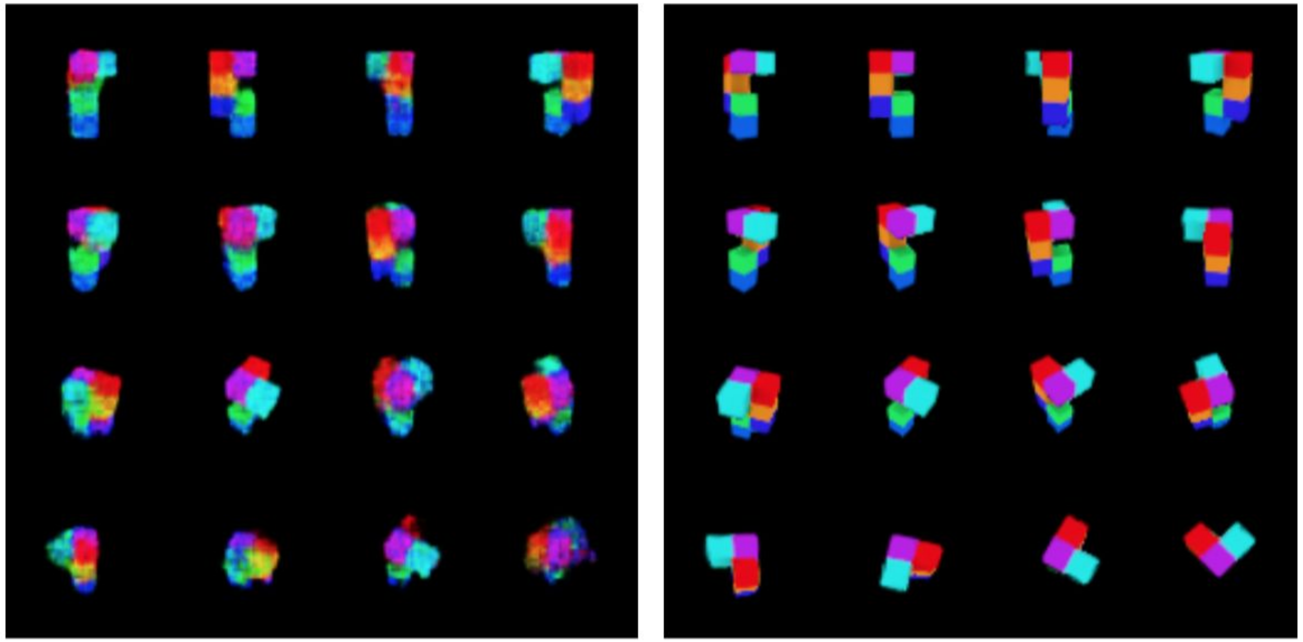


Figure 2: Shepard-Metzler object from 1k-object training set, 15 observations each. SRNs (right) outperform dGQN (left) on this small dataset.

ShapeNet

- Cars and Chairs
- Novel view synthesis on:
  - Training set.
  - Few-shot on official test objects.

