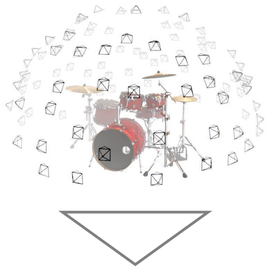
Neural Networks as PRIOR-BASED INFERENCE ALGORITHMS

6.S980 | 2022-05-19

Admin things

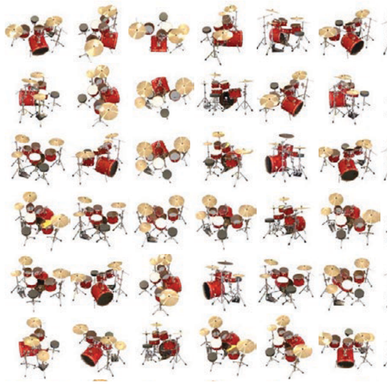
- Bugs in Section 2.1 of Homework 2:
  - Asked you to have a sigmoid as final layer. Should've been just a linear layer.
  - Should have 3 input dimensions.
- Still some students missing on paper session.

Recap: Differentiable Rendering

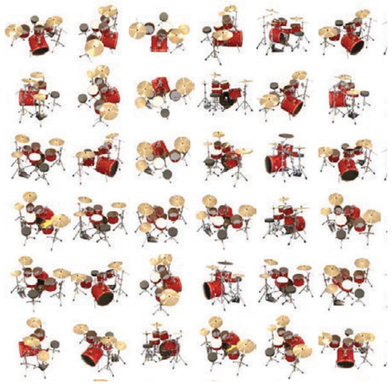


Scene Representation

Neural Renderer



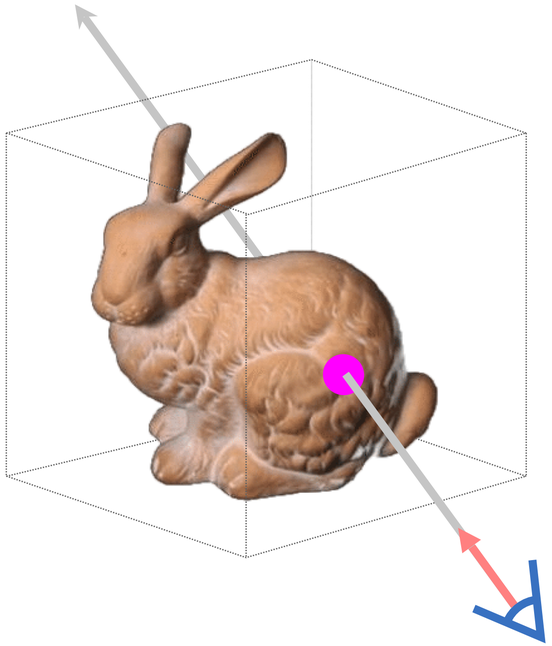
Rendered Images

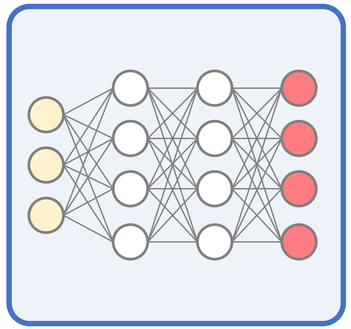


GT Images

Optimization via SGD

General structure of Neural Renderers for 3D Fields

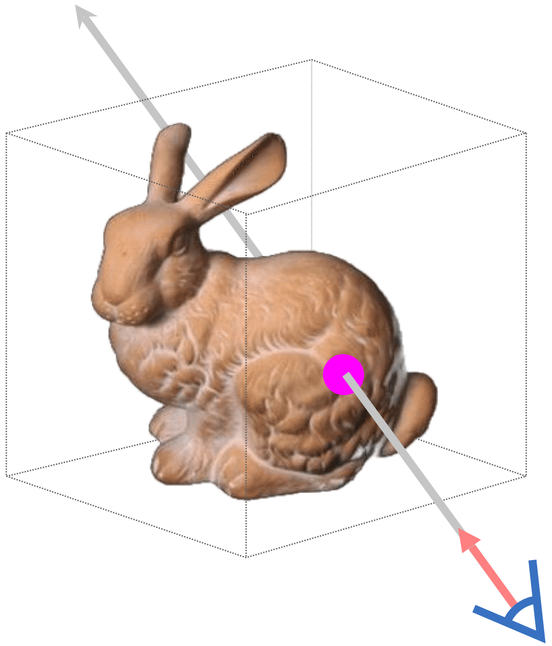


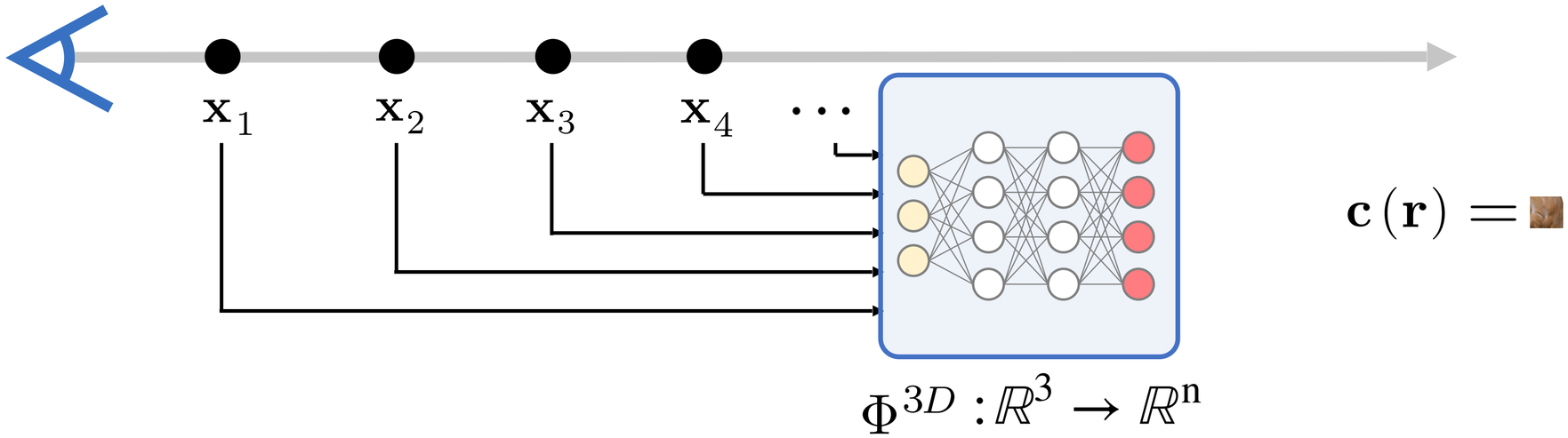


$$\Phi^{3D} : \mathbb{R}^3 \rightarrow \mathbb{R}^n$$

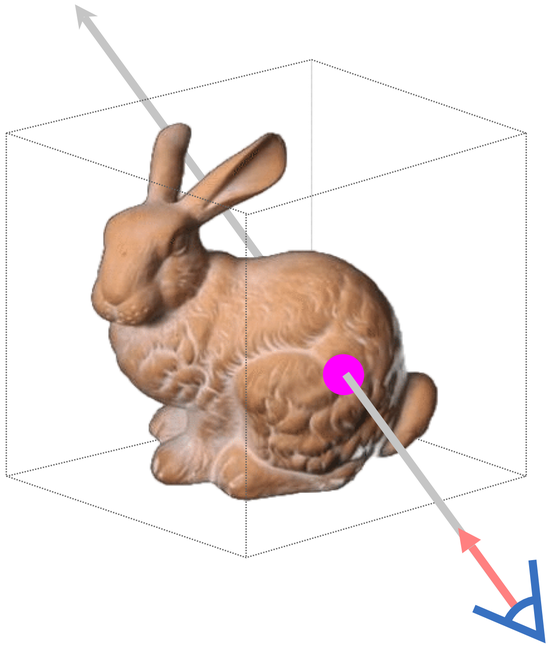
$$\mathbf{c}(\mathbf{r}) =$$ 

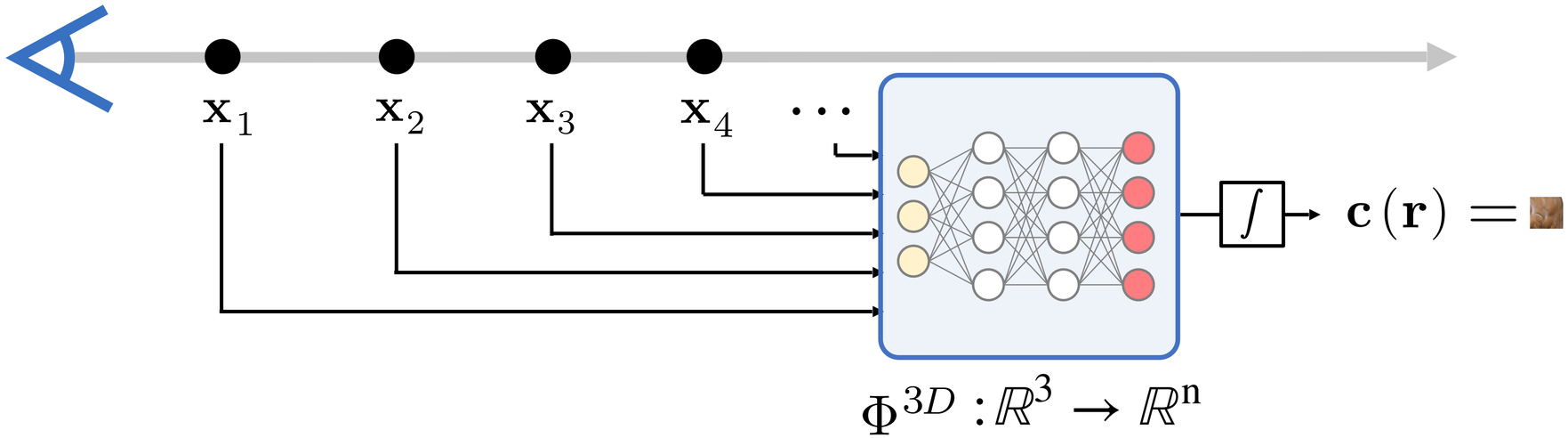
General structure of Neural Renderers for 3D Fields



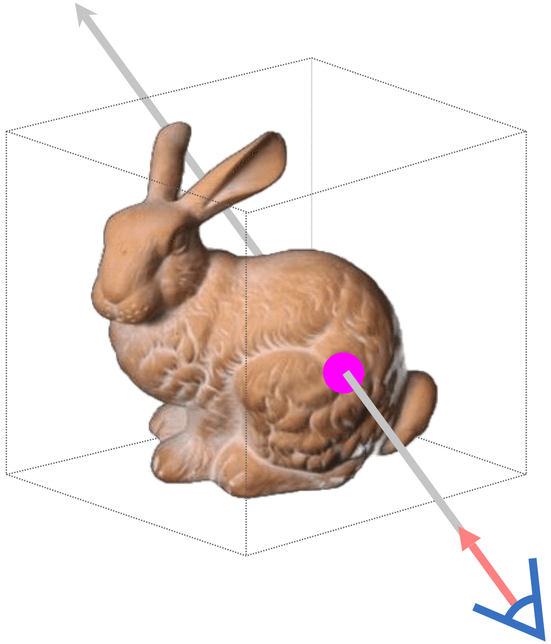


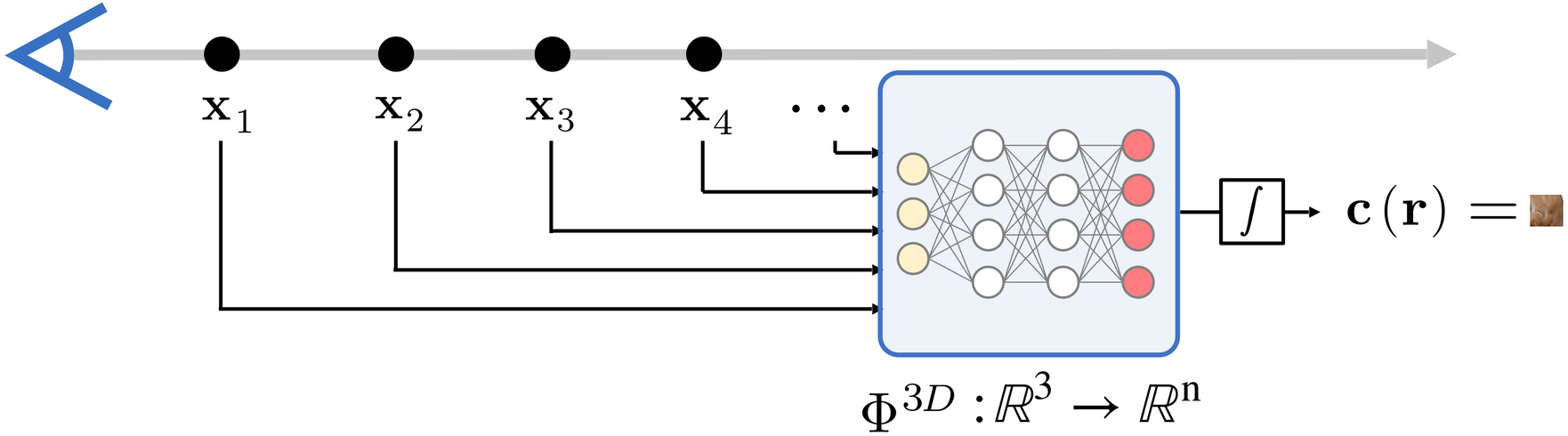
General structure of Neural Renderers for 3D Fields





General structure of Neural Renderers for 3D Fields





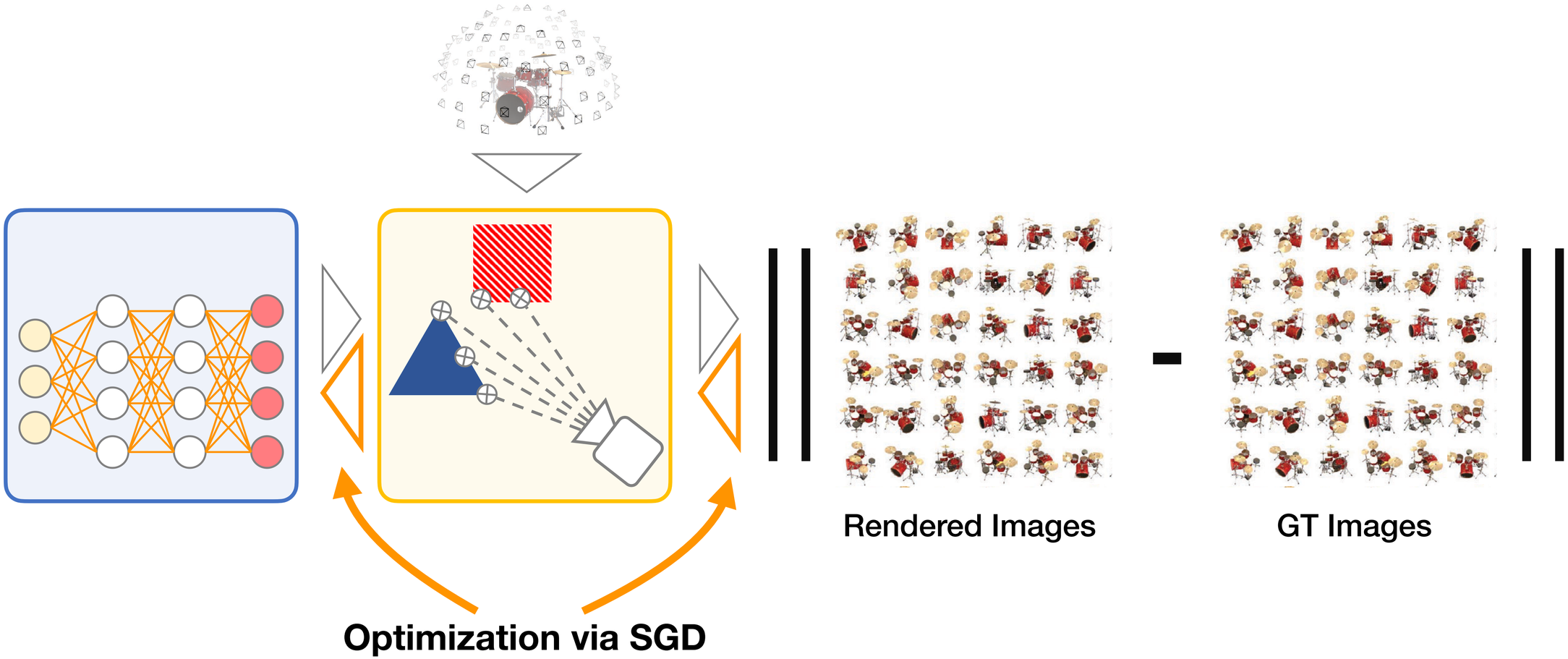
Sphere-Tracing [JC Hart, 1996]

Volumetric

Hybrid implicit-volumetric

Learned aggregation

What did we do here?

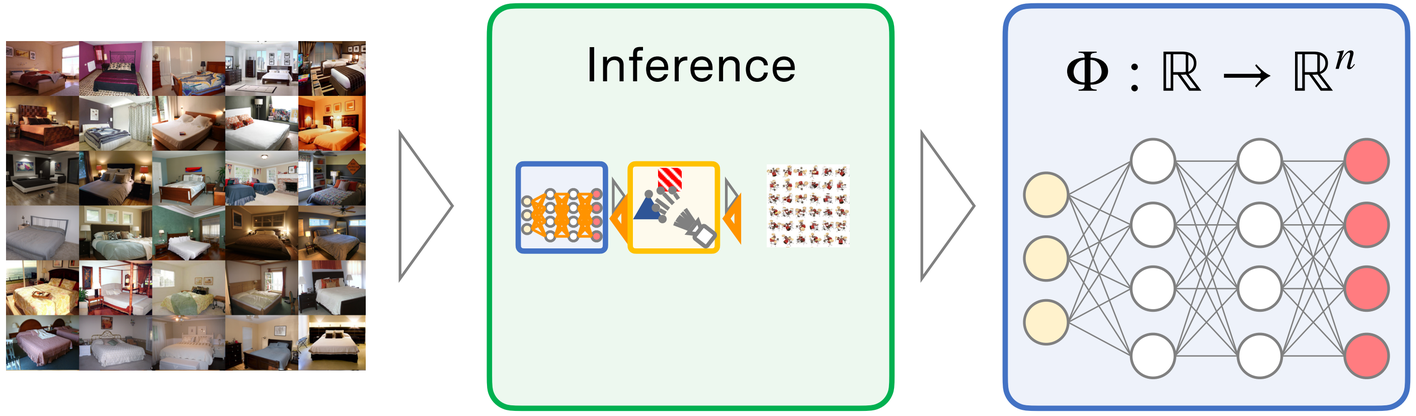


What did we do here?

Observations

What did we do here?

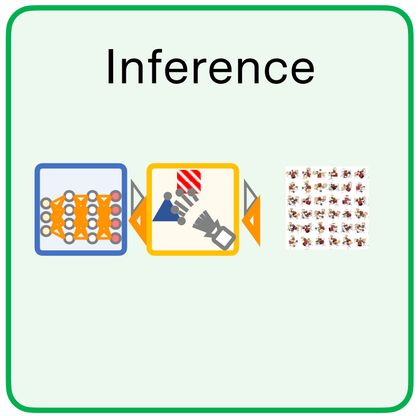
Observations

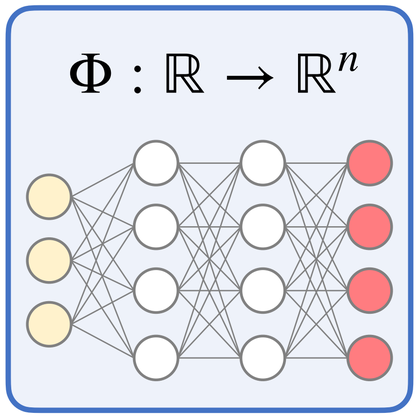


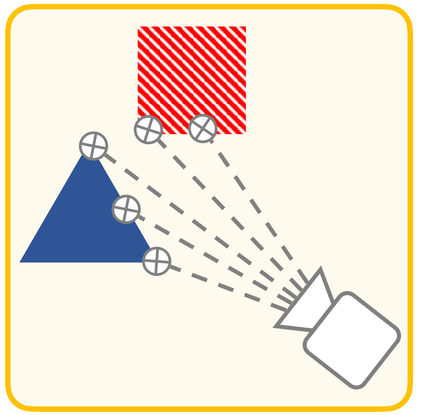
We inferred “hidden/latent” variables 3D appearance and geometry, given “observable” variables RGB images and camera poses.

What did we do here?

Observations



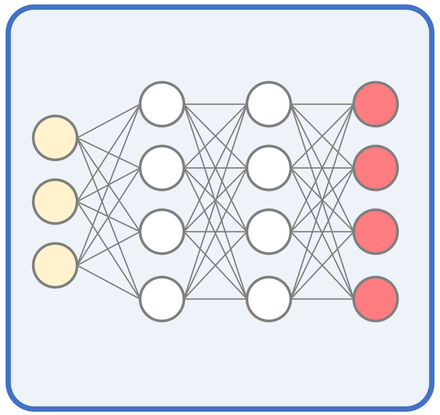




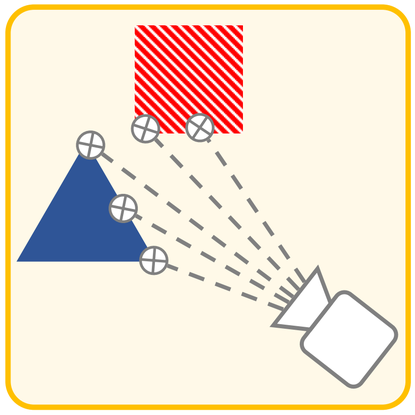
Renderings

We inferred “hidden/latent” variables 3D appearance and geometry, given “observable” variables RGB images and camera poses.

What if observations don't constrain scene representation?

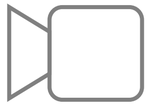


Srn_{ϕ}



Render_{θ}

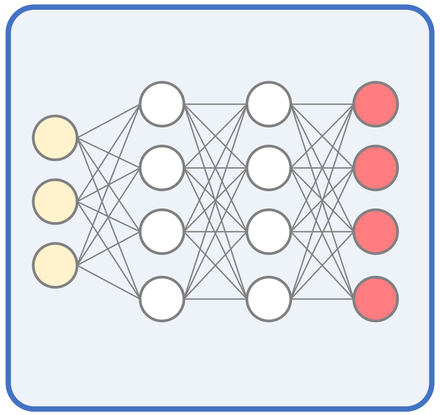
$+$



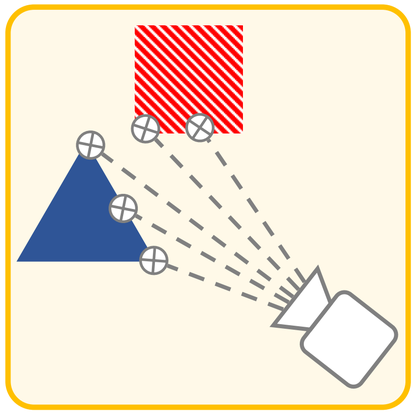
$(\mathcal{I}, \xi)$

$$\underset{\phi,\theta}{\text{argmin}} \left\| \text{Render}_{\theta}(\text{Srn}_{\phi}, \xi) - \mathcal{I} \right\|$$

What if observations don't constrain scene representation?



Srn_{ϕ}



Render_{θ}

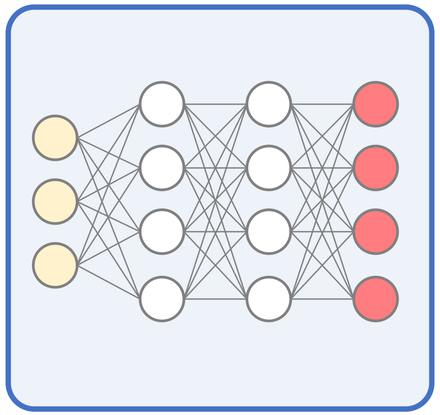
$+$

$(\mathcal{I}, \xi)$

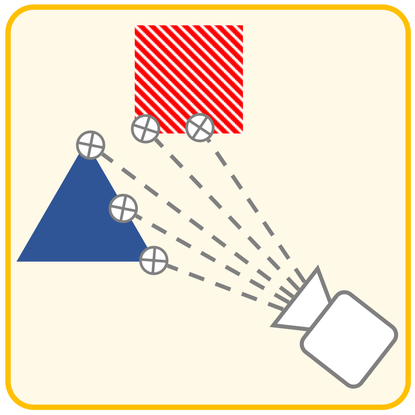
No 3D inform.

$$\underset{\phi,\theta}{\text{argmin}} \left\| \text{Render}_{\theta}(\text{Srn}_{\phi}, \xi) - \mathcal{I} \right\|$$

What if observations don't constrain scene representation?



Srn_{ϕ}



Render_{θ}

$+$

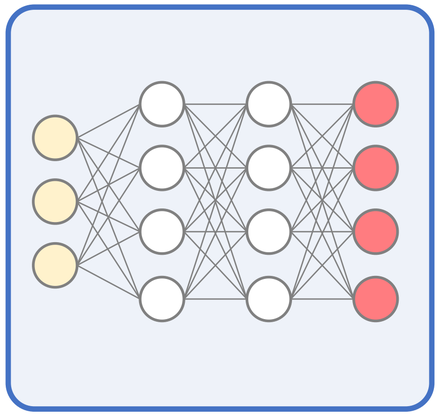
$(\mathcal{I}, \xi)$

No 3D inform.

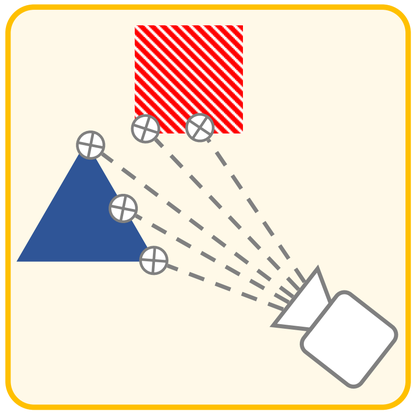
Input

$$\underset{\phi,\theta}{\text{argmin}} \left\| \text{Render}_{\theta}(\text{Srn}_{\phi}, \xi) - \mathcal{I} \right\|$$

What if observations don't constrain scene representation?



Srn_{ϕ}



Render_{θ}

$+$

$(\mathcal{I}, \xi)$

No 3D inform.

$$\underset{\phi,\theta}{\text{argmin}} \left\| \text{Render}_{\theta}(\text{Srn}_{\phi}, \xi) - \mathcal{I} \right\|$$

Input

Normal map

RGB

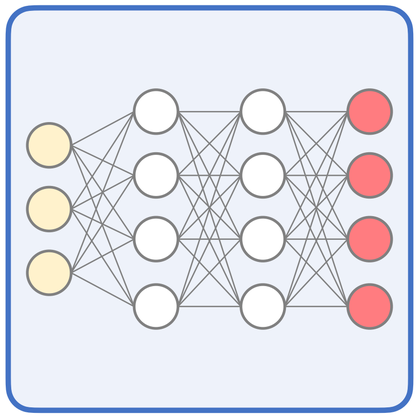
GT

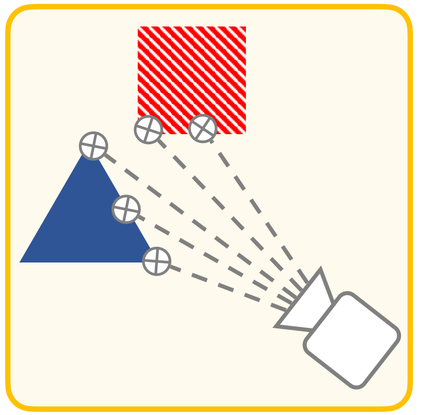
Today: *Learned* inference algorithms!

Observations

Inference

?

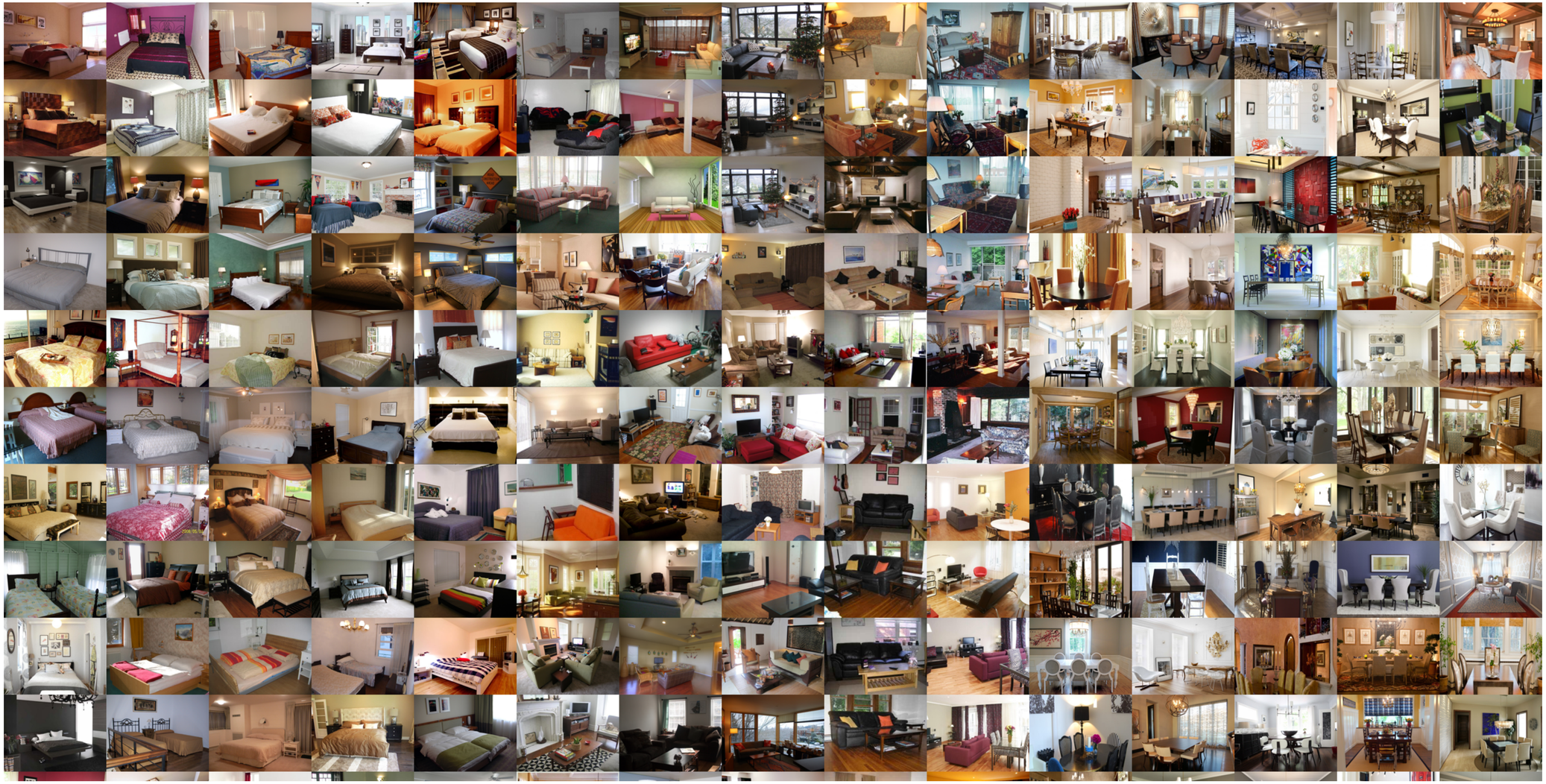




Renderings

Why?	**We humans can reconstruct 3D from incomplete observations, by using knowledge that we have learned about the world. Deep learning is the best way we know to date to learn such priors from data.**
What you'll learn.	**How to express priors over 3D scenes using deep learning, different ways of doing inference (encoding, auto-decoding)**

Unsupervised Learning: What priors can we learn from *observations only?*



Auto-Encoding

$\mathbf{x}$

Image

Auto-Encoding

$\mathbf{x}$

Image

"Auto-Encoder"

$\mathcal{F}$

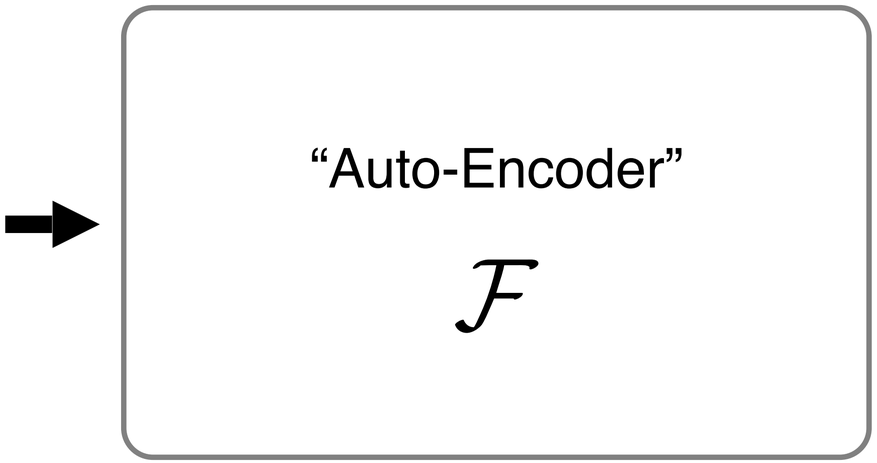
$\hat{\mathbf{x}}$

Reconstructed image

Auto-Encoding

$\mathbf{X}$

Image



$\hat{\mathbf{X}}$

Reconstructed image

$$\arg\min_{\mathcal{F}} \mathbb{E}_{\mathbf{X}}[||\mathcal{F}(\mathbf{X}) - \mathbf{X}||]$$

Auto-Encoding

$\mathbf{x}$

Image

$\hat{\mathbf{x}}$

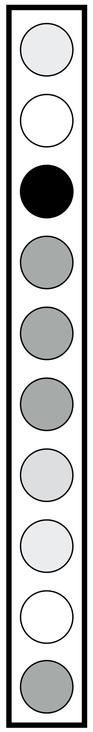
Reconstructed image

[e.g., Hinton & Salakhutdinov, Science 2006]

Auto-Encoding

$\mathbf{x}$

Image



$\hat{\mathbf{x}}$

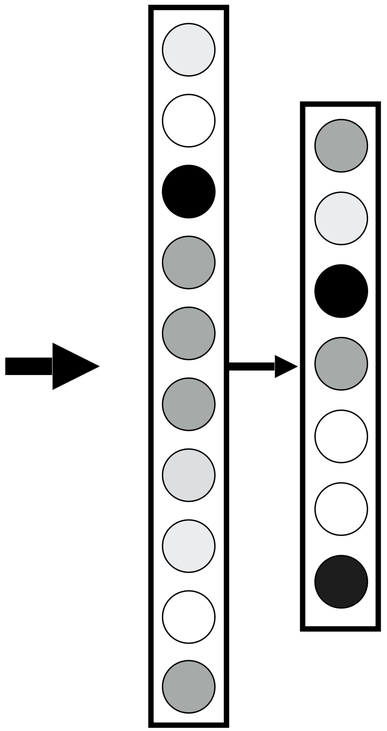
Reconstructed image

[e.g., Hinton & Salakhutdinov, Science 2006]

Auto-Encoding

$\mathbf{x}$

Image



$\hat{\mathbf{x}}$

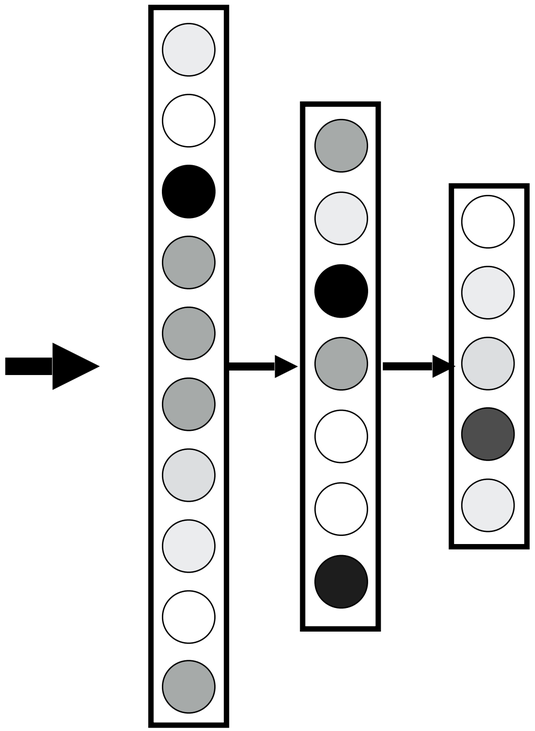
Reconstructed image

[e.g., Hinton & Salakhutdinov, Science 2006]

Auto-Encoding

$\mathbf{x}$

Image



$\hat{\mathbf{x}}$

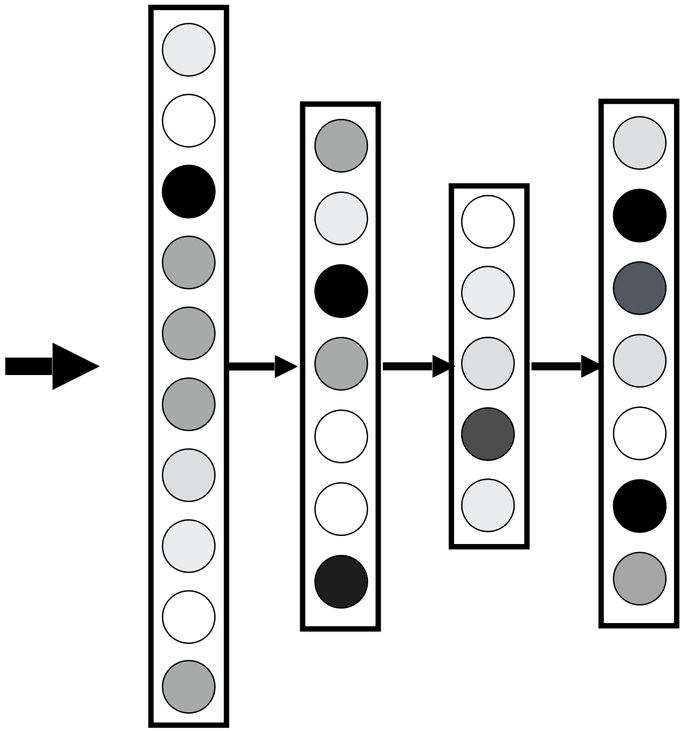
Reconstructed image

[e.g., Hinton & Salakhutdinov, Science 2006]

Auto-Encoding

$\mathbf{x}$

Image



$\hat{\mathbf{x}}$

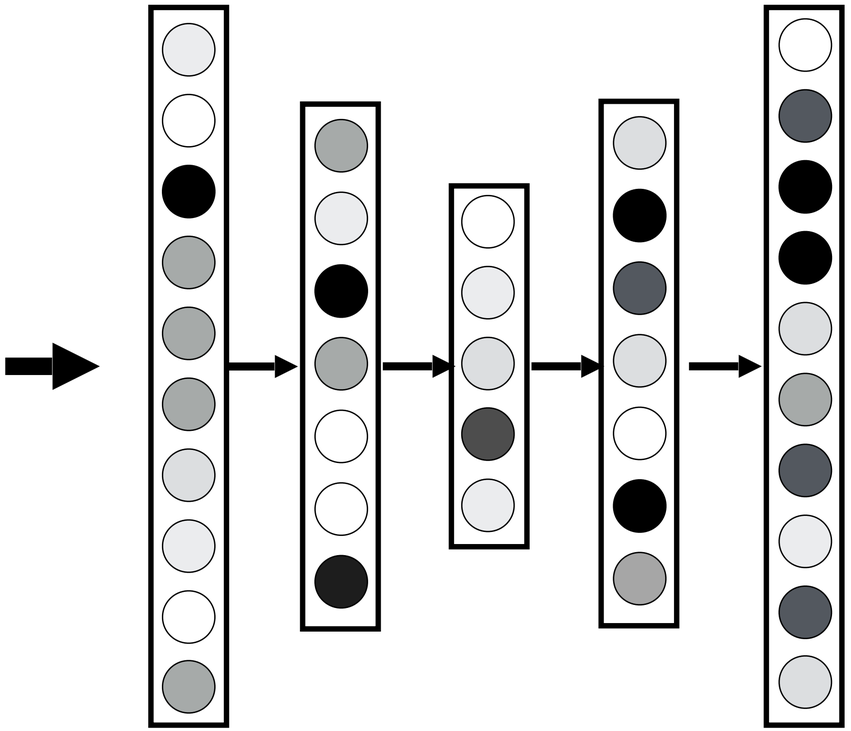
Reconstructed image

[e.g., Hinton & Salakhutdinov, Science 2006]

Auto-Encoding

$\mathbf{x}$

Image

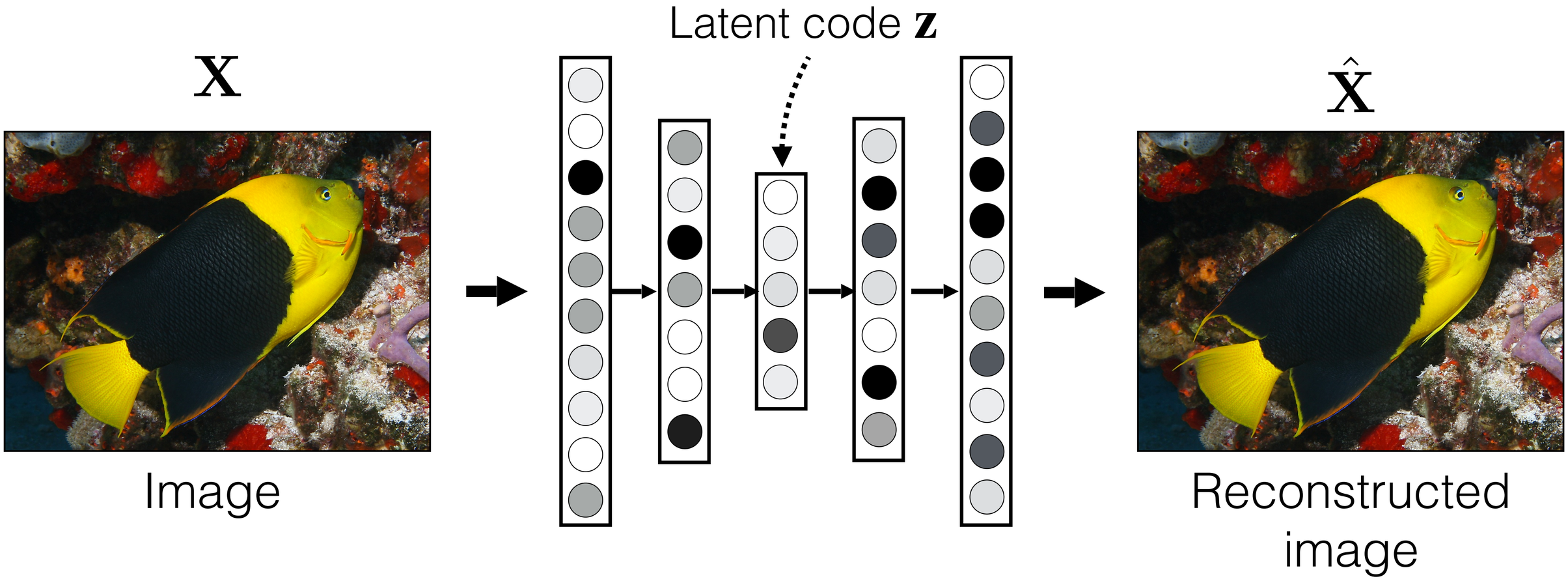


$\hat{\mathbf{x}}$

Reconstructed image

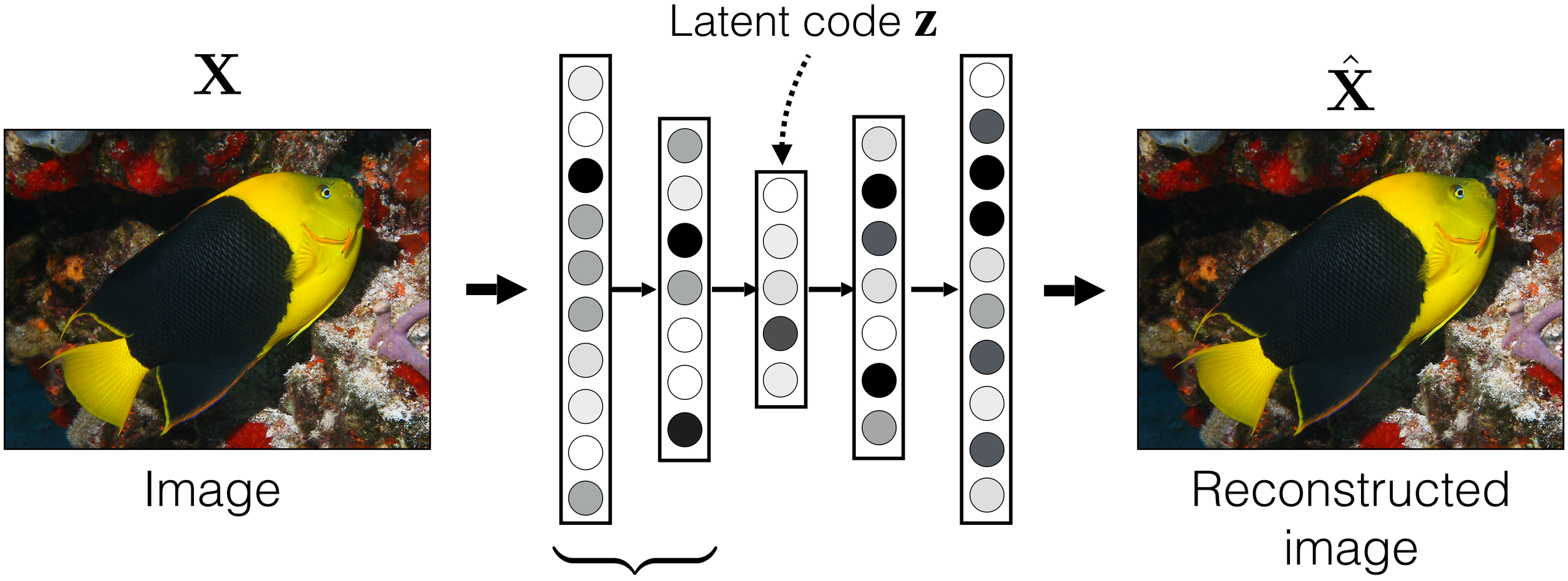
[e.g., Hinton & Salakhutdinov, Science 2006]

Auto-Encoding



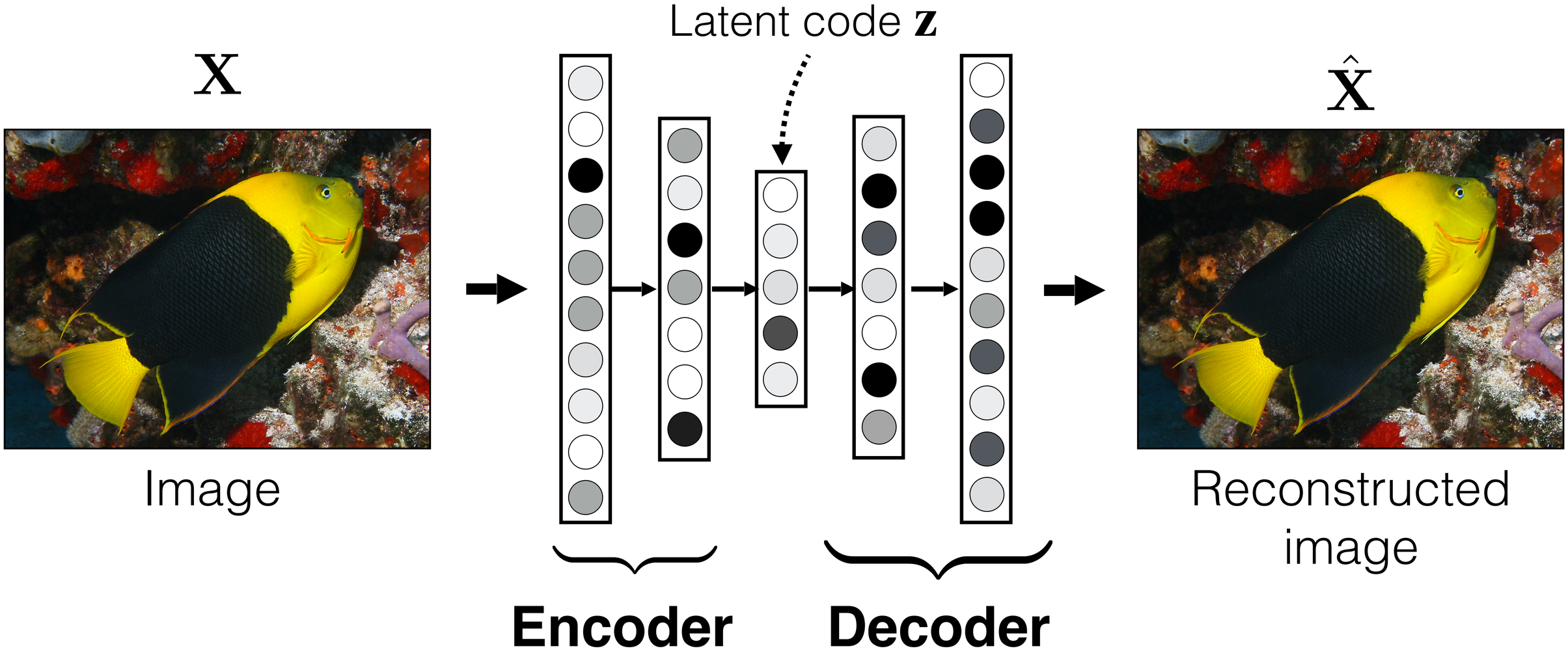
[e.g., Hinton & Salakhutdinov, Science 2006]

Auto-Encoding

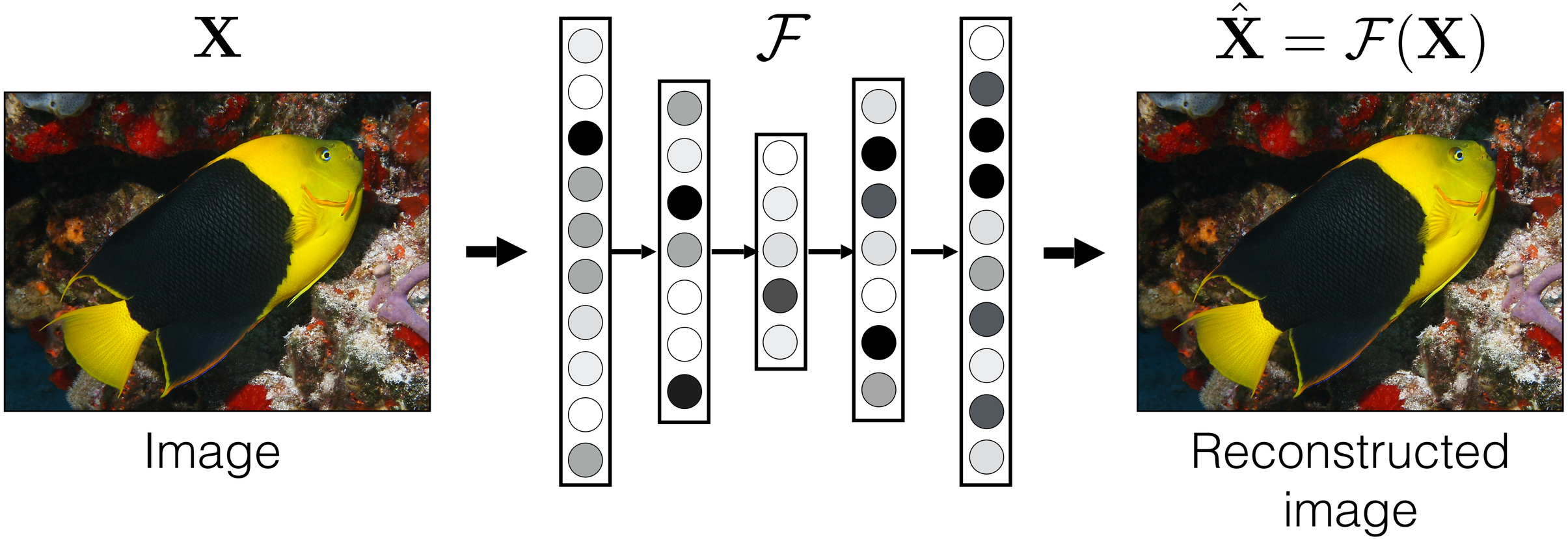


[e.g., Hinton & Salakhutdinov, Science 2006]

