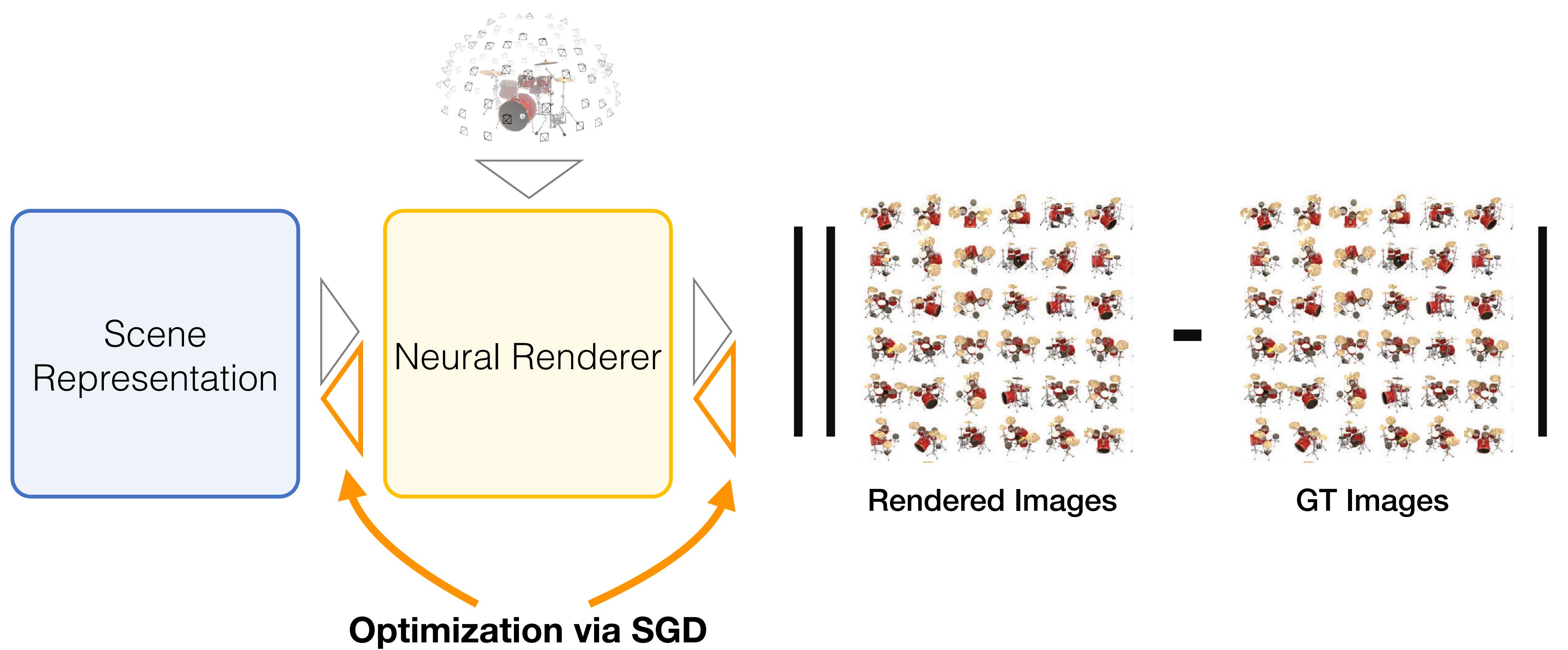


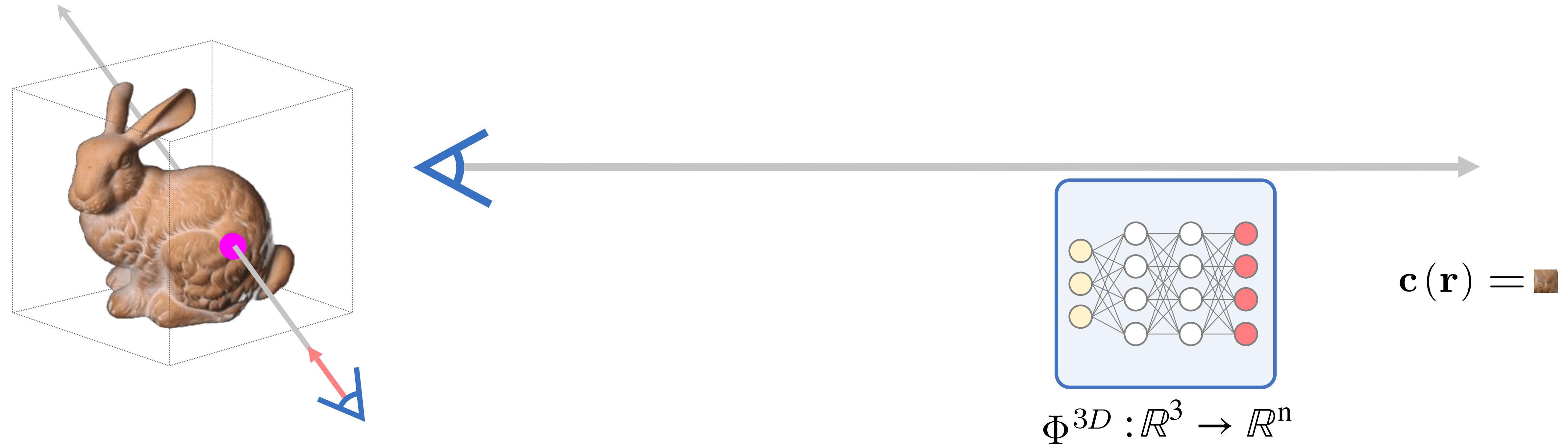
Neural Networks as **PRIOR-BASED** **INFERENCE ALGORITHMS**