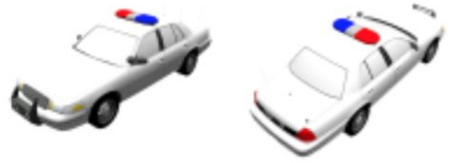


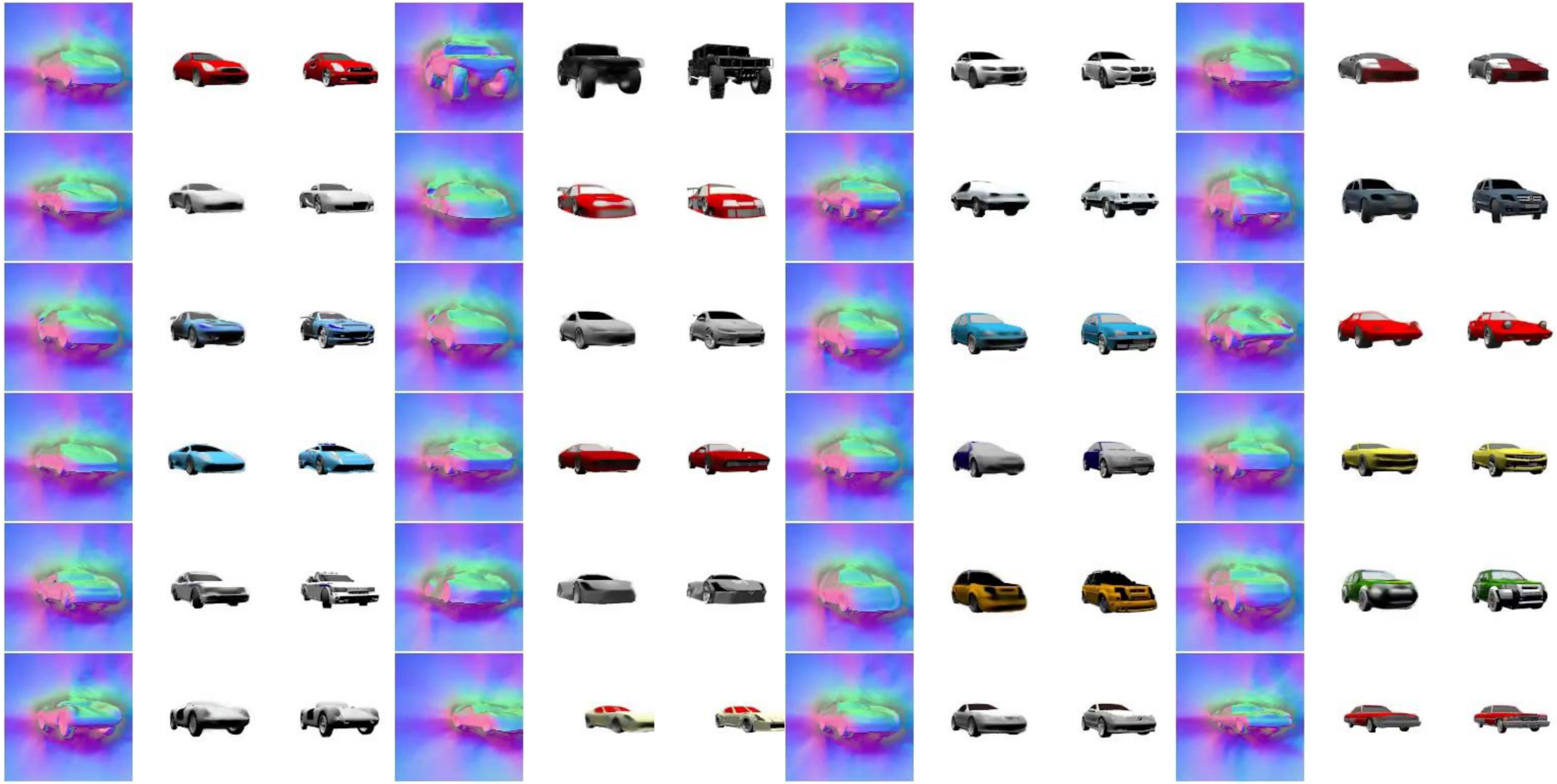
Figure 7: Single- (left) and two-shot (both) reference views.