Auto-Encoding



[e.g., Hinton & Salakhutdinov, Science 2006]



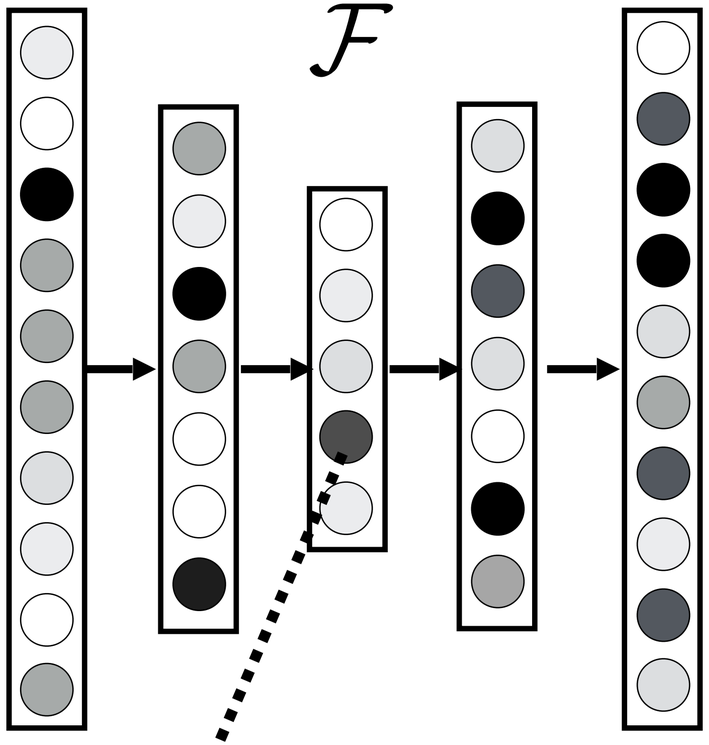
Learning Signal: **Forced Compression**

$\mathbf{X}$

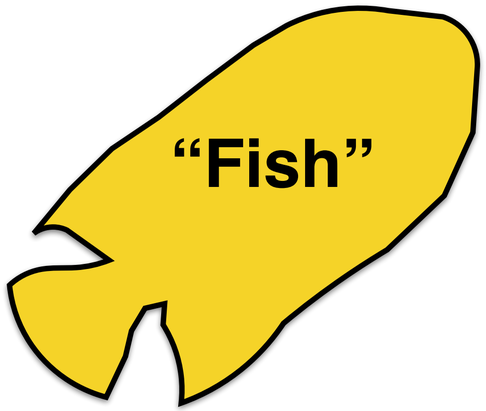
$\mathcal{F}$

$\hat{\mathbf{X}} = \mathcal{F}(\mathbf{X})$

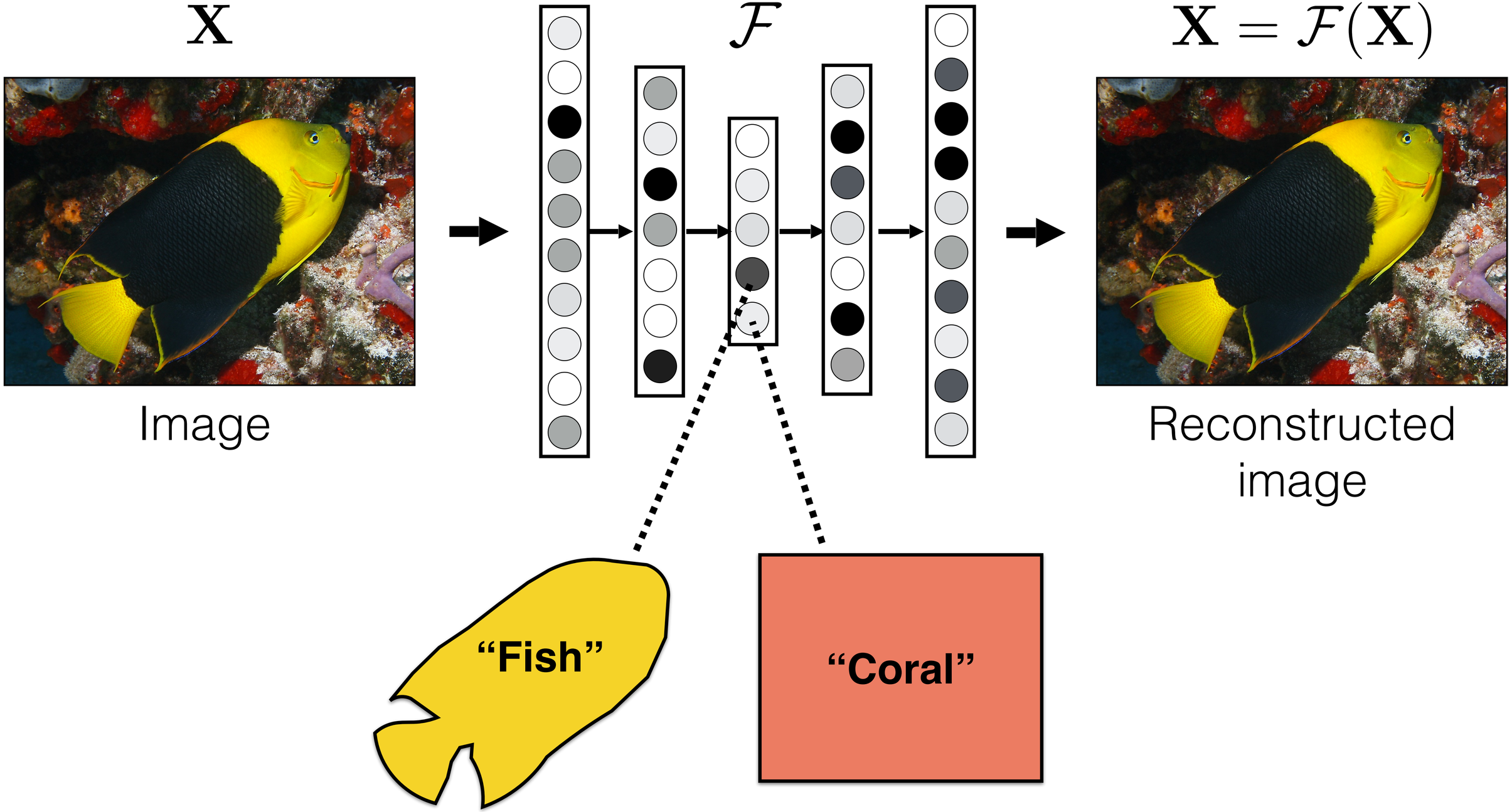
Image



Reconstructed image

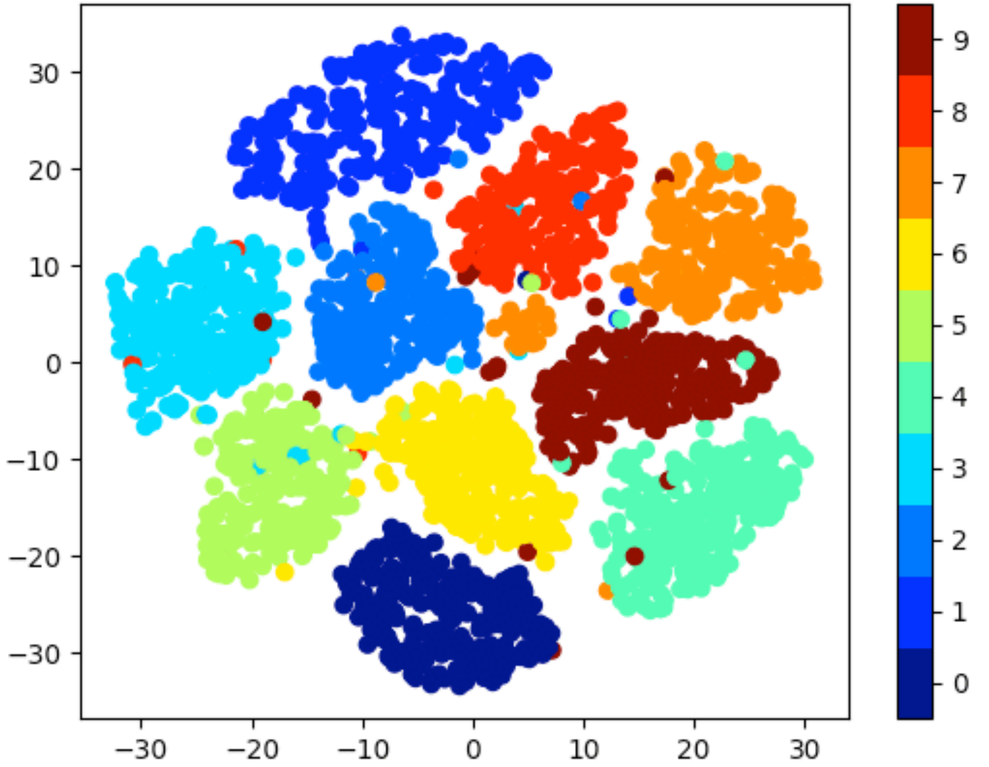


Learning Signal: **Forced Compression**



Learning Signal: **Forced Compression**

On MNIST Digits: Clustering Latent Variables $\mathbf{z}$



Self-supervised Scene Representation Learning

Latent 3D Scenes

{  ,  , . . . }

Self-supervised Scene Representation Learning

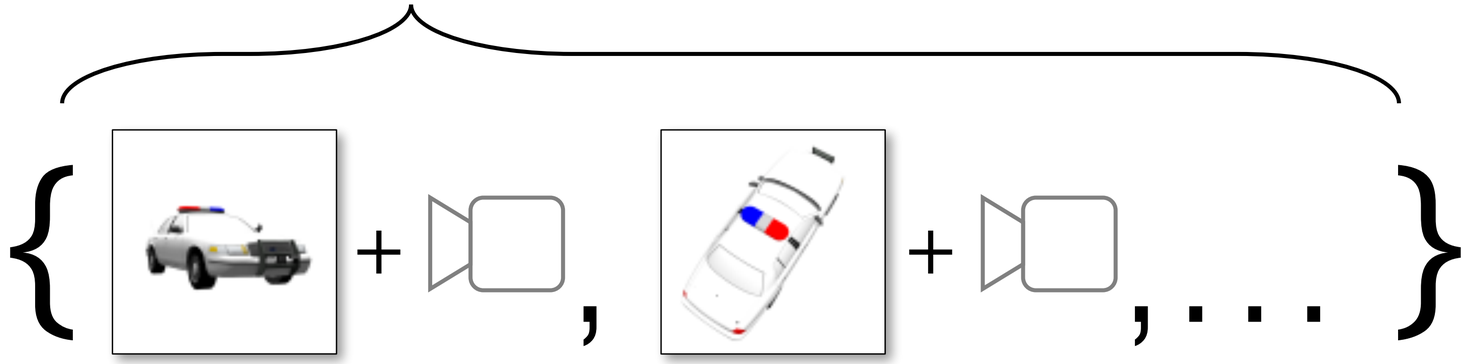
Latent 3D Scenes

{  ,  , . . . }

Self-supervised Scene Representation Learning

Latent 3D Scenes

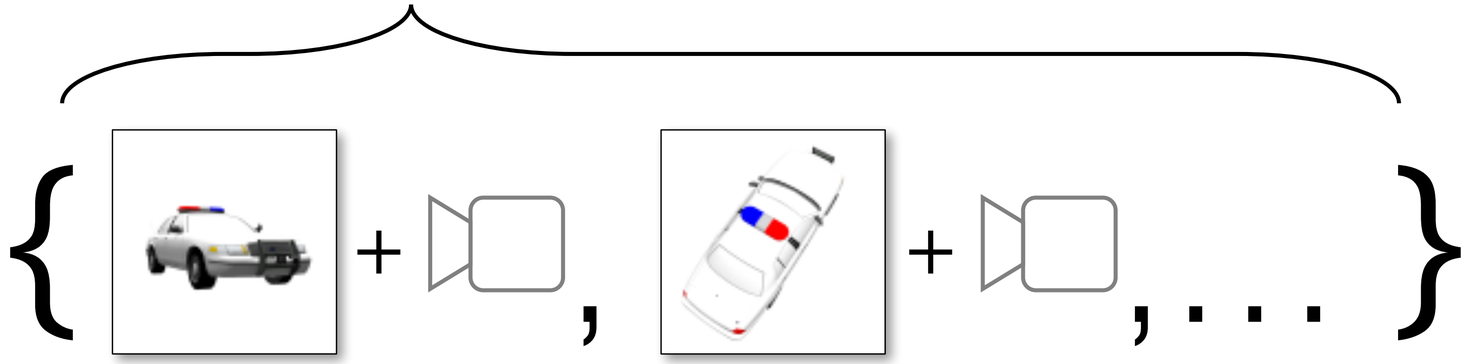
Observations
Image + Pose & Intrinsics



Self-supervised Scene Representation Learning

Latent 3D Scenes

Observations
Image + Pose & Intrinsics



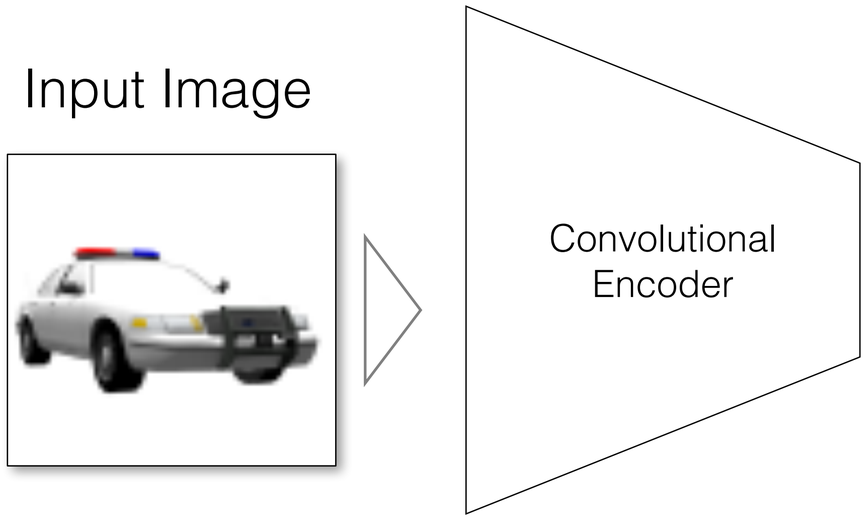
What can we learn about latent 3D scenes from observations?

Vision: Learn rich representations just by watching video!

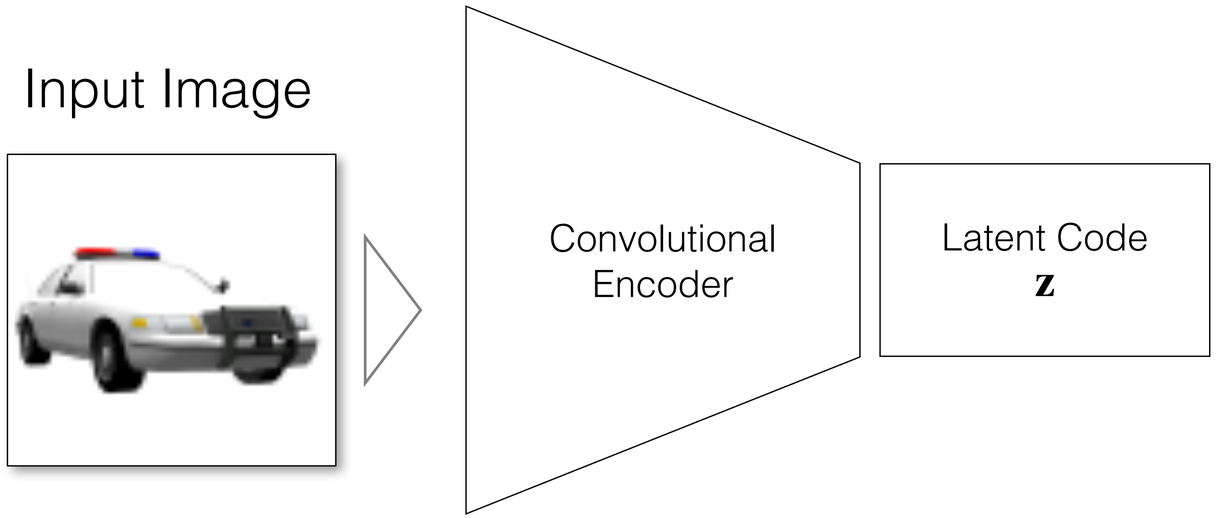
2D baseline: Auto-Encoder-Like model

Input Image

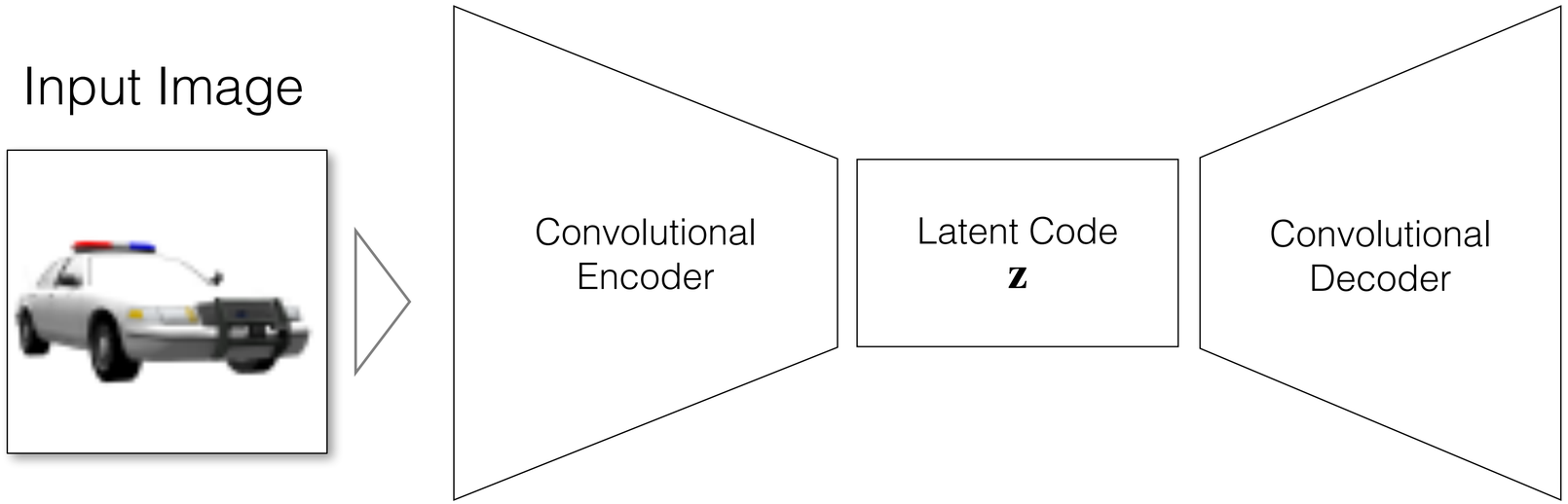
2D baseline: Auto-Encoder-Like model



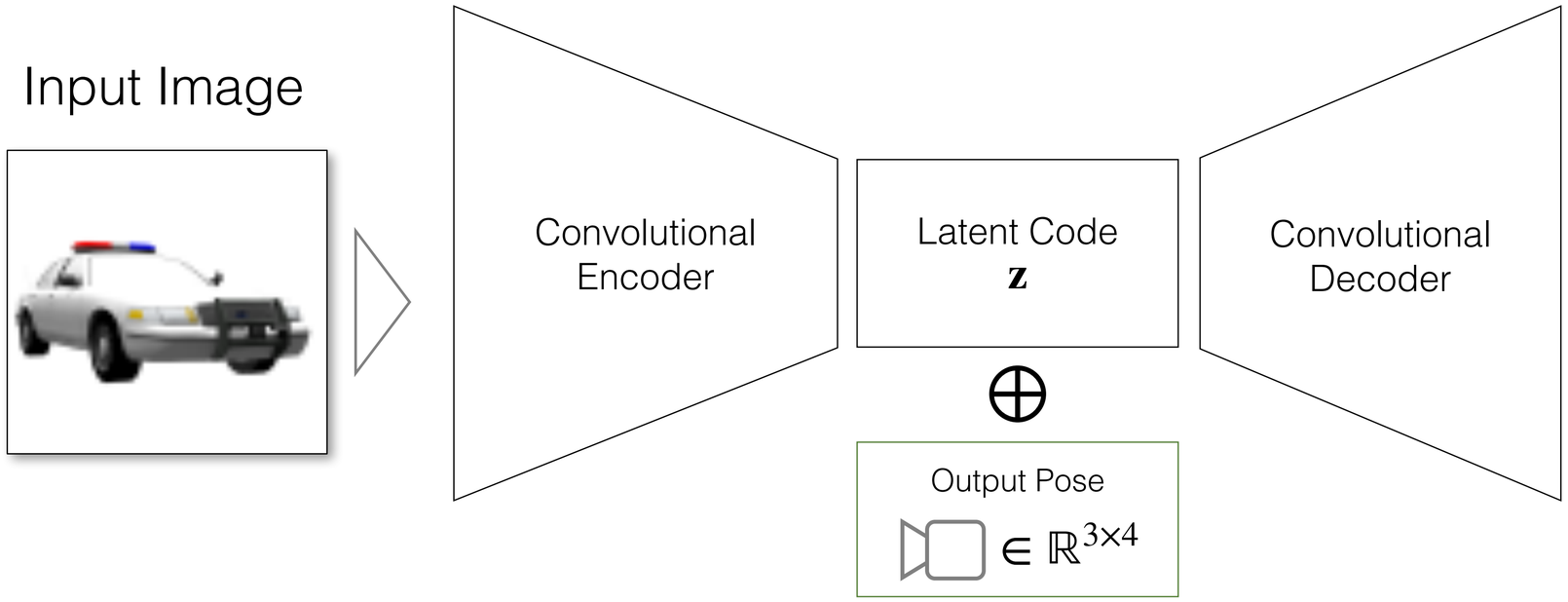
2D baseline: Auto-Encoder-Like model



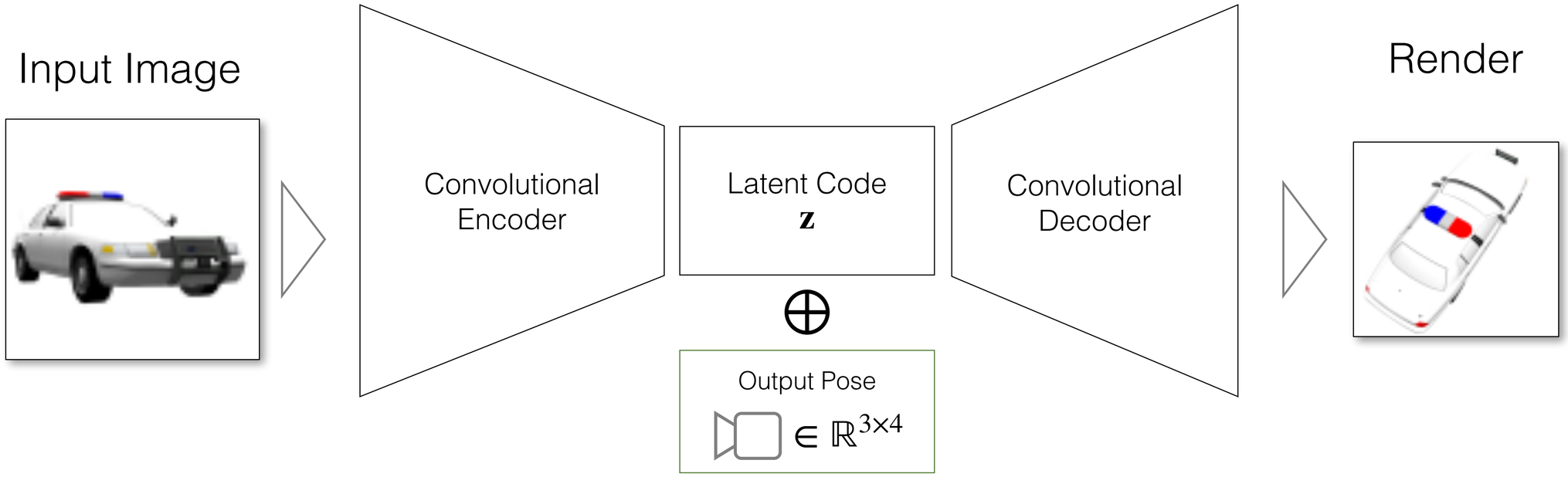
2D baseline: Auto-Encoder-Like model



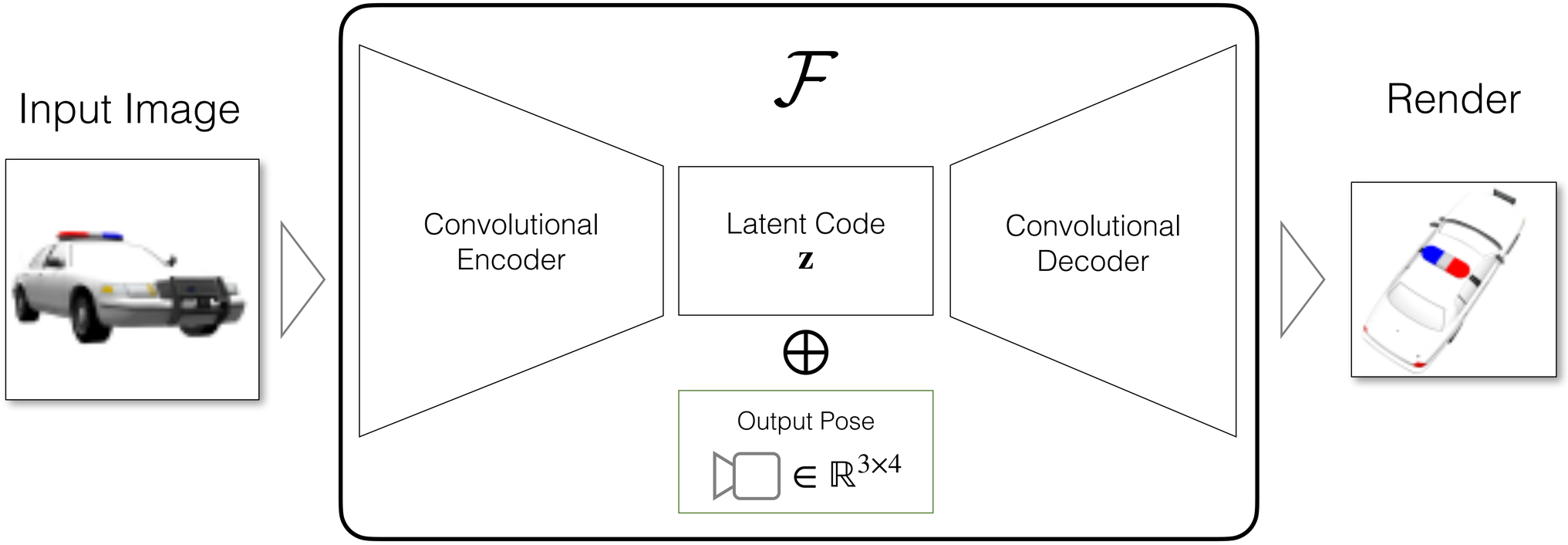
2D baseline: Auto-Encoder-Like model



2D baseline: Auto-Encoder-Like model

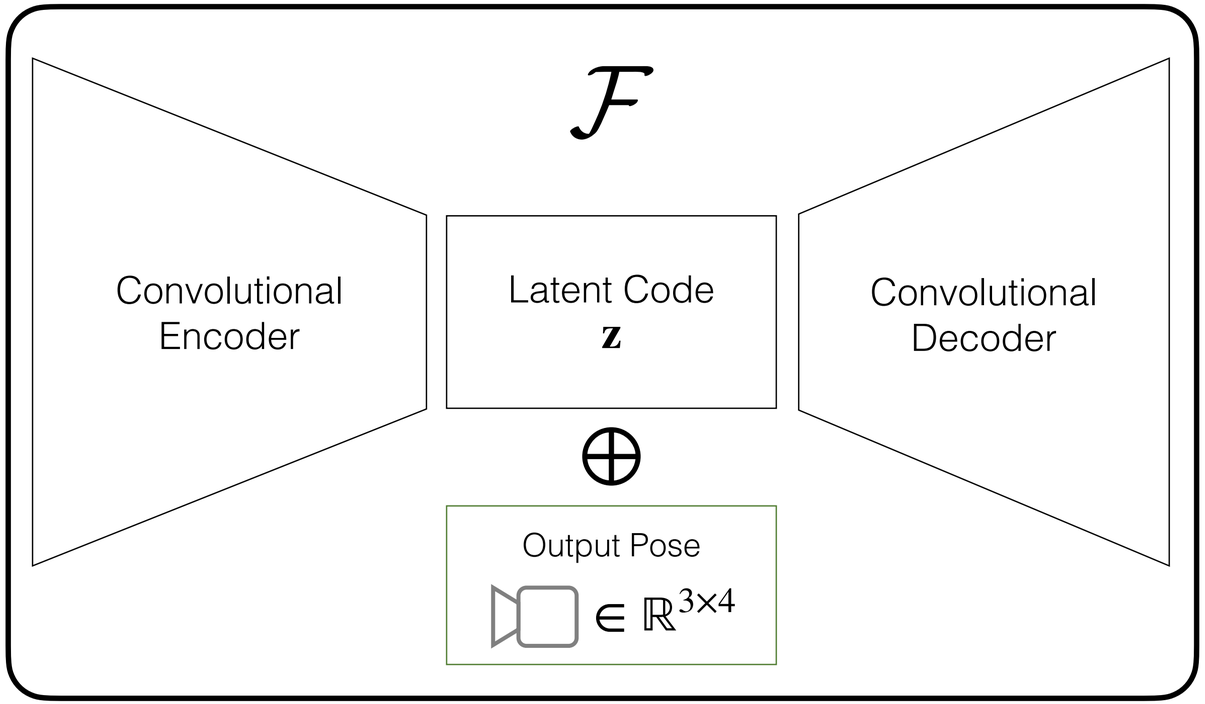


2D baseline: Auto-Encoder-Like model

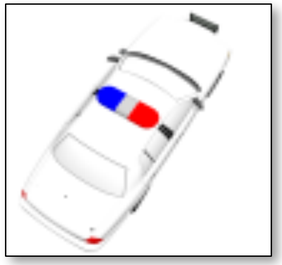


2D baseline: Auto-Encoder-Like model

Input Image

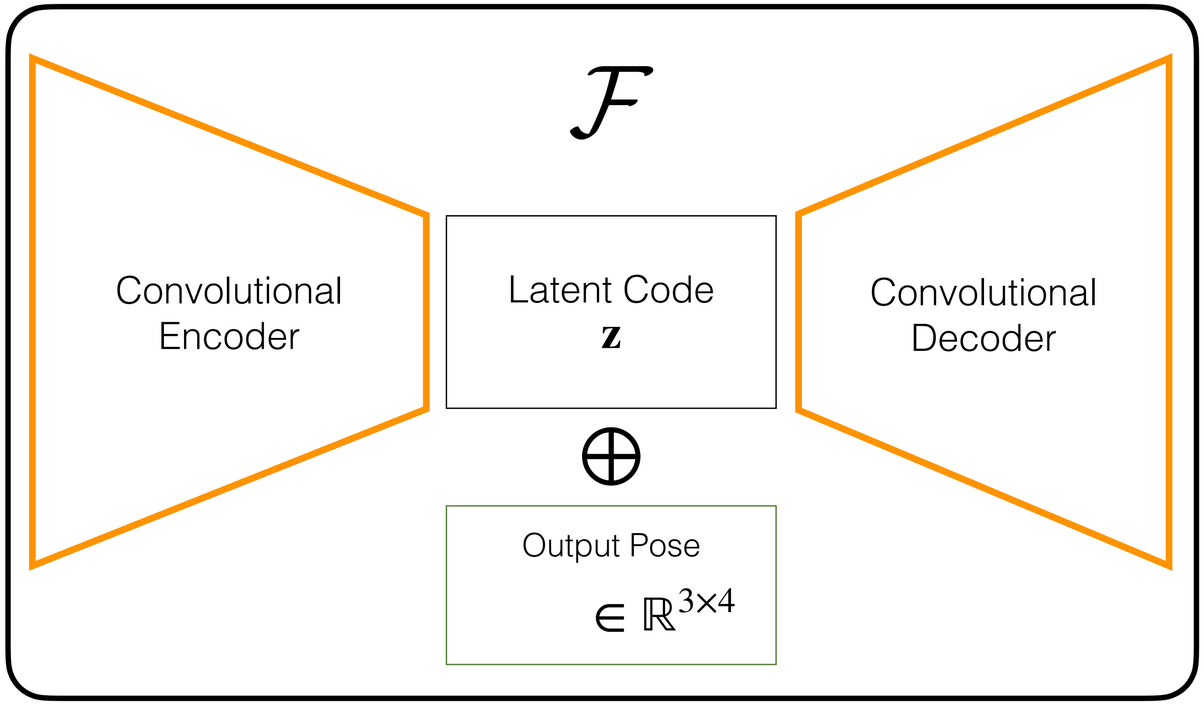


Render



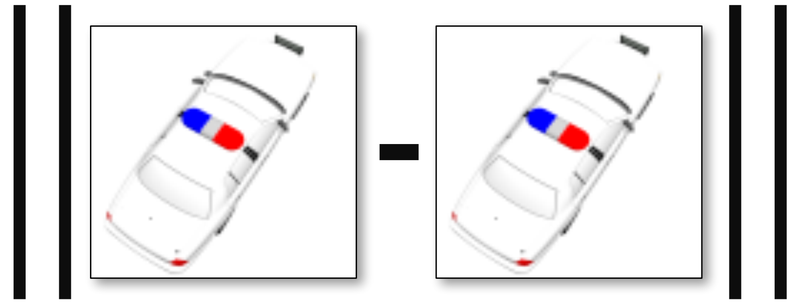
2D baseline: Auto-Encoder-Like model

Input Image

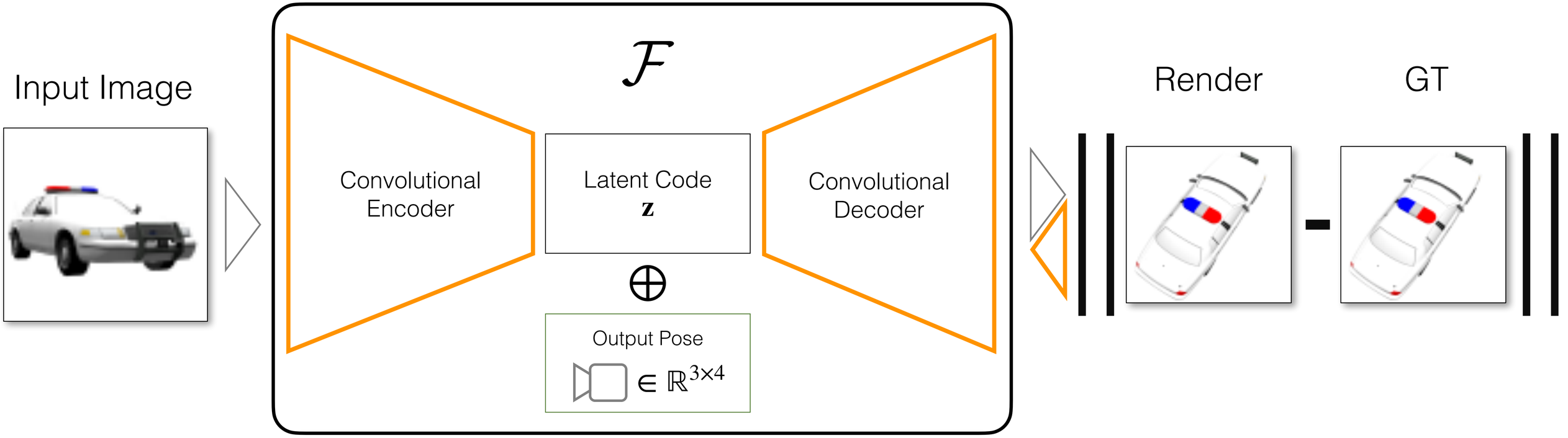


Render

GT



2D baseline: Auto-Encoder-Like model



Doesn’t capture 3D properties of scenes.

Trained on ~2500 shapenet cars with 50 observations each.

Doesn’t capture 3D properties of scenes.

Trained on ~2500 shapenet cars with 50 observations each.

Doesn’t capture 3D properties of scenes.

Trained on ~2500 shapenet cars with 50 observations each.

Doesn’t appear to have discovered 3D.
Why?

Doesn’t capture 3D properties of scenes.

Trained on ~2500 shapenet cars with 50 observations each.

Doesn’t capture 3D properties of scenes.

Trained on ~2500 shapenet cars with 50 observations each.

Doesn’t capture 3D properties of scenes.

Trained on ~2500 shapenet cars with 50 observations each.

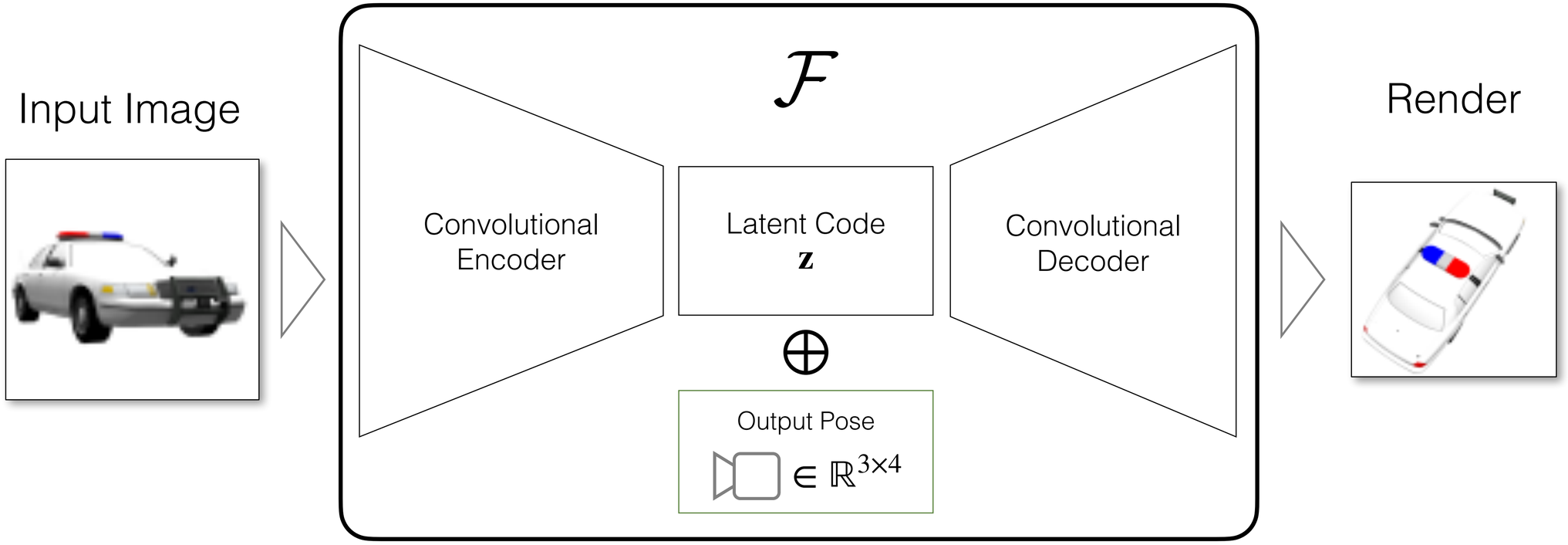
Infinite number of possible image-to-image functions to explain training set.

Only one of them is correct!

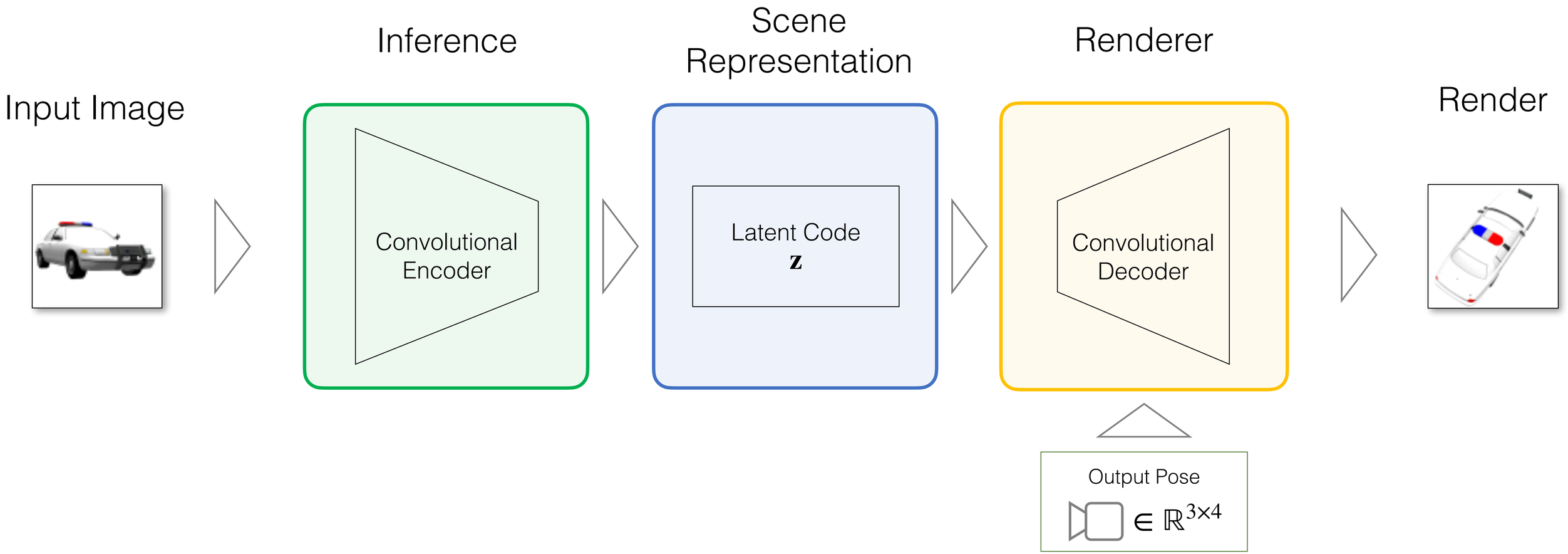
CNN does not have any 3D inductive bias - works by “matching” 2D features.

How to fix?