6.S980 | 2022-05-19

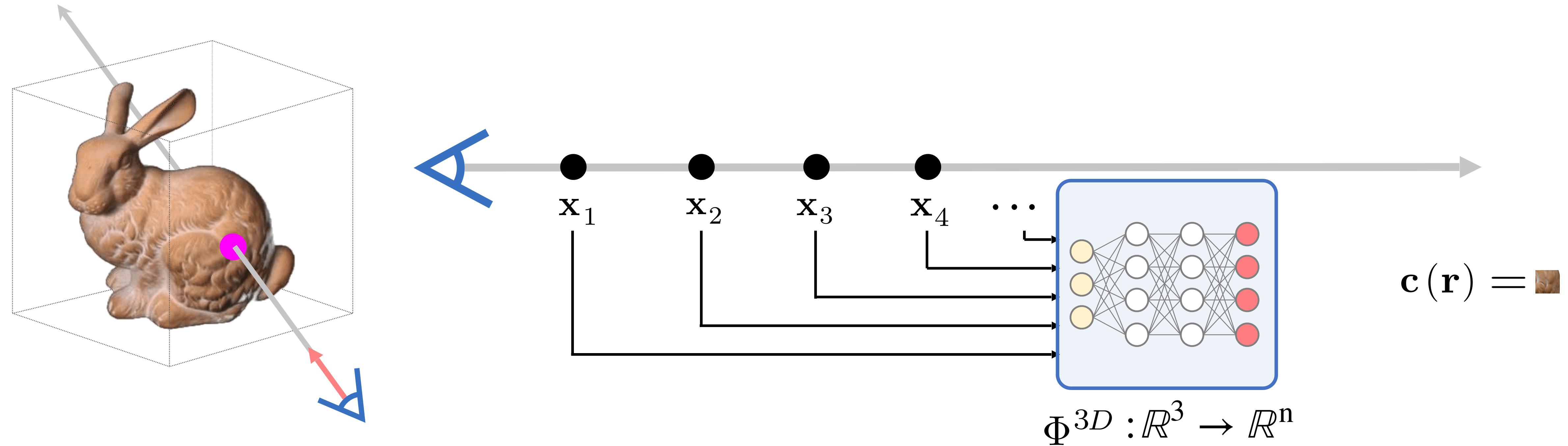
Recap: Differentiable Rendering



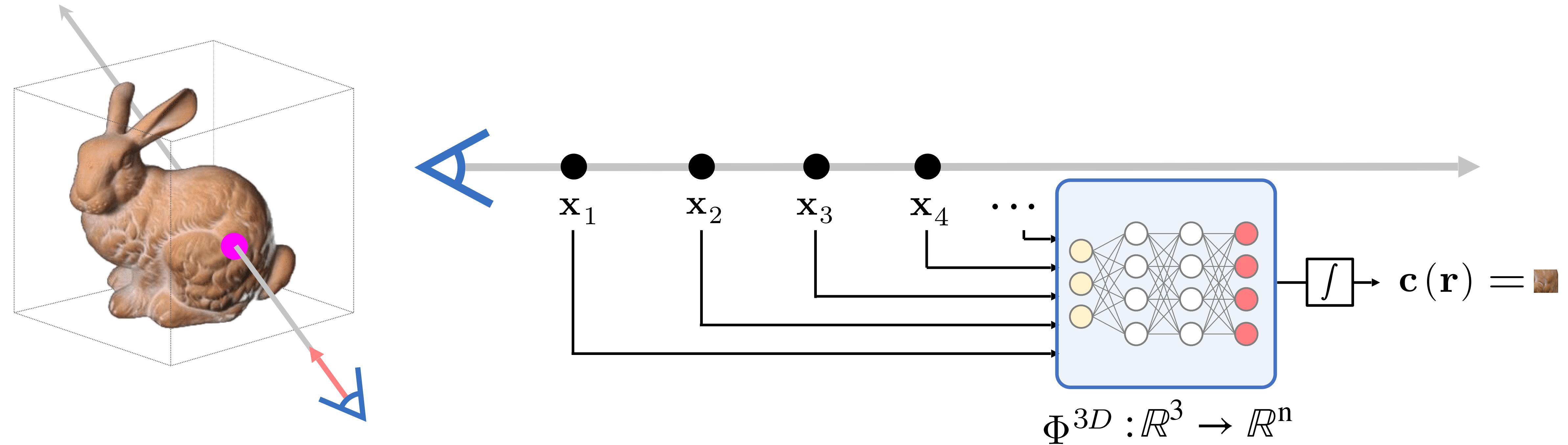
General structure of Neural Renderers for 3D Fields