	50 images (training set)		2 images		Single image	
	Chairs	Cars	Chairs	Cars	Chairs	Cars
TCO [1]	24.31 / 0.92	20.38 / 0.83	21.33 / 0.88	18.41 / 0.80	21.27 / 0.88	18.15 / 0.79
WRL [4]	24.57 / 0.93	19.16 / 0.82	22.28 / 0.90	17.20 / 0.78	22.11 / 0.90	16.89 / 0.77
dGQN [2]	22.72 / 0.90	19.61 / 0.81	22.36 / 0.89	18.79 / 0.79	21.59 / 0.87	18.19 / 0.78
SRNs	**26.23 / 0.95**	**26.32 / 0.94**	**24.48 / 0.92**	**22.94 / 0.88**	**22.89 / 0.91**	**20.72 / 0.85**

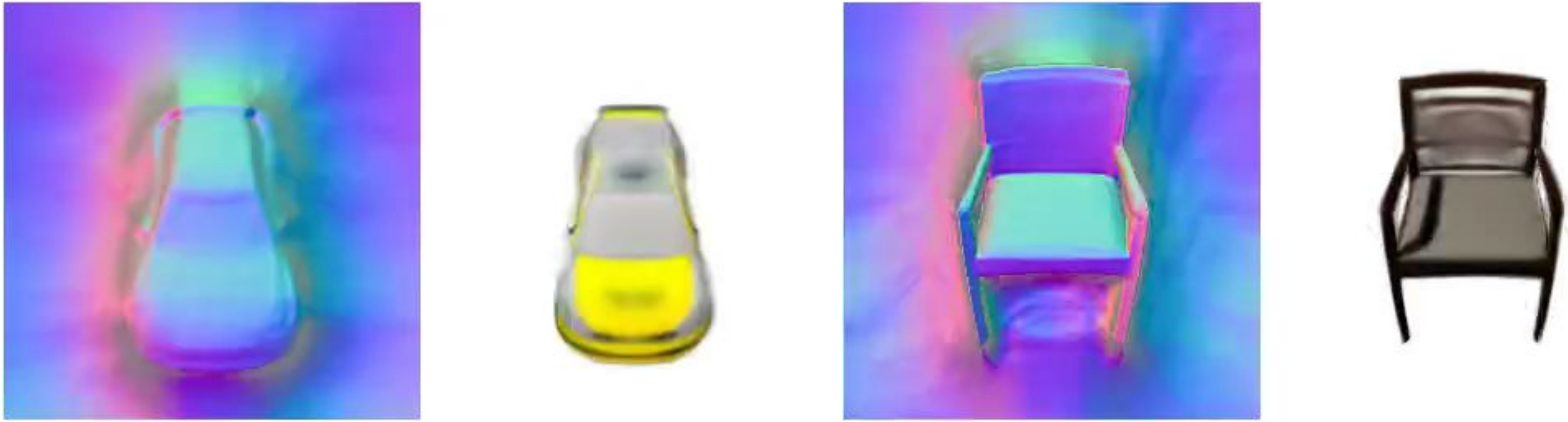
ShapeNet


dGQN
Tatarchenko et al.
SRNs
Ground Truth
50-shot
2-Shot
1-Shot

ShapeNet

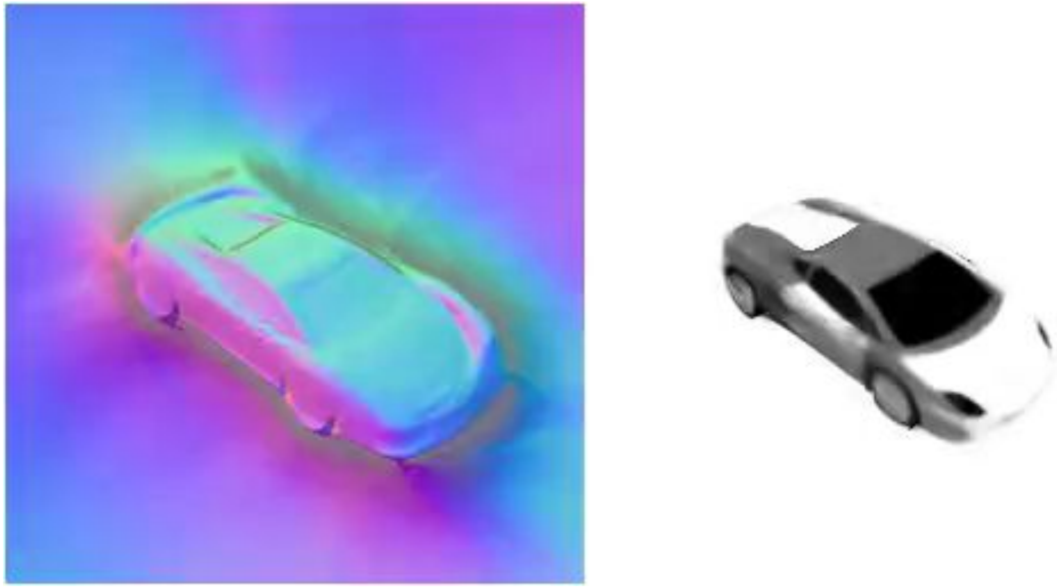


Latent space interpolation



Camera pose extrapolation

Camera zoom

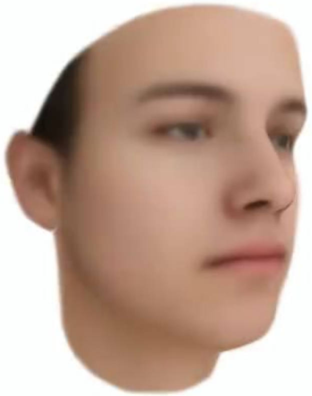


Camera rotation

Basel face model

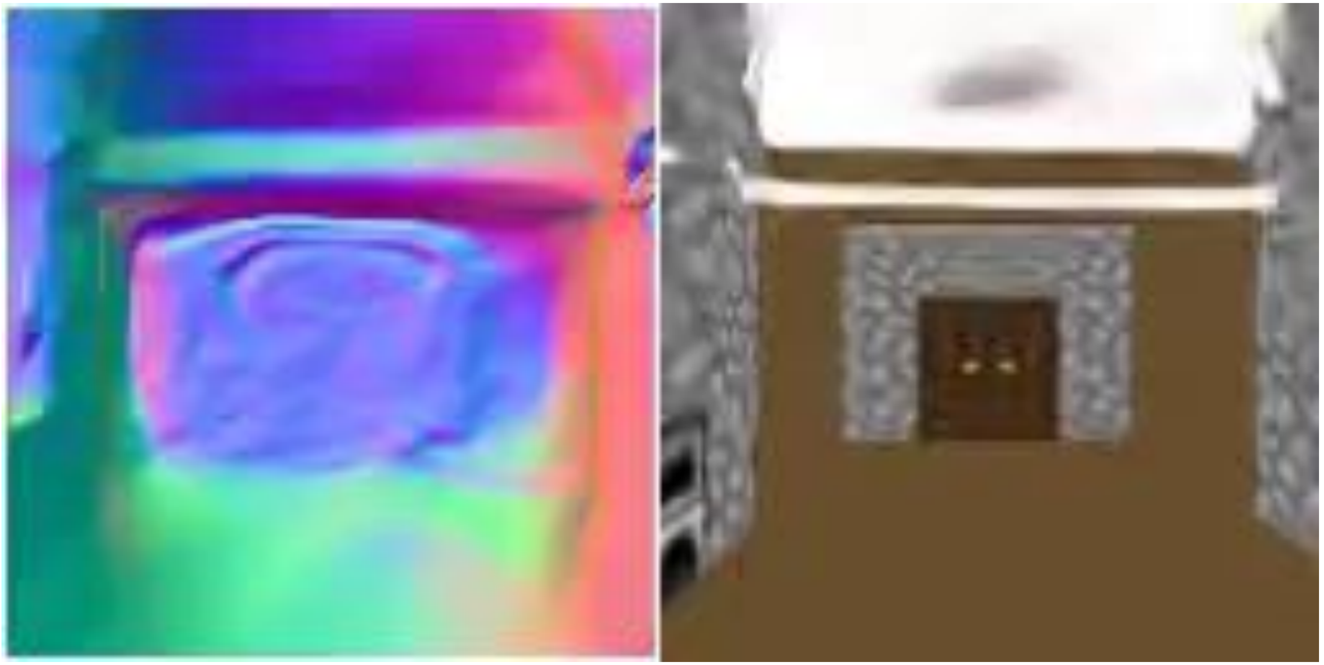
- Available disentangled latent:
  - Identity
  - Expression