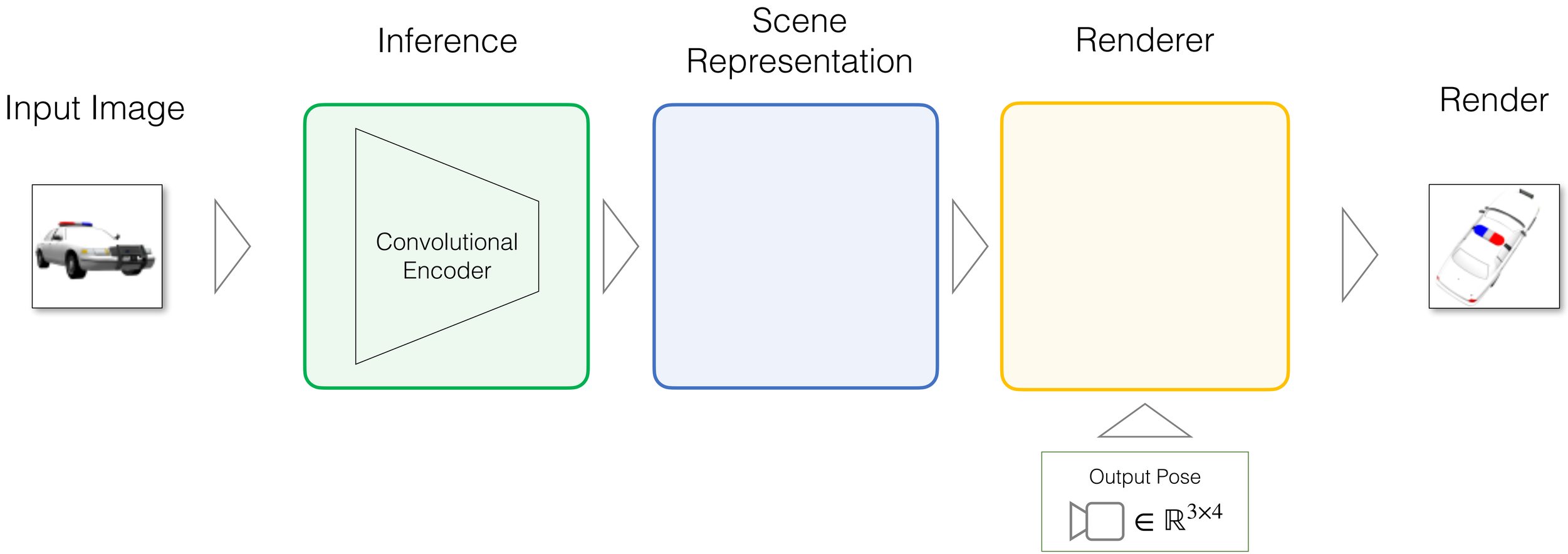
2D baseline: Auto-Encoder-Like model



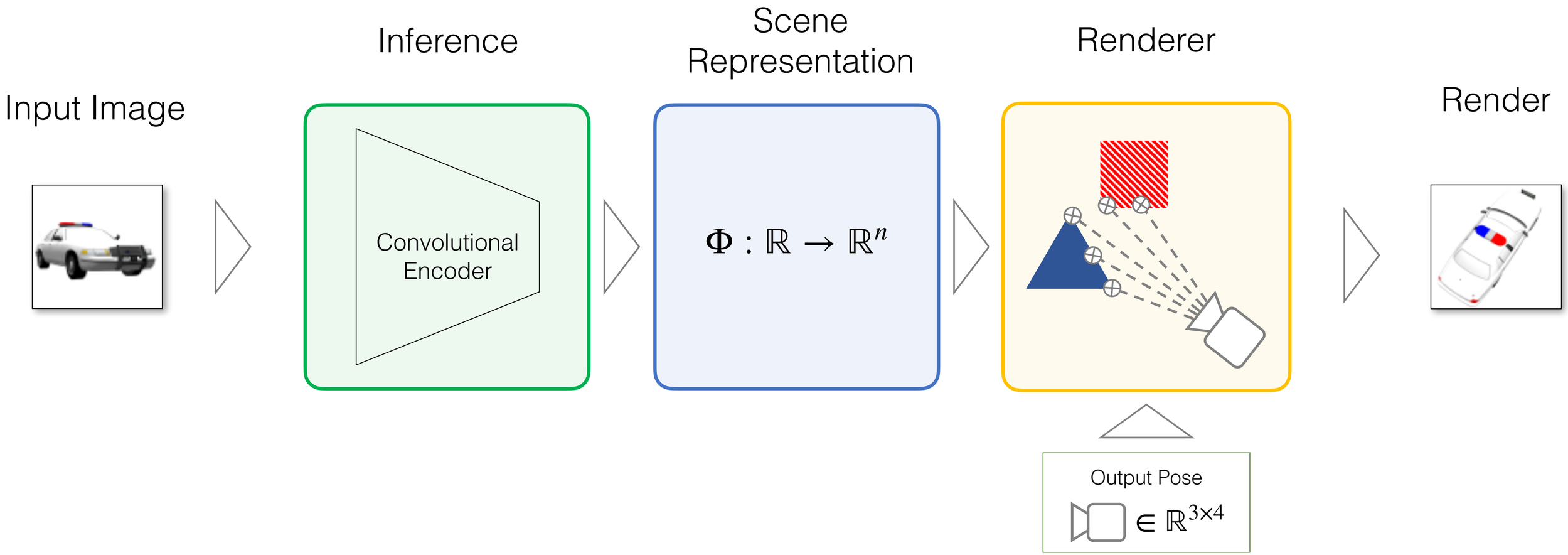
2D baseline: Auto-Encoder-Like model



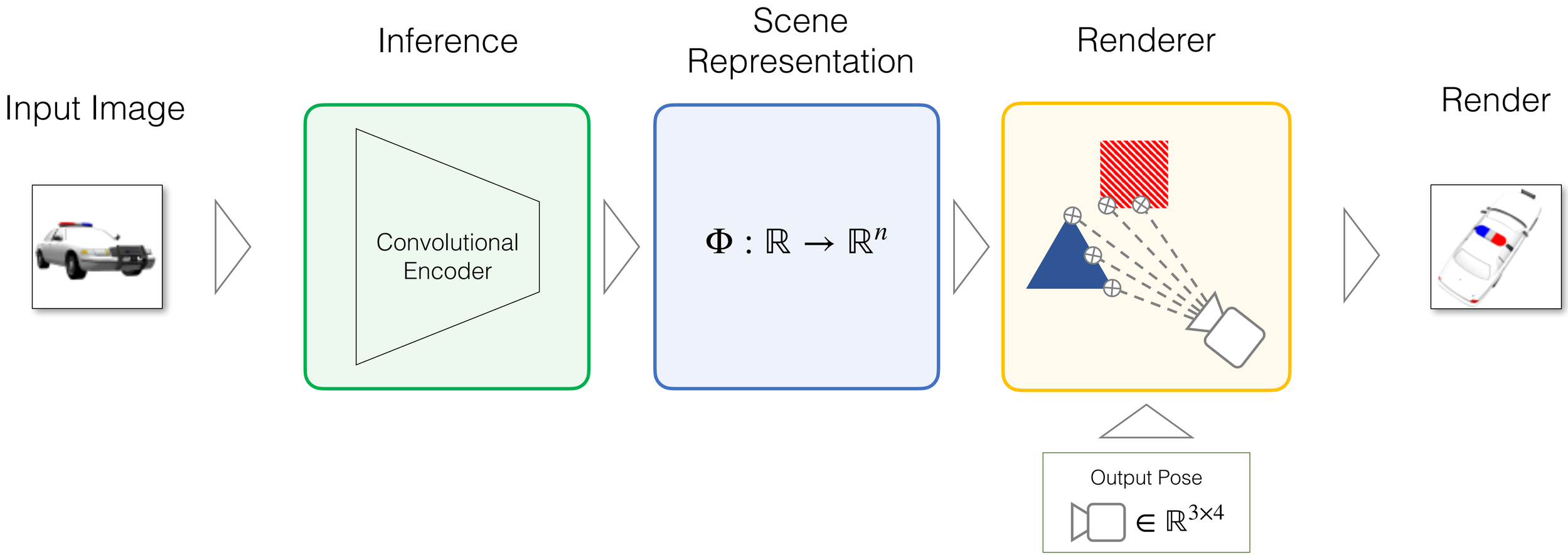
2D baseline: Auto-Encoder-Like model



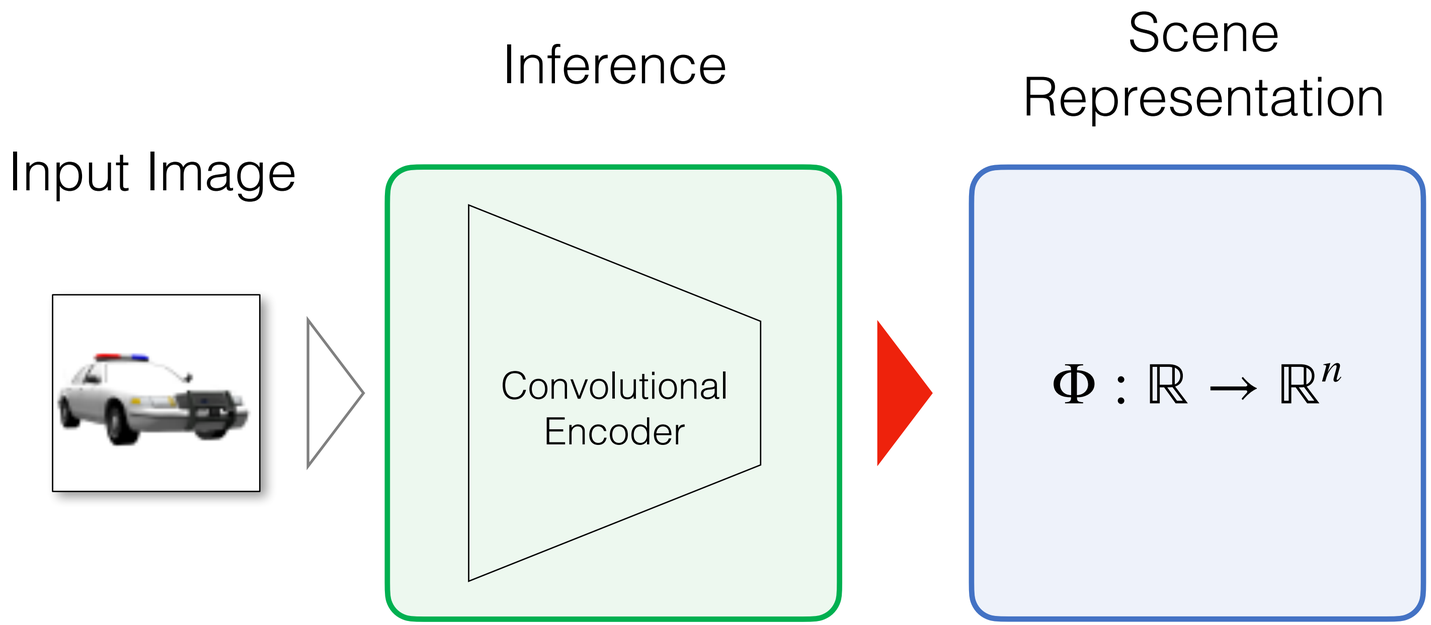
3D-Structured Decoder



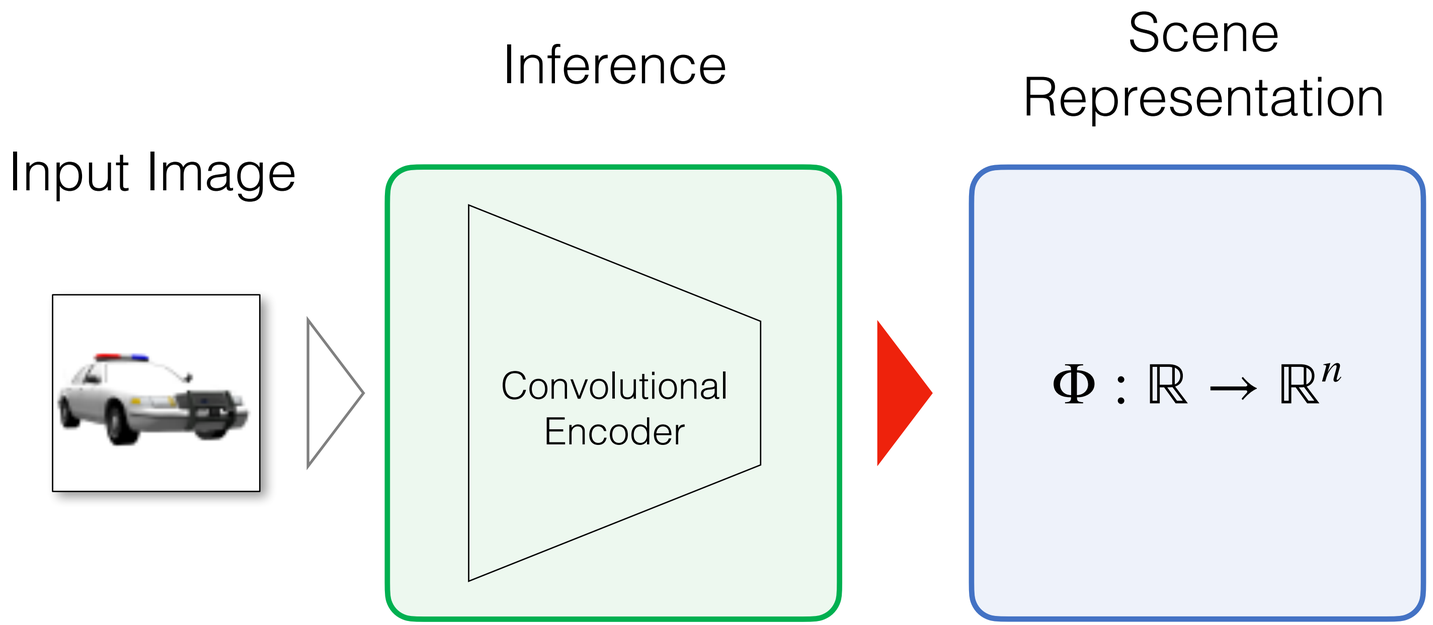
3D-Structured Decoder



3D-Structured Decoder

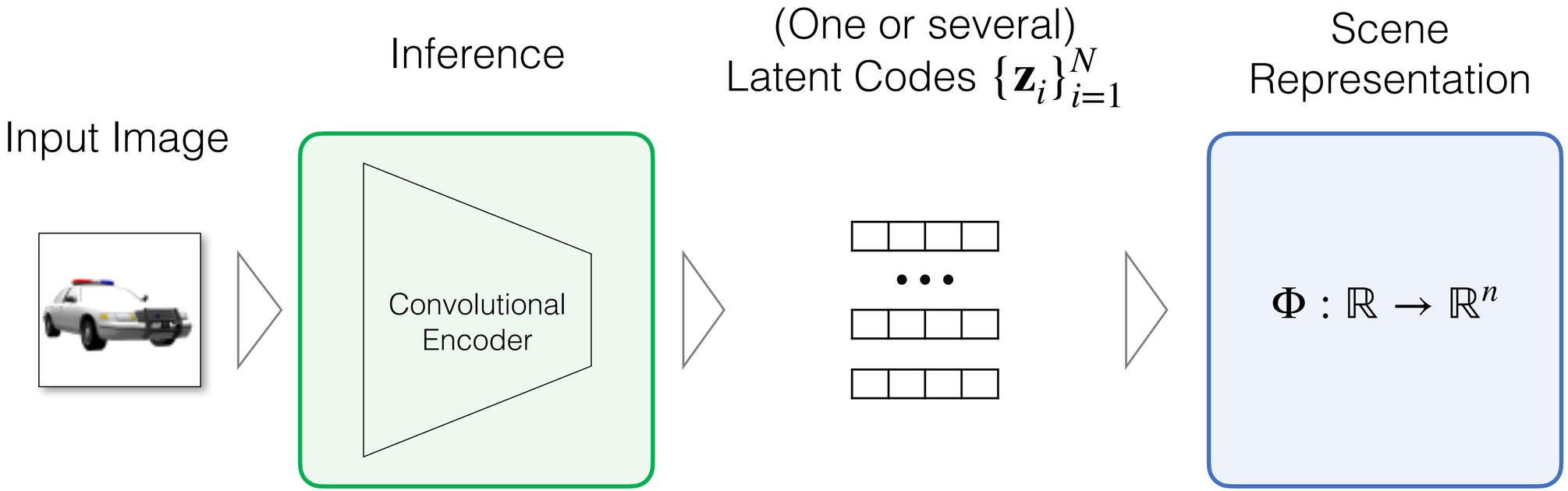


3D-Structured Decoder

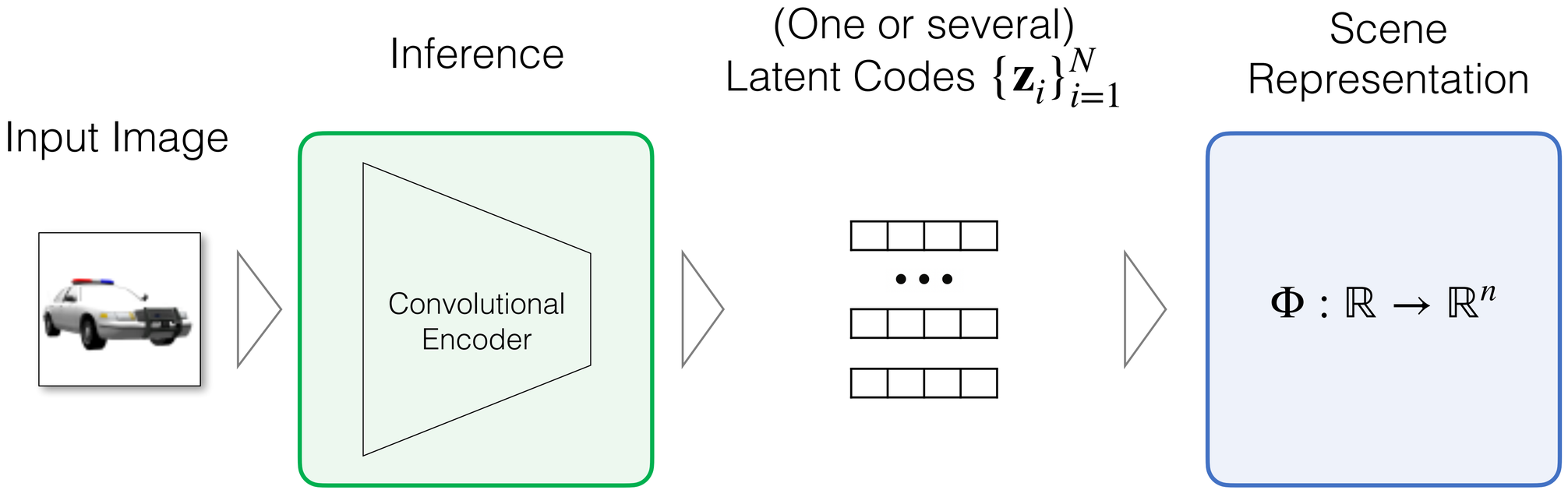


How can we have the convolution encoder “output” the Scene Representation?

3D-Structured Decoder



3D-Structured Decoder

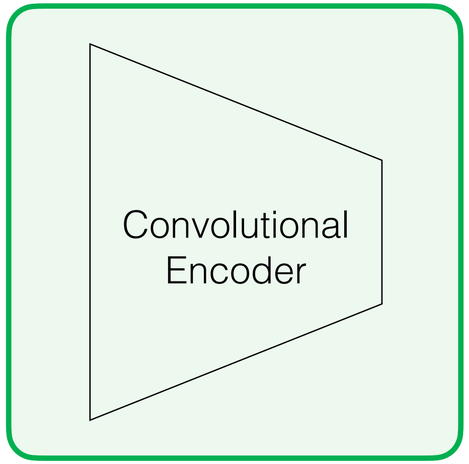


Encoder will output one (or several, will discuss later) latent codes $\mathbf{z}_i$

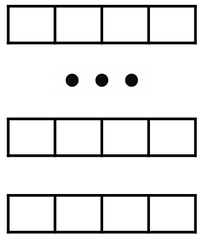
3D-Structured Decoder

Input Image

Inference



(One or several)
Latent Codes $\{\mathbf{z}_i\}_{i=1}^{N}$

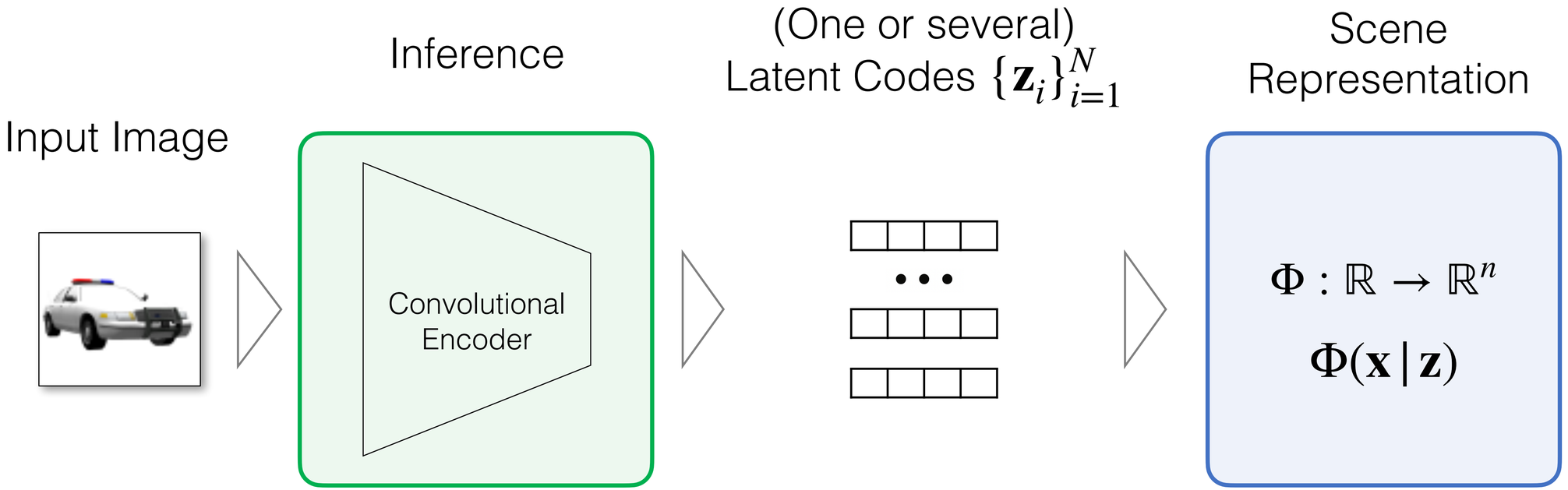


Scene
Representation

$$\Phi : \mathbb{R} \rightarrow \mathbb{R}^n$$

$$\Phi(\mathbf{x} \mid \mathbf{z})$$

3D-Structured Decoder



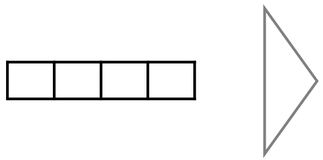
Encoder will output one (or several, will discuss later) latent codes $\mathbf{z}_i$

We will *condition* Scene Representation on latent code, denoted as $\Phi(\mathbf{x} \mid \mathbf{z})$

How?

Conditioning: Predicting Parameters

Latent Code $\mathbf{z}$



Conditioning Network
AKA “Hypernetwork”

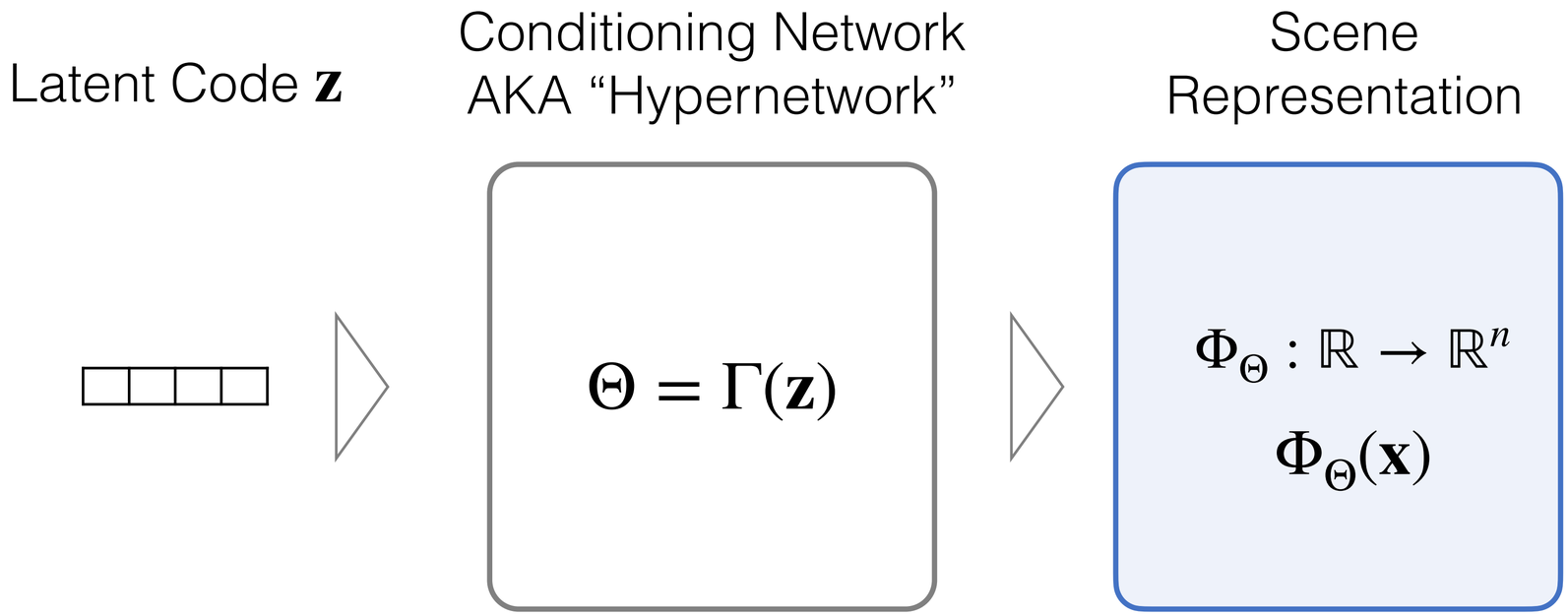
$$\Theta = \Gamma(\mathbf{z})$$

Scene
Representation

$$\Phi_\Theta : \mathbb{R} \rightarrow \mathbb{R}^n$$

$$\Phi_\Theta(\mathbf{x})$$

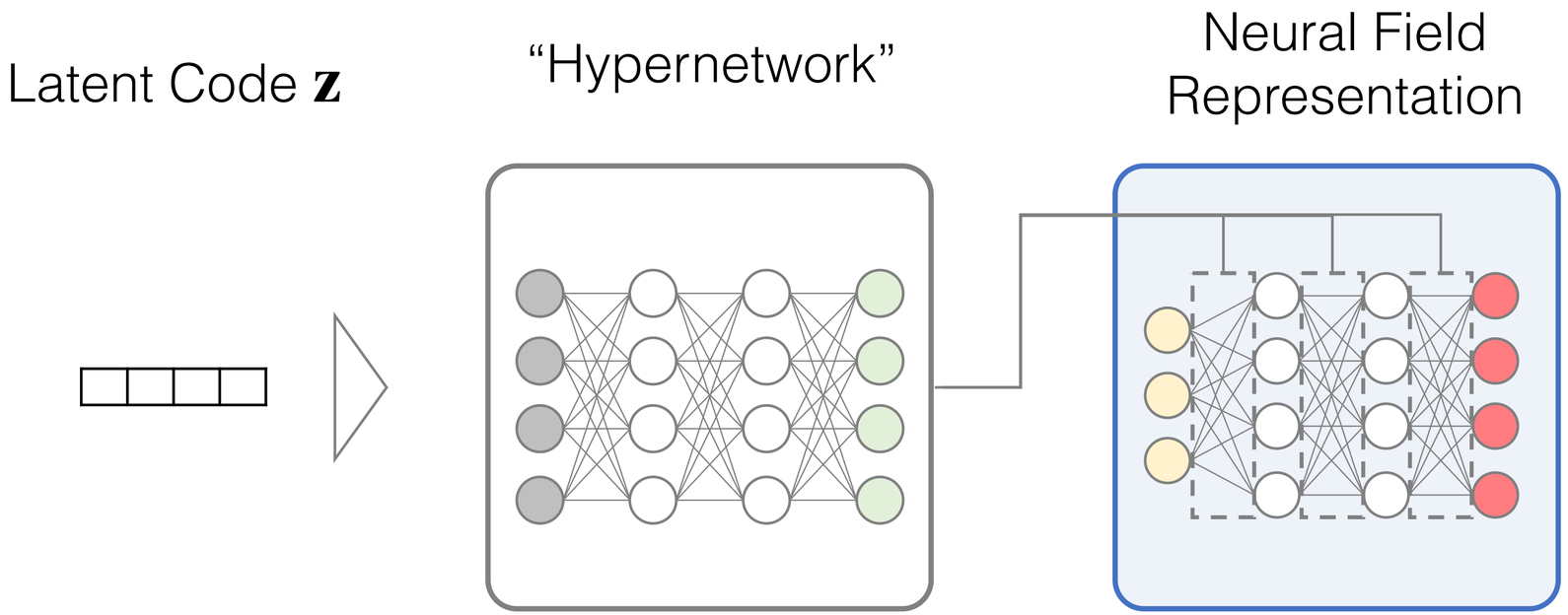
Conditioning: Predicting Parameters



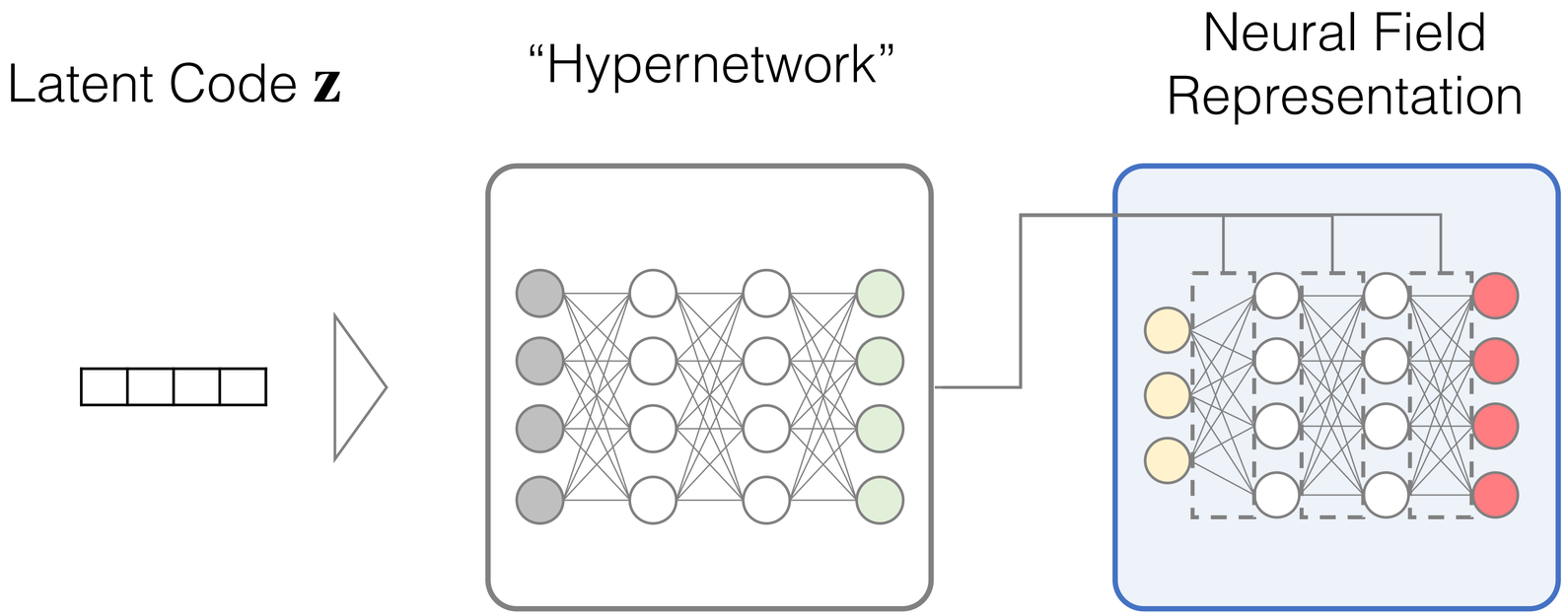
Any Scene Representation has parameters, denoted as Θ: Voxelgrid has voxel contents, Neural Field (MLP) has weights & biases, …

We can map latent codes to these parameters via a neural network (fully connected, deconvolutional, transformer, …)

Conditioning Neural Fields by predicting parameters

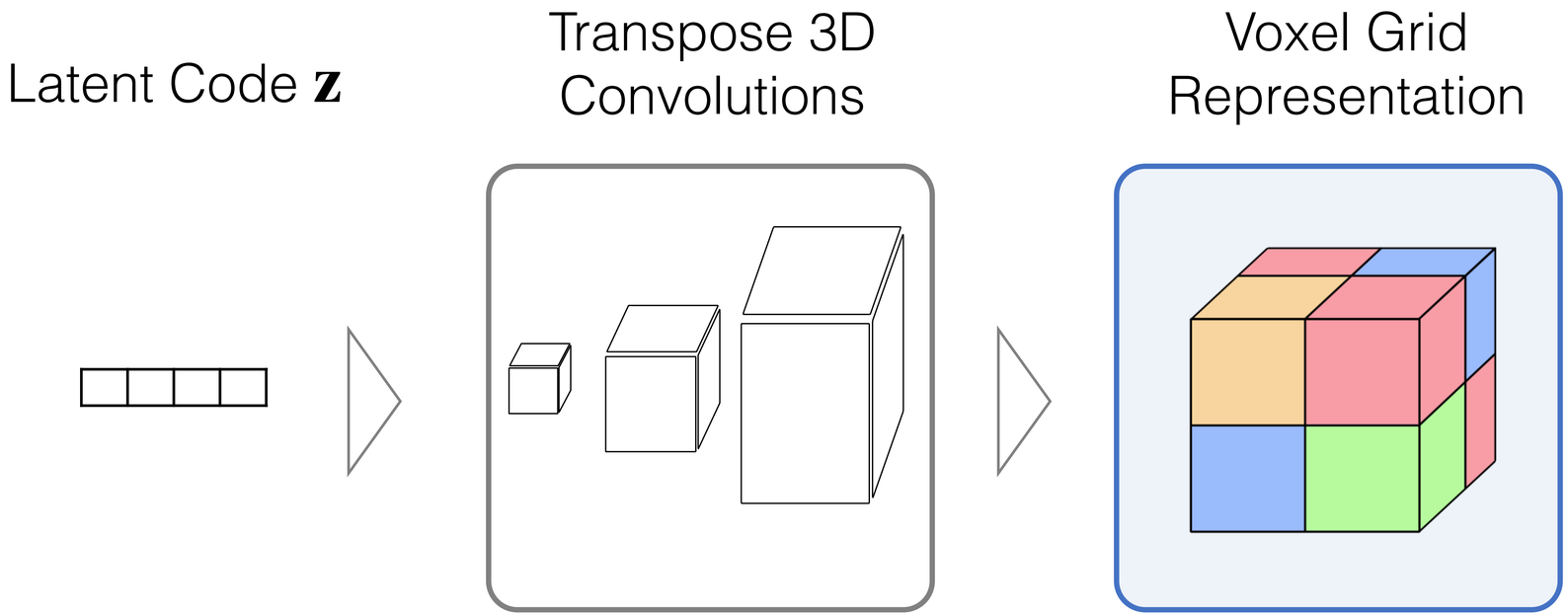


Conditioning Neural Fields by predicting parameters

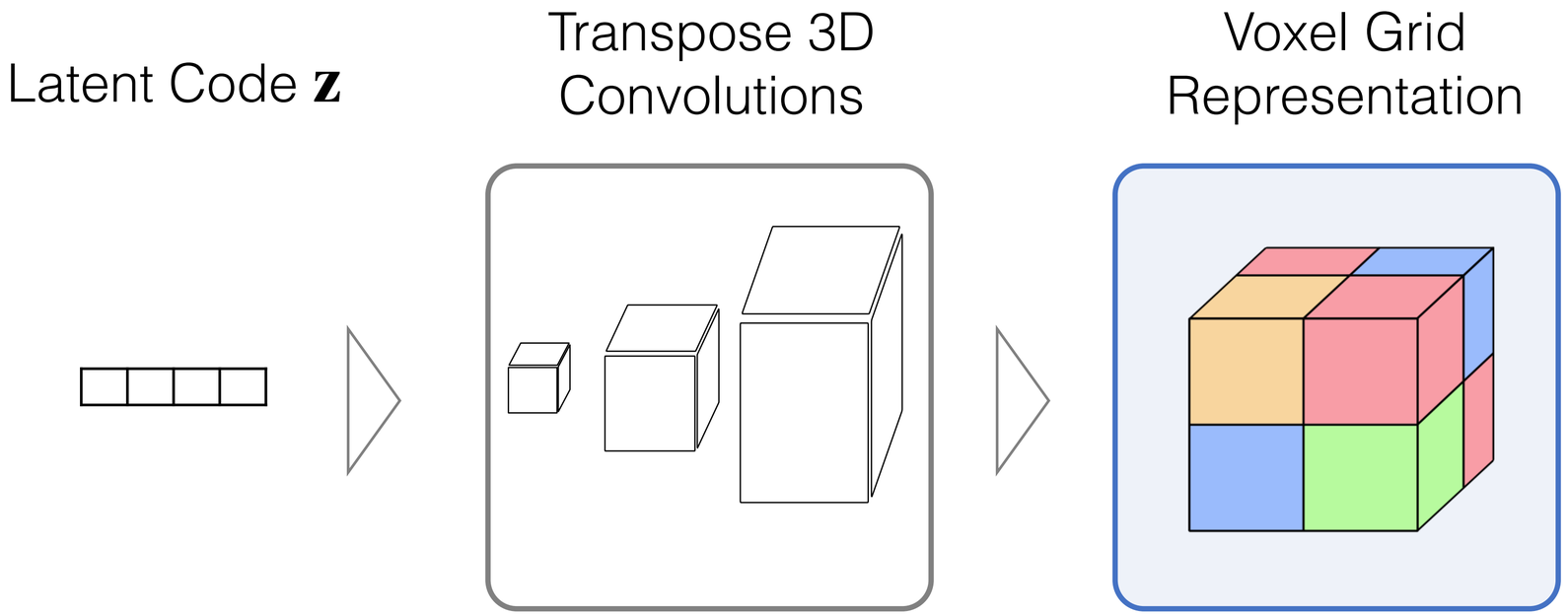


To condition neural field, can literally have MLP that takes as input $\mathbf{z}$ and outputs *all* of the parameters (weights, biases) of the neural field.

Conditioning Voxel Grids by predicting parameters

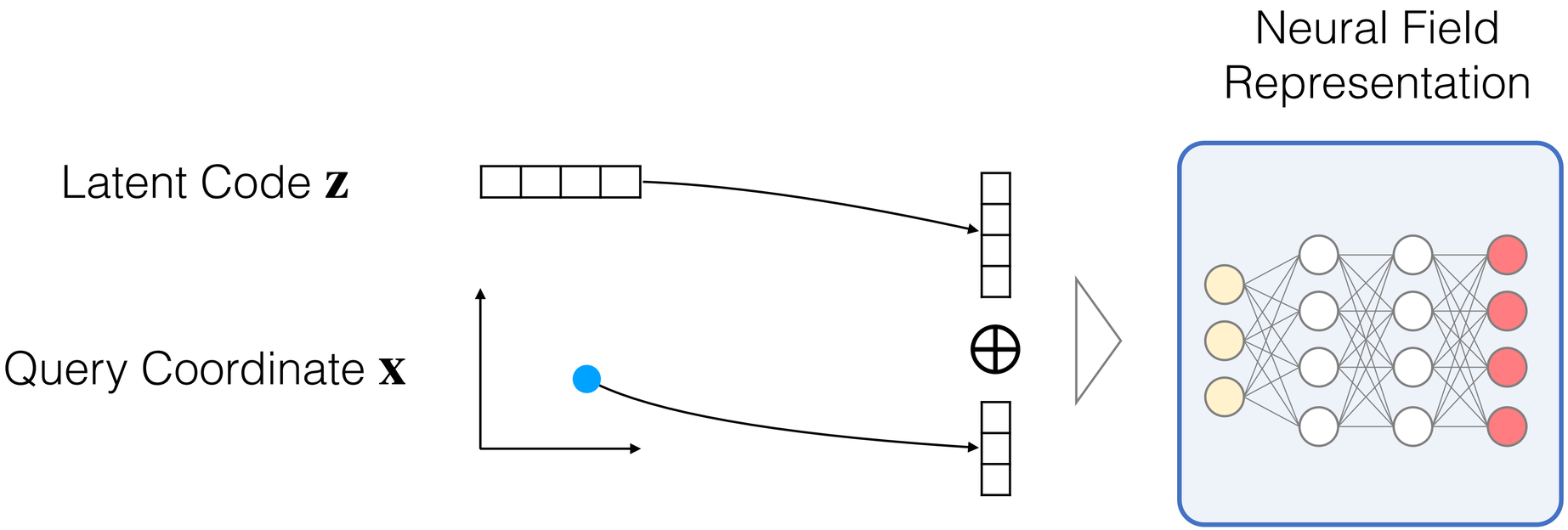


Conditioning Voxel Grids by predicting parameters

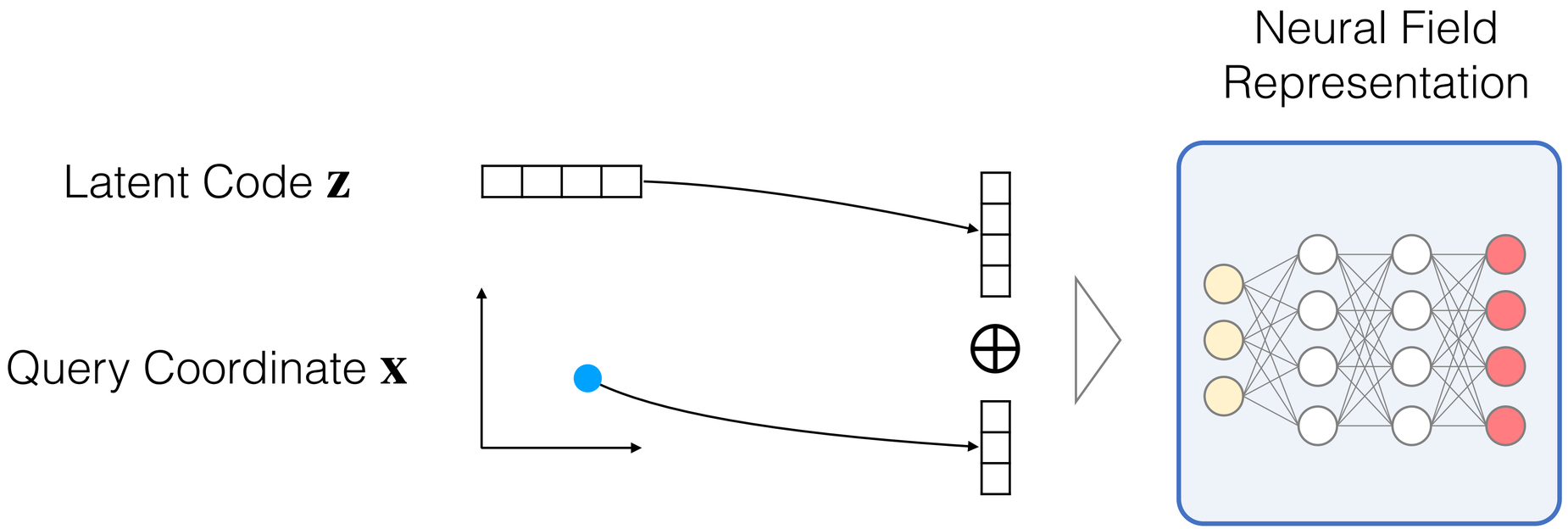


To condition voxel grid, can apply 3D de-convolutional neural network (aka transpose convolutions) to upsample $1 \times 1 \times 1 \times ch$ latent code to $N \times N \times N \times ch$.

Conditioning Neural Fields: Conditioning via Concatenation



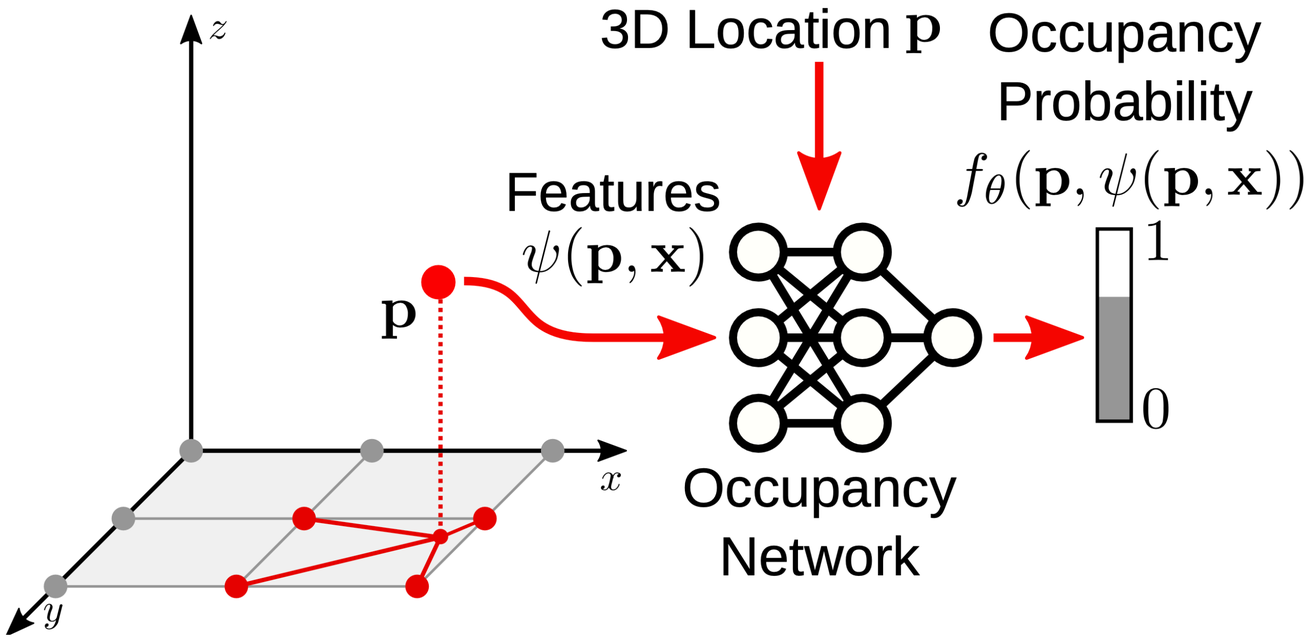
Conditioning Neural Fields: Conditioning via Concatenation



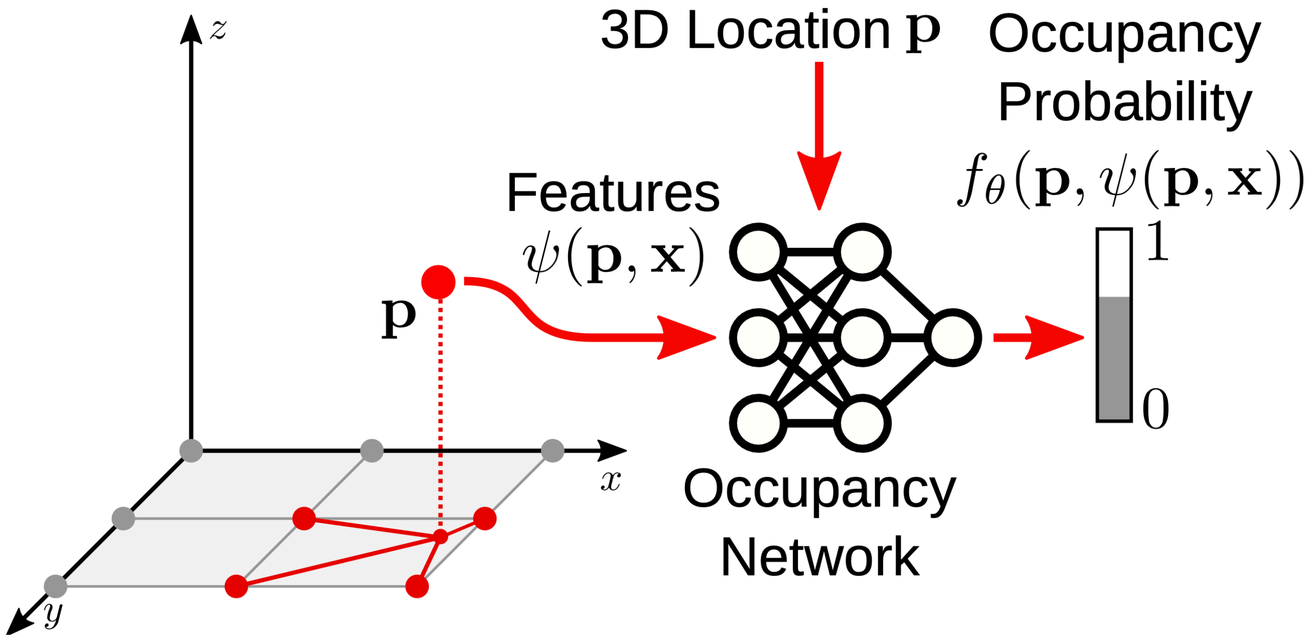
For neural fields, can concatenate latent code and query coordinate as input.

Less expressive than hyper net, but also often fine - for more alternatives, see section 2.1.3. in Neural Fields in Visual Computing and Beyond, Xie et al.

Conditioning Neural Fields: Conditioning via Concatenation

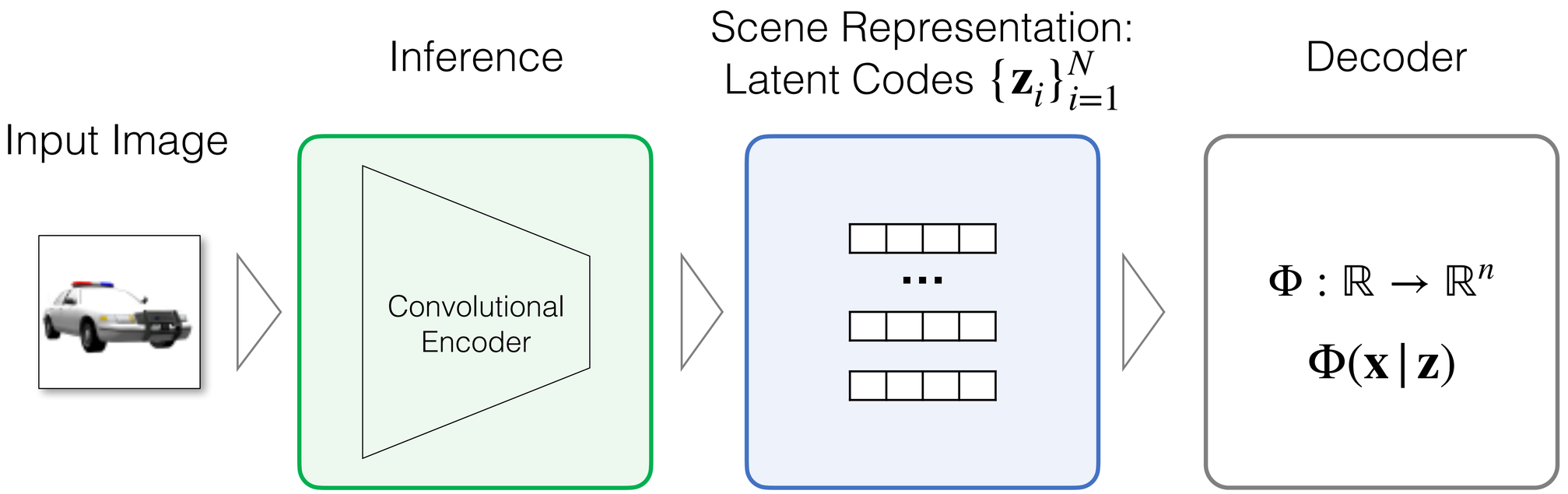


Conditioning Neural Fields: Conditioning via Concatenation

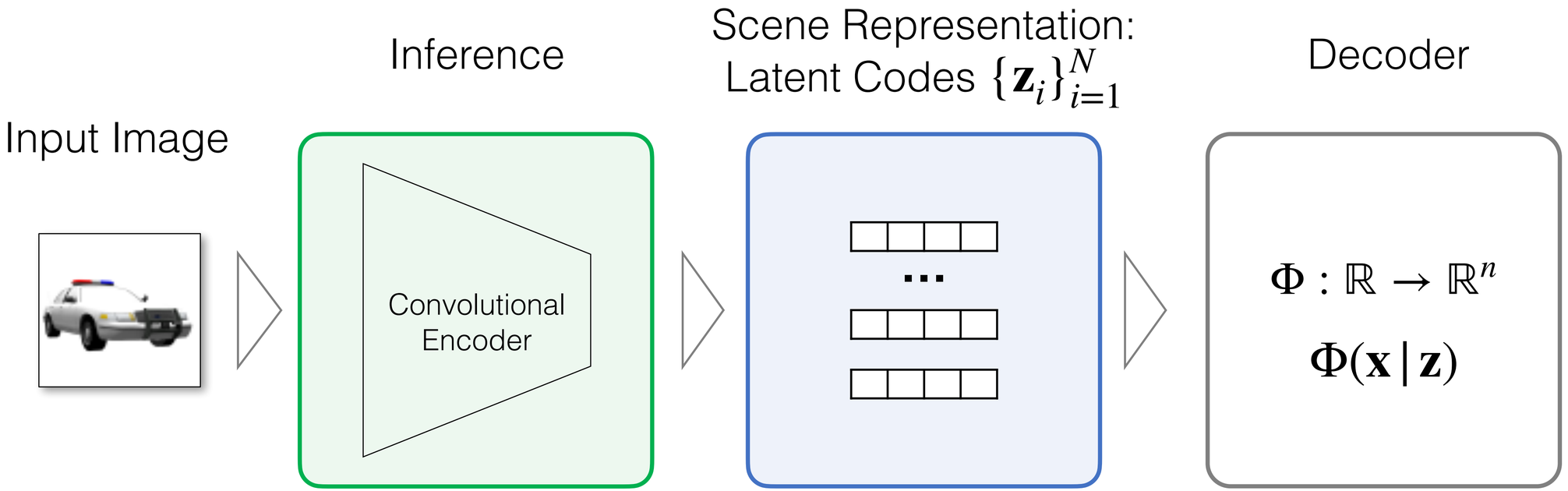


You already know this: That's what's usually done in hybrid discrete-continuous Representations!

Note on Terminology



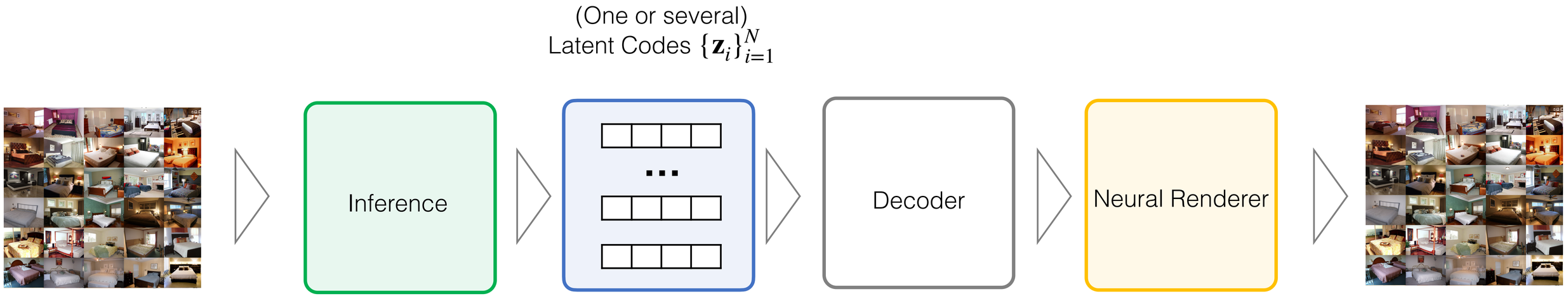
Note on Terminology



In fact, the **Latent Codes** are what's actually encoding the scene!

What we previously called "Scene Representation" is now simply a **way of decoding the latents, just as the 2D convolutions were before.**

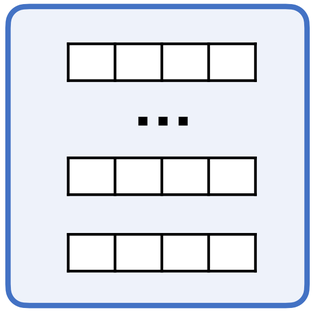
Summary: General Framework



Summary: General Framework

(One or several)
Latent Codes $\{\mathbf{z}_i\}_{i=1}^{N}$

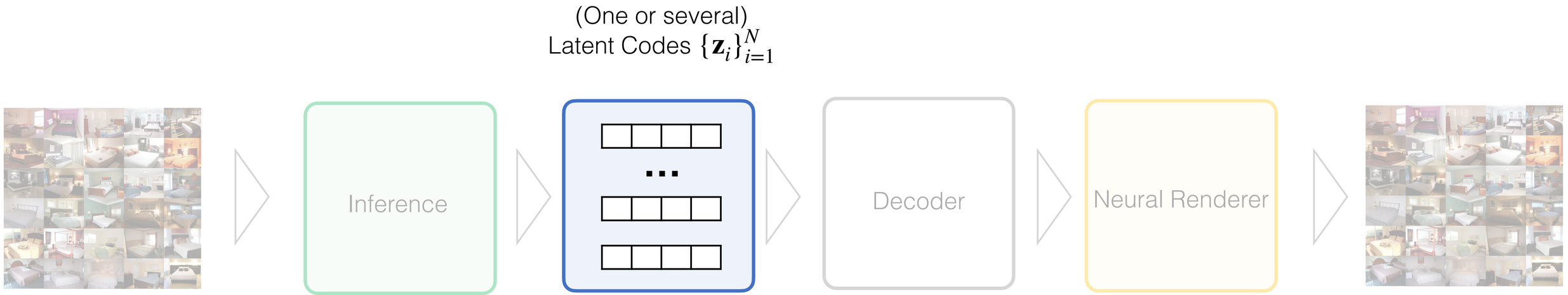
Inference



Decoder

Neural Renderer

Summary: General Framework

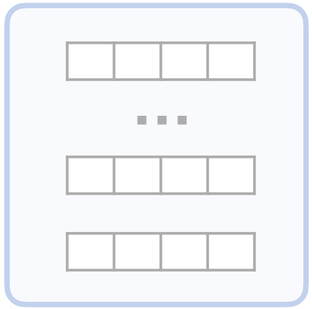


What are the **Latent Codes and how many should there be?**

Summary: General Framework

(One or several)
Latent Codes $\{\mathbf{z}_i\}_{i=1}^{N}$

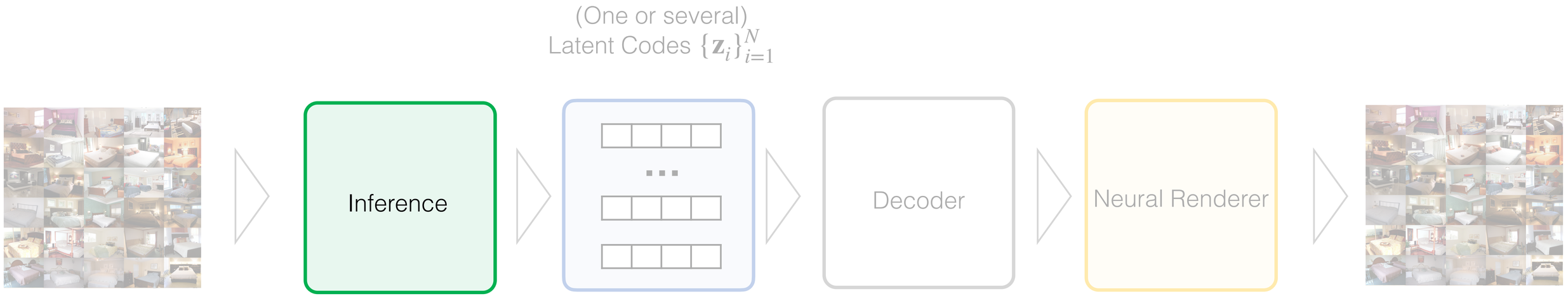
Inference



Decoder

Neural Renderer

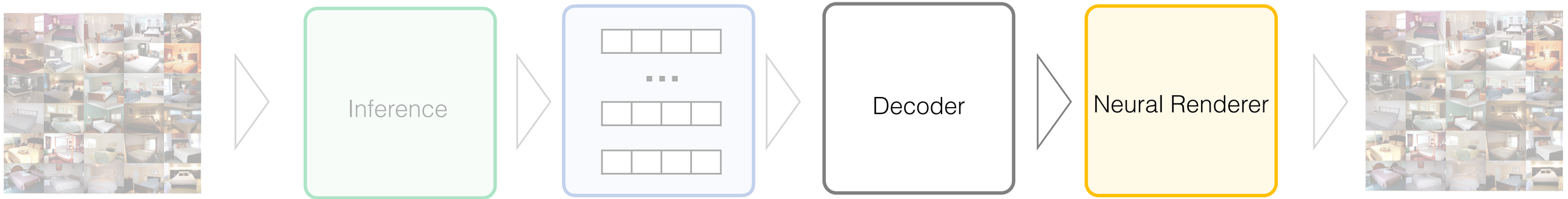
Summary: General Framework



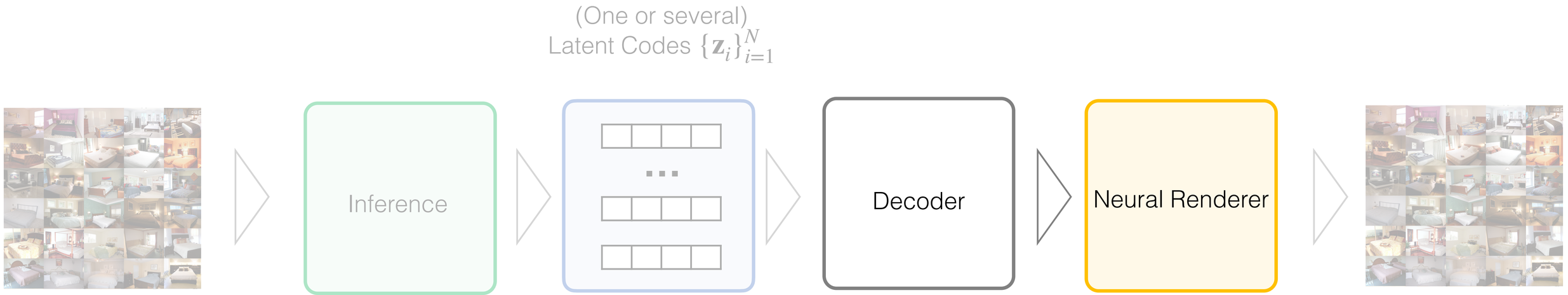
How to get the latent codes from images?