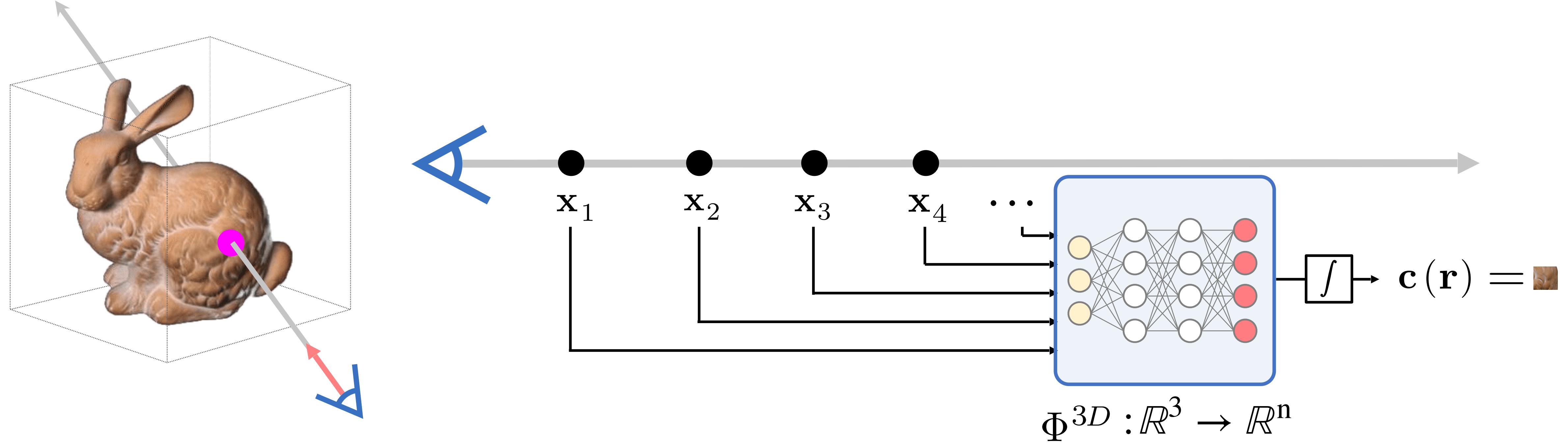
General structure of Neural Renderers for 3D Fields