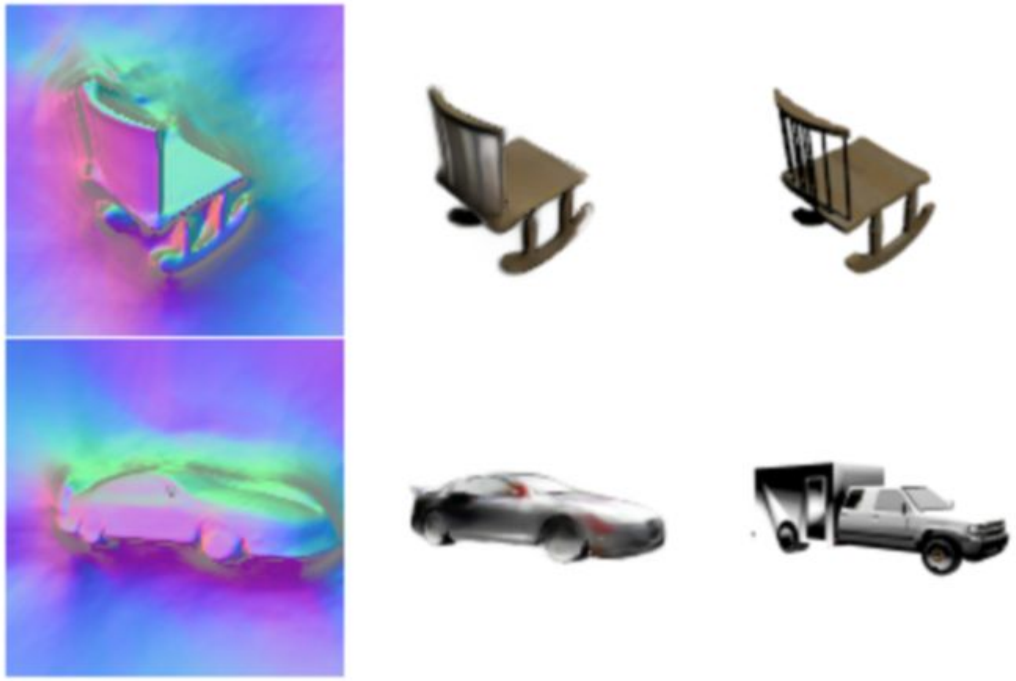
Minecraft room

Room scale scene



Critique / Limitations / Open Issues

- Availability of camera pose?
- Effects of view or lighting?
- Failure cases.



Critique / Limitations / Open Issues

- Modeling and architectural choices:
  - Pixel generator:
    - MLP vs CNN.
    - Texture details (Using positional encoding or sinusoidal activation function(Siren))
  - Computationally expensive hypernetwork ($\approx 10^7$ parameters).
    - What if we use Auto-Encoder? (instead of Auto-Decoder).
    - Meta learning (MetaSDF).
  - Ray marching
    - Expensive feed-forward of scene function for each step.
    - Weak convergence.

Contributions (Recap)

- A continuous, 3D structure aware, neural scene representation encoding geometry and appearance a multi-view consistent manner.
  - Along with a Differentiable ray marching algorithm for rendering.

- End-to-end training without explicit 3D supervision.

- Generalizable to other geometry or appearance.

- Evaluation in:
  - Novel view synthesis.
  - Few-shot reconstruction.
  - ...

Thank you!