How to guarantee generalization?

Summary: General Framework



Summary: General Framework

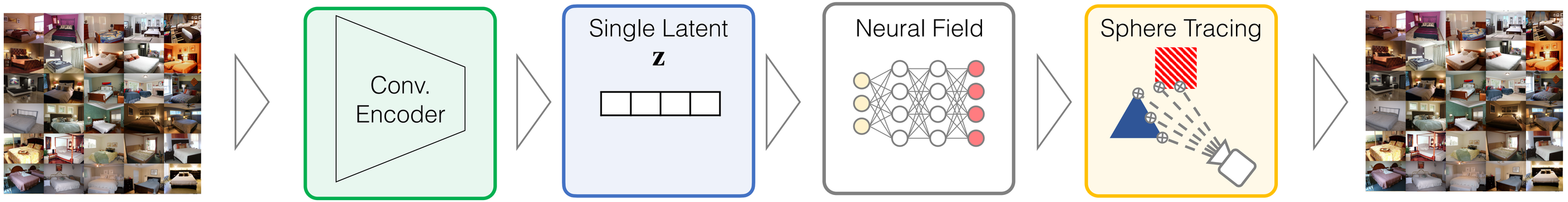


What forms & shapes can the decoder take?

Does the Scene Representation always have to be 3D / volumetric?

Do we need 3D volumetric renderers?

Specific Model



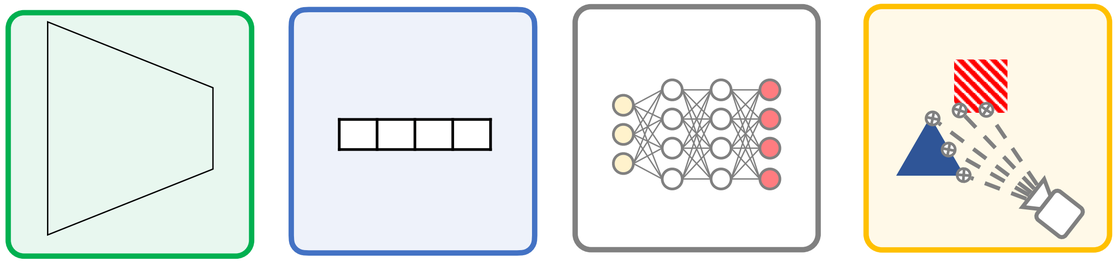
Learn inference model from many images of many scenes.

Learn inference model from many images of many scenes.

Model

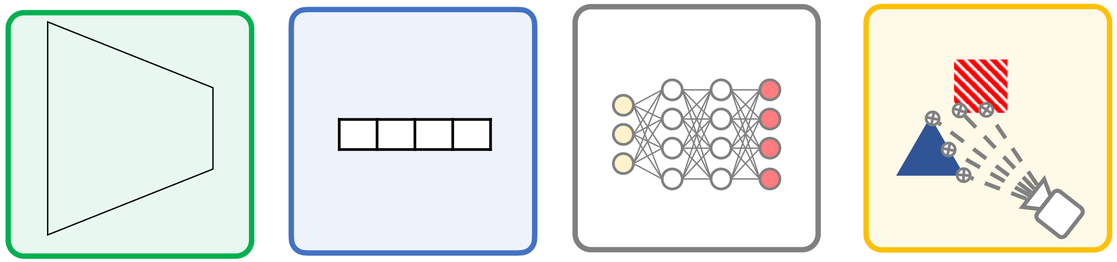
Learn inference model from many images of many scenes.

Model



Learn inference model from many images of many scenes.

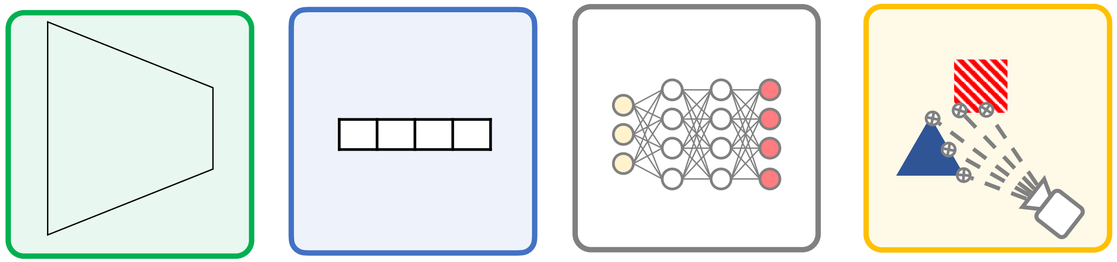
Model



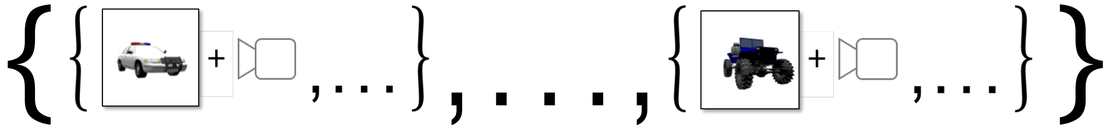
Dataset

Learn inference model from many images of many scenes.

Model

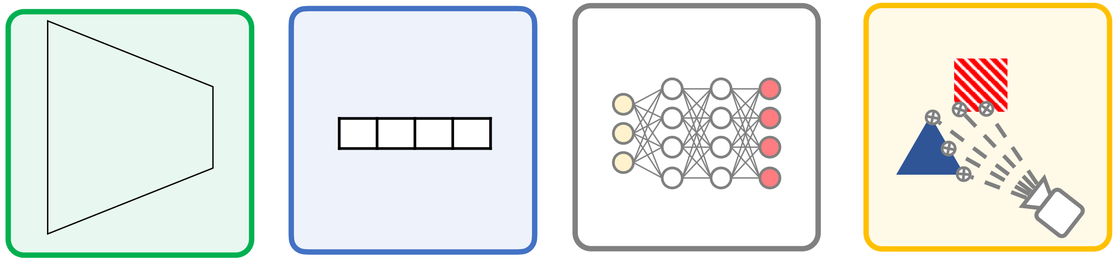


Dataset



Learn inference model from many images of many scenes.

Model

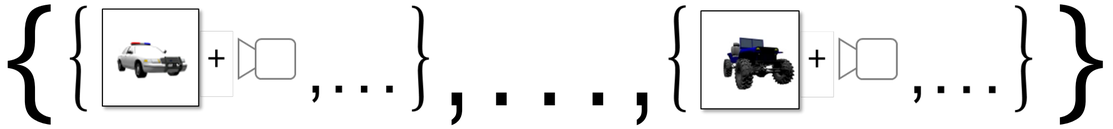


Input view

Normal map

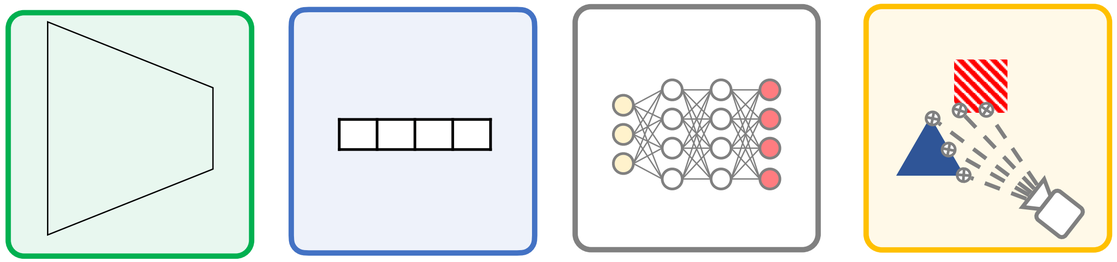
RGB

Dataset



Learn inference model from many images of many scenes.

Model

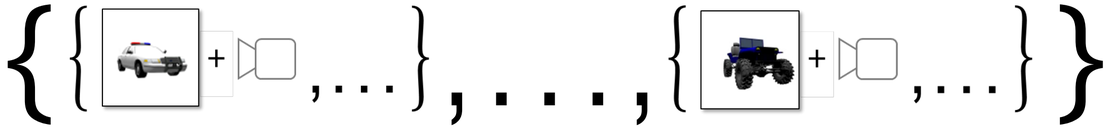


Input view

Normal map

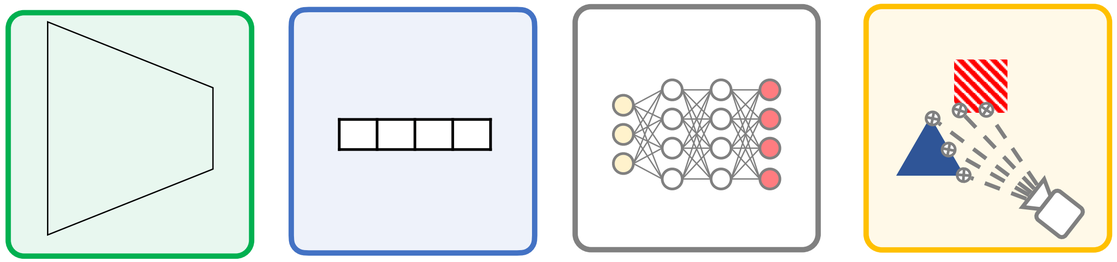
RGB

Dataset

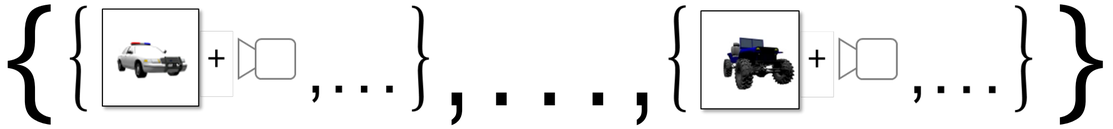


Learn inference model from many images of many scenes.

Model

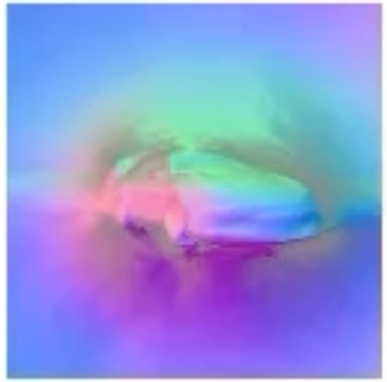


Dataset



Input view

Normal map



RGB

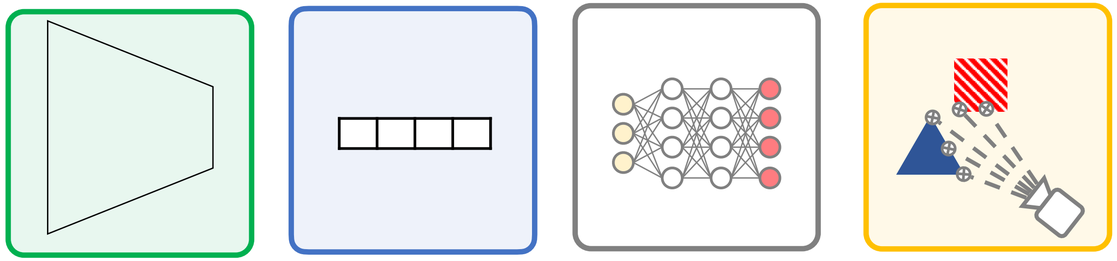
Out-of-distribution
Input view

Normal map

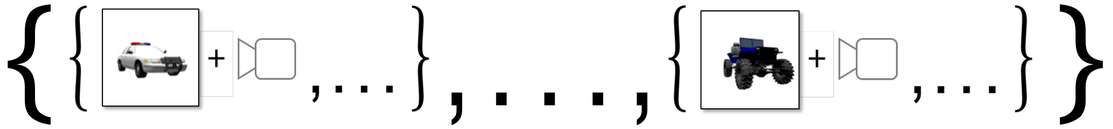
RGB

Learn inference model from many images of many scenes.

Model

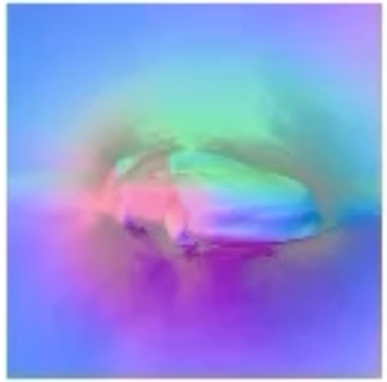


Dataset



Input view

Normal map



RGB

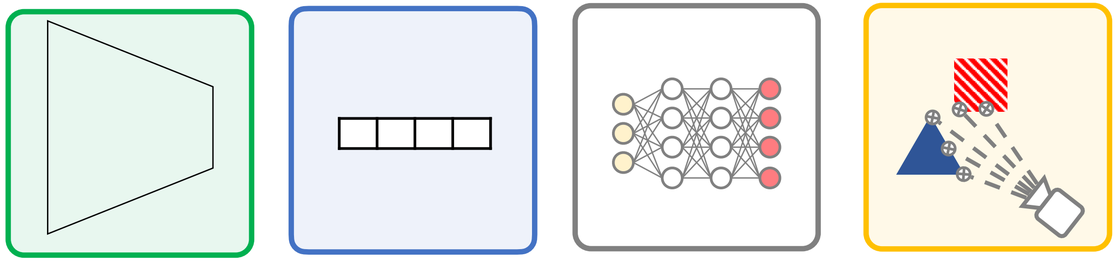
Out-of-distribution
Input view

Normal map

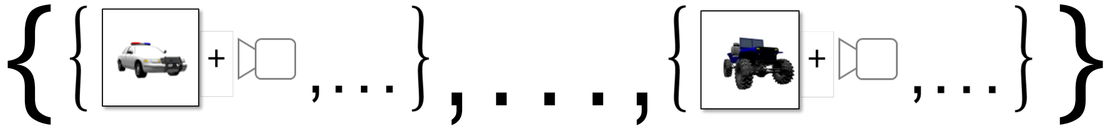
RGB

Learn inference model from many images of many scenes.

Model



Dataset



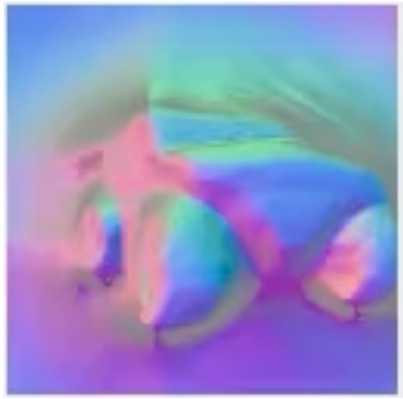
Input view

Normal map

RGB

Out-of-distribution
Input view

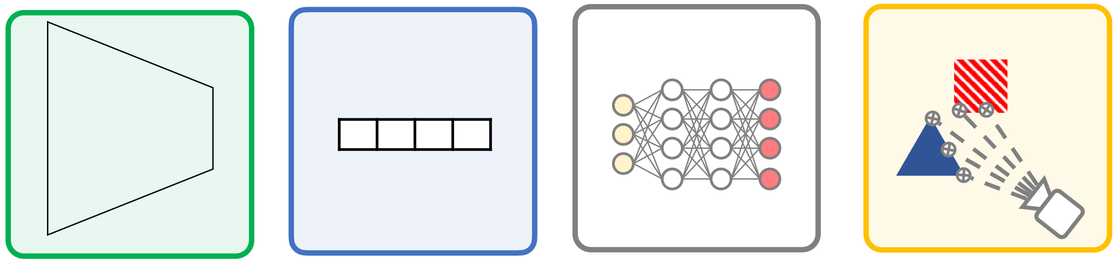
Normal map



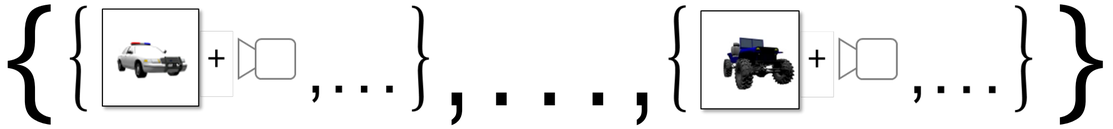
RGB

Learn inference model from many images of many scenes.

Model



Dataset



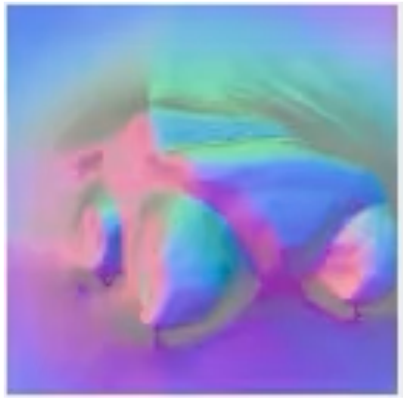
Input view

Normal map

RGB

Out-of-distribution
Input view

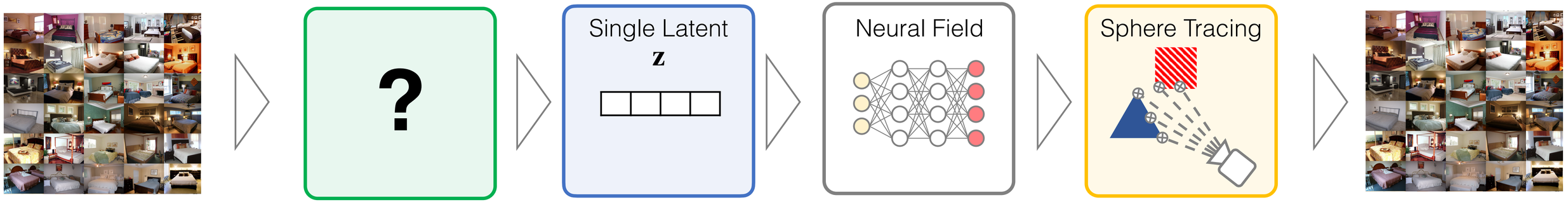
Normal map



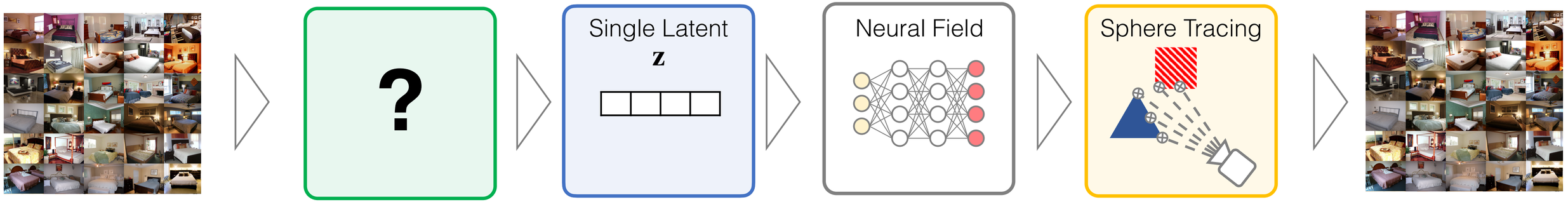
RGB

Wrong window color, wrong tires, wrong shape…

Specific Model



Specific Model



Can we think of an alternative inference method?

Training: Initialize one embedding per scene

embedding z_0

embedding z_1

+

○ ○ ○

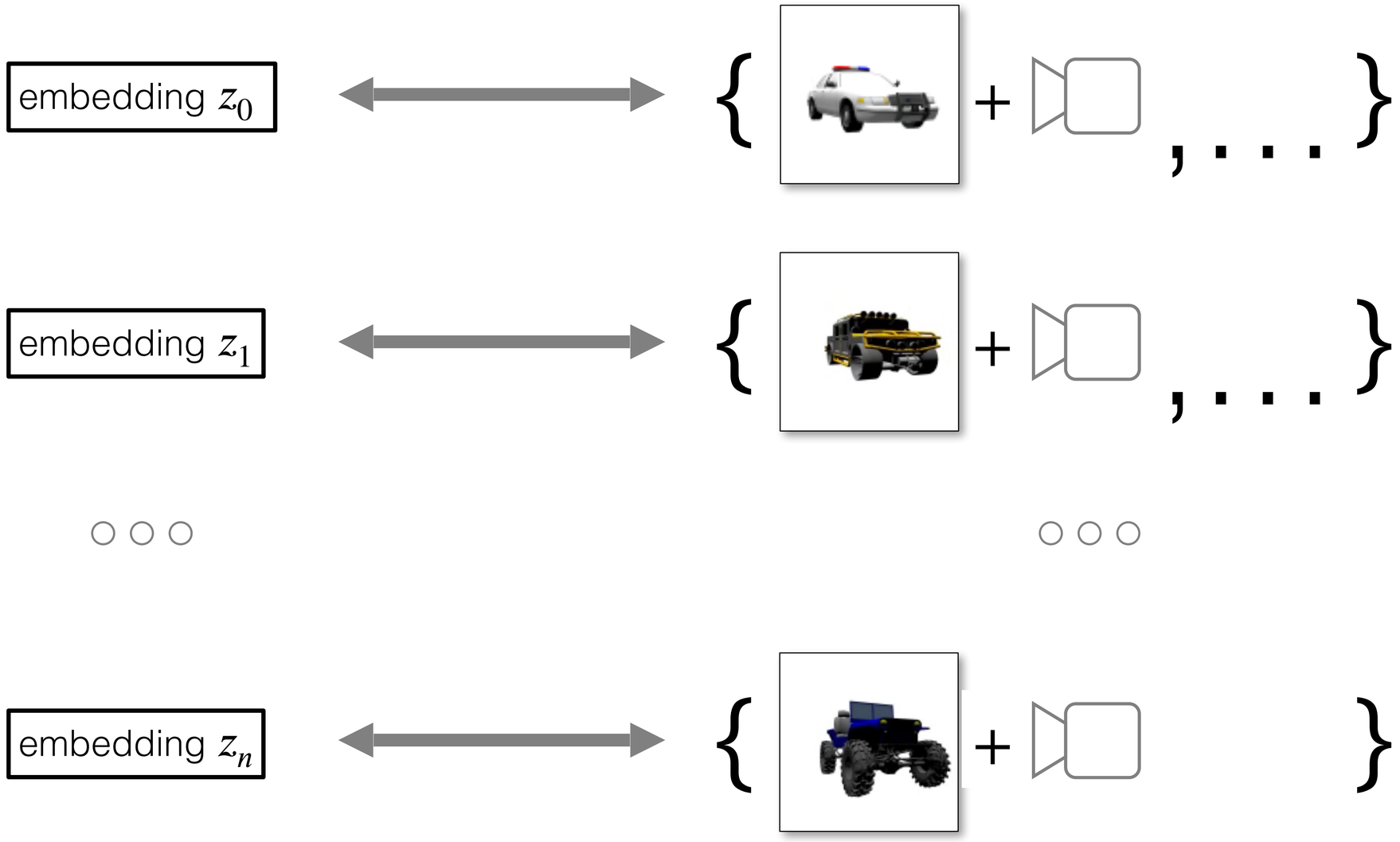
○ ○ ○

embedding z_n

+

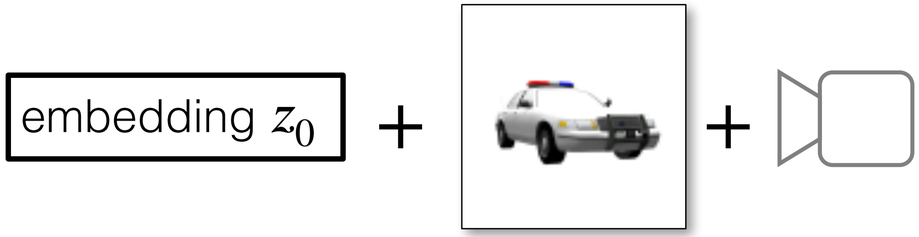
$$z_{j=0,\ldots,n} \sim \mathcal{N}(0, \sigma^2)$$

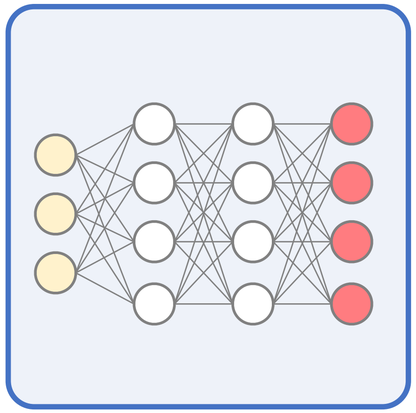
Training: Initialize one embedding per scene



$$z_{j=0,\ldots,n} \sim \mathcal{N}(0, \sigma^2)$$

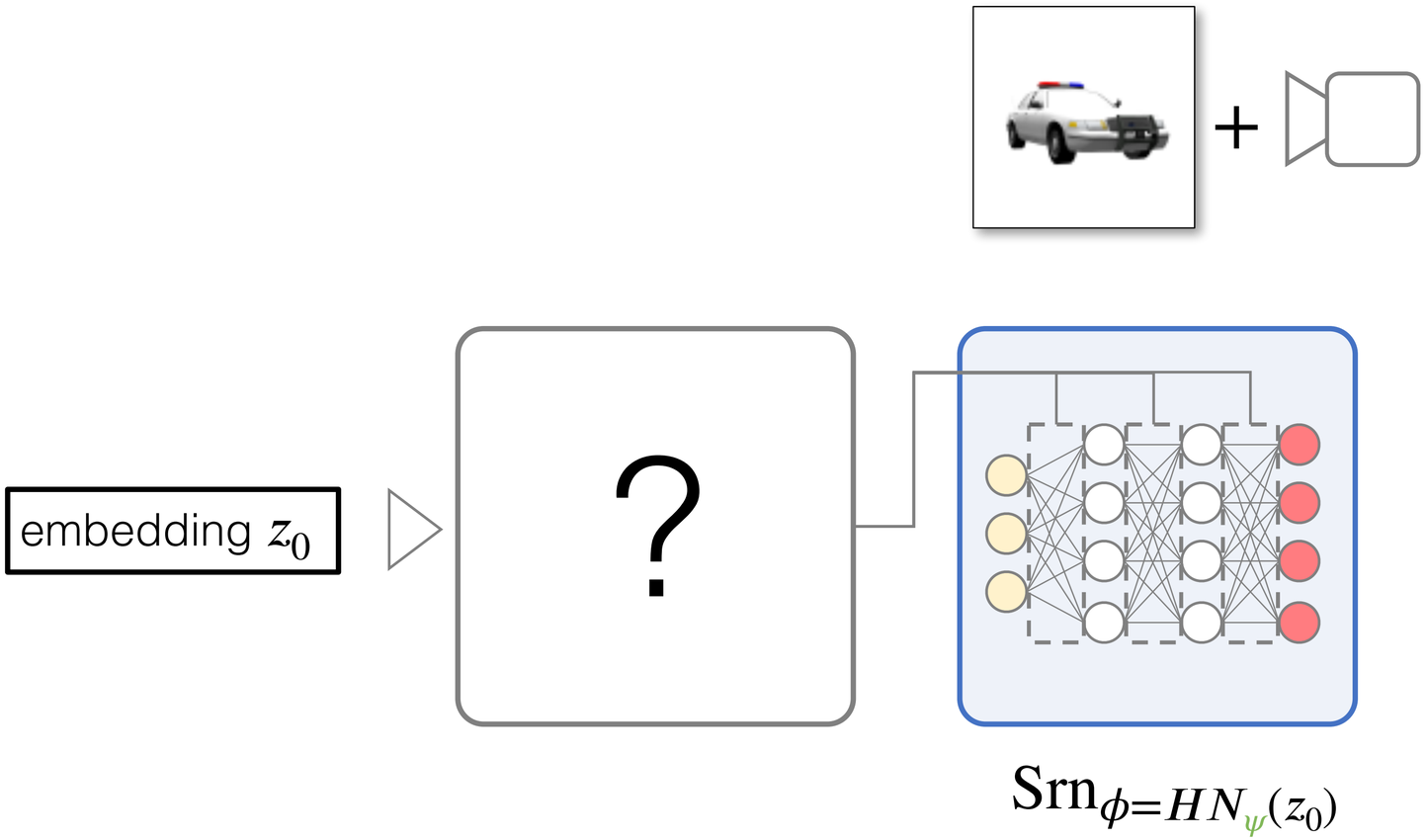
Decode embedding into scene representation



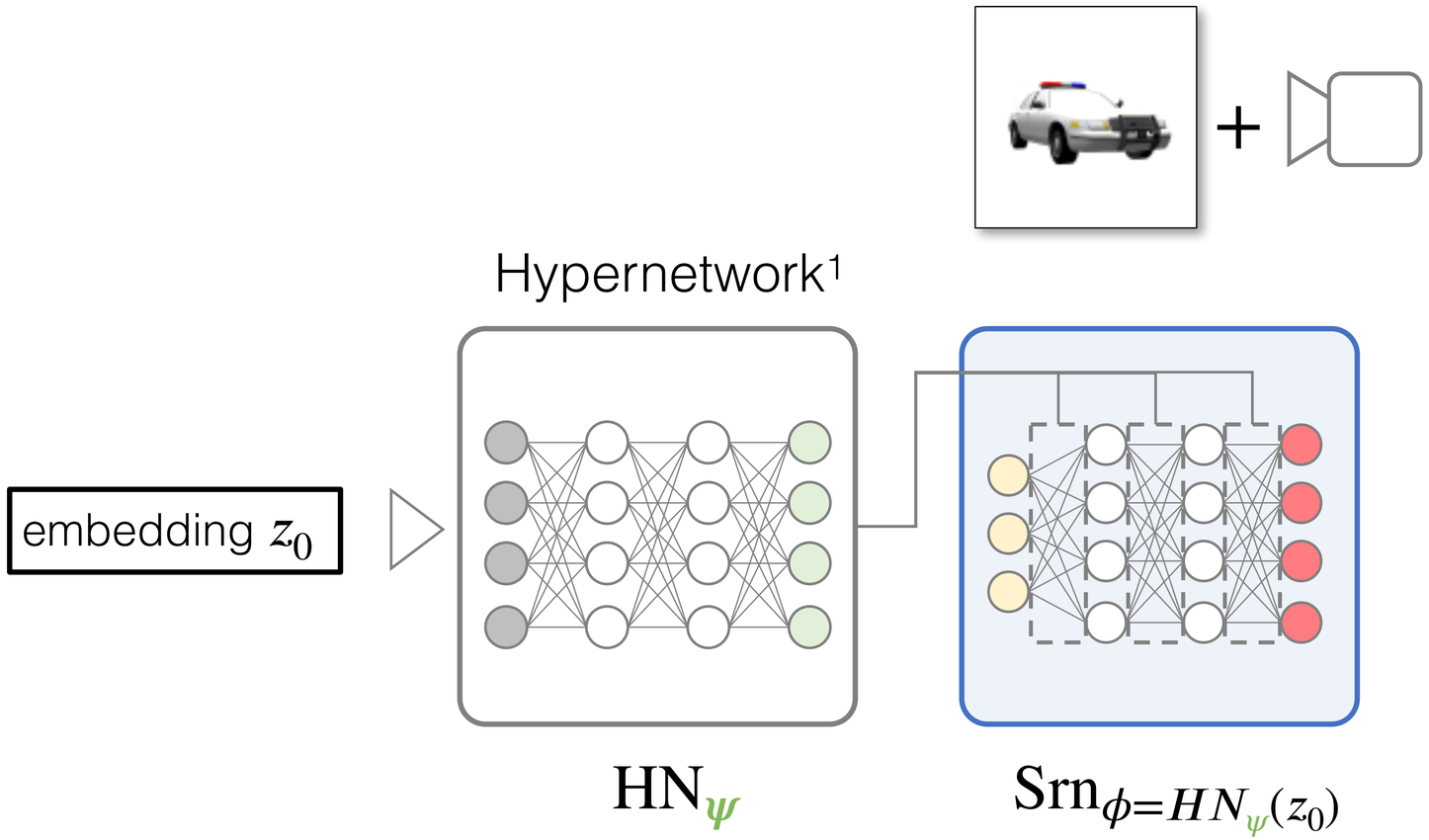


$$\text{Srn}_{\phi=HN_{\psi}(z_0)}$$

Decode embedding into scene representation

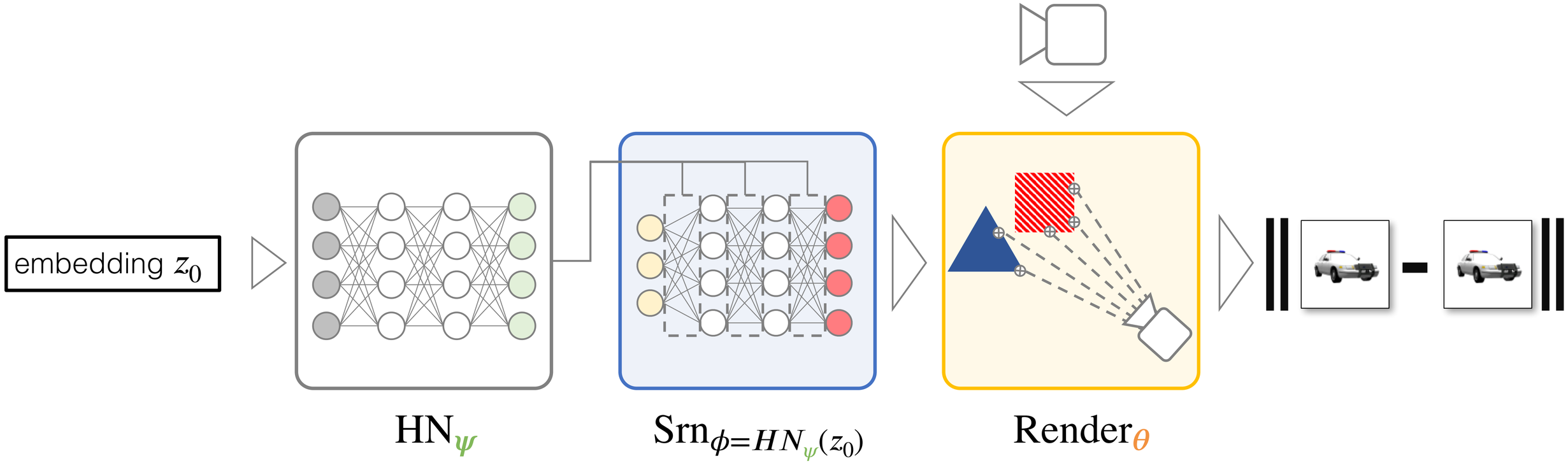


Decode embedding into scene representation

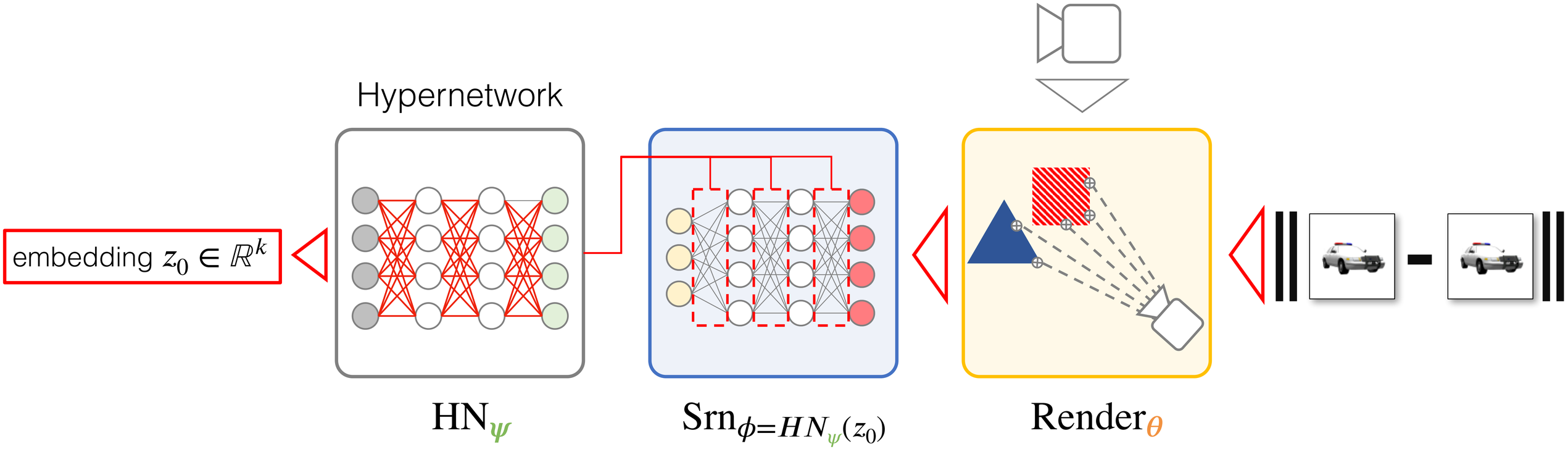


[1][Schmidhuber et al. 1992, Schmidhuber et al. 1993, Stanley et al. 2009, Ha et al., 2016]

Render training view

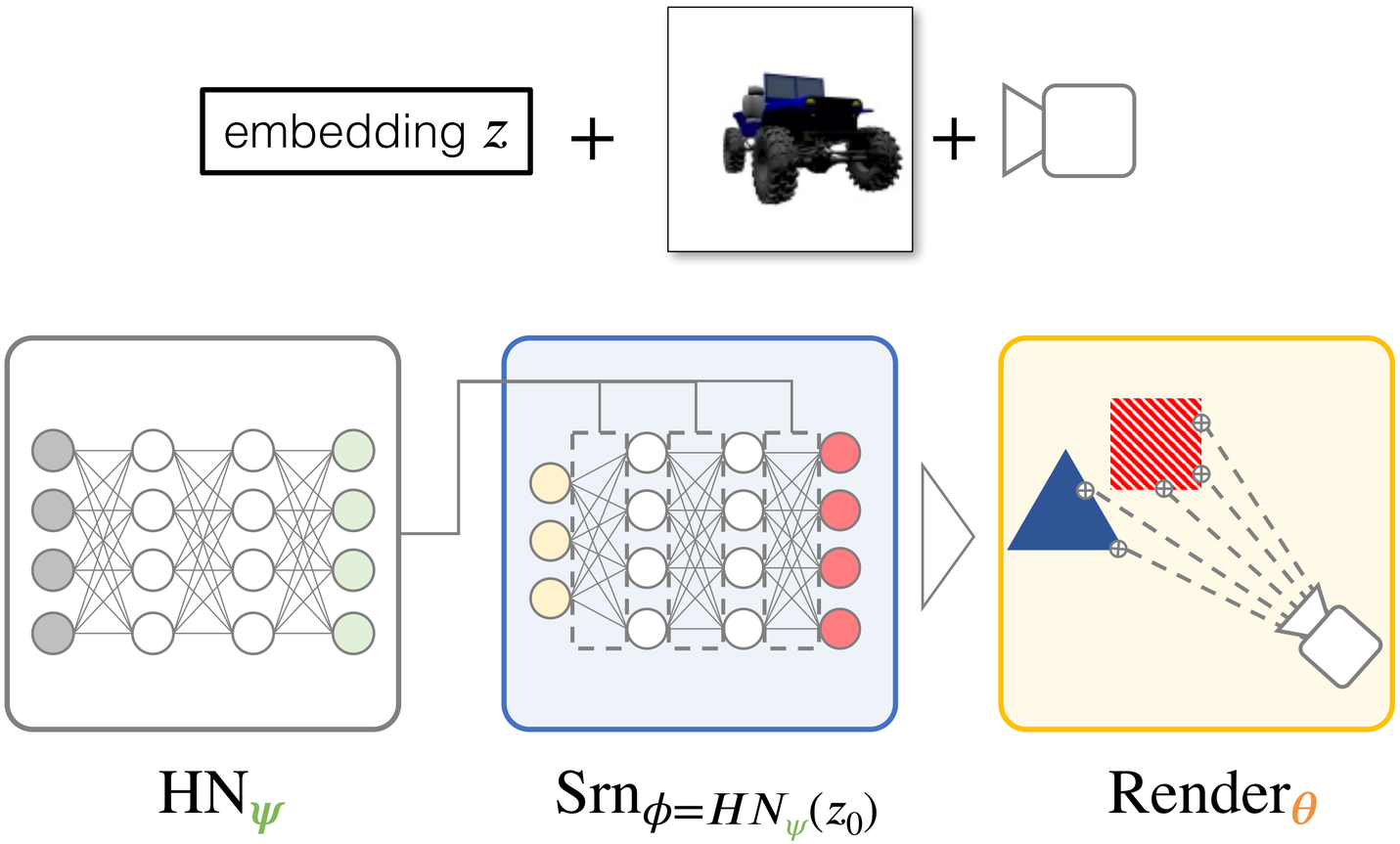


Co-optimize embeddings and model weights!

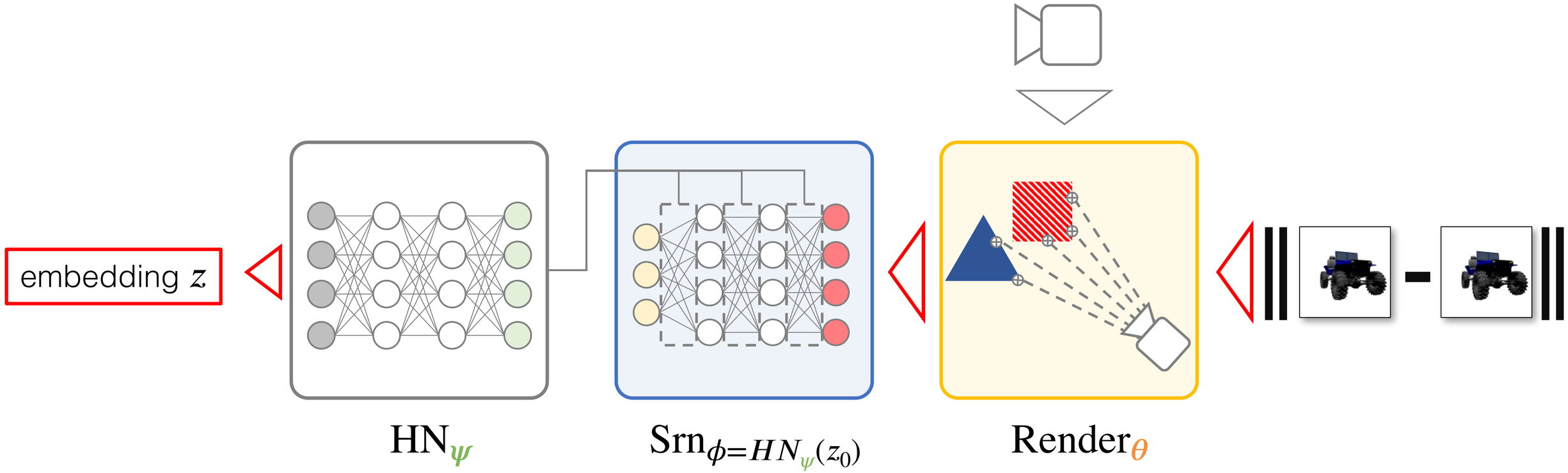


$$\arg \min_{\{z_j\}_{j=1}^{M},\psi,\theta} \sum_j \sum_i \left\| \text{Render}_{\theta}(\text{Srn}_{\phi=HN_{\psi}(z_j)}, \xi_i) - \mathcal{I}_i^j \right\|$$

Test time: initialize new embedding



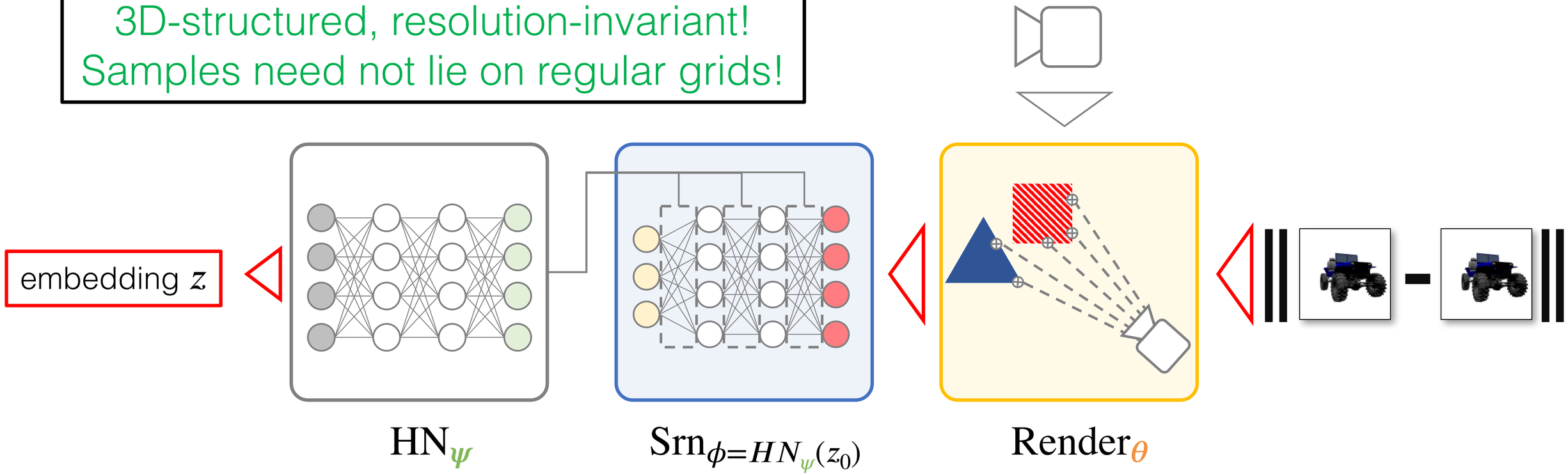
Inference: Freeze model weights, backprop into embedding only!



$$\hat{z} = \underset{z}{\text{argmin}} \left\| \text{Render}_{\theta}(\text{Srn}_{\phi=HN_{\psi}(z)}, \xi) - \mathcal{I} \right\|$$

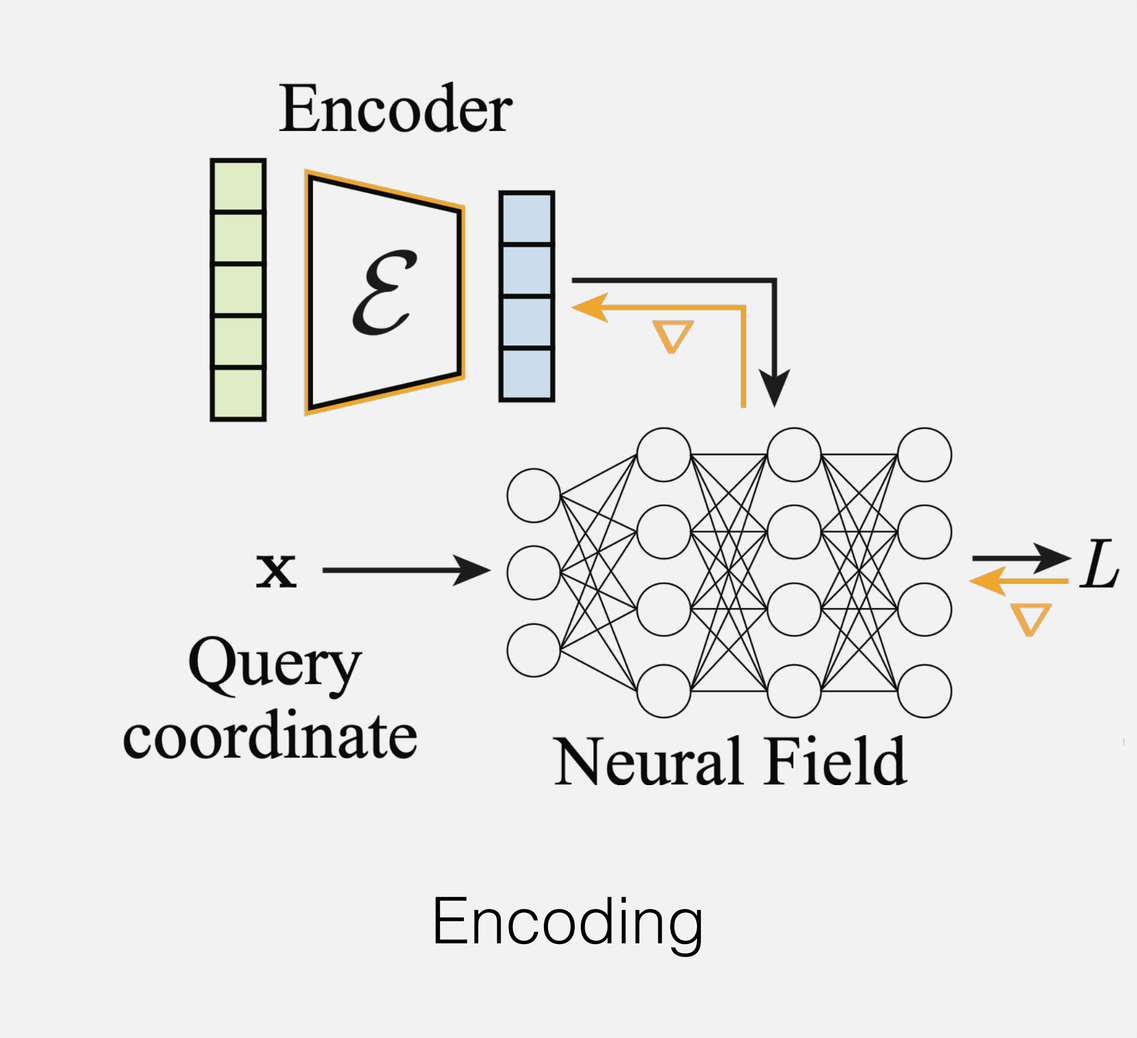
Inference: Freeze model weights, backprop into embedding only!