General structure of Neural Renderers for 3D Fields



General structure of Neural Renderers for 3D Fields



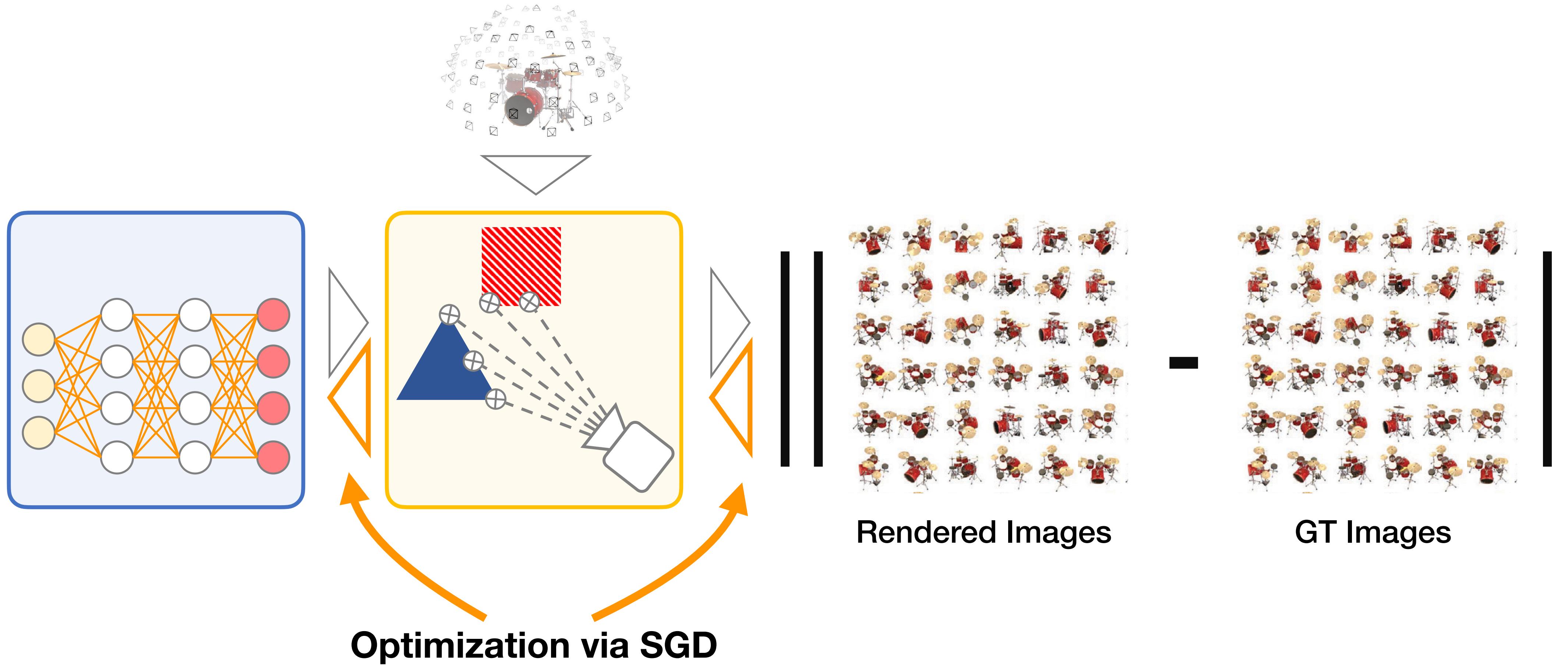
Sphere-Tracing
[JC Hart, 1996]

Volumetric

Hybrid implicit-volumetric

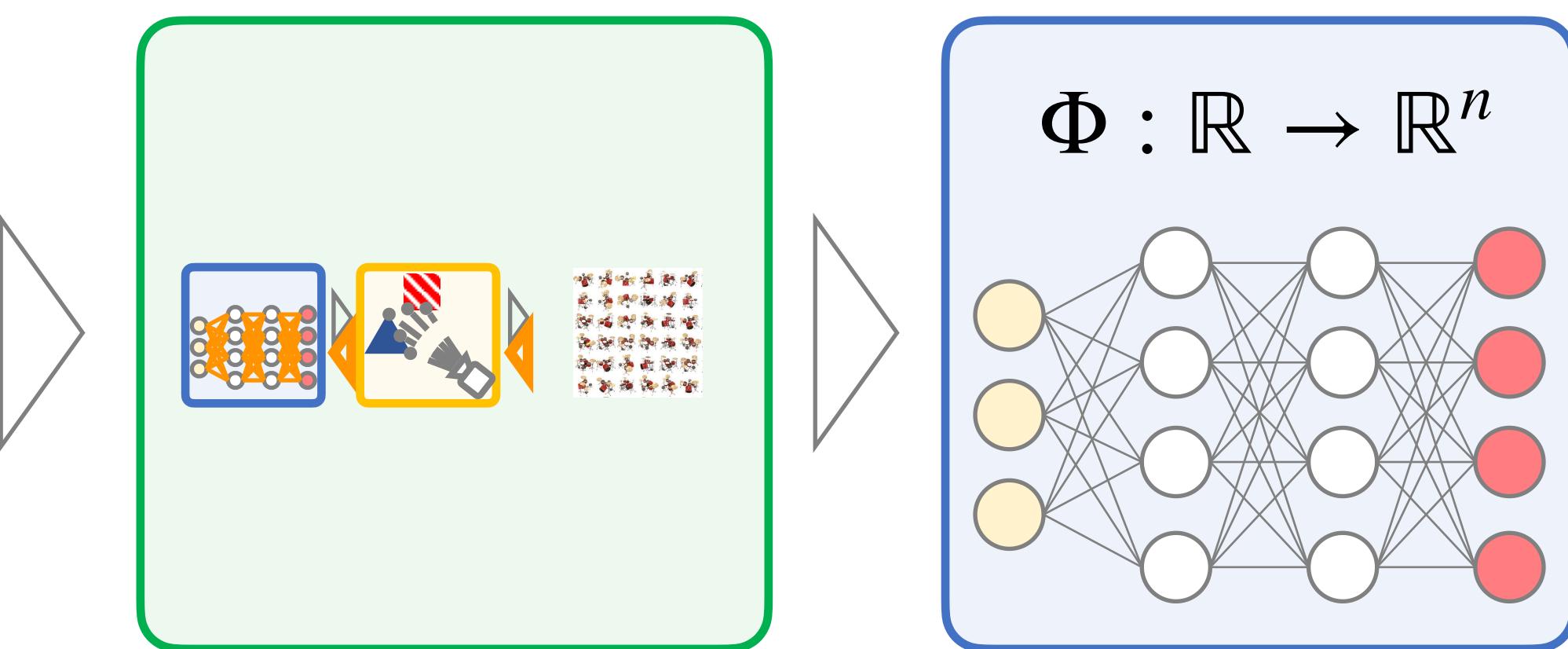
Learned aggregation

What did we do here?



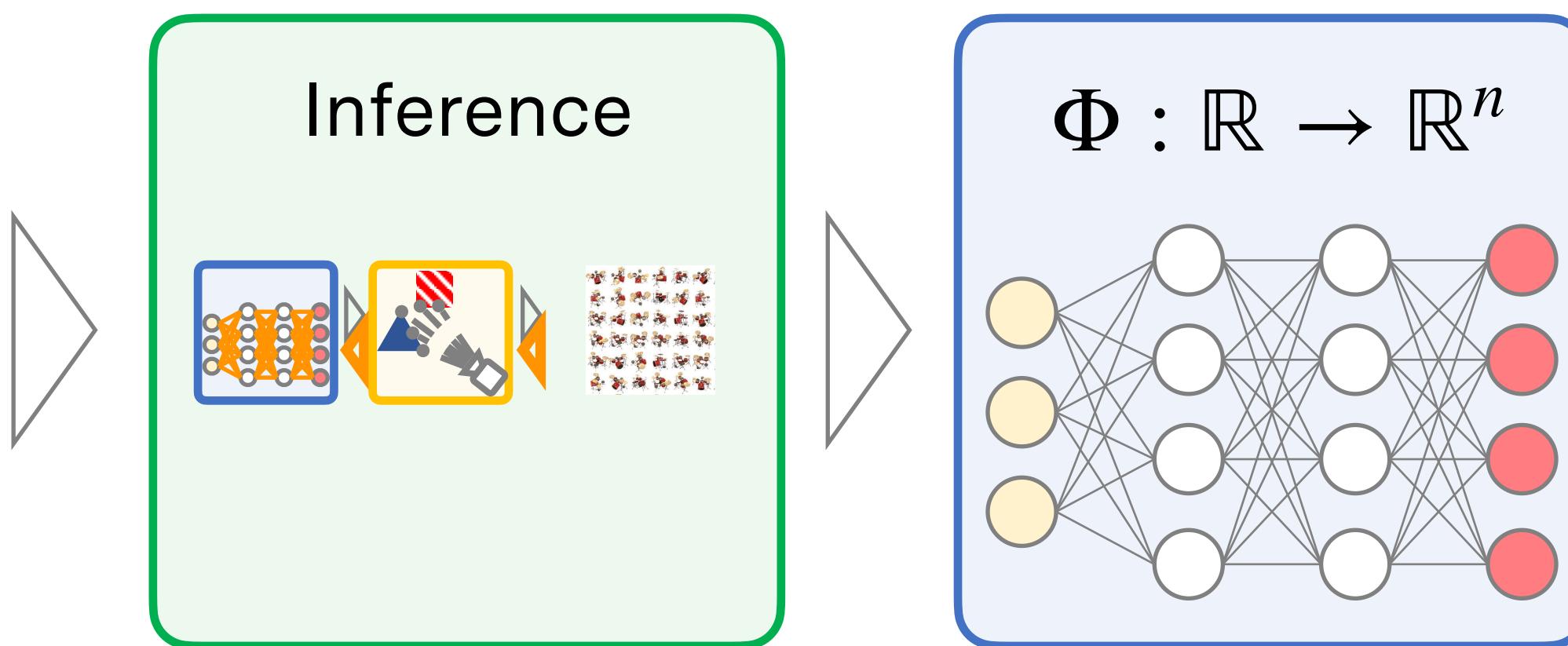
What did we do here?