3D-structured, resolution-invariant!
Samples need not lie on regular grids!



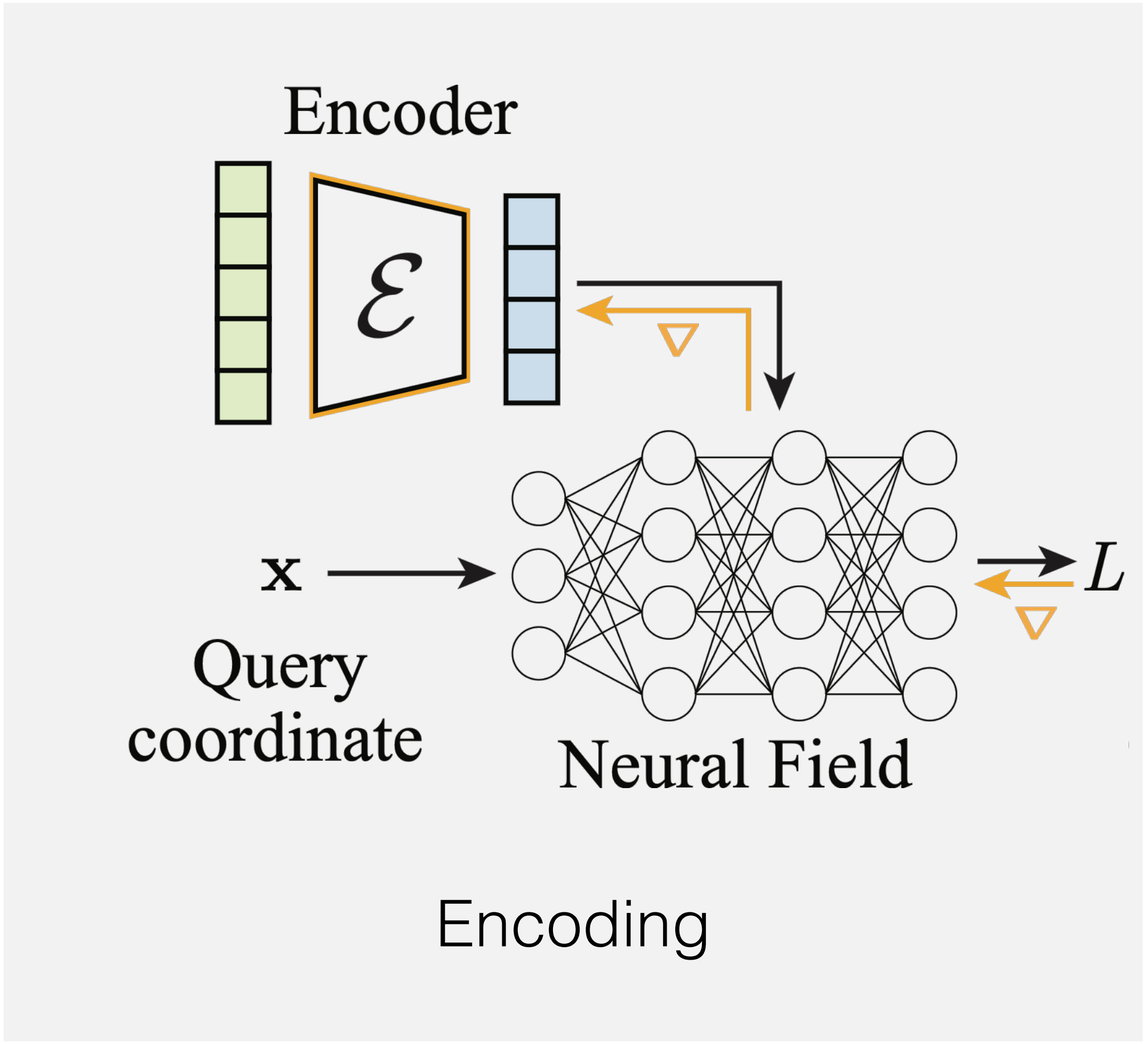
$$\hat{z} = \underset{z}{\text{argmin}} \left\| \text{Render}_{\theta}(\text{Srn}_{\phi=HN_{\psi}(z)}, \xi) - \mathcal{I} \right\|$$

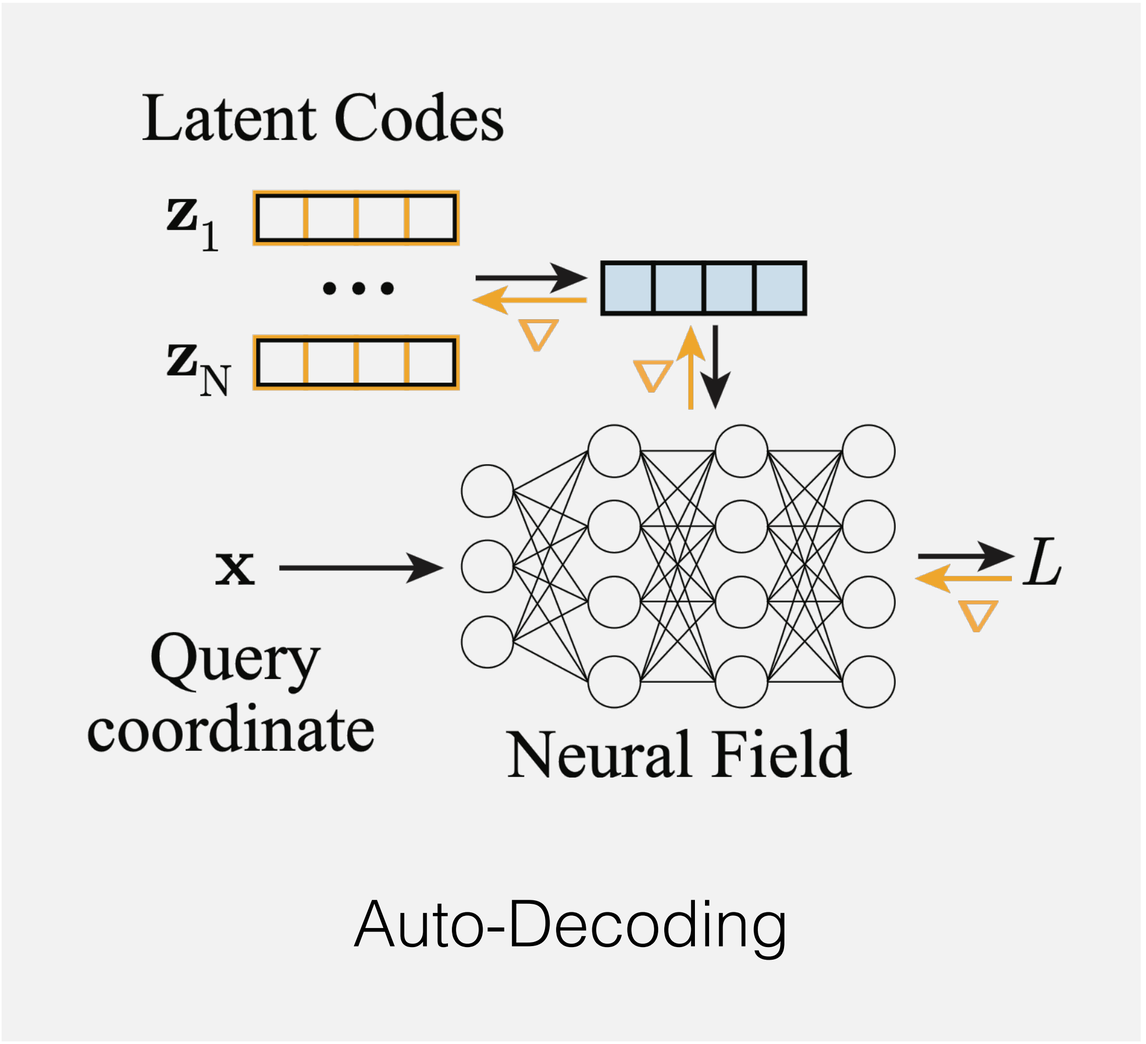
Encoding & Auto-Decoding



Neural Fields in Visual Computing and Beyond, Xie et al., EG STAR 2022

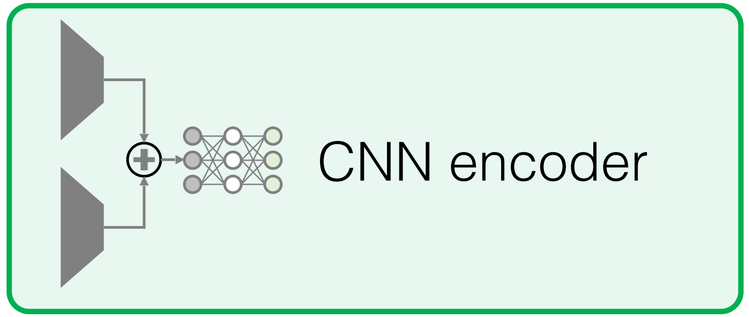
Encoding & Auto-Decoding





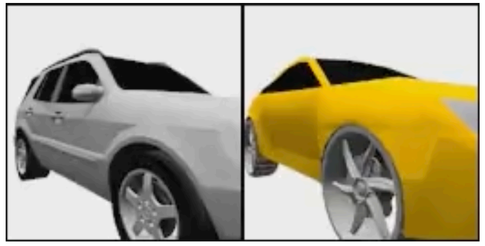
Neural Fields in Visual Computing and Beyond, Xie et al., EG STAR 2022

Out-of-distribution generalization

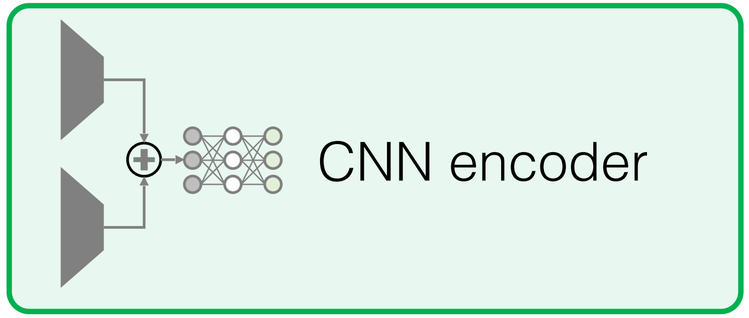


Input

Reconstructions

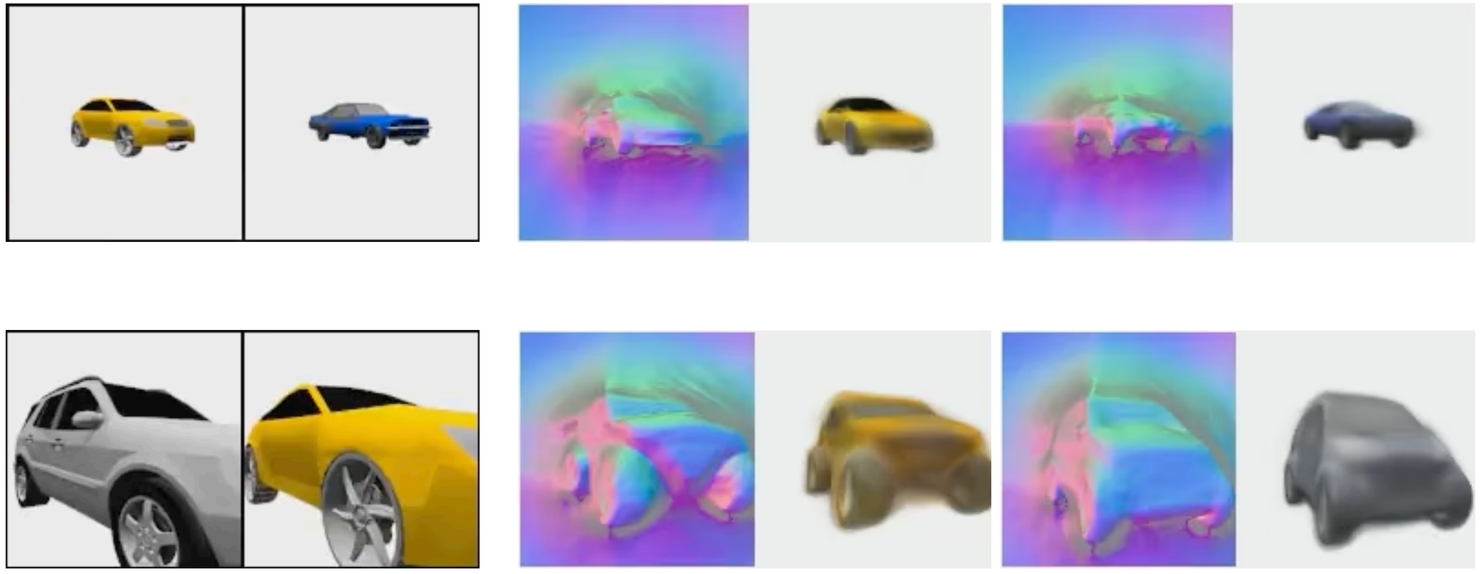


Out-of-distribution generalization

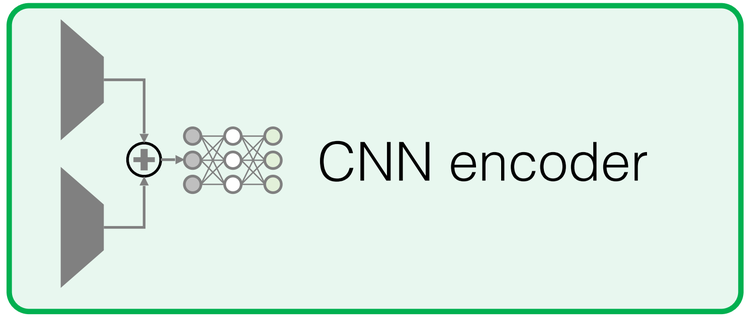


Input

Reconstructions



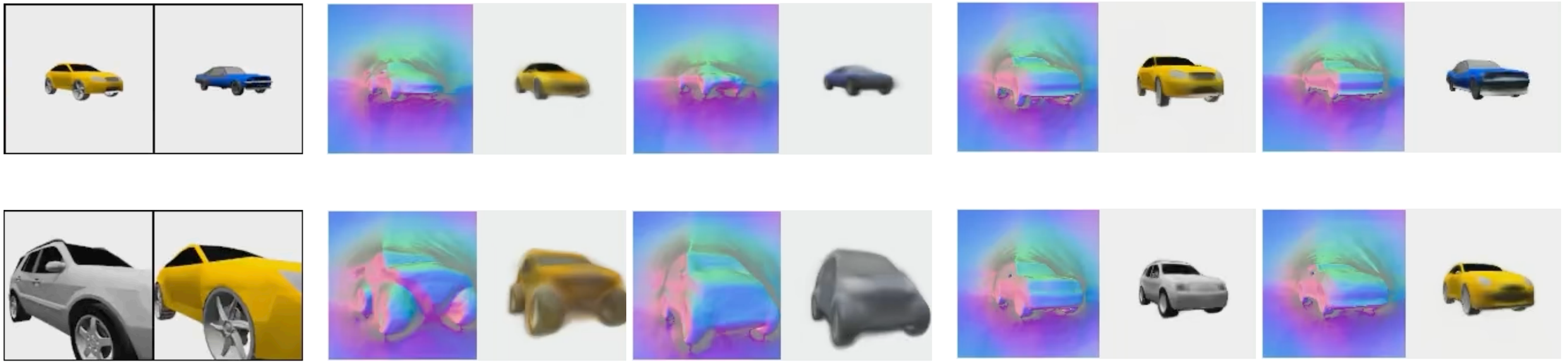
Out-of-distribution generalization



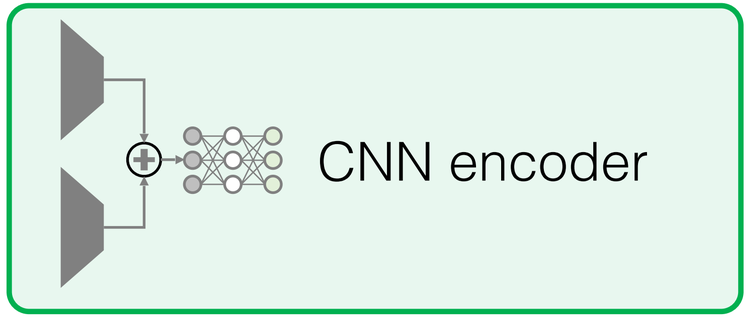
$$argmin_z ||Render(SRN, \xi) - \mathcal{I}||$$

Input

Reconstructions



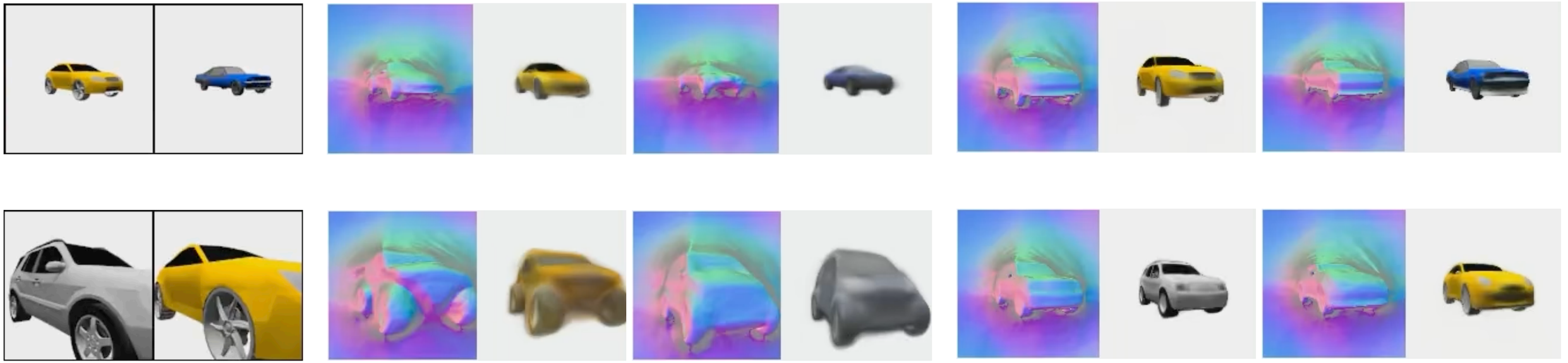
Out-of-distribution generalization



$$argmin_z ||Render(SRN, \xi) - \mathcal{I}||$$

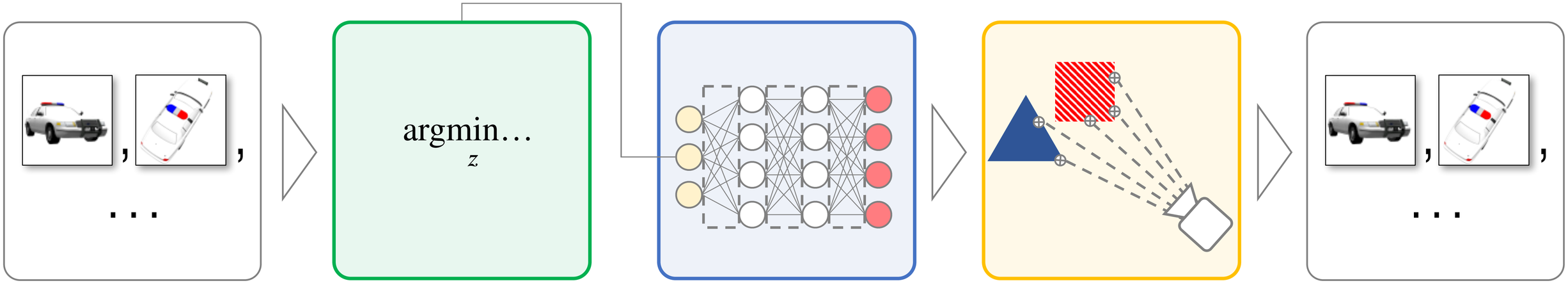
Input

Reconstructions

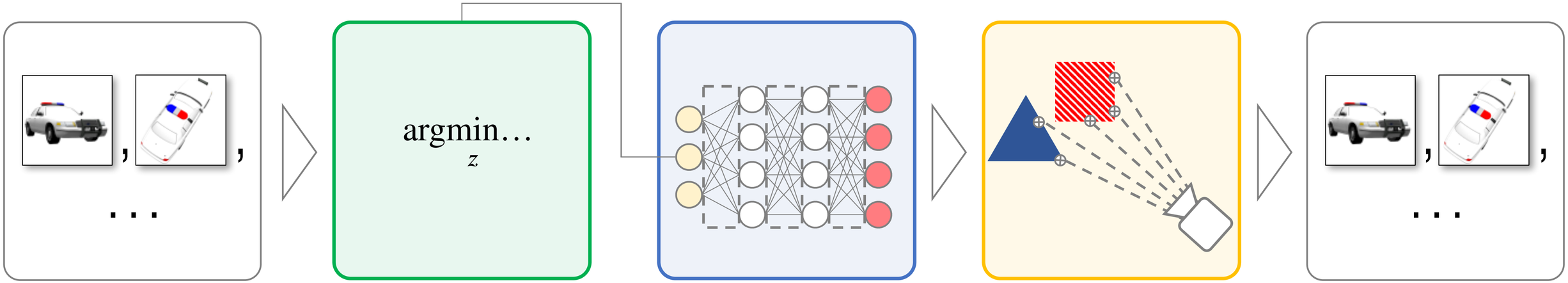


3D structure enables <u>generalization to out-of-distribution camera poses</u>!

Scene Representation Networks (Sitzmann et al., Neurips 2019)

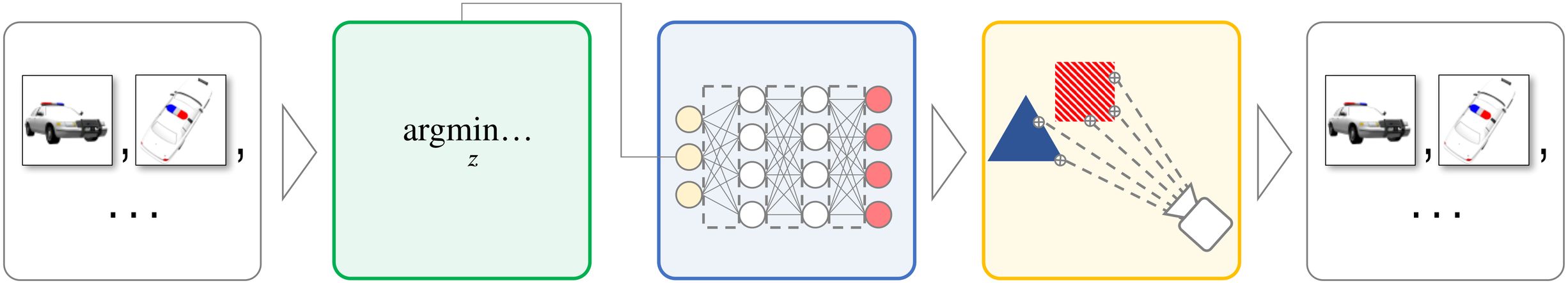


Scene Representation Networks (Sitzmann et al., Neurips 2019)



Enable reconstruction from incomplete observations!

Scene Representation Networks (Sitzmann et al., Neurips 2019)



Enable reconstruction from incomplete observations!

Scene understanding!

Single-shot reconstruction of held-out test objects

Input observation

SRNs (Ours)

Tatarchenko et al. 2015

Worrall et al. 2017

Deterministic GQN, adapted Eslami et al. 2018

Scene Representation Networks: Continuous 3D-structure Aware Neural Scene Representations, Sitzmann et al., NeurIPS

Single-shot reconstruction of held-out test objects

Input observation

SRNs (Ours)

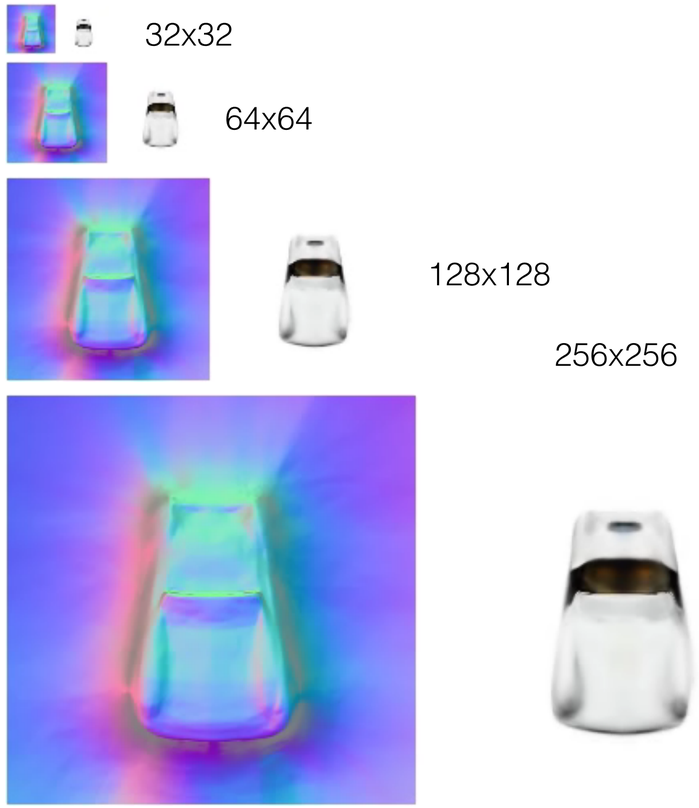
Tatarchenko et al. 2015

Worrall et al. 2017

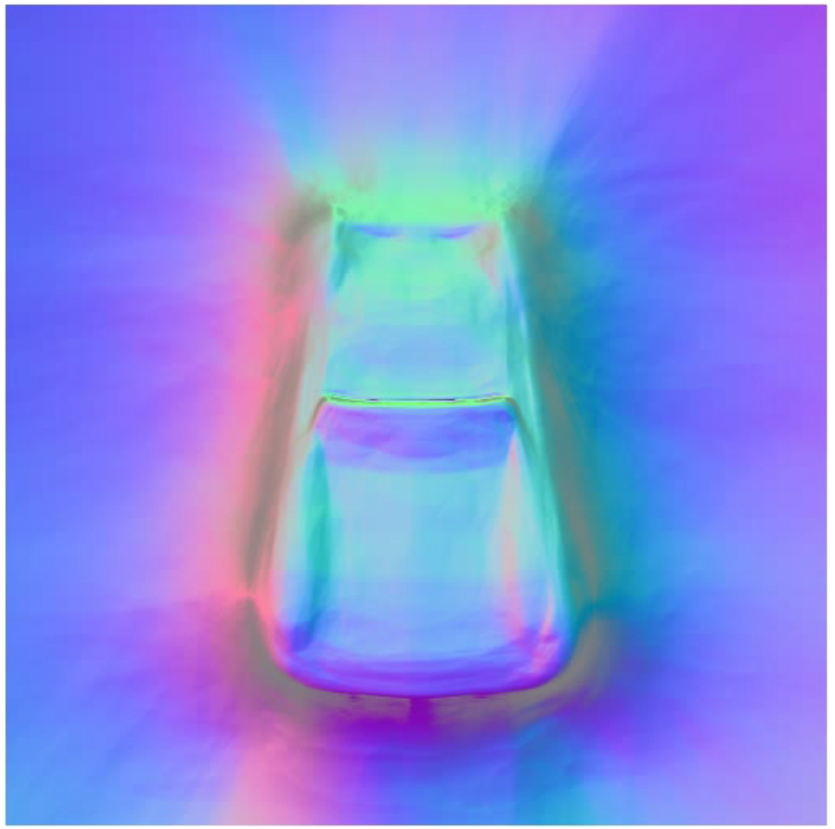
Deterministic GQN, adapted Eslami et al. 2018

Scene Representation Networks: Continuous 3D-structure Aware Neural Scene Representations, Sitzmann et al., NeurIPS

Sampling at arbitrary resolutions



512x512

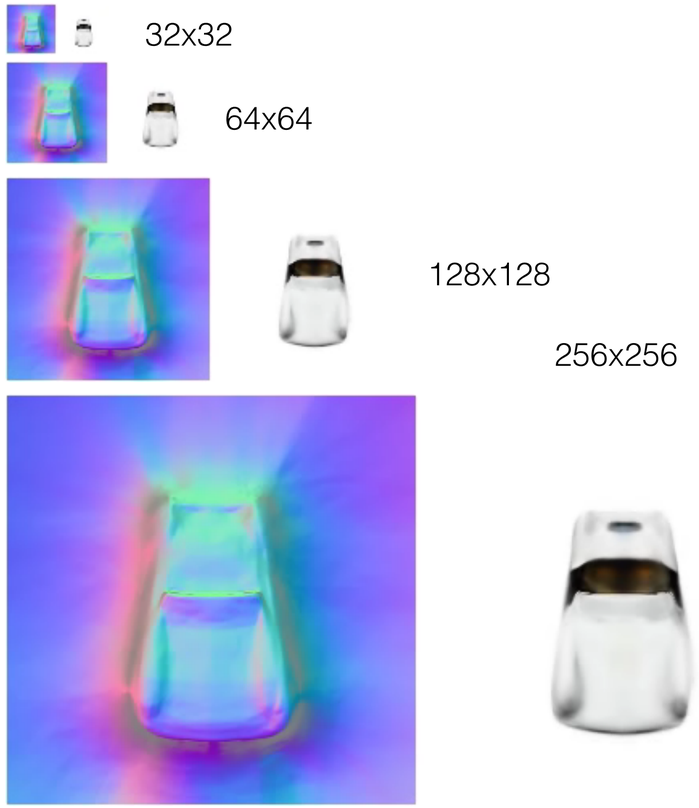


Surface Normals

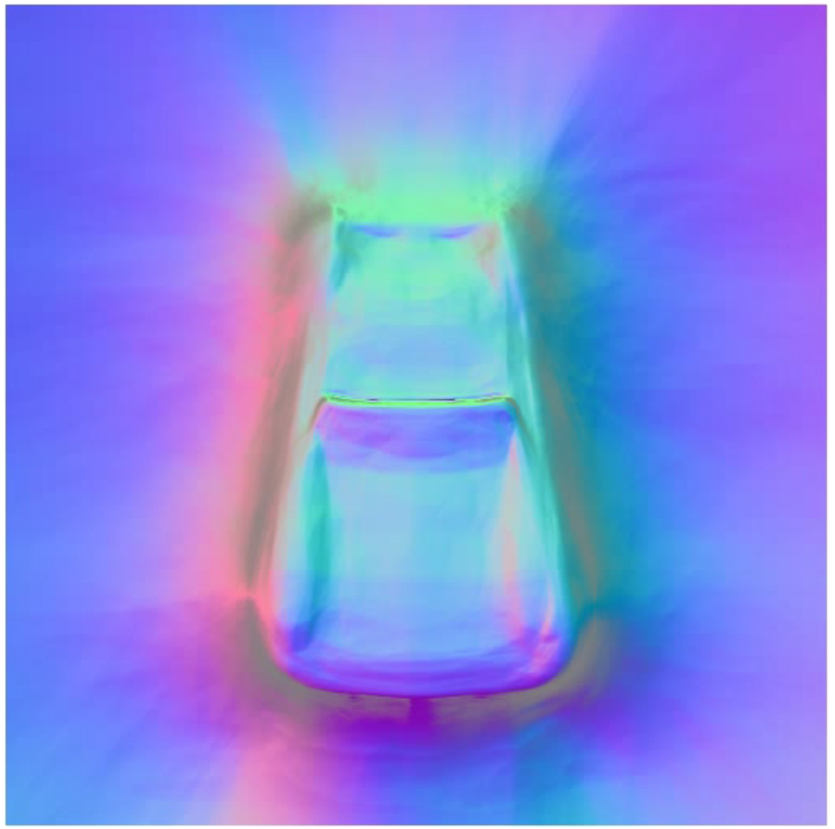
RGB

Scene Representation Networks: Continuous 3D-structure Aware Neural Scene Representations, Sitzmann et al., NeurIPS

Sampling at arbitrary resolutions



512x512



Surface Normals

RGB

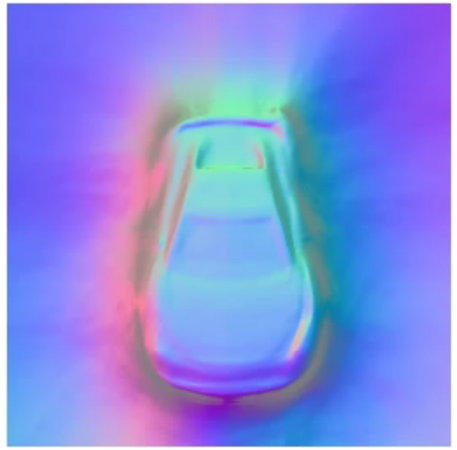
Scene Representation Networks: Continuous 3D-structure Aware Neural Scene Representations, Sitzmann et al., NeurIPS

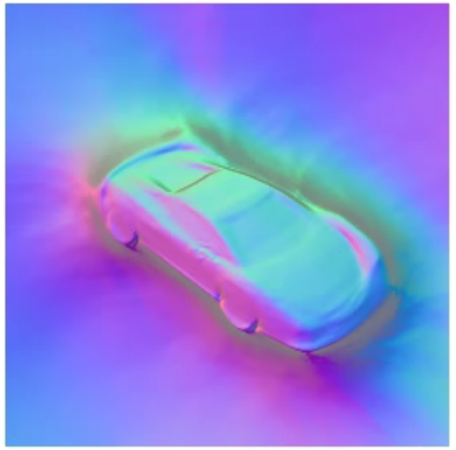
Generalization to unseen camera poses

Camera close-up

Camera Roll

SRNs





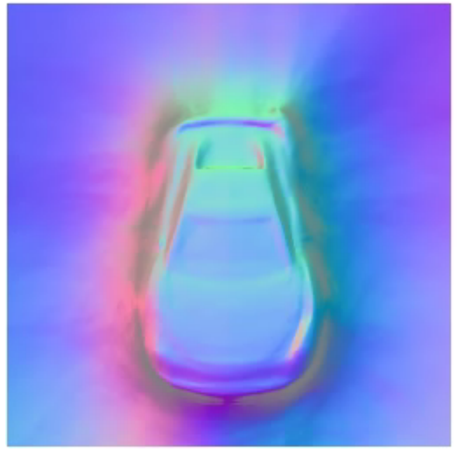
Scene Representation Networks: Continuous 3D-structure Aware Neural Scene Representations, Sitzmann et al., NeurIPS

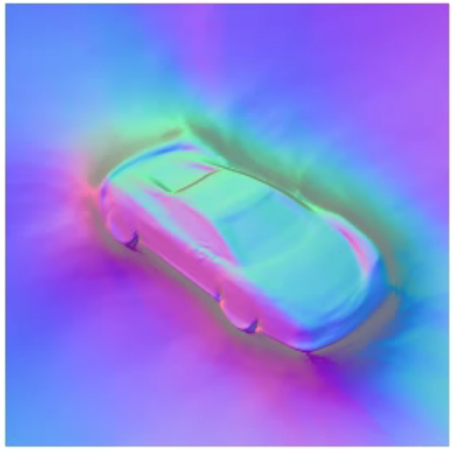
Generalization to unseen camera poses

Camera close-up

Camera Roll

SRNs





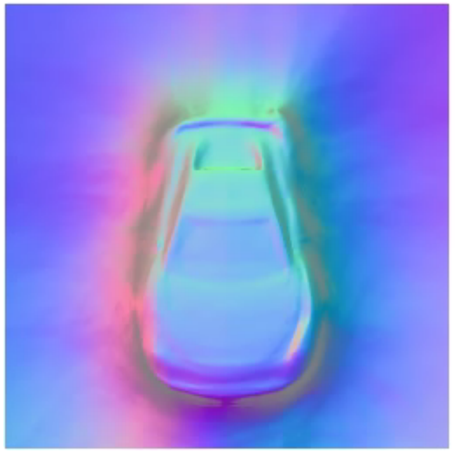
Scene Representation Networks: Continuous 3D-structure Aware Neural Scene Representations, Sitzmann et al., NeurIPS

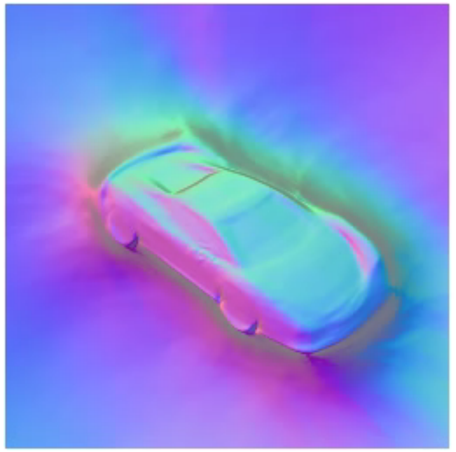
Generalization to unseen camera poses

Camera close-up

Camera Roll

SRNs





Tatarchenko et al.

Doesn’t reconstruct geometry

Doesn’t reconstruct geometry

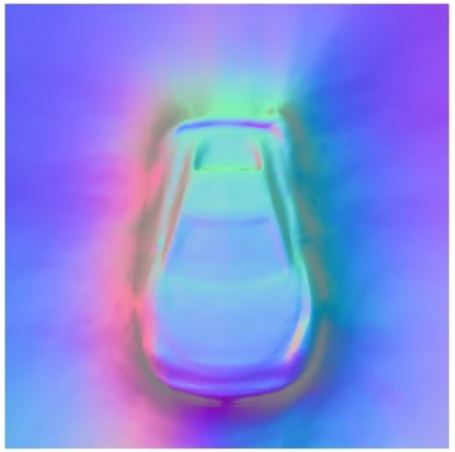
Scene Representation Networks: Continuous 3D-structure Aware Neural Scene Representations, Sitzmann et al., NeurIPS

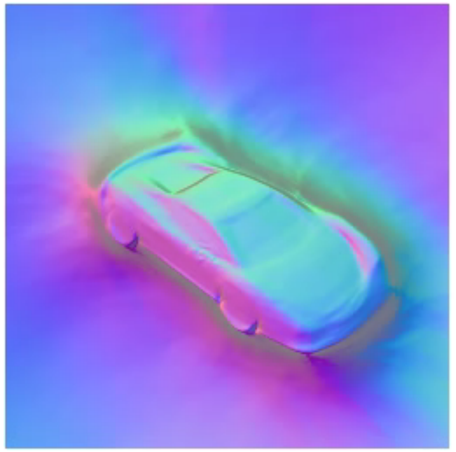
Generalization to unseen camera poses

Camera close-up

Camera Roll

SRNs





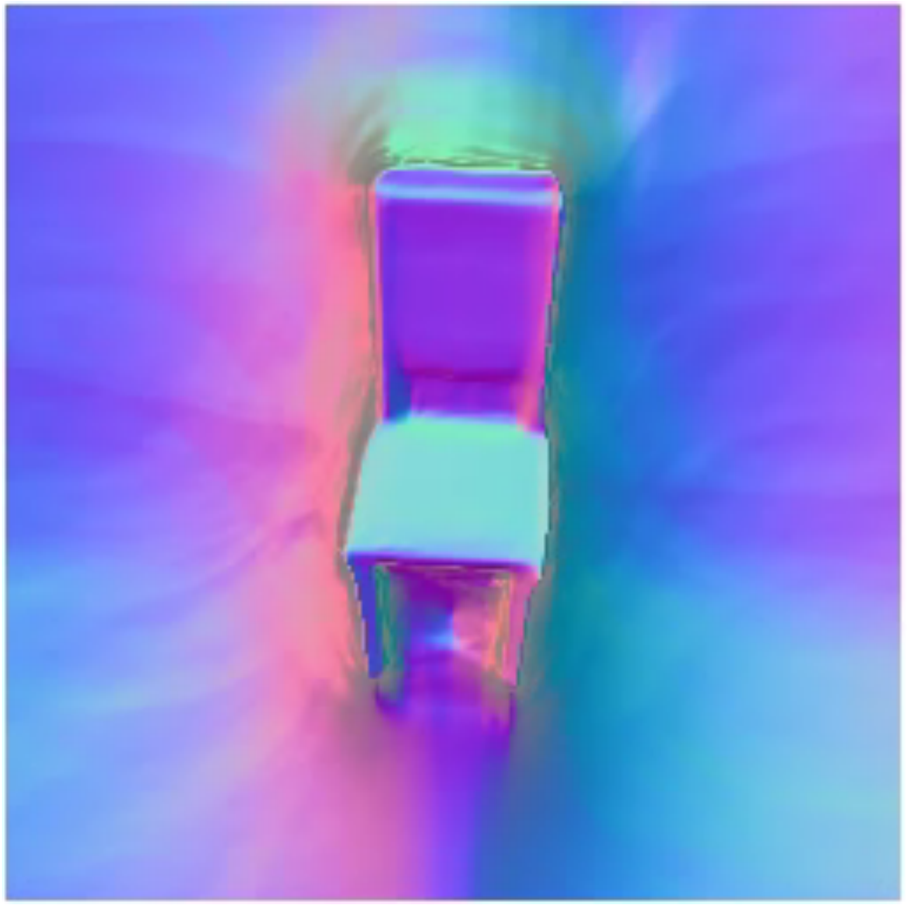
Tatarchenko et al.

Doesn’t reconstruct geometry

Doesn’t reconstruct geometry

Scene Representation Networks: Continuous 3D-structure Aware Neural Scene Representations, Sitzmann et al., NeurIPS

Latent code interpolation

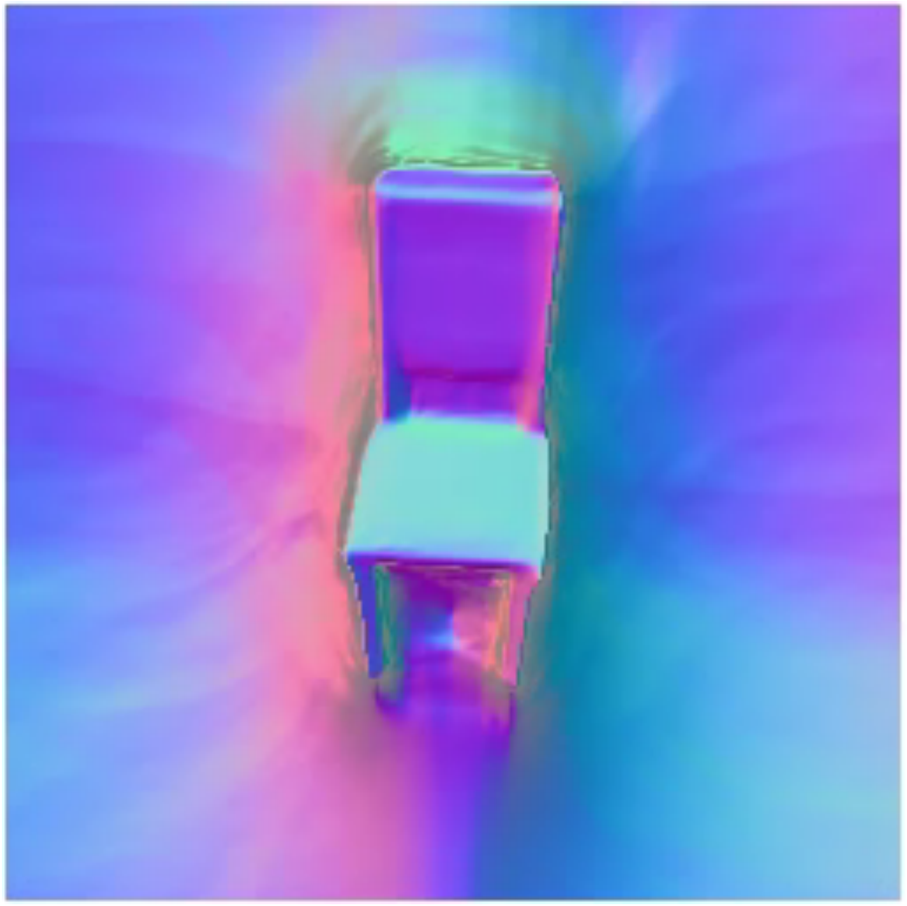


Surface Normals

RGB

Scene Representation Networks: Continuous 3D-structure Aware Neural Scene Representations, Sitzmann et al., NeurIPS

Latent code interpolation

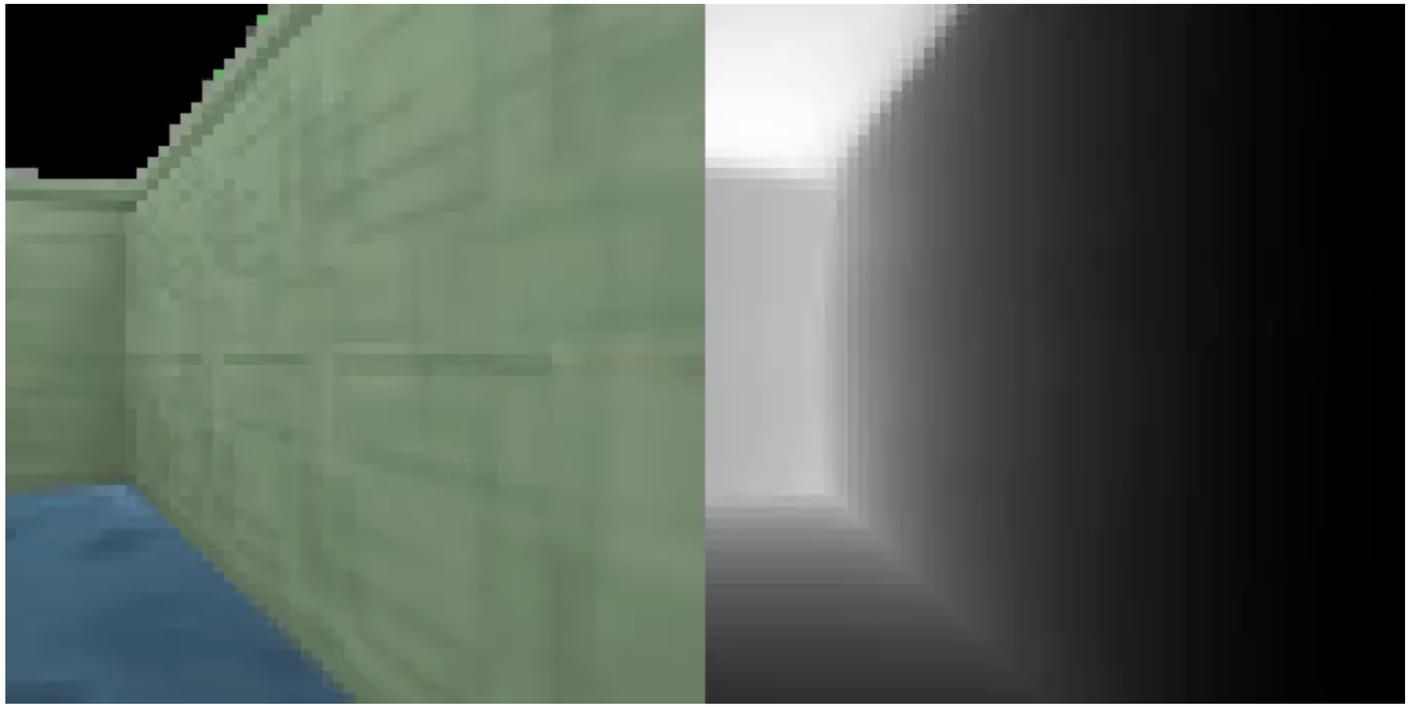


Surface Normals

RGB

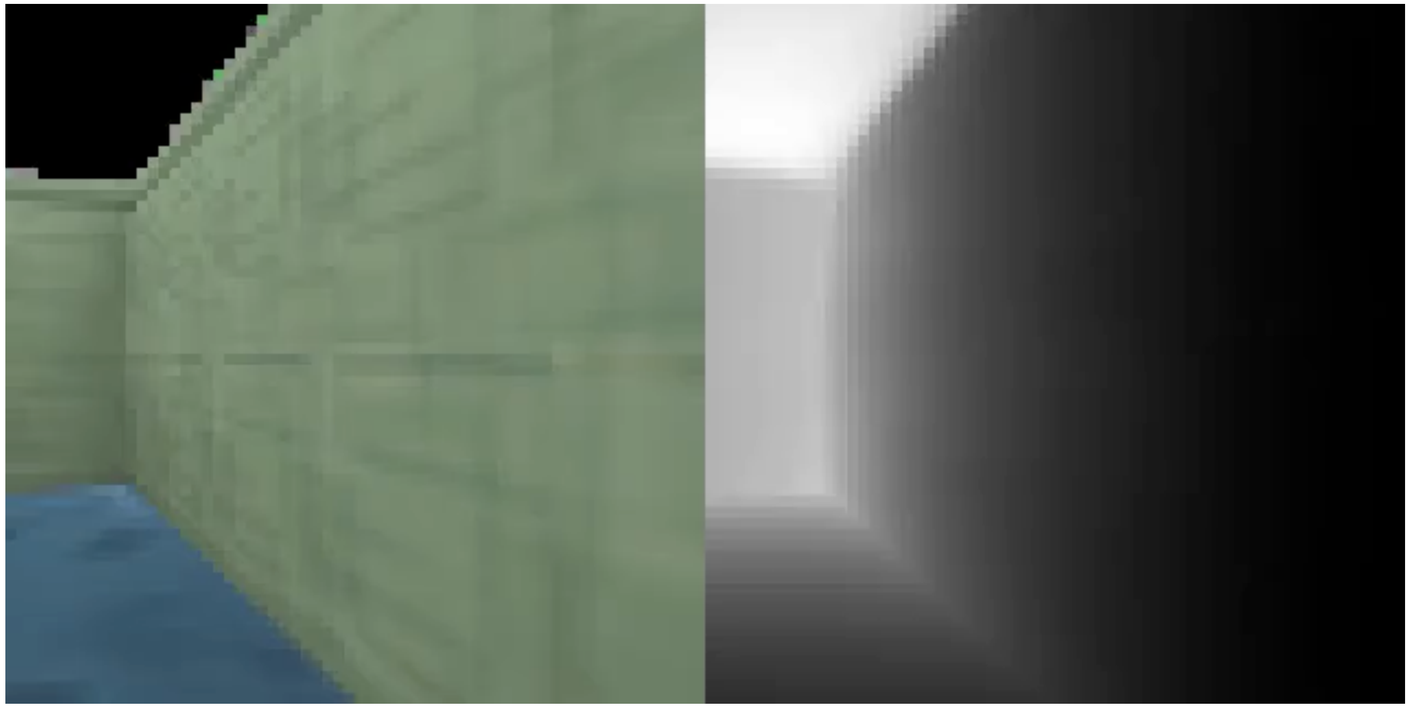
Scene Representation Networks: Continuous 3D-structure Aware Neural Scene Representations, Sitzmann et al., NeurIPS

Can generalize across simple room-scale scenes.



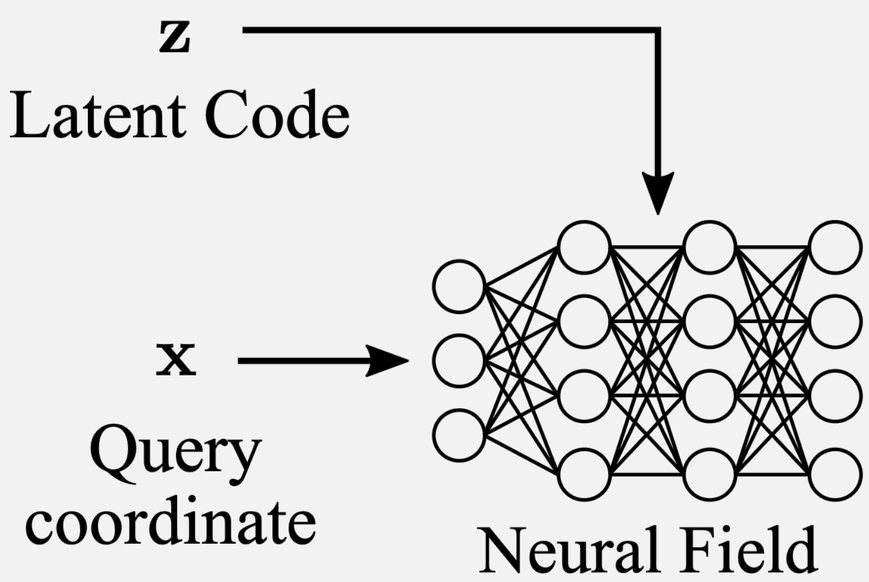
Scene Representation Networks: Continuous 3D-structure Aware Neural Scene Representations, Sitzmann et al., NeurIPS

Can generalize across simple room-scale scenes.

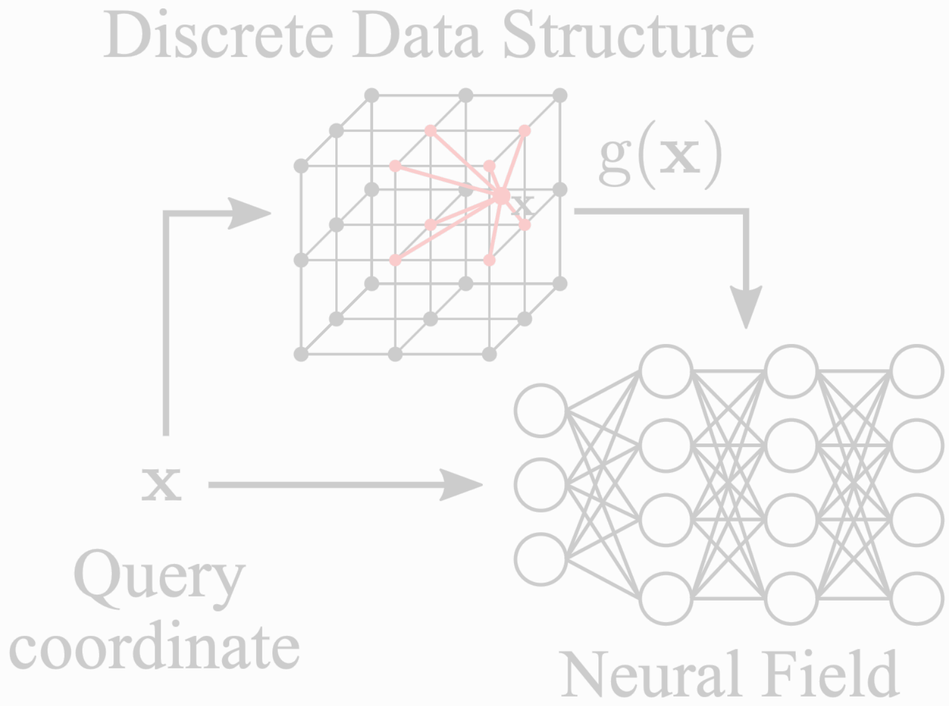


Scene Representation Networks: Continuous 3D-structure Aware Neural Scene Representations, Sitzmann et al., NeurIPS

Global Conditioning: Single Latent Code for whole 3D Scene



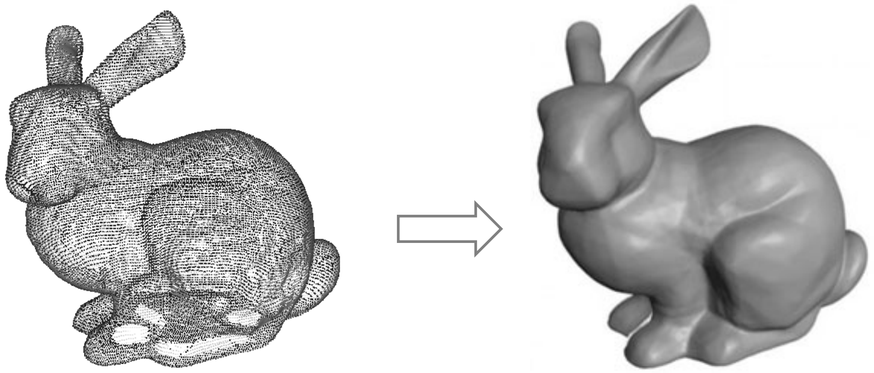
Global Conditioning



Local Conditioning

Neural Fields in Visual Computing and Beyond, Xie et al., EG STAR 2022

Global Latent Codes: Enables reconstruction from partial observations!

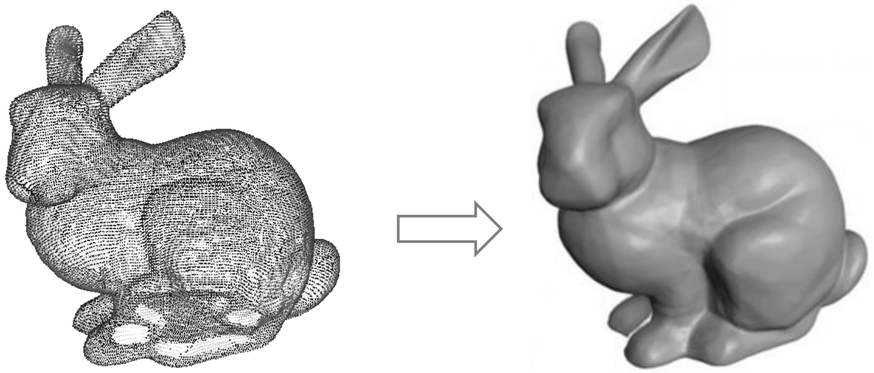


DeepSDF, Occupancy Networks, IM-Net

Scene Representation Networks: Continuous
3D-Structure-Aware Neural Scene Representations, NeurIPS 2019.

Differential Volumetric Rendering,
Niemeyer et al., CVPR 2020

Global Latent Codes: Enables reconstruction from partial observations!



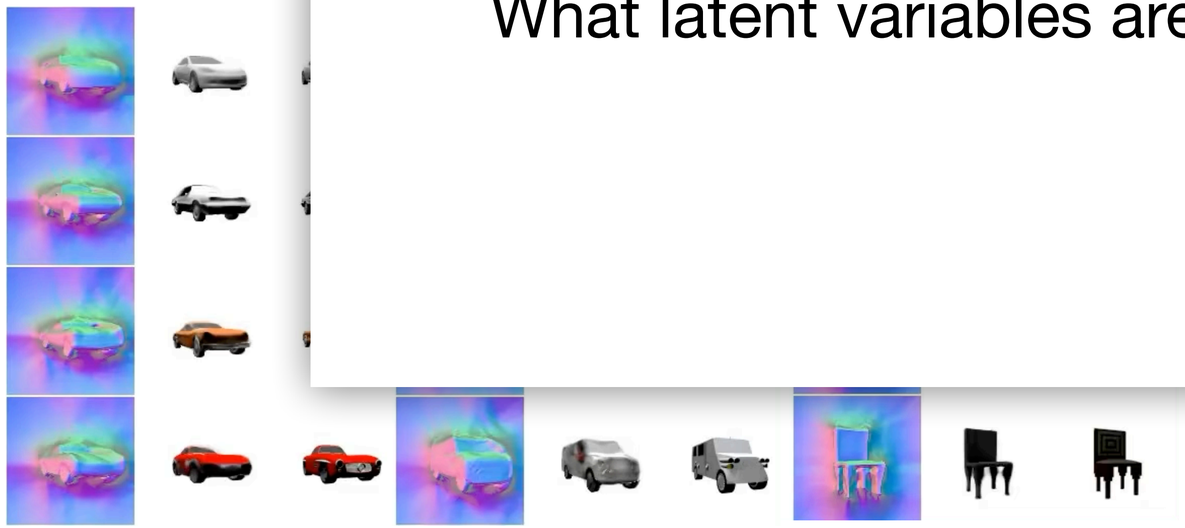
DeepSDF, Occupancy Networks, IM-Net

Scene Representation Networks: Continuous
3D-Structure-Aware Neural Scene Representations, NeurIPS 2019.

Differential Volumetric Rendering,
Niemeyer et al., CVPR 2020

Global Latent Codes: Enables reconstruction from partial observations!

Dee

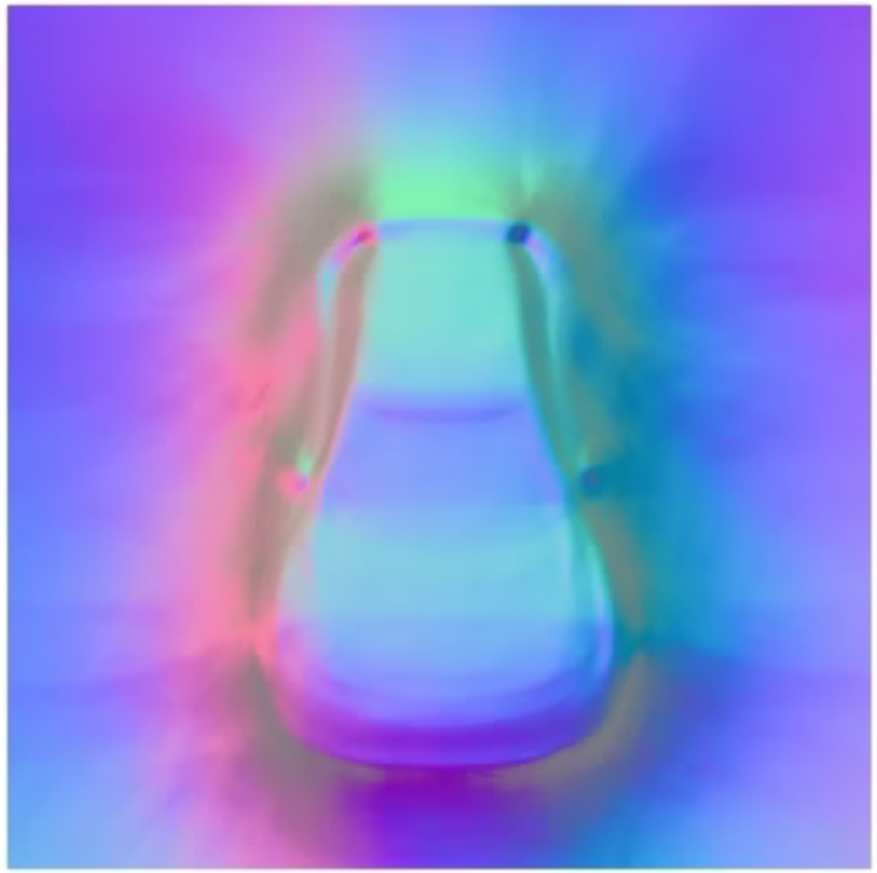


What latent variables are discovered by the model?

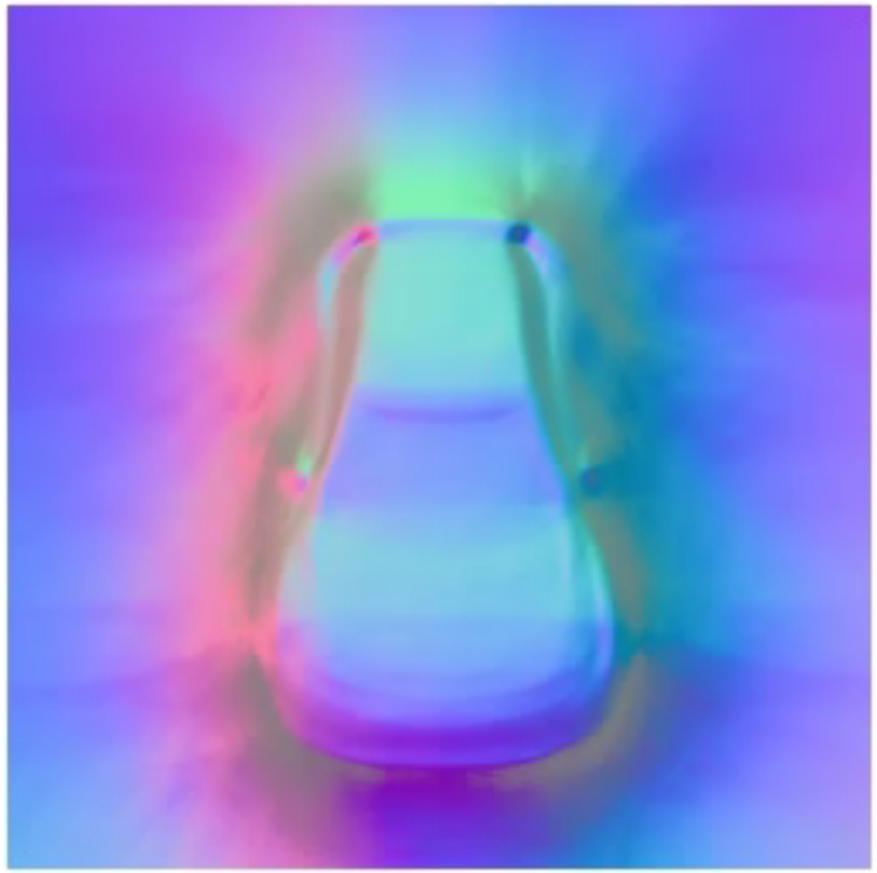
Scene Representation Networks: Continuous 3D-Structure-Aware Neural Scene Representations, NeurIPS 2019.

Differential Volumetric Rendering, Niemeyer et al., CVPR 2020

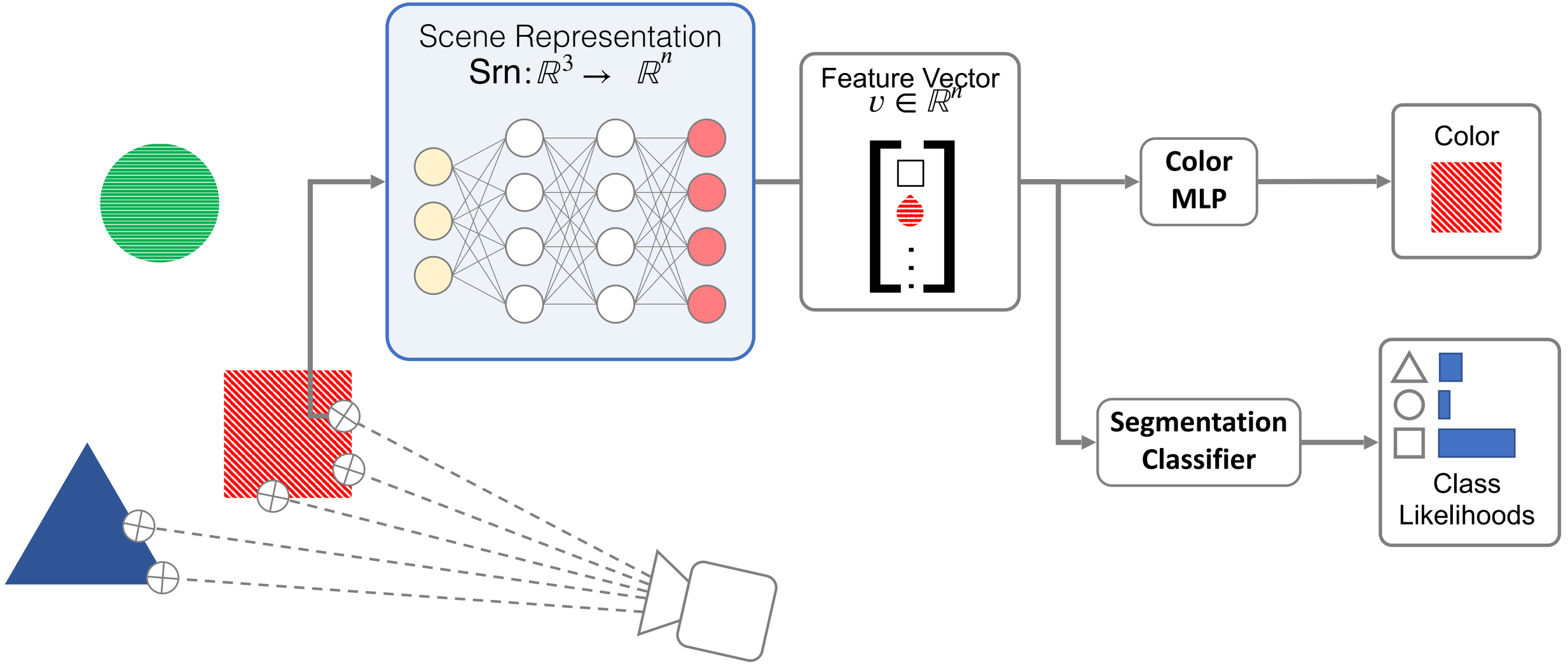
Latent Space Interpolation



Latent Space Interpolation

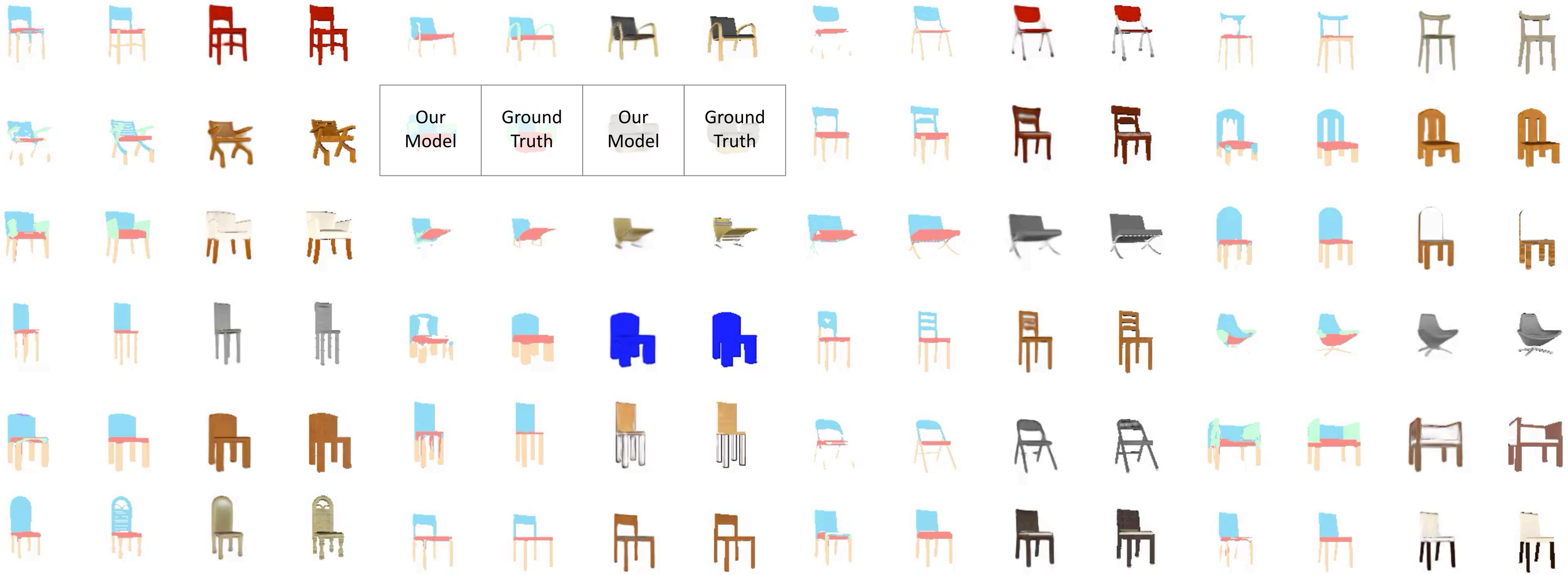


Learned features enable semantic segmentation from few labels.



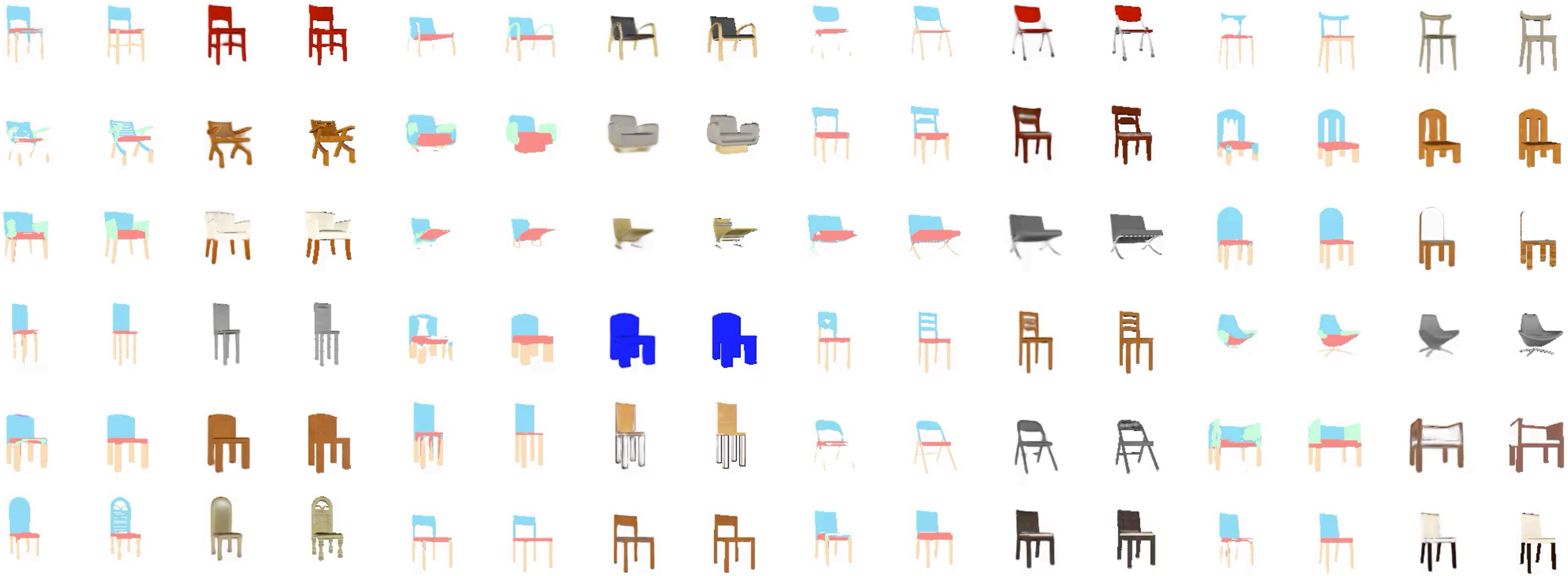
3D Representation Learning for Semantic Segmentation, Kohli et al., 3DV, 2020

Learned features enable semantic segmentation from few labels.



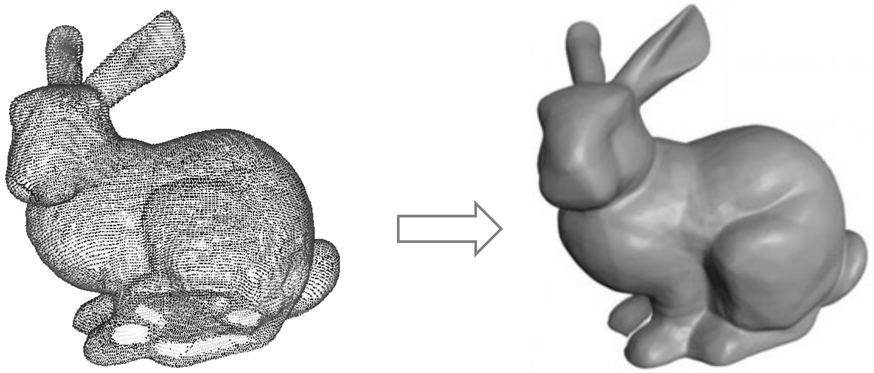
Kohli et al., 3DV, 2020

Learned features enable semantic segmentation from few labels.



Kohli et al., 3DV, 2020

Global Latent Codes: Enables reconstruction from partial observations!

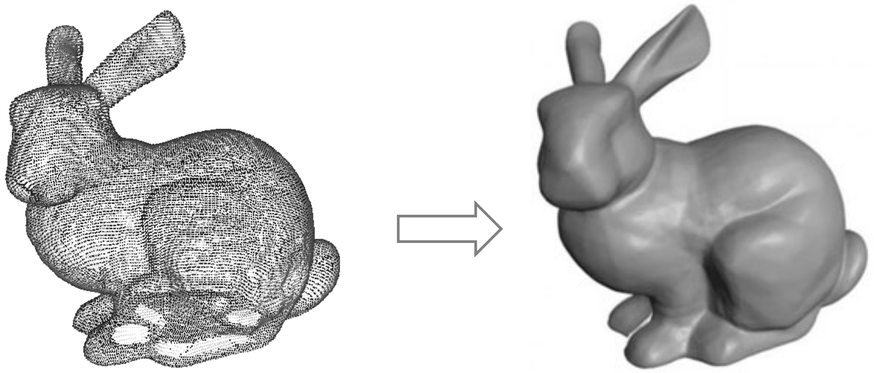


DeepSDF, Occupancy Networks, IM-Net

Scene Representation Networks: Continuous
3D-Structure-Aware Neural Scene Representations, NeurIPS 2019.

Differential Volumetric Rendering,
Niemeyer et al., CVPR 2020

Global Latent Codes: Enables reconstruction from partial observations!

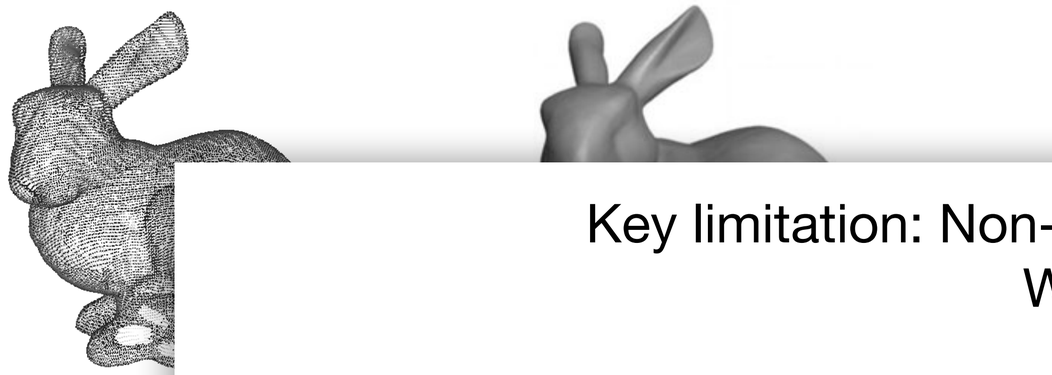


DeepSDF, Occupancy Networks, IM-Net

Scene Representation Networks: Continuous
3D-Structure-Aware Neural Scene Representations, NeurIPS 2019.

Differential Volumetric Rendering,
Niemeyer et al., CVPR 2020

Global Latent Codes: Enables reconstruction from partial observations!



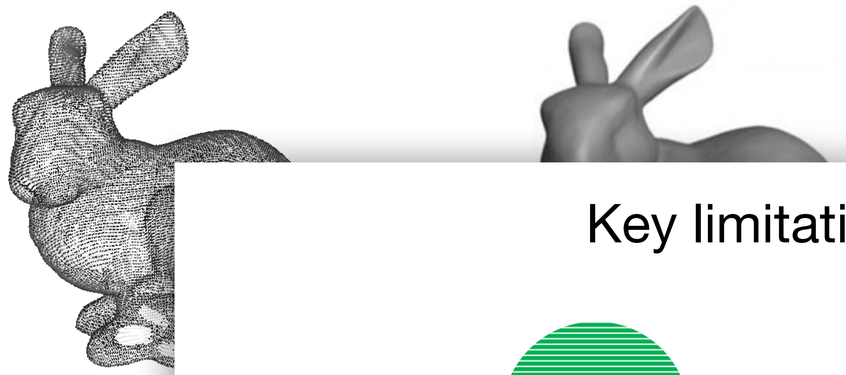
Dee

Key limitation: Non-compositional scenes.
Why?

Scene Representation Networks: Continuous 3D-Structure-Aware Neural Scene Representations, NeurIPS 2019.

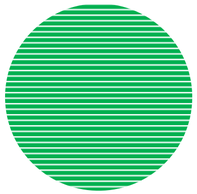
Differential Volumetric Rendering, Niemeyer et al., CVPR 2020

Global Latent Codes: Enables reconstruction from partial observations!



Dee

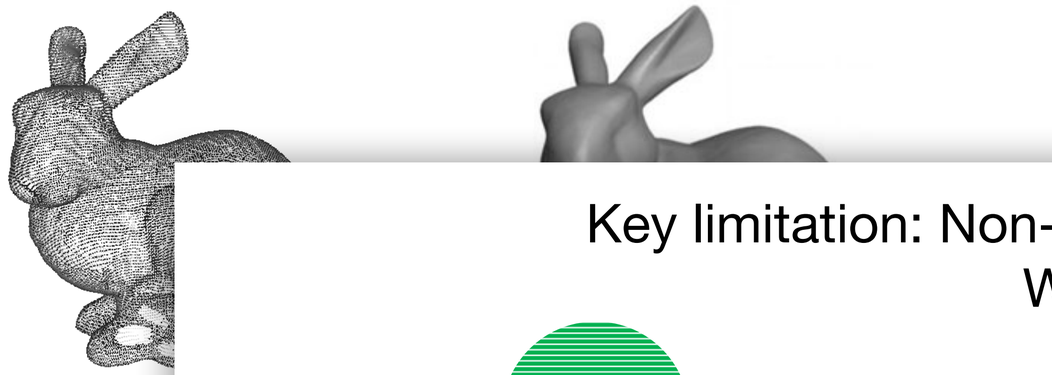
Key limitation: Non-compositional scenes.
Why?



Scene Representation Networks: Continuous 3D-Structure-Aware Neural Scene Representations, NeurIPS 2019.

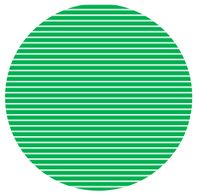
Differential Volumetric Rendering, Niemeyer et al., CVPR 2020

Global Latent Codes: Enables reconstruction from partial observations!



Dee

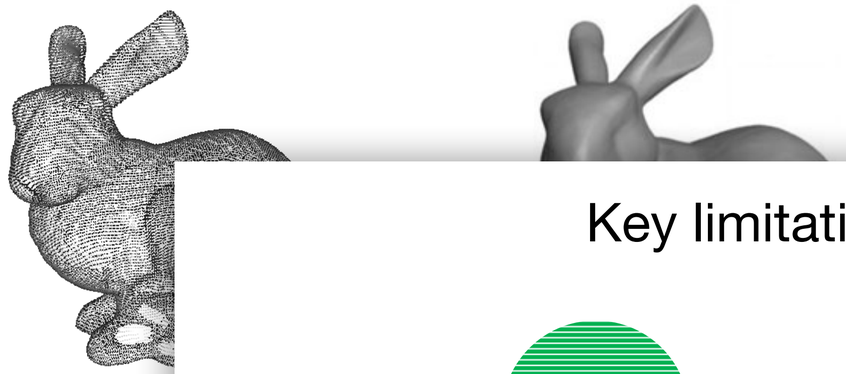
Key limitation: Non-compositional scenes.
Why?



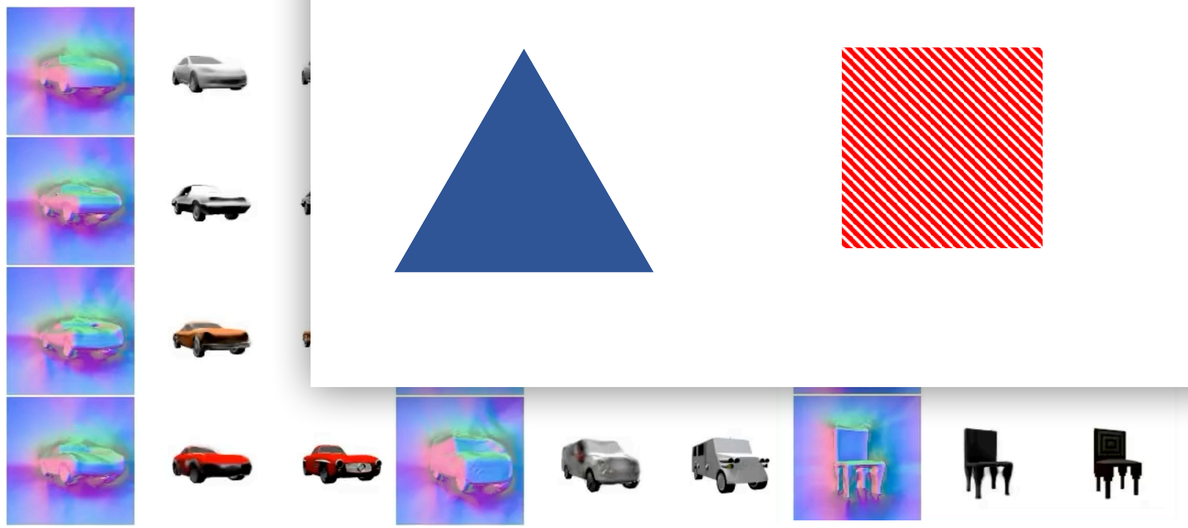
Scene Representation Networks: Continuous 3D-Structure-Aware Neural Scene Representations, NeurIPS 2019.

Differential Volumetric Rendering, Niemeyer et al., CVPR 2020

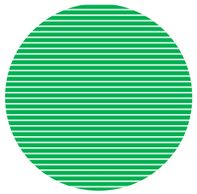
Global Latent Codes: Enables reconstruction from partial observations!



Dee



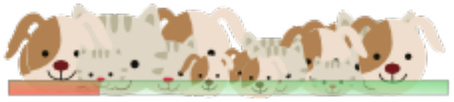
Key limitation: Non-compositional scenes.
Why?



Scene Representation Networks: Continuous 3D-Structure-Aware Neural Scene Representations, NeurIPS 2019.

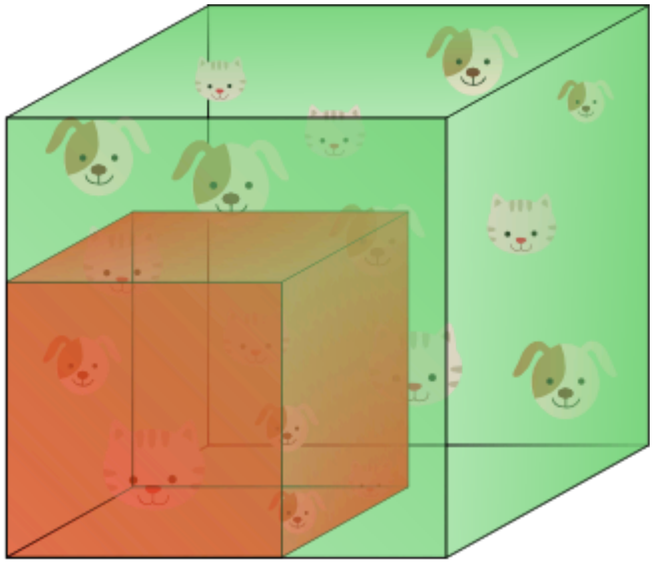
Differential Volumetric Rendering, Niemeyer et al., CVPR 2020

Reminder: Curse of Dimensionality



1D

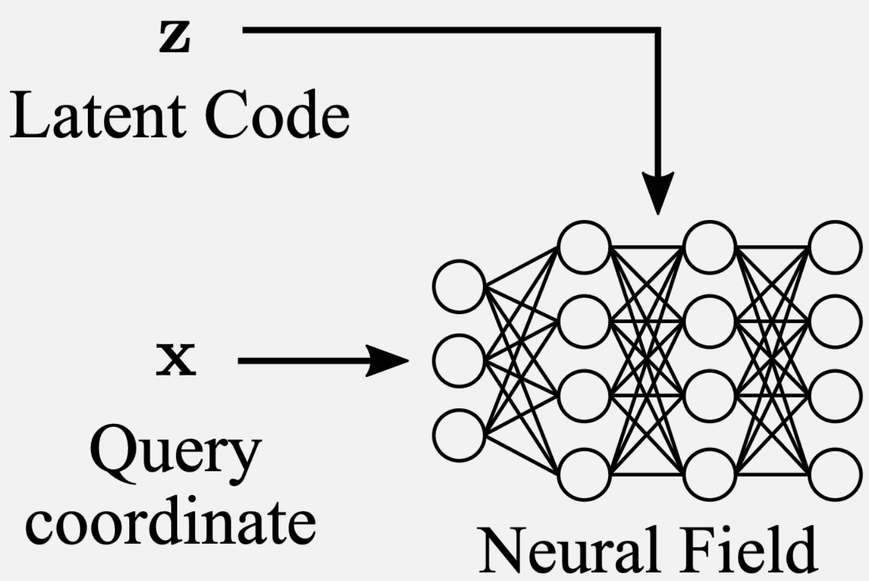
2D



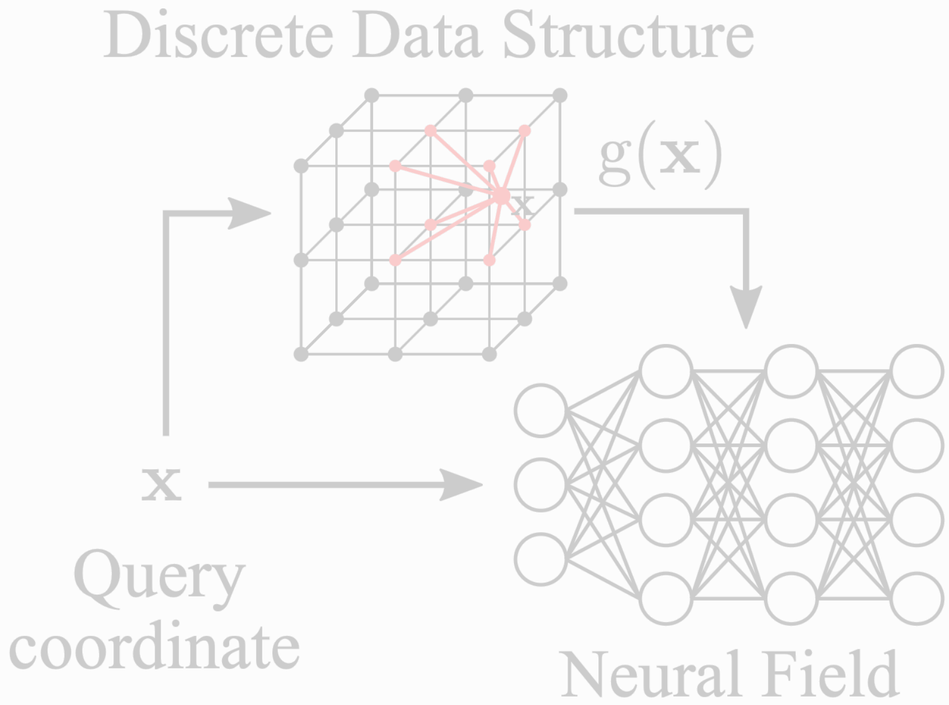
3D

...

Global Latent Codes



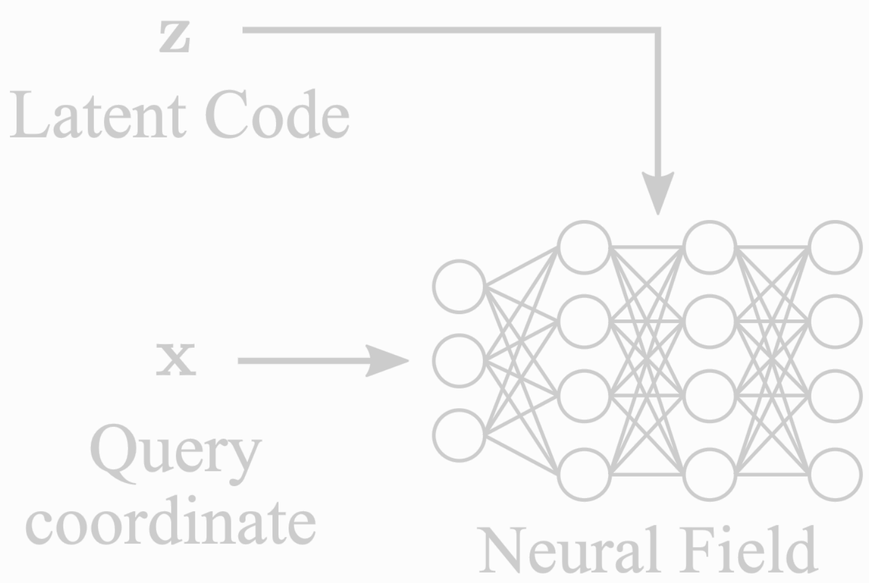
Global Conditioning



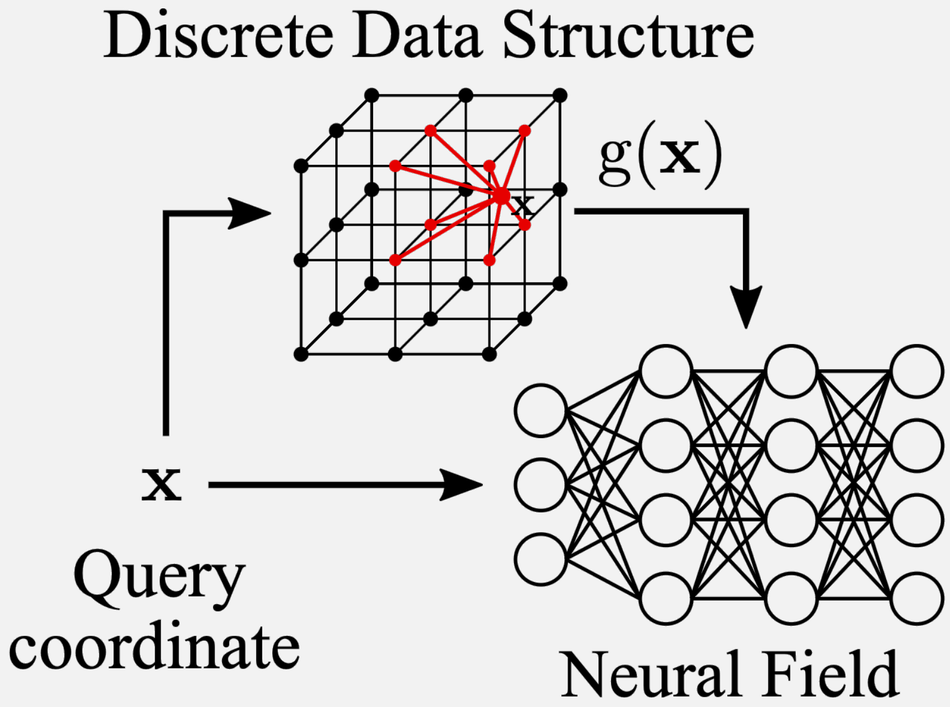
Local Conditioning

Neural Fields in Visual Computing and Beyond, Xie et al., EG STAR 2022

Local Latent Codes



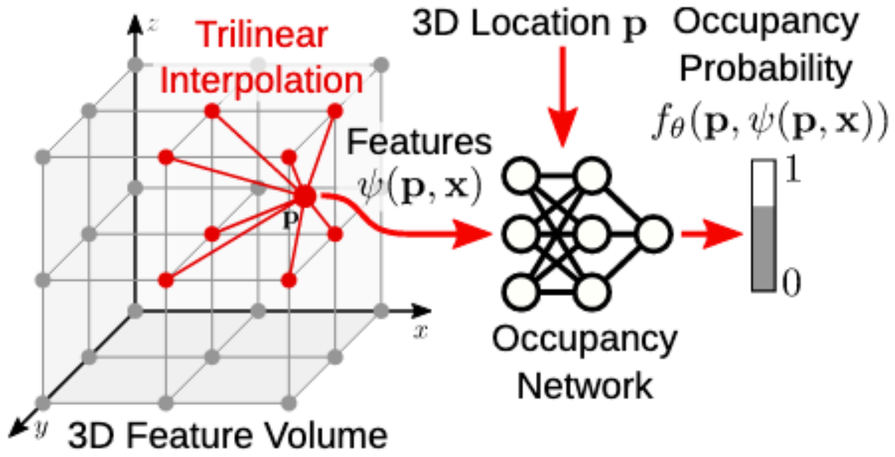
Global Conditioning



Local Conditioning

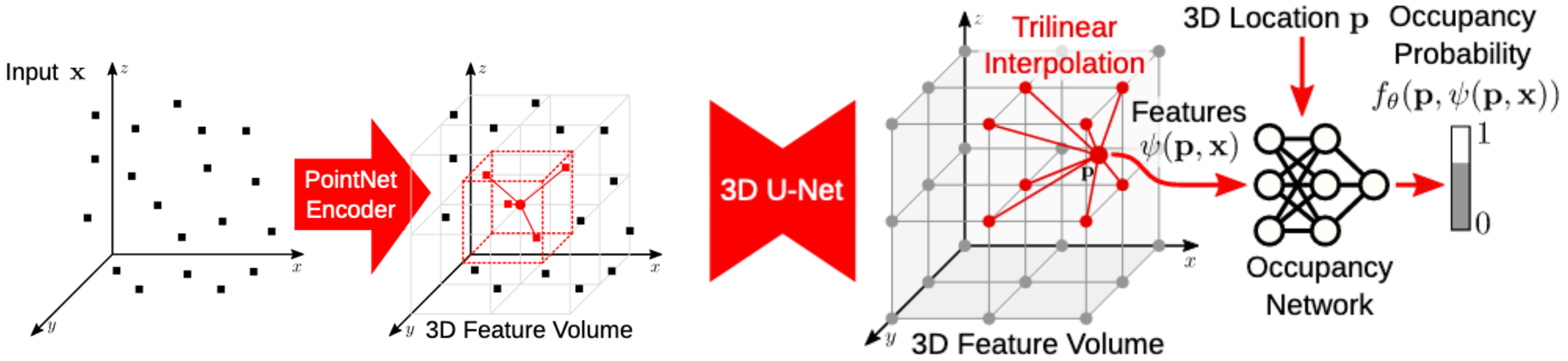
Neural Fields in Visual Computing and Beyond, Xie et al., EG STAR 2022

From point clouds: Conditioning on Feature Voxel grids



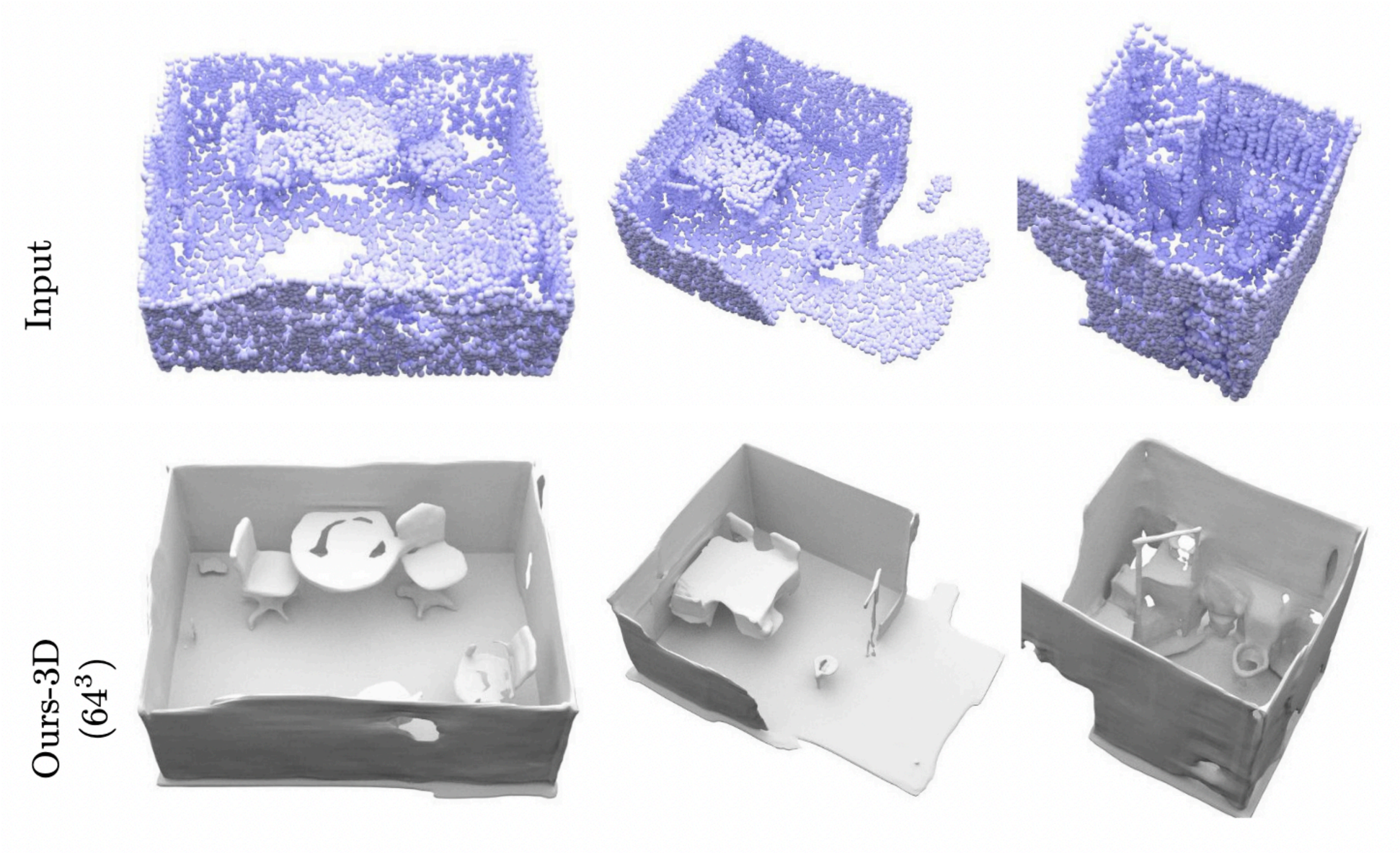
Convolutional Occupancy Networks [Peng et al. 2020]
Local Implicit Grid Representations for 3D Scenes [Jiang et al. 2020]
Implicit Functions in Feature Space for 3D Shape Reconstruction and Completion [Chabra et al. 2020]
Deep Local Shapes: Learning Local SDF Priors for Detailed 3D Reconstruction [Chibane et al. 2020]