Observations



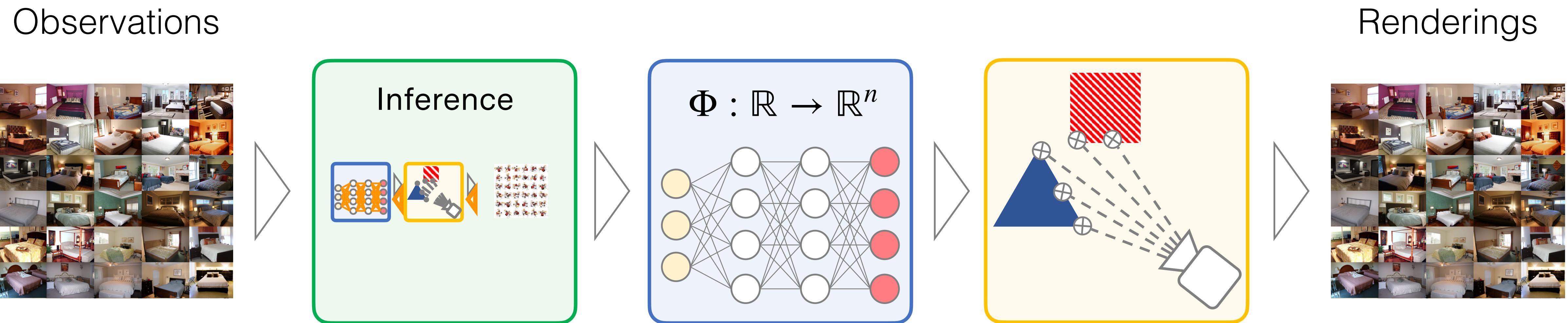
What did we do here?

Observations



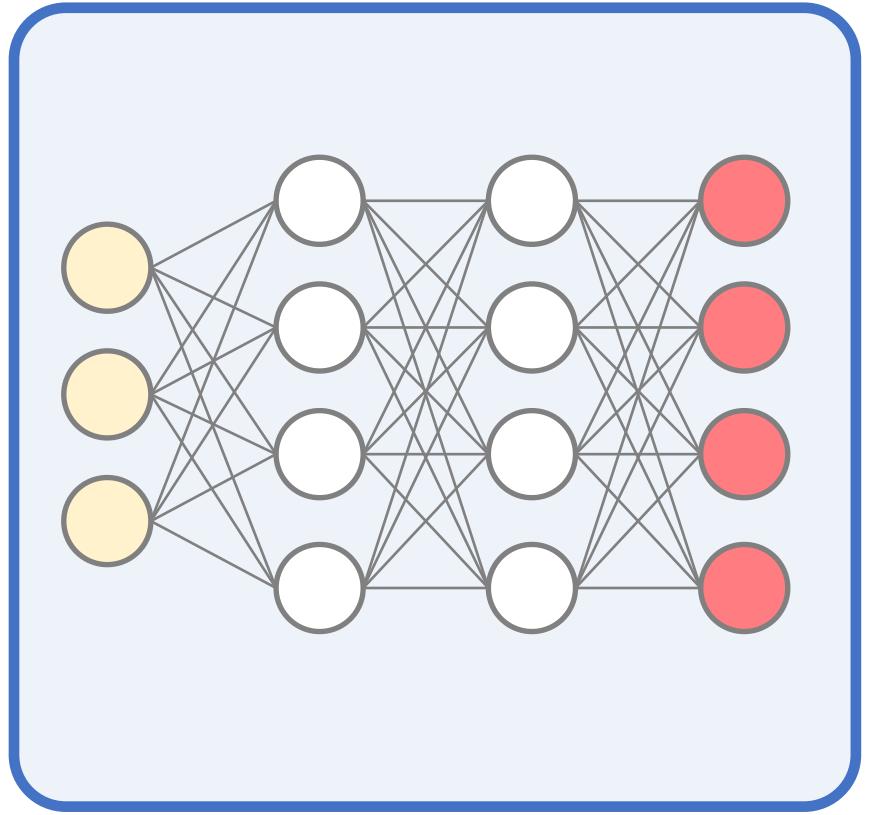
We inferred “hidden/latent” variables 3D appearance and geometry, given “observable” variables RGB images and camera poses.

What did we do here?

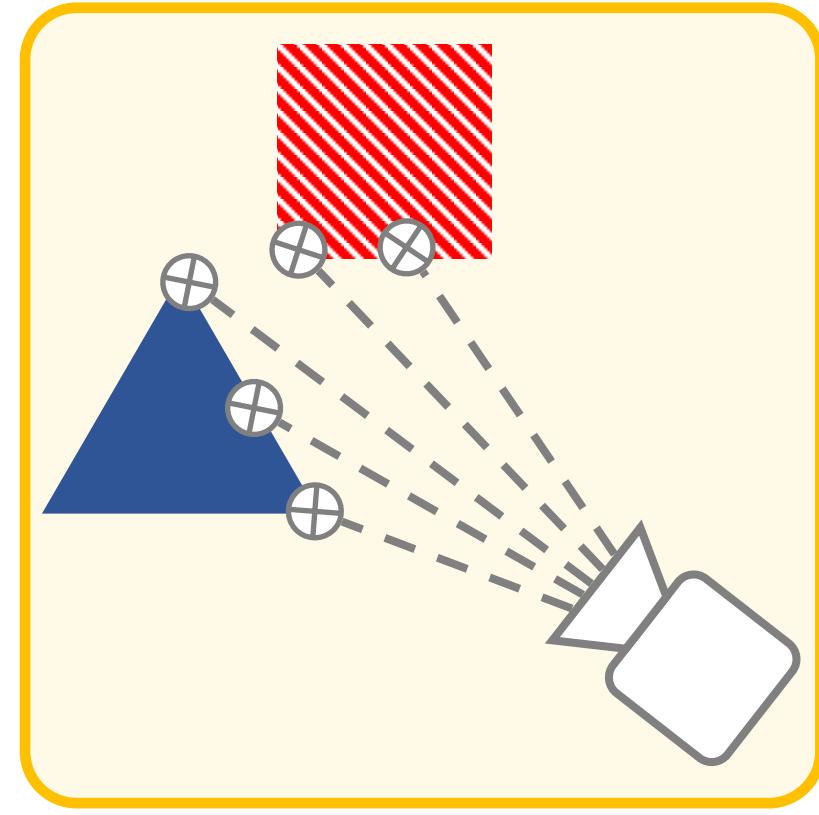


We inferred “hidden/latent” variables 3D appearance and geometry, given “observable” variables RGB images and camera poses.

What if observations don't constrain scene representation?



Srn $_{\phi}$



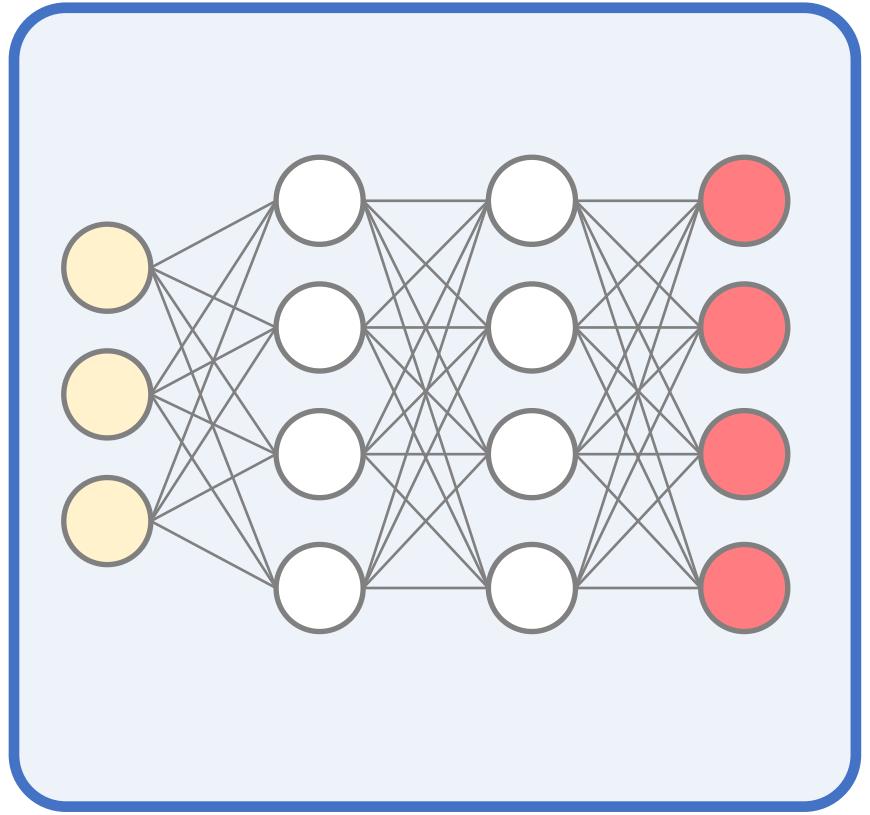
Render $_{\theta}$



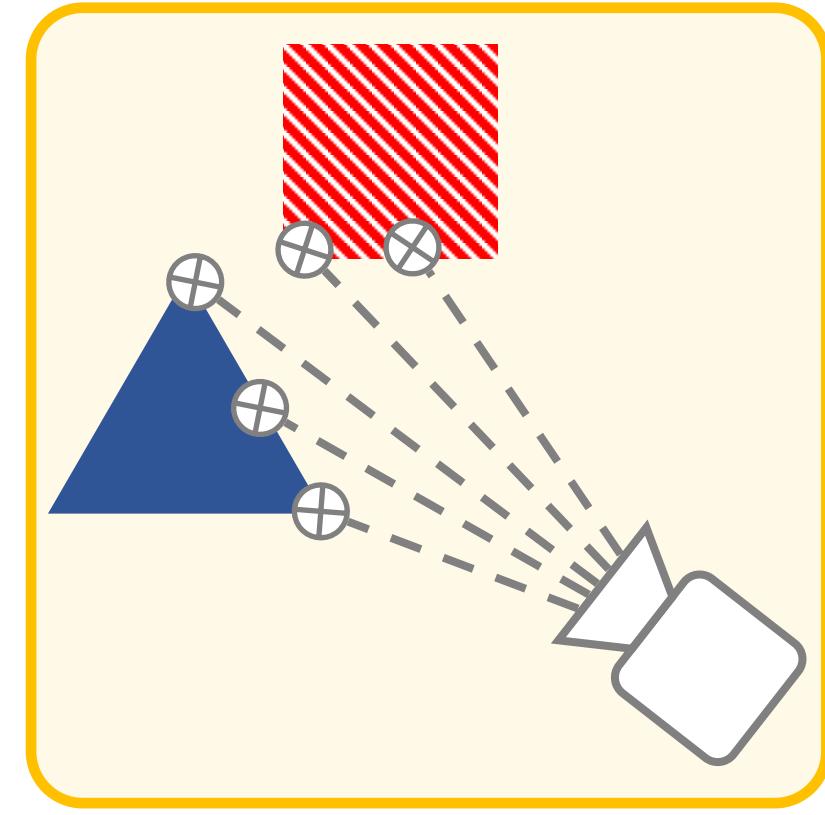
$$(\mathcal{I}, \xi)$$

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

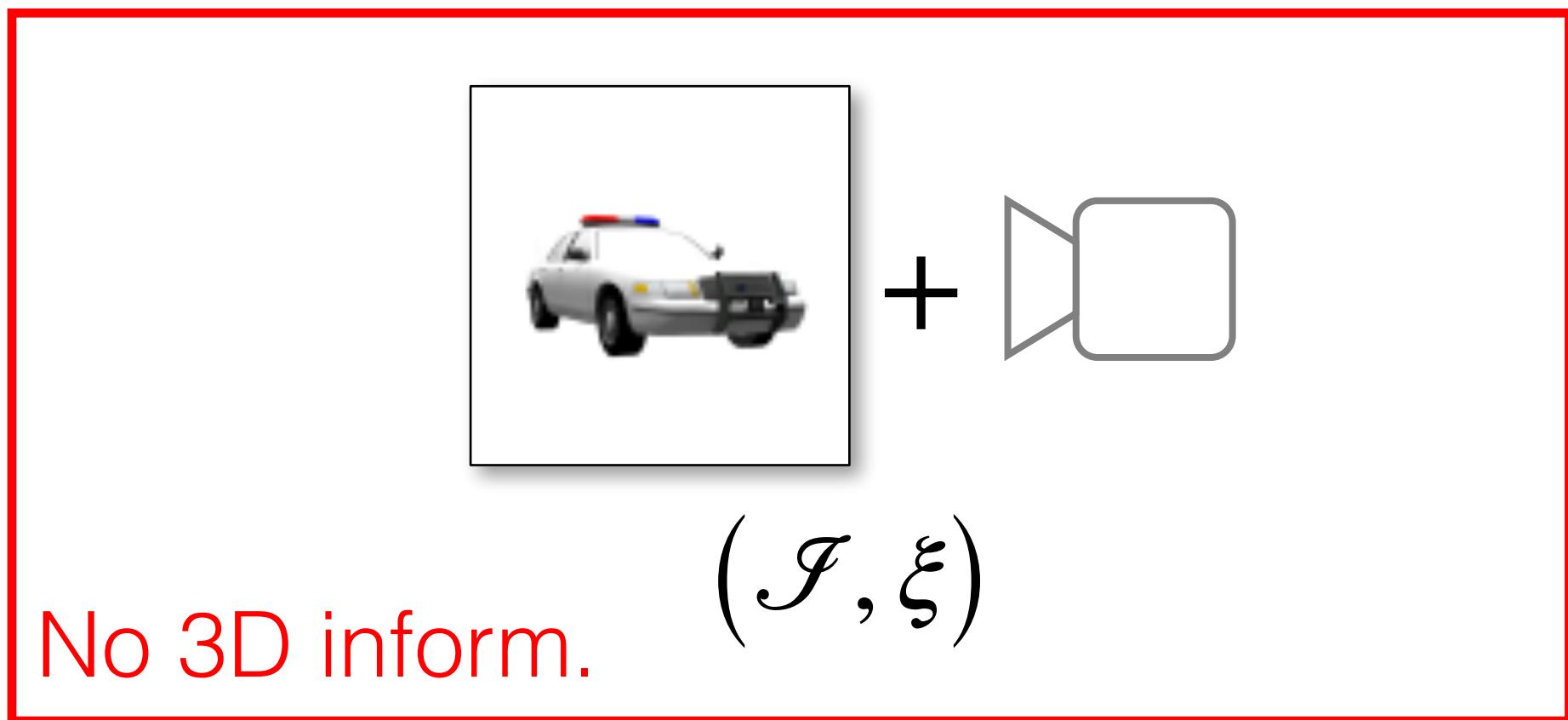
What if observations don't constrain scene representation?



Srn_ϕ