From point clouds: Conditioning on Feature Voxel grids

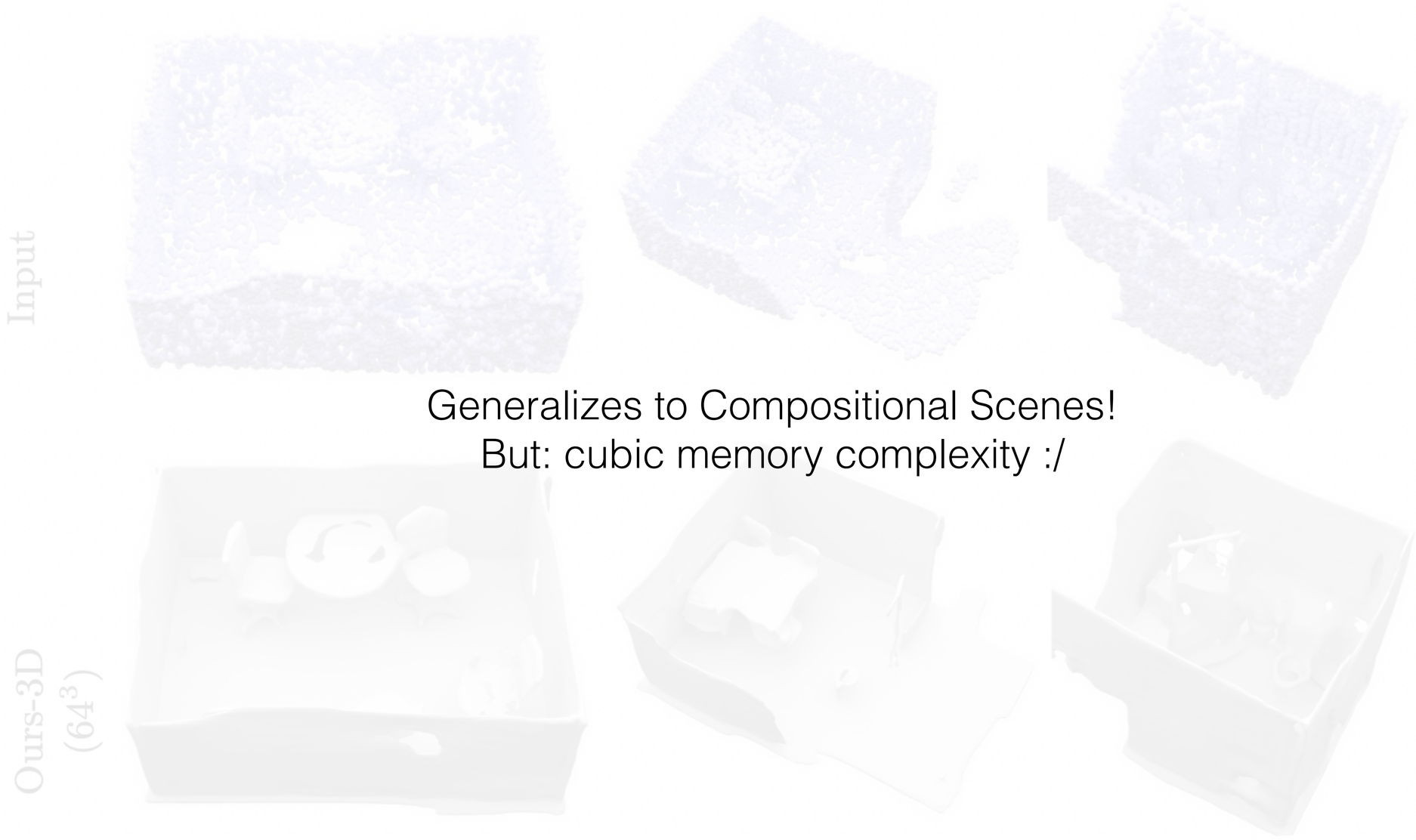


Convolutional Occupancy Networks [Peng et al. 2020]
Local Implicit Grid Representations for 3D Scenes [Jiang et al. 2020]
Implicit Functions in Feature Space for 3D Shape Reconstruction and Completion [Chabra et al. 2020]
Deep Local Shapes: Learning Local SDF Priors for Detailed 3D Reconstruction [Chibane et al. 2020]

From point clouds: Conditioning on Feature Voxel grids

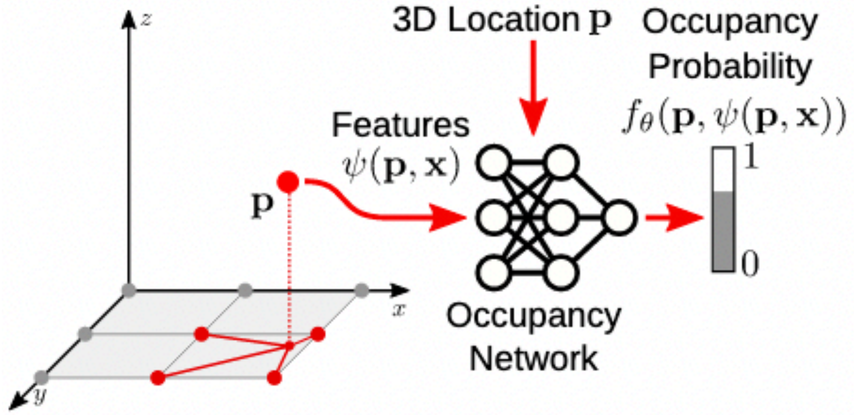


From point clouds: Conditioning on Feature Voxel grids



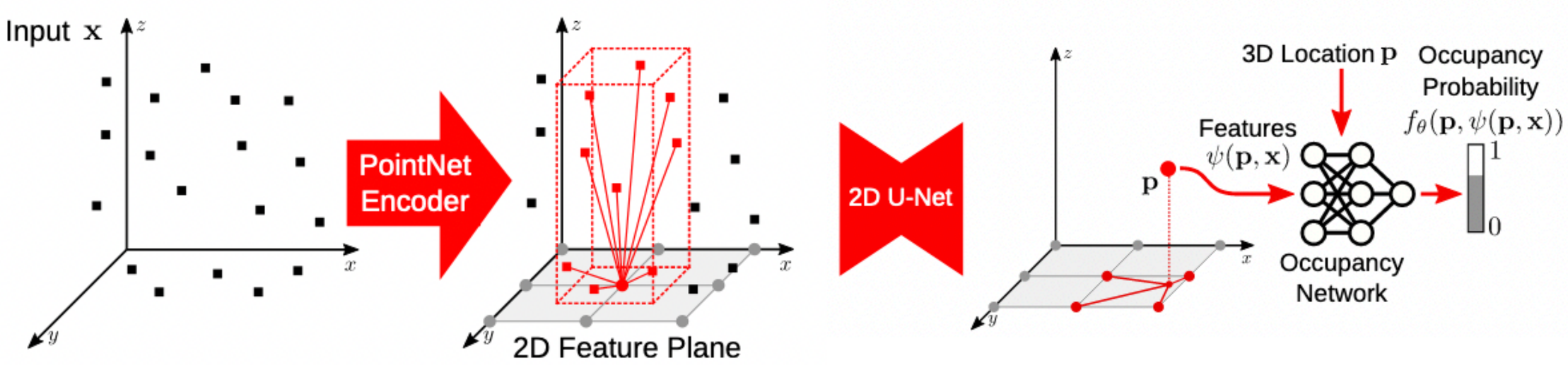
Generalizes to Compositional Scenes!
But: cubic memory complexity :/

From Point clouds: Ground-plan and Tri-plane factorizations



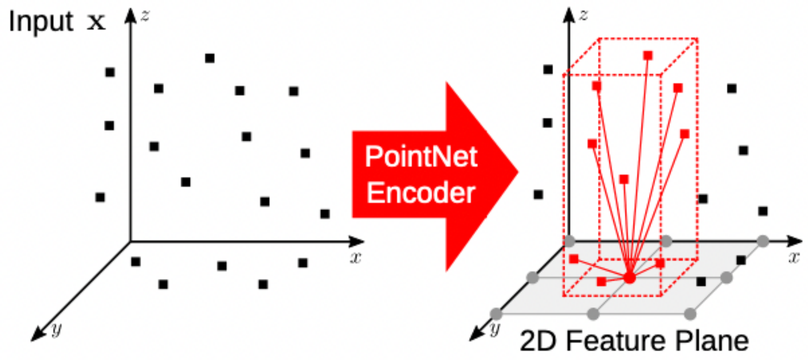
Convolutional Occupancy Networks [Peng et al. 2020]

From Point clouds: Ground-plan and Tri-plane factorizations



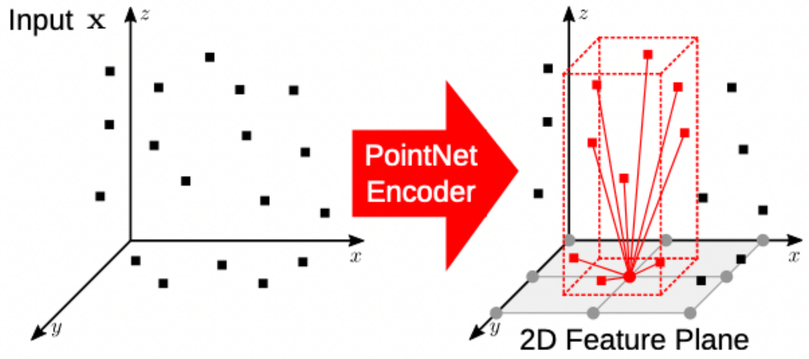
Convolutional Occupancy Networks [Peng et al. 2020]

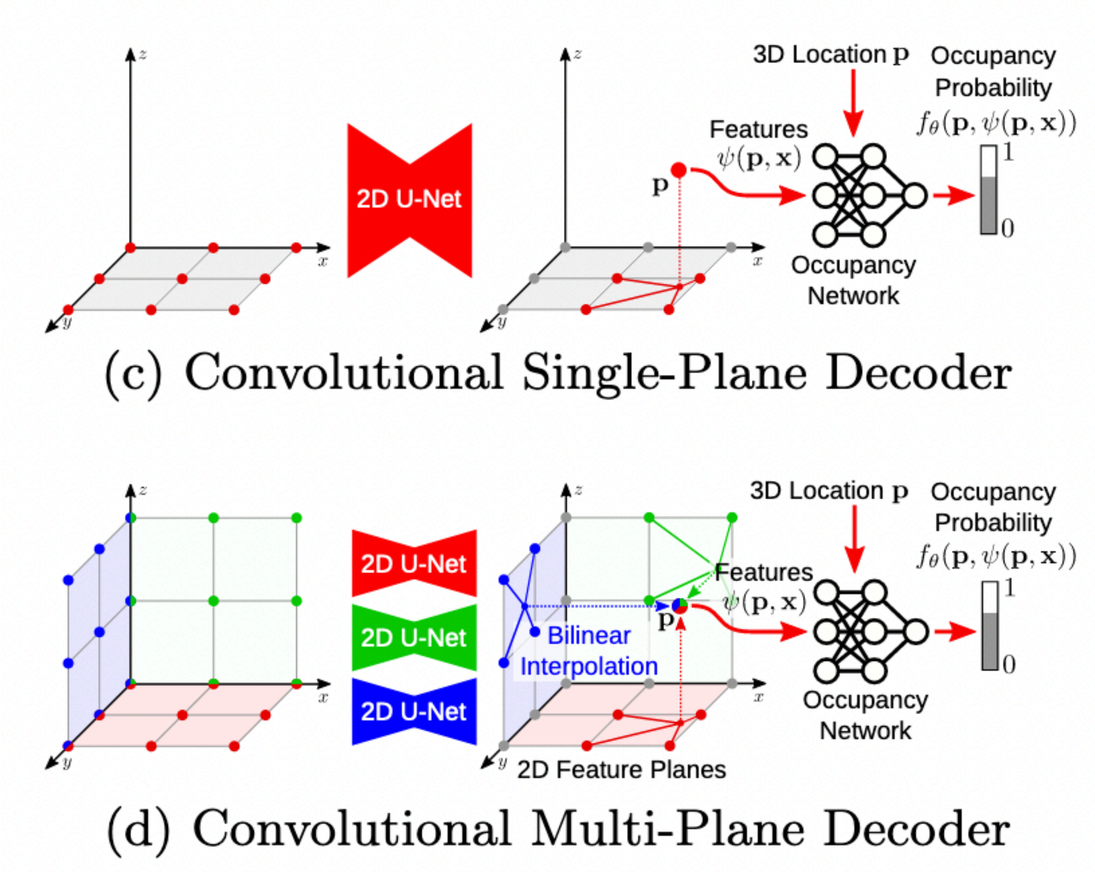
From Point clouds: Ground-plan and Tri-plane factorizations



Convolutional Occupancy Networks [Peng et al. 2020]

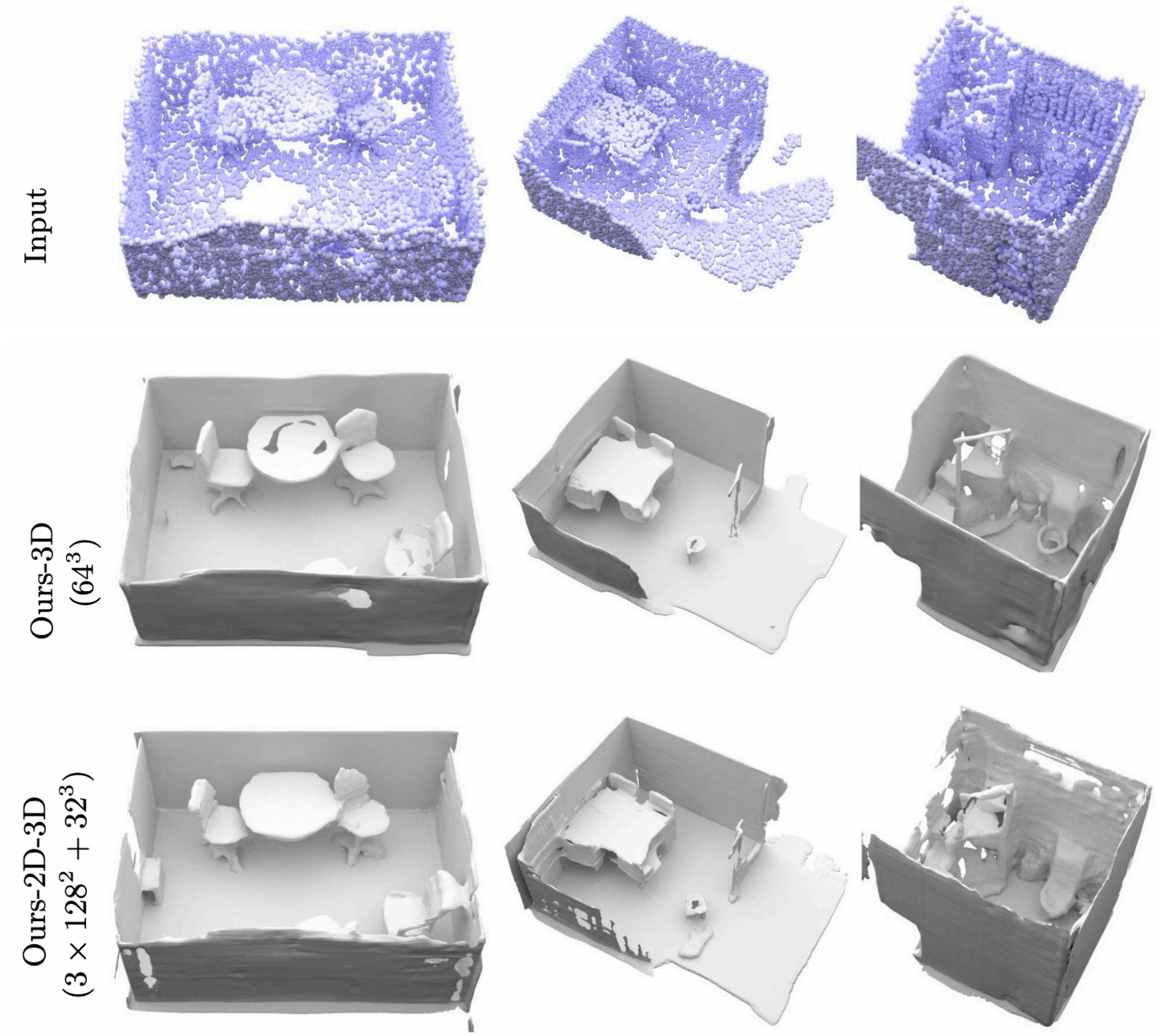
From Point clouds: Ground-plan and Tri-plane factorizations





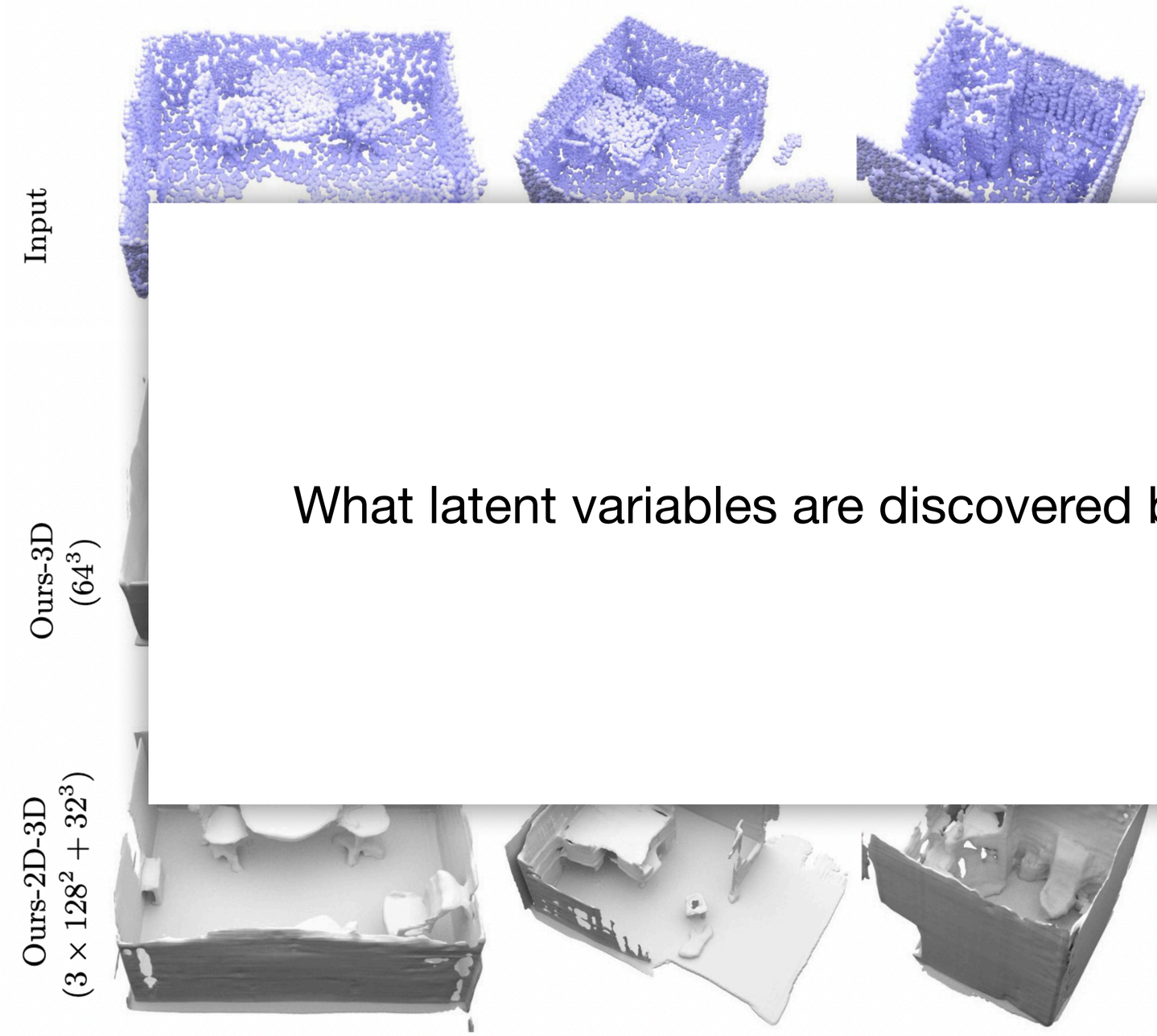
Convolutional Occupancy Networks [Peng et al. 2020]

From point clouds: Conditioning on Reconstructed Voxelgrids



5x less memory!

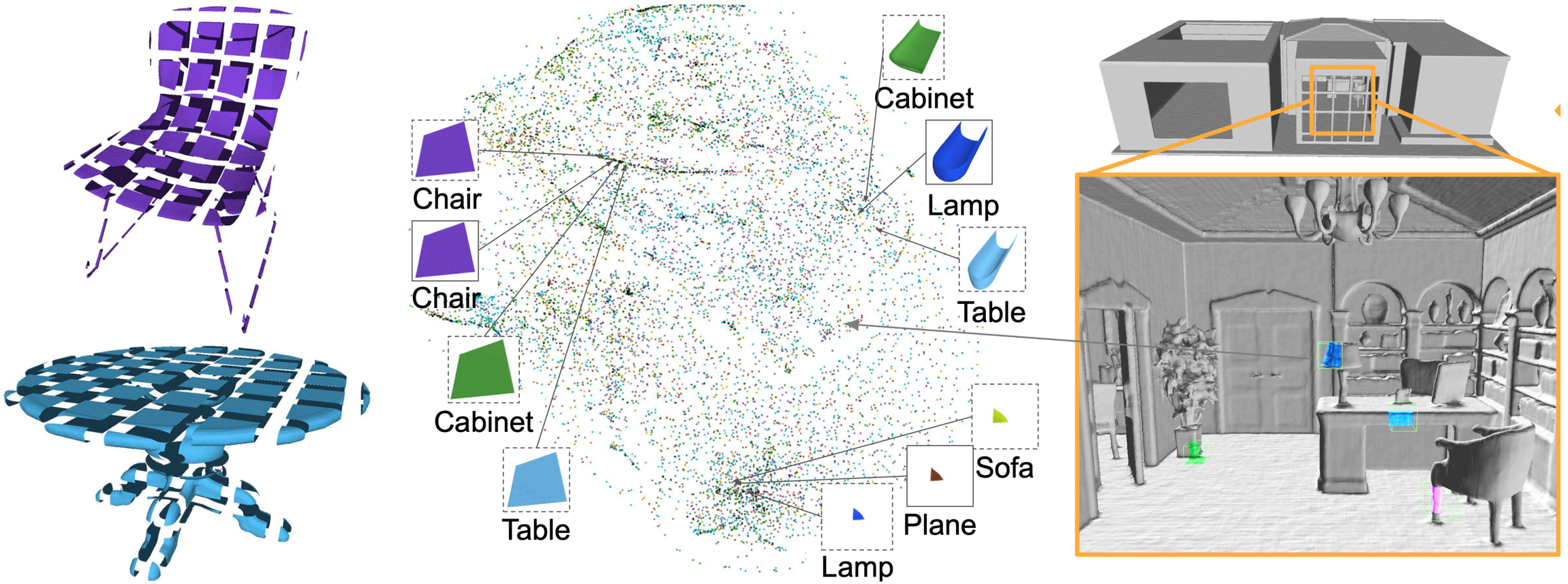
From point clouds: Conditioning on Reconstructed Voxelgrids



What latent variables are discovered by the model?

5x less memory!

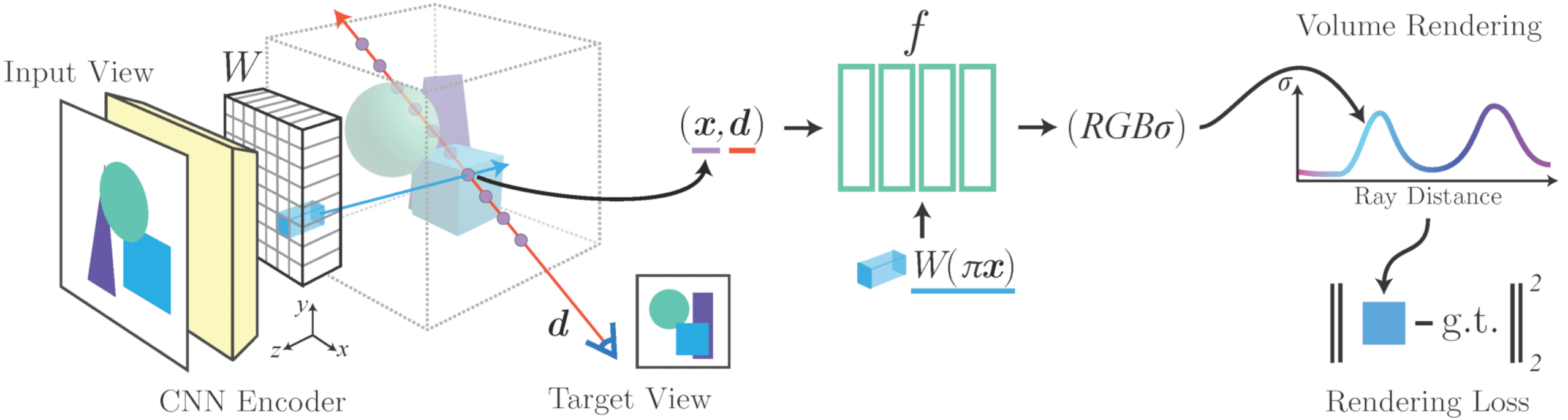
Locally Conditioned Latent Space



Local Implicit Grid Representations for 3D Scenes, Jiang et al. 2020

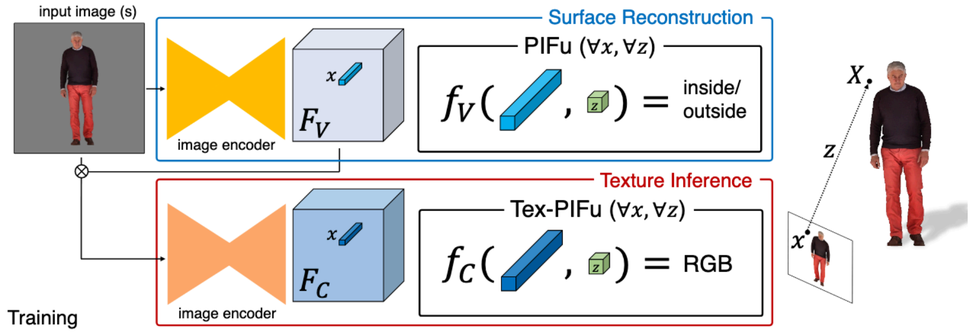
How to locally condition if sensor domain different than field domain?

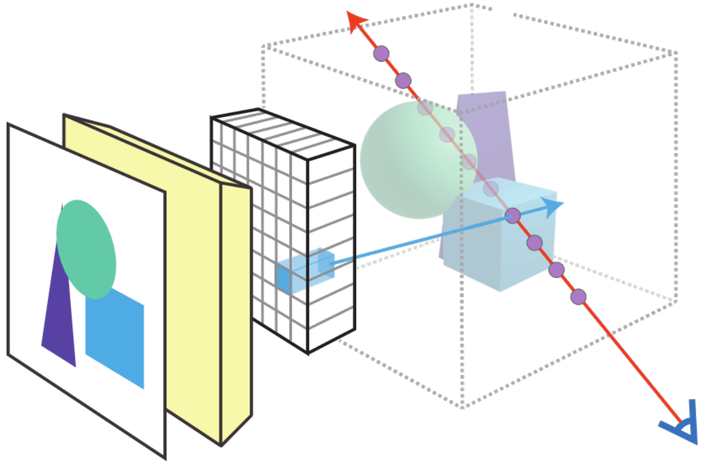
Local Conditioning: Pixel-Aligned Features.



PiFU, Saito et al., ICCV 2019.
PixelNeRF, Yu et al., CVPR 2021
Grf: Learning a general radiance field…, Trevithick et al.

Local Conditioning: Pixel-Aligned Features.

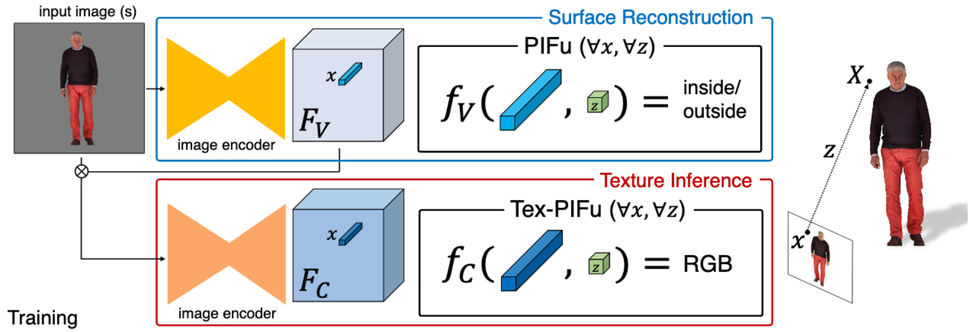


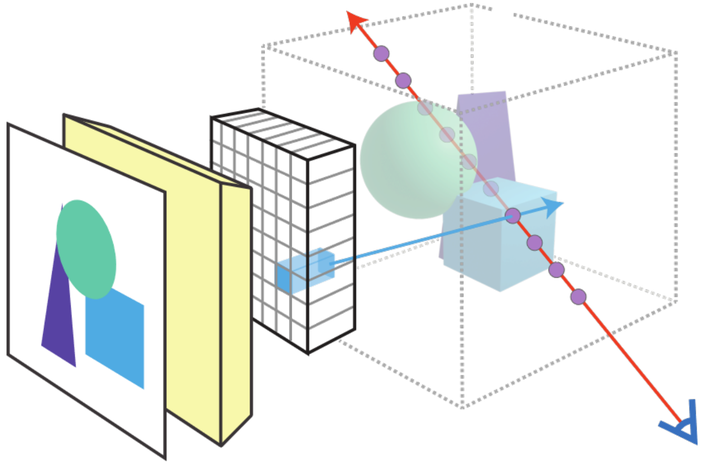


PiFU, Saito et al., ICCV 2019.
PixelNeRF, Yu et al., CVPR 2021
Grf: Learning a general radiance field…, Trevithick et al.

SRNs PixelNeRF

Local Conditioning: Pixel-Aligned Features.

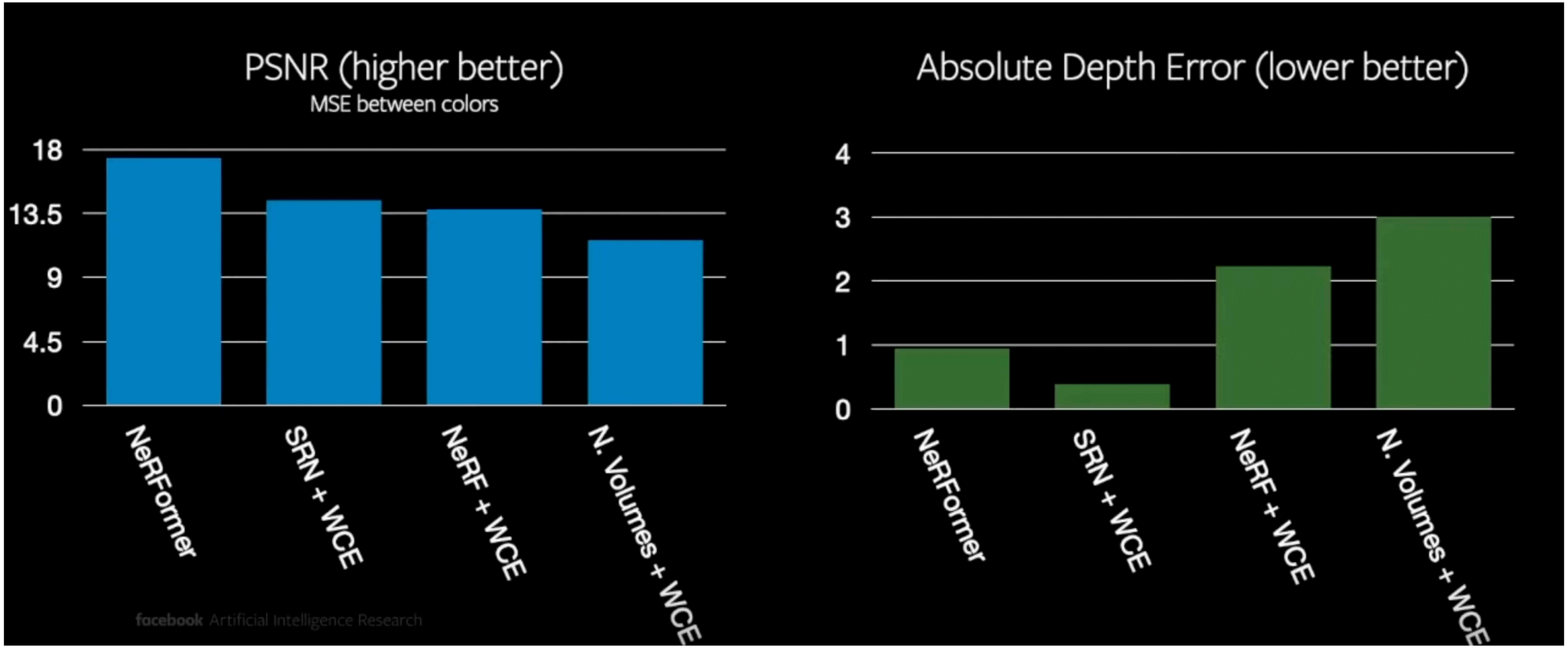




PiFU, Saito et al., ICCV 2019.
PixelNeRF, Yu et al., CVPR 2021
Grf: Learning a general radiance field…, Trevithick et al.

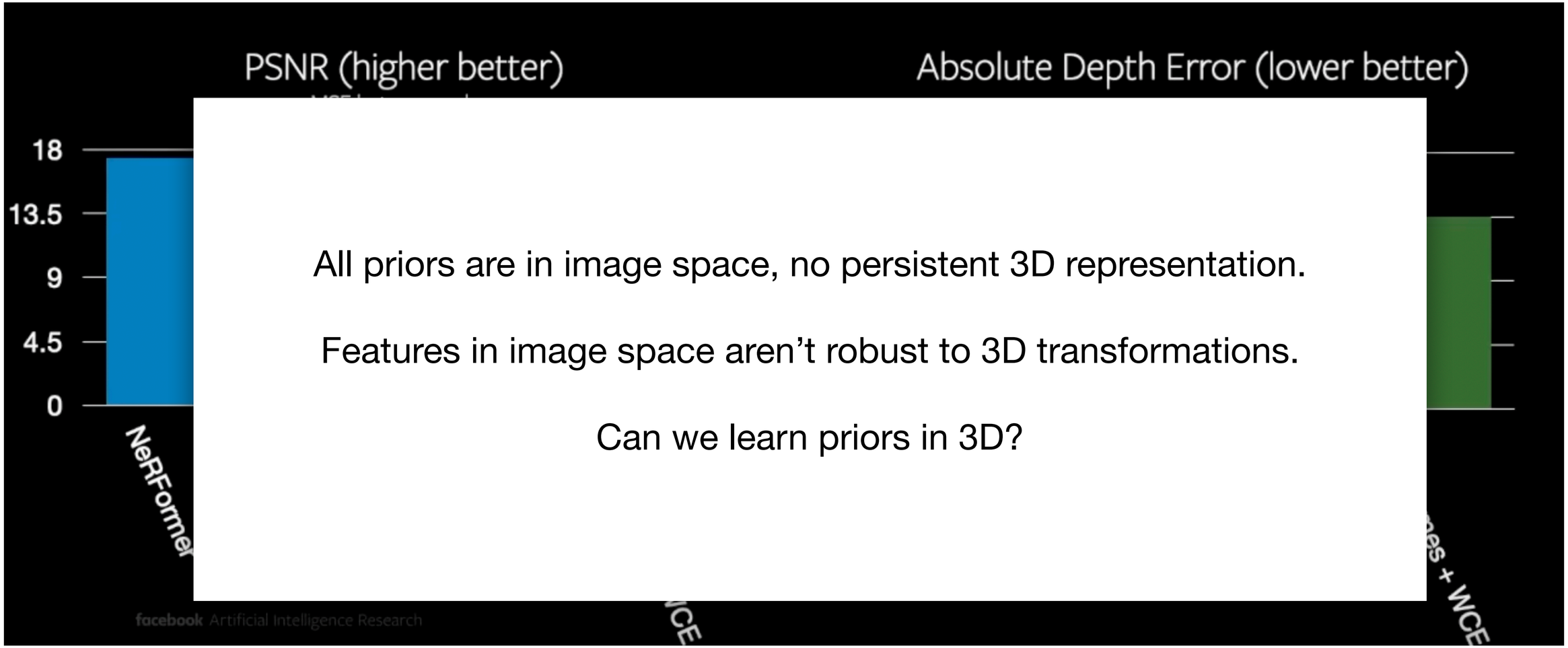
SRNs PixelNeRF

Can combine Scene Representation Networks with pixel-aligned features as well!



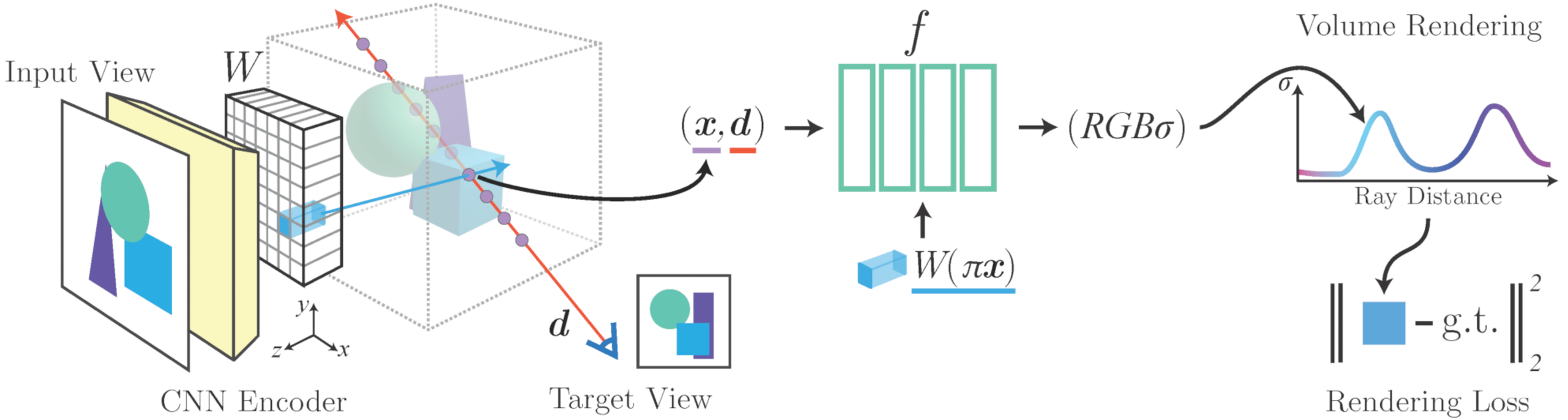
Common Objects in 3D: Large-Scale Learning and Evaluation of Real-life 3D Category Reconstruction

Can combine Scene Representation Networks with pixel-aligned features as well!



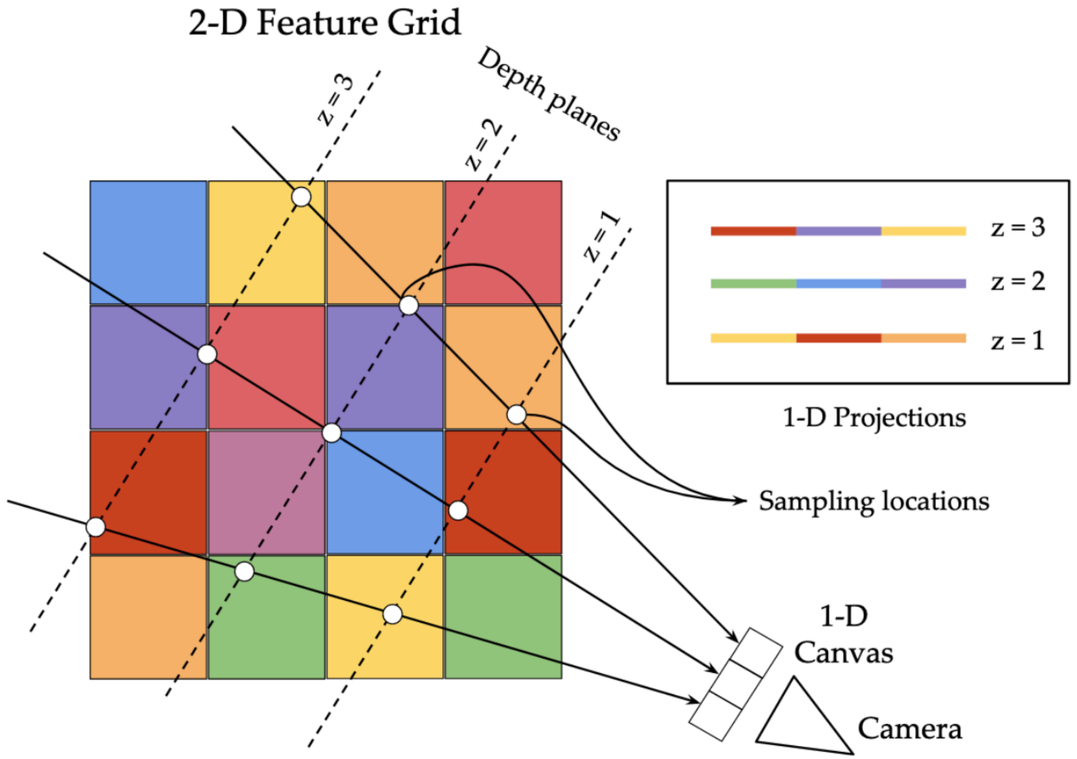
Common Objects in 3D: Large-Scale Learning and Evaluation of Real-life 3D Category Reconstruction

Local Conditioning: Pixel-Aligned Features.



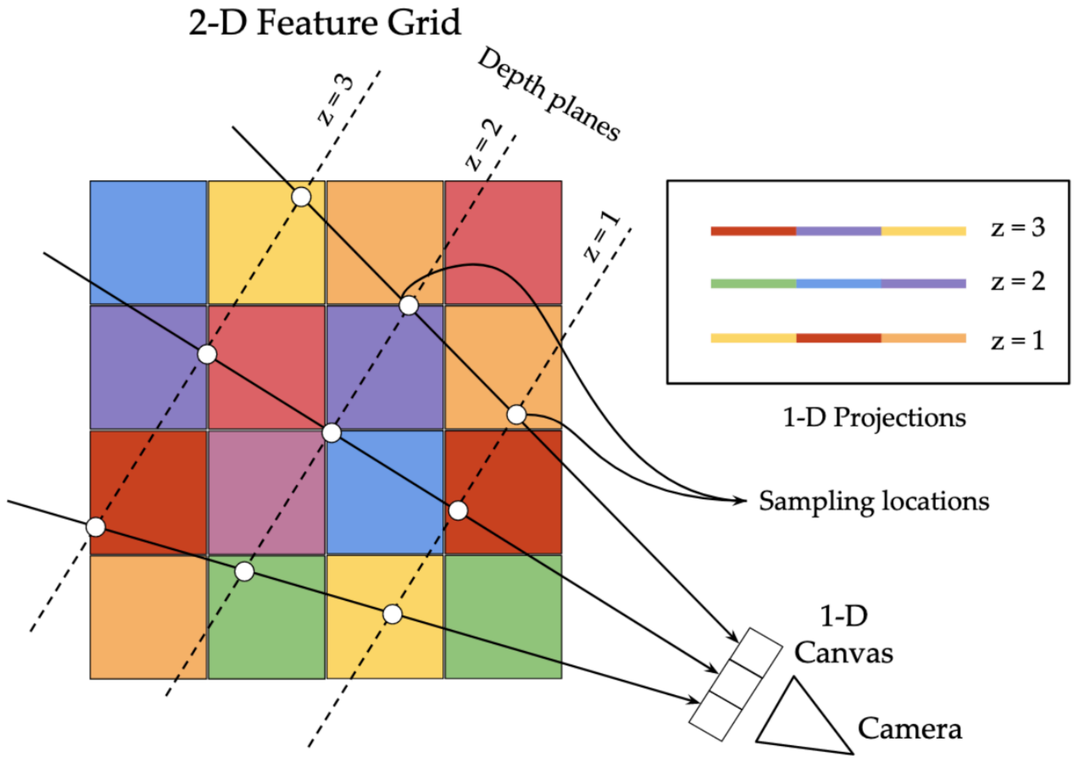
PiFU, Saito et al., ICCV 2019.
PixelNeRF, Yu et al., CVPR 2021
Grf: Learning a general radiance field…, Trevithick et al.

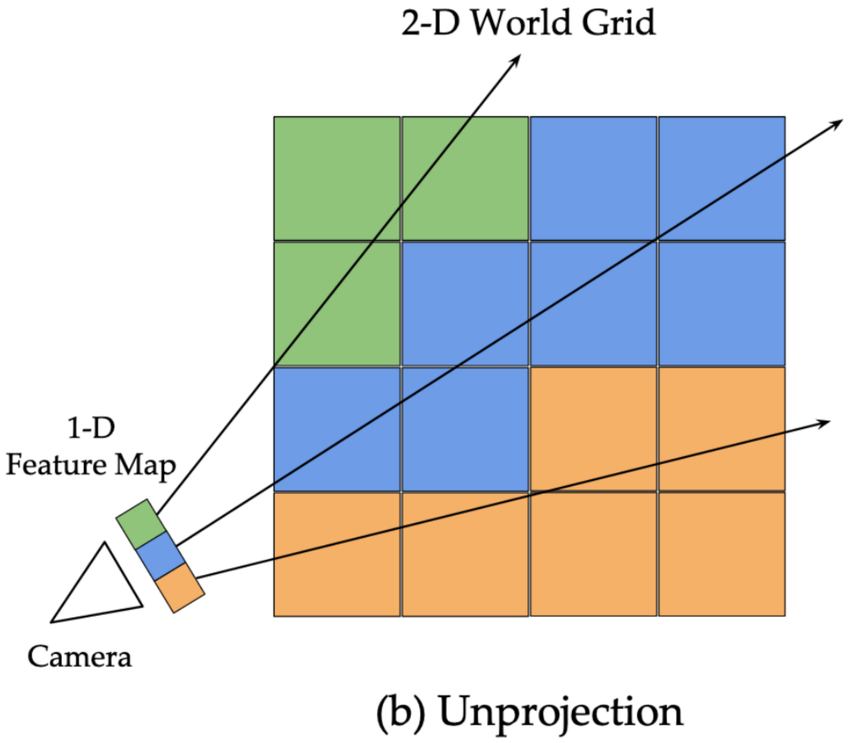
Preliminary Knowledge: “Lifting” or “Unprojecting” Features from a 2D Image Plane to a 3D Volume



Learning a Multi-View Stereo Machine, Kar et al. 2017
DeepVoxels, Sitzmann et al. 2019
Learning Spatial Common Sense with Geometry-Aware Recurrent Networks, Tung et al. 2019

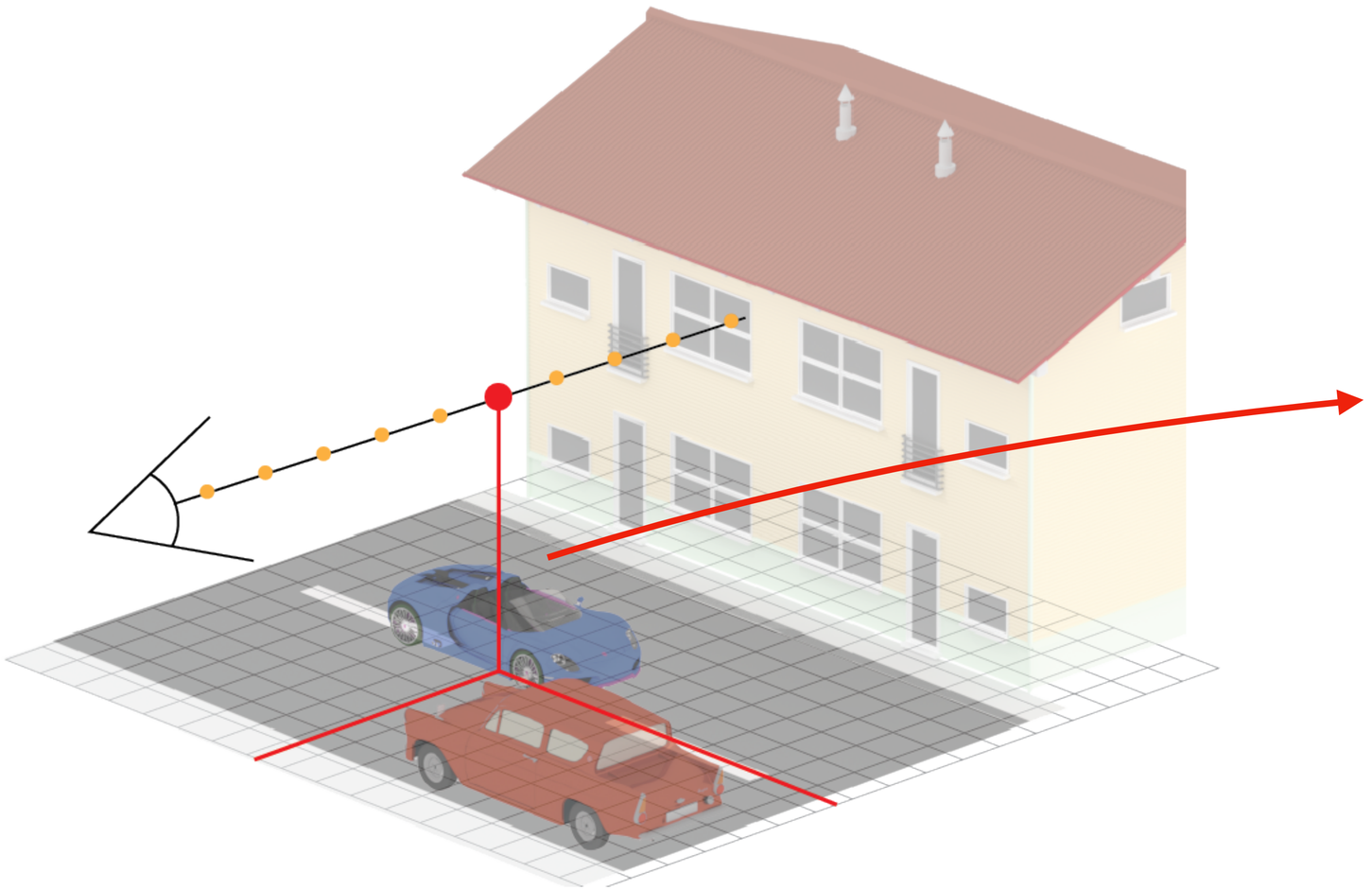
Preliminary Knowledge: “Lifting” or “Unprojecting” Features from a 2D Image Plane to a 3D Volume

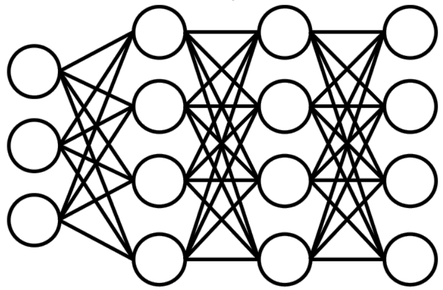




Learning a Multi-View Stereo Machine, Kar et al. 2017
DeepVoxels, Sitzmann et al. 2019
Learning Spatial Common Sense with Geometry-Aware Recurrent Networks, Tung et al. 2019

Conditional Ground Plans for Single-Image 3D Reconstruction



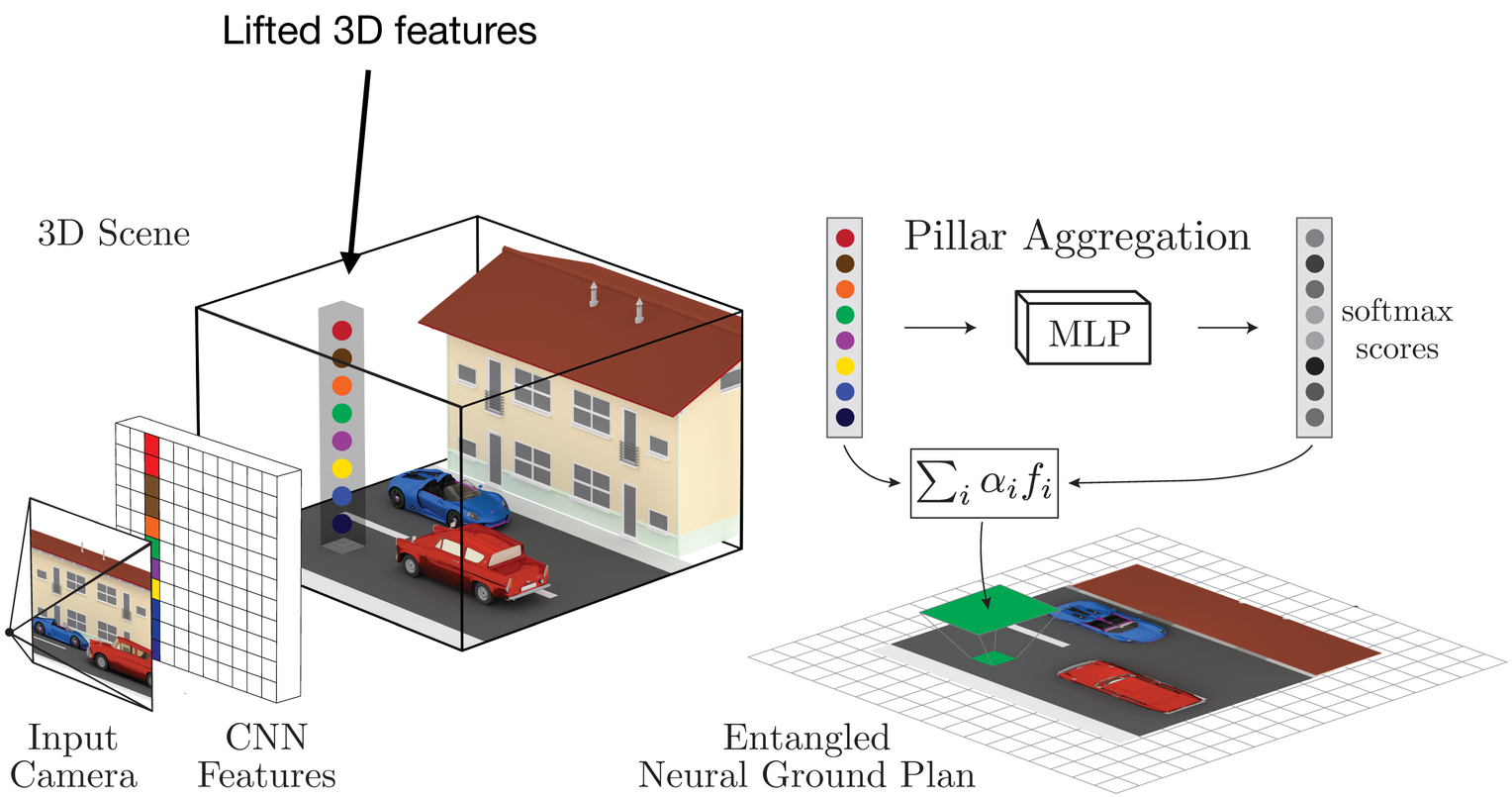


Neural Field

$$(\sigma, C)$$

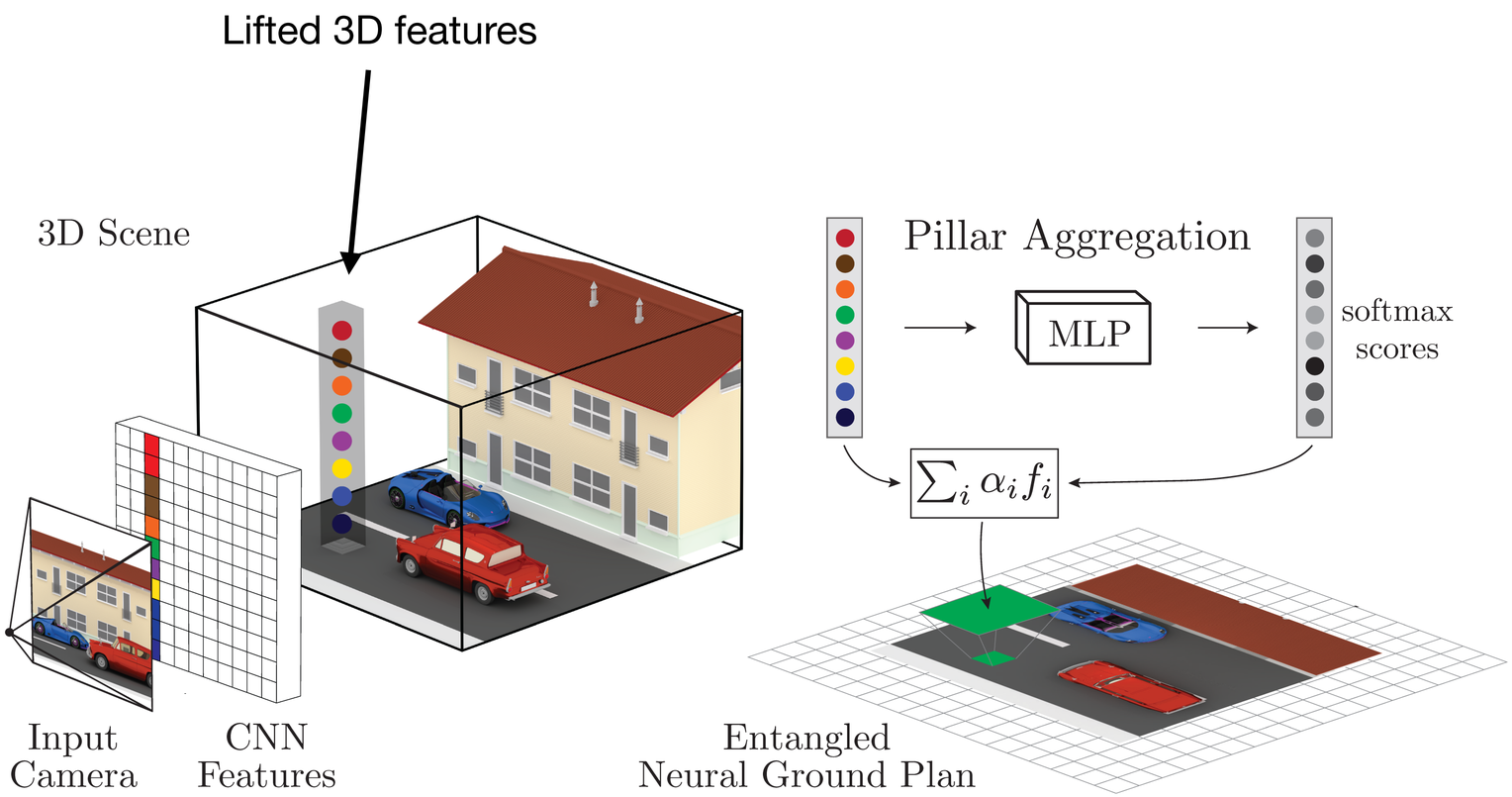
Seeing 3D Objects in a Single Image via Self-Supervised Static-Dynamic Disentanglement, Sharma et al. 2022

Conditional Ground Plans for Single-Image 3D Reconstruction



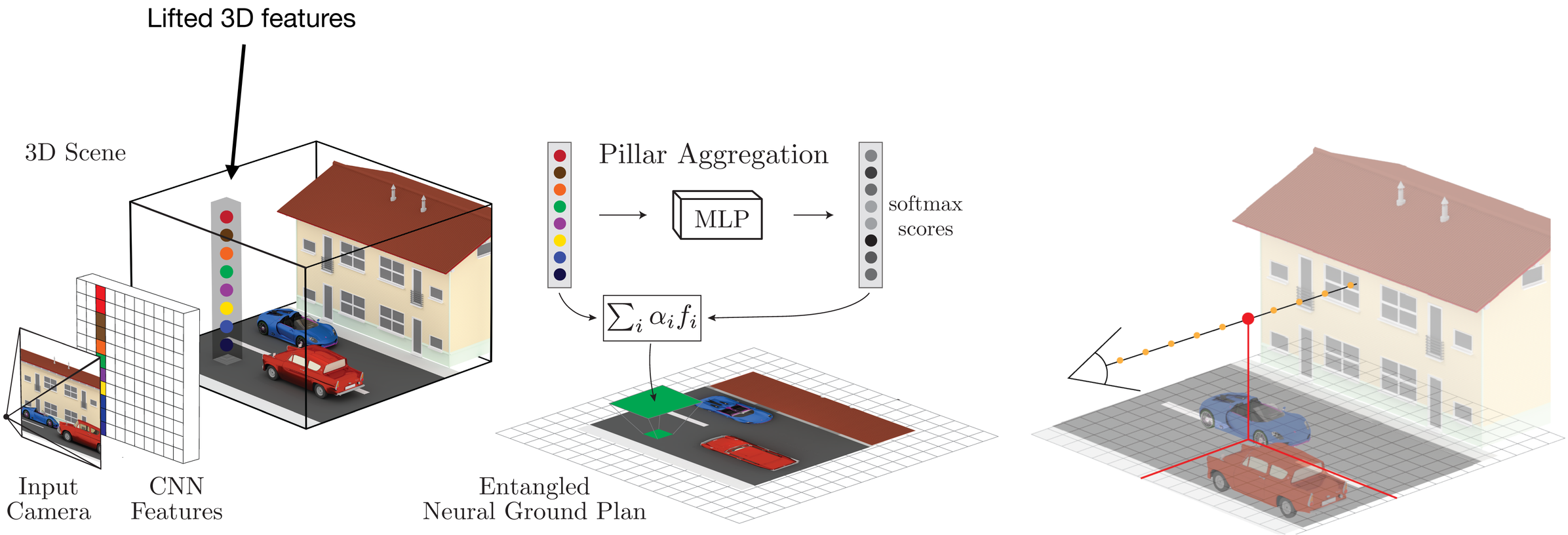
Seeing 3D Objects in a Single Image via Self-Supervised Static-Dynamic Disentanglement, Sharma et al. 2022

Conditional Ground Plans for Single-Image 3D Reconstruction



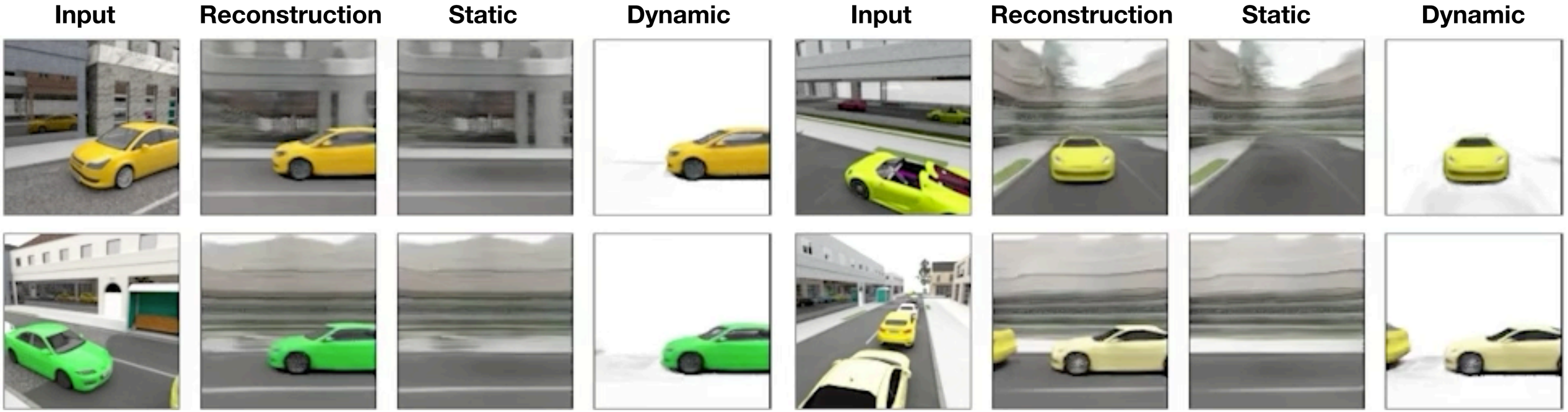
Seeing 3D Objects in a Single Image via Self-Supervised Static-Dynamic Disentanglement, Sharma et al. 2022

Conditional Ground Plans for Single-Image 3D Reconstruction



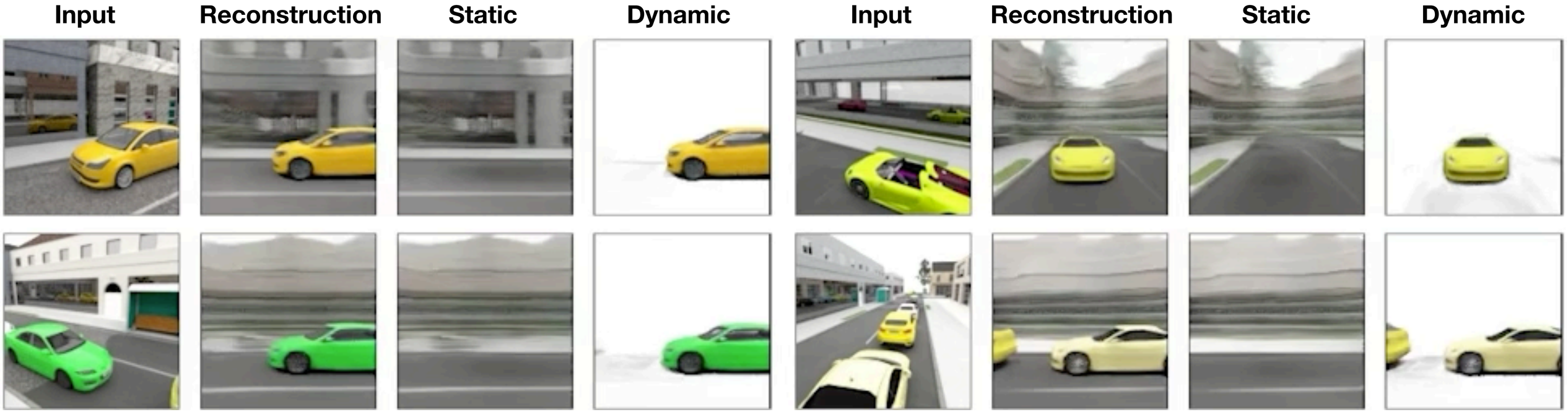
Seeing 3D Objects in a Single Image via Self-Supervised Static-Dynamic Disentanglement, Sharma et al. 2022

Static-dynamic disentanglement from a **single** image!



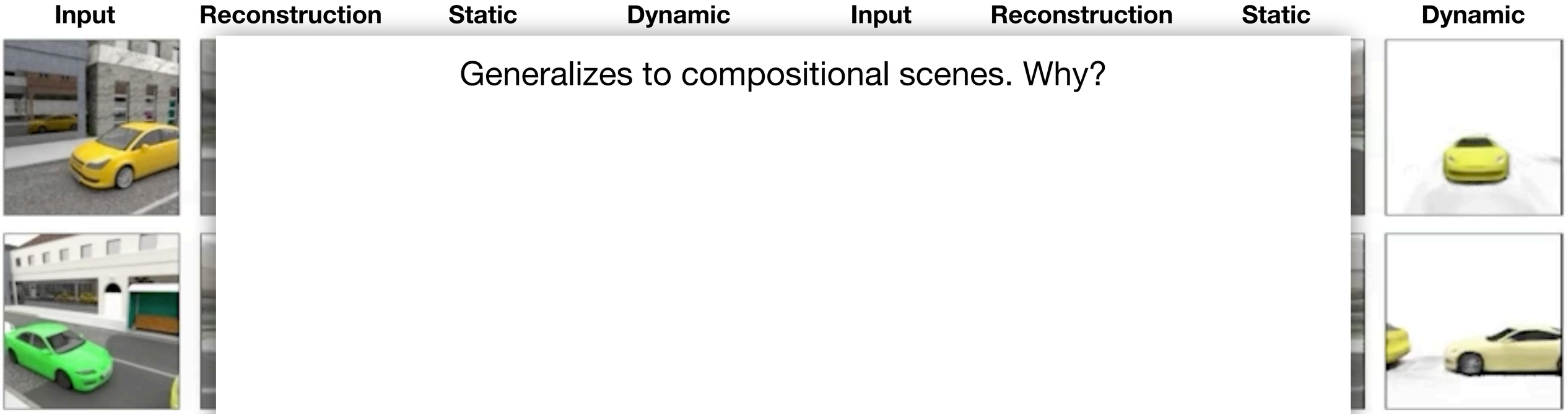
Seeing 3D Objects in a Single Image via Self-Supervised Static-Dynamic Disentanglement, Sharma et al. 2022

Static-dynamic disentanglement from a **single** image!



Seeing 3D Objects in a Single Image via Self-Supervised Static-Dynamic Disentanglement, Sharma et al. 2022

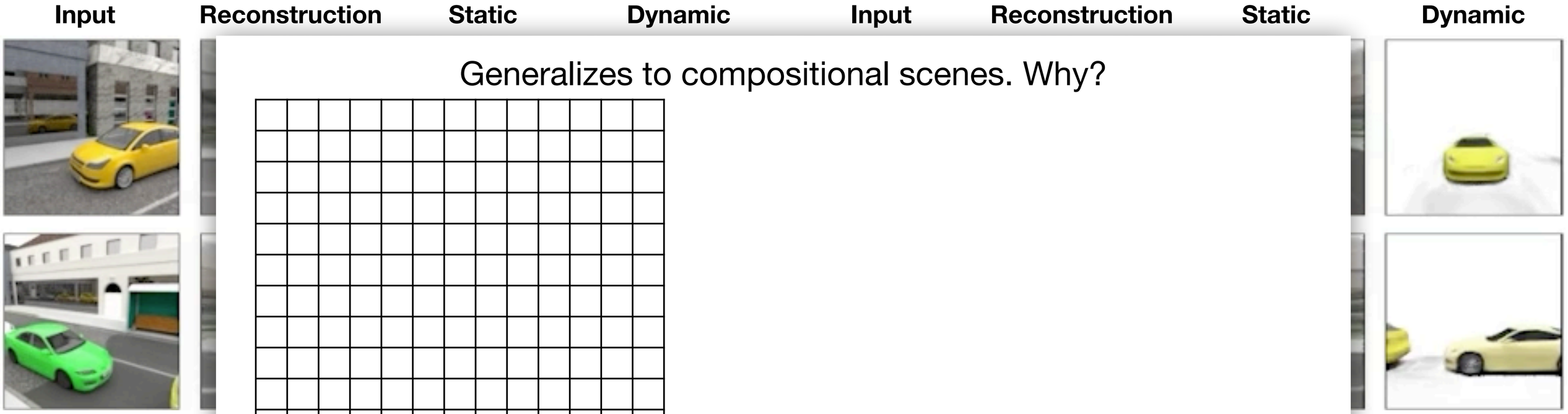
Static-dynamic disentanglement from a **single** image!



Generalizes to compositional scenes. Why?

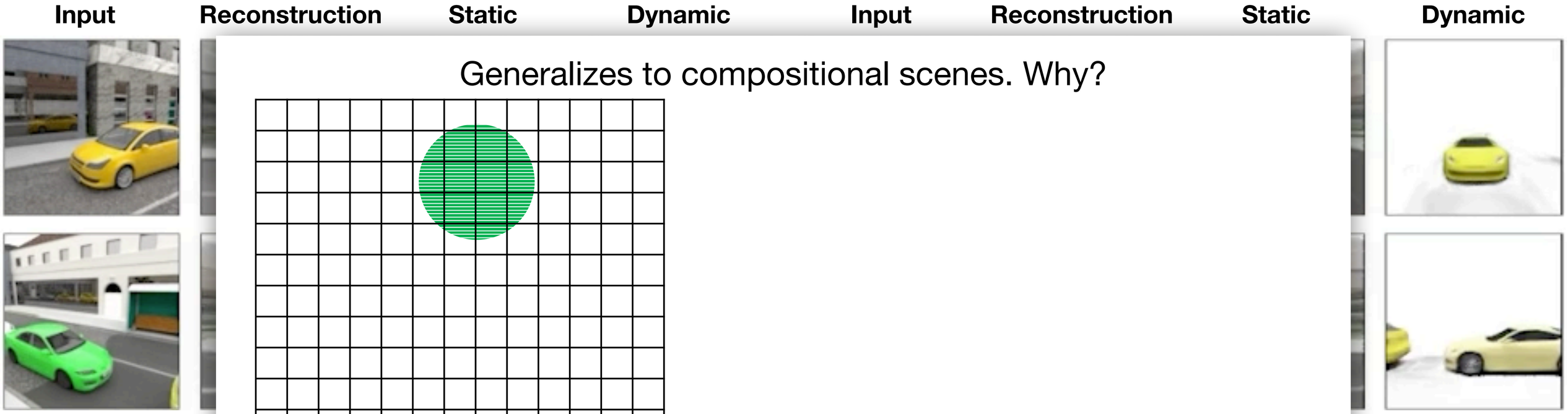
Seeing 3D Objects in a Single Image via Self-Supervised Static-Dynamic Disentanglement, Sharma et al. 2022

Static-dynamic disentanglement from a **single** image!



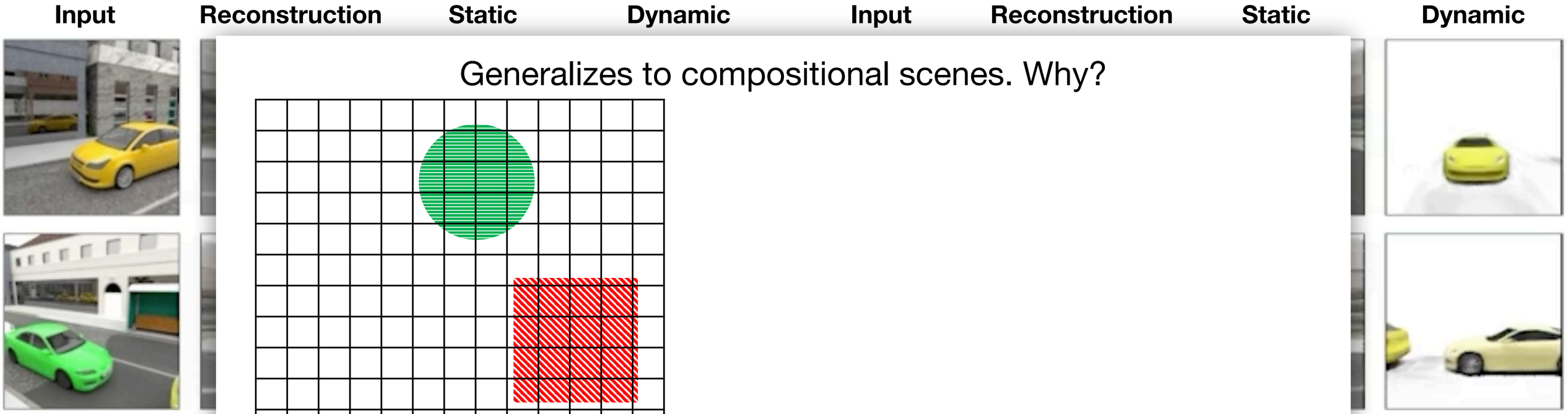
Seeing 3D Objects in a Single Image via Self-Supervised Static-Dynamic Disentanglement, Sharma et al. 2022

Static-dynamic disentanglement from a **single** image!



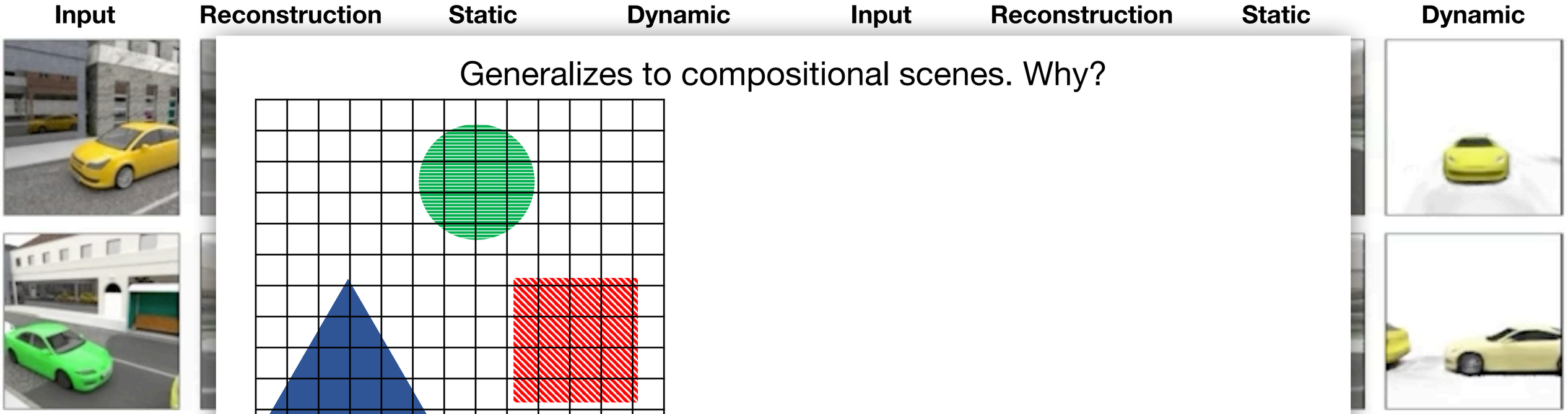
Seeing 3D Objects in a Single Image via Self-Supervised Static-Dynamic Disentanglement, Sharma et al. 2022

Static-dynamic disentanglement from a **single** image!



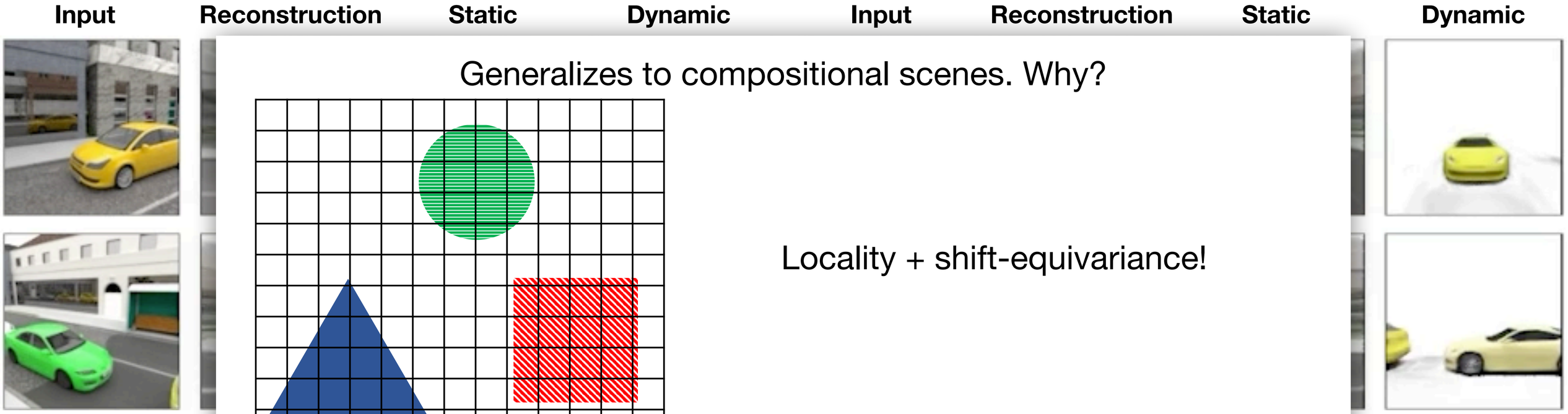
Seeing 3D Objects in a Single Image via Self-Supervised Static-Dynamic Disentanglement, Sharma et al. 2022

Static-dynamic disentanglement from a **single** image!



Seeing 3D Objects in a Single Image via Self-Supervised Static-Dynamic Disentanglement, Sharma et al. 2022

Static-dynamic disentanglement from a **single** image!



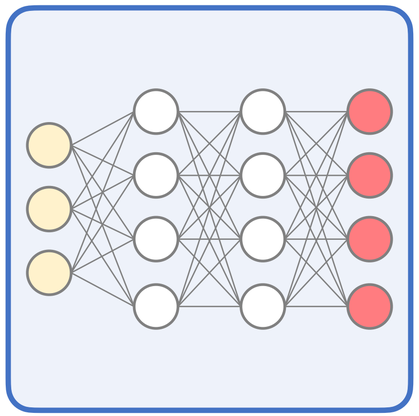
Seeing 3D Objects in a Single Image via Self-Supervised Static-Dynamic Disentanglement, Sharma et al. 2022

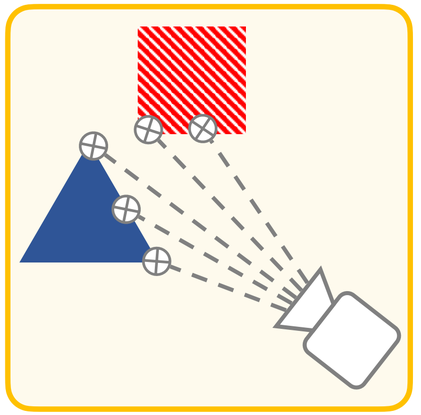
Today: *Learned* inference algorithms!

Observations

Inference

?

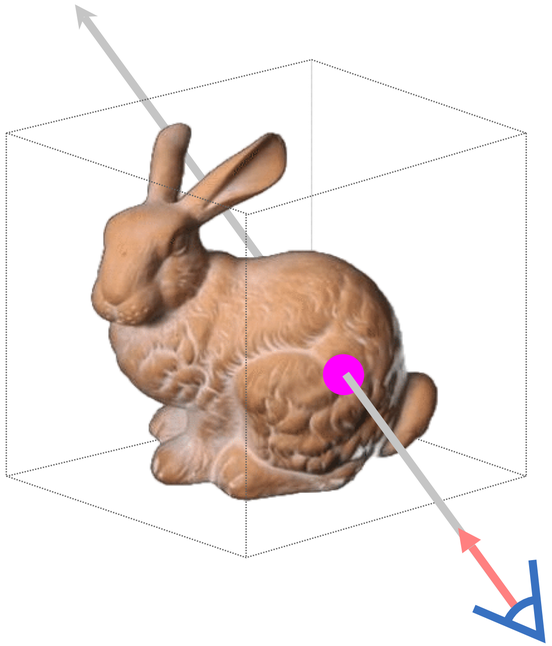


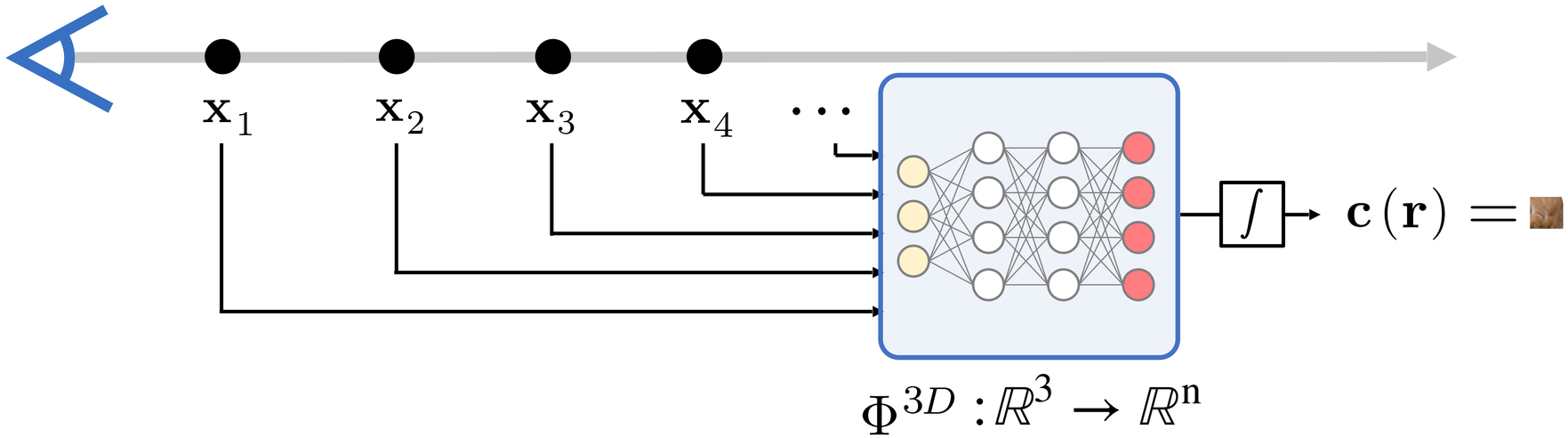


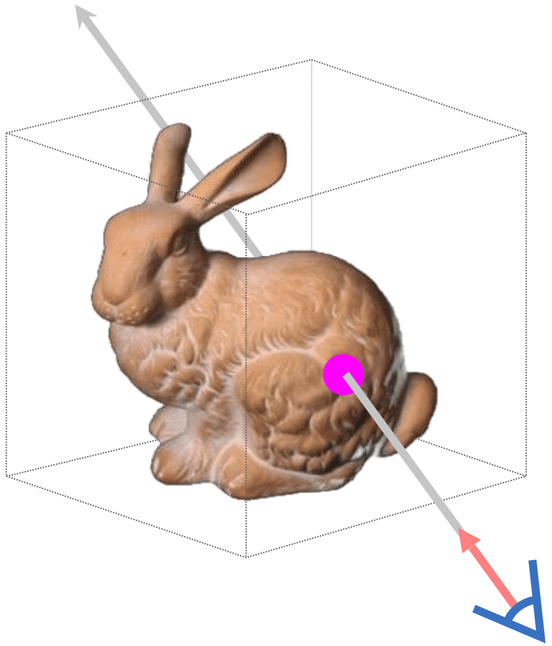
Renderings

Why?	**We humans can reconstruct 3D from incomplete observations, by using knowledge that we have learned about the world. Deep learning is the best way we know to date to learn such priors from data.**
What you'll learn.	**How to express priors over 3D scenes using deep learning, different ways of doing inference (encoding, auto-decoding)**

General structure of Neural Renderers for 3D Fields

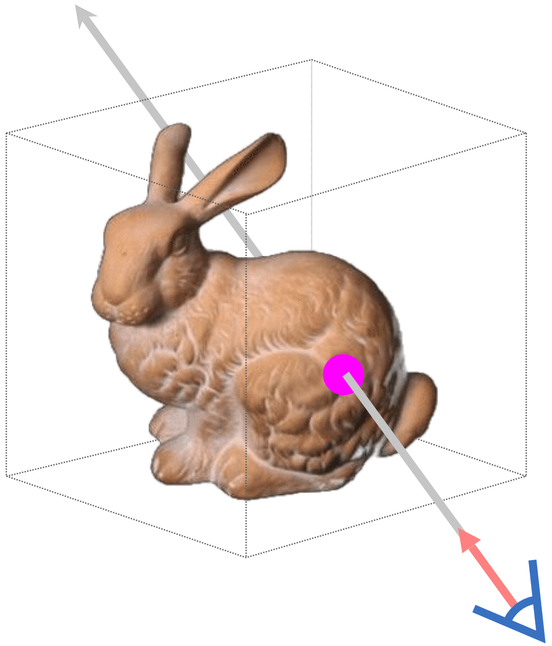






Light Field

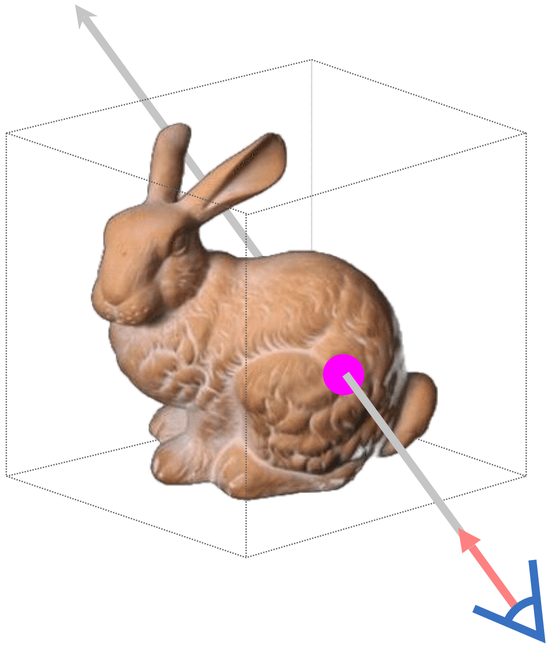
[Adelson et al. 1991, Levoy et al. 1996, Gortler et al. 1996]

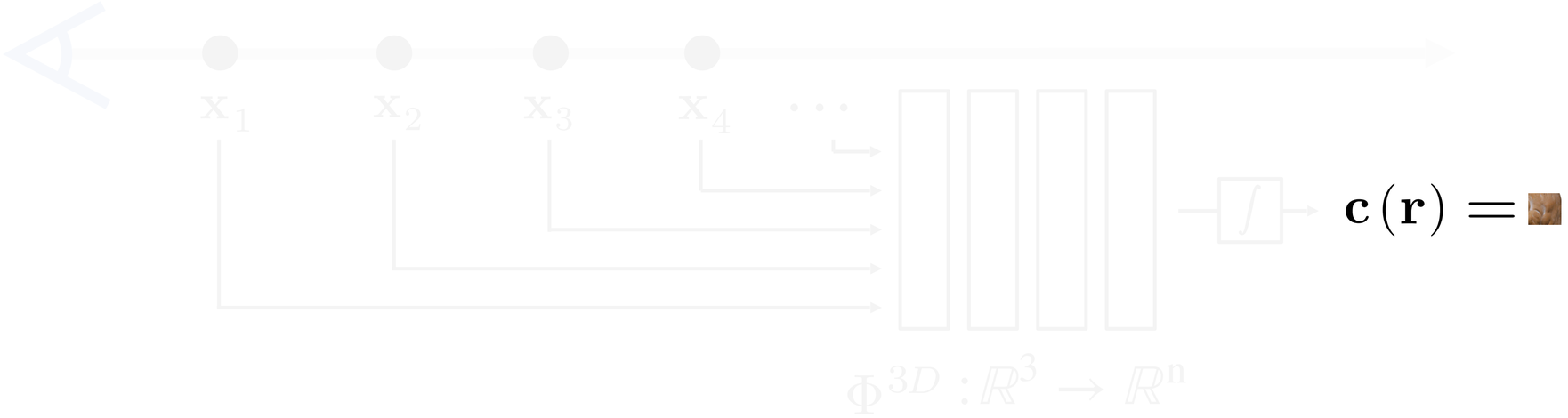


Light Field Networks

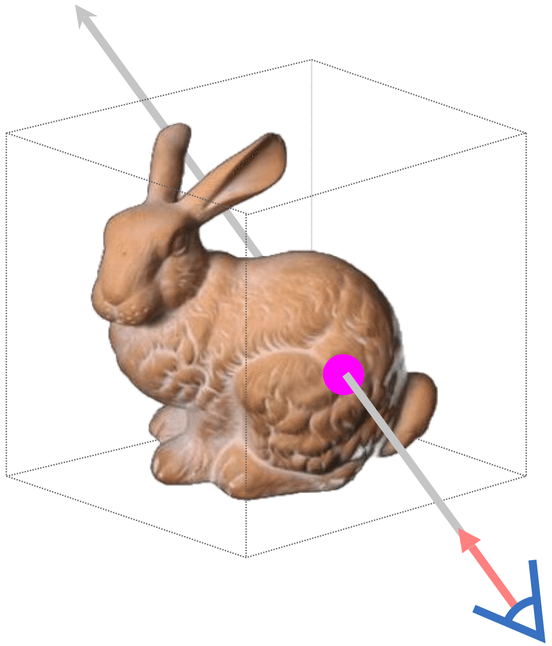
$\mathbf{c}(\mathbf{r}) =$

Light Field Networks





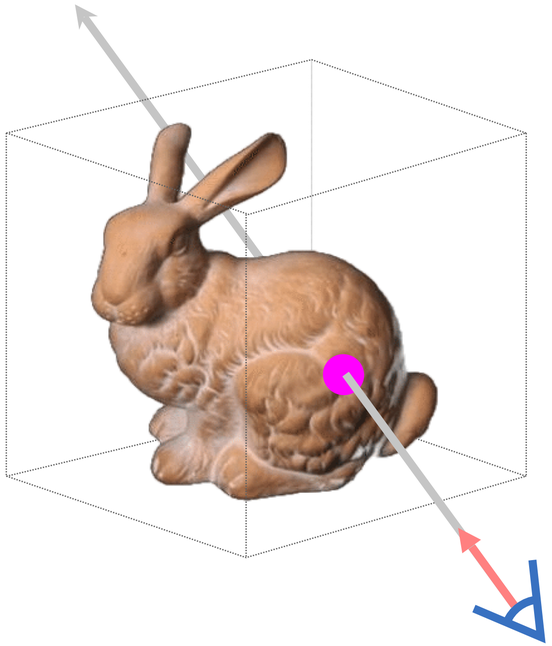
Light Field Networks



$$\mathbf{c}(\mathbf{r}) =$$

An Alternative Scene Representation

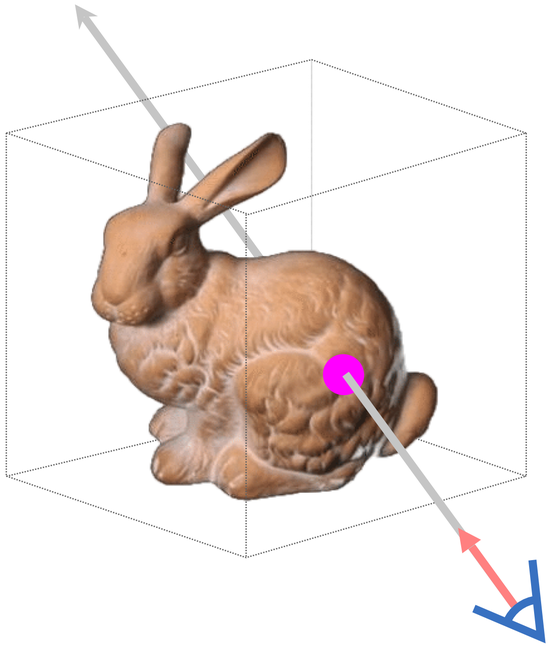
Light Field Networks

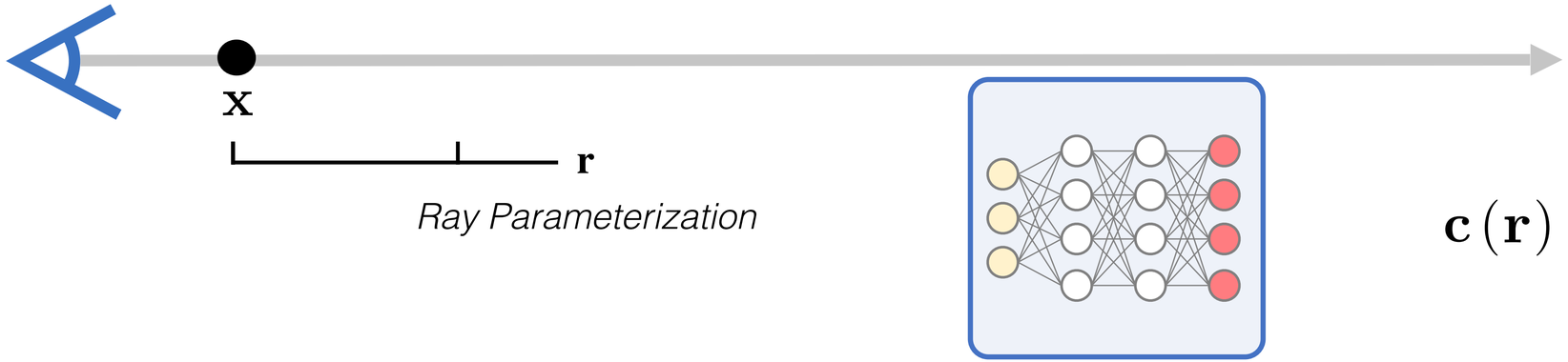


$\mathbf{c}(\mathbf{r}) =$ 

An Alternative Scene Representation

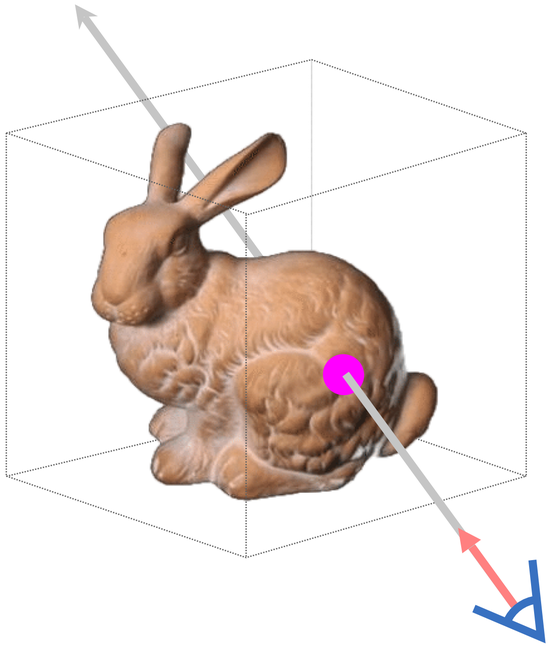
Light Field Networks

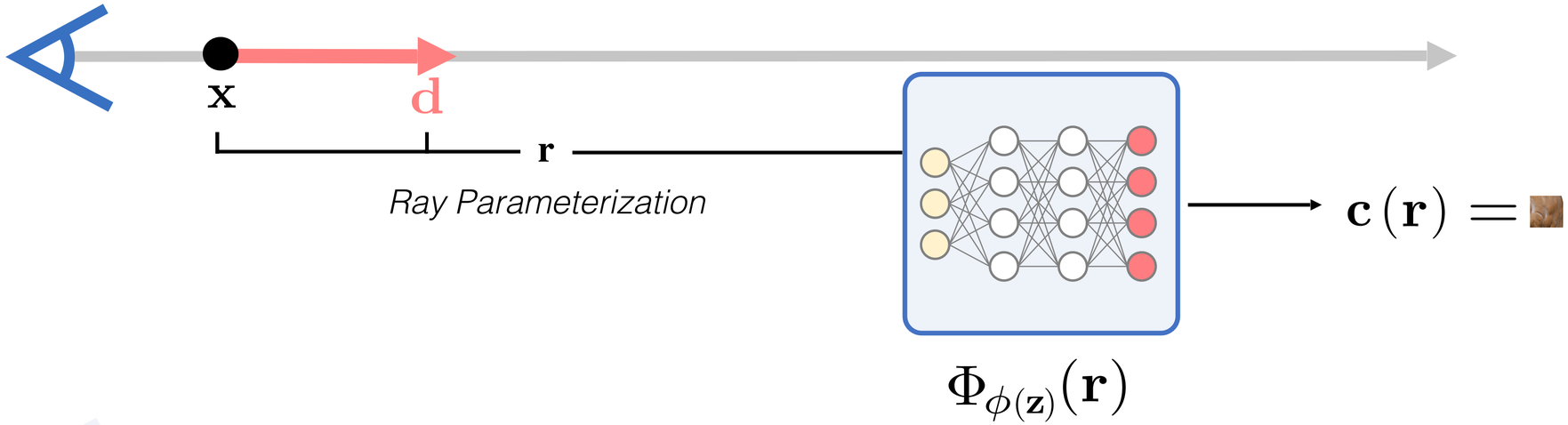




An Alternative Scene Representation

Light Field Networks





An Alternative Scene Representation

Overfitting doesn’t work out-of-the-box: No built-in multi-view consistency!

Context Views

Overfitting doesn’t work out-of-the-box: No built-in multi-view consistency!

Context Views

Overfitting doesn’t work out-of-the-box: No built-in multi-view consistency!

Context Views

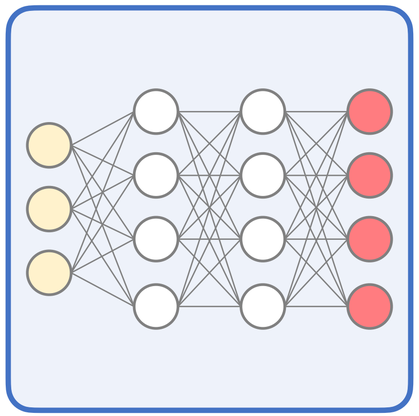
Intermediate Views

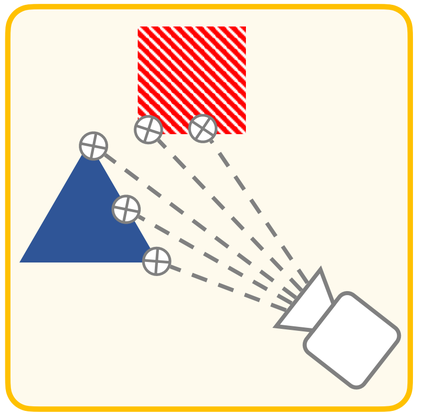
Today: *Learned* inference algorithms!

Observations

Inference

?

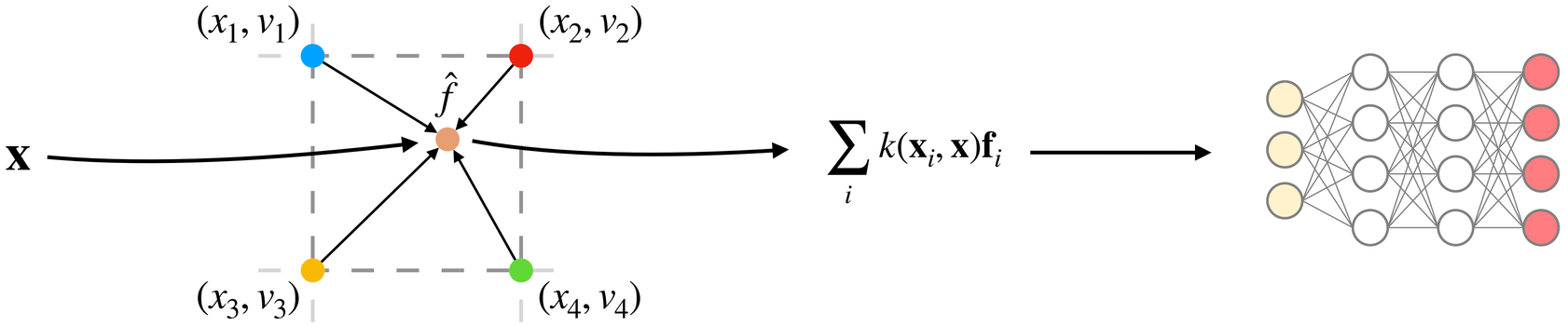


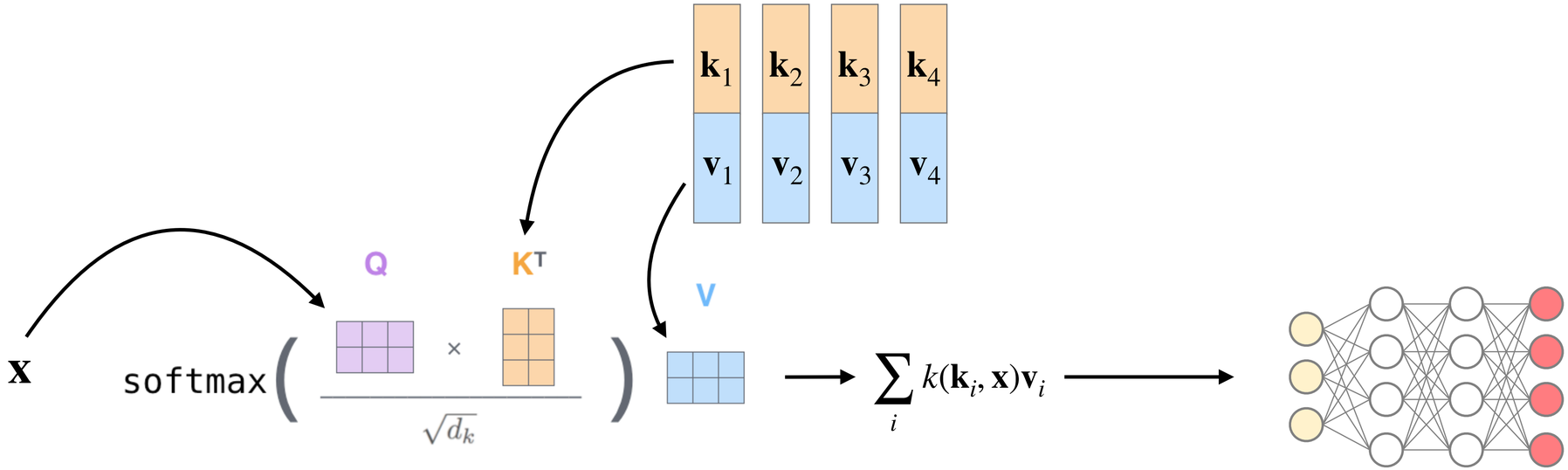


Renderings

Why?	**We humans can reconstruct 3D from incomplete observations, by using knowledge that we have learned about the world. Deep learning is the best way we know to date to learn such priors from data.**
What you'll learn.	**How to express priors over 3D scenes using deep learning, different ways of doing inference (encoding, auto-decoding)**

Next time: Advanced Inference Topics





Alammar, J (2018). The Illustrated Transformer [Blog post]. Retrieved from https://jalammar.github.io/illustrated-transformer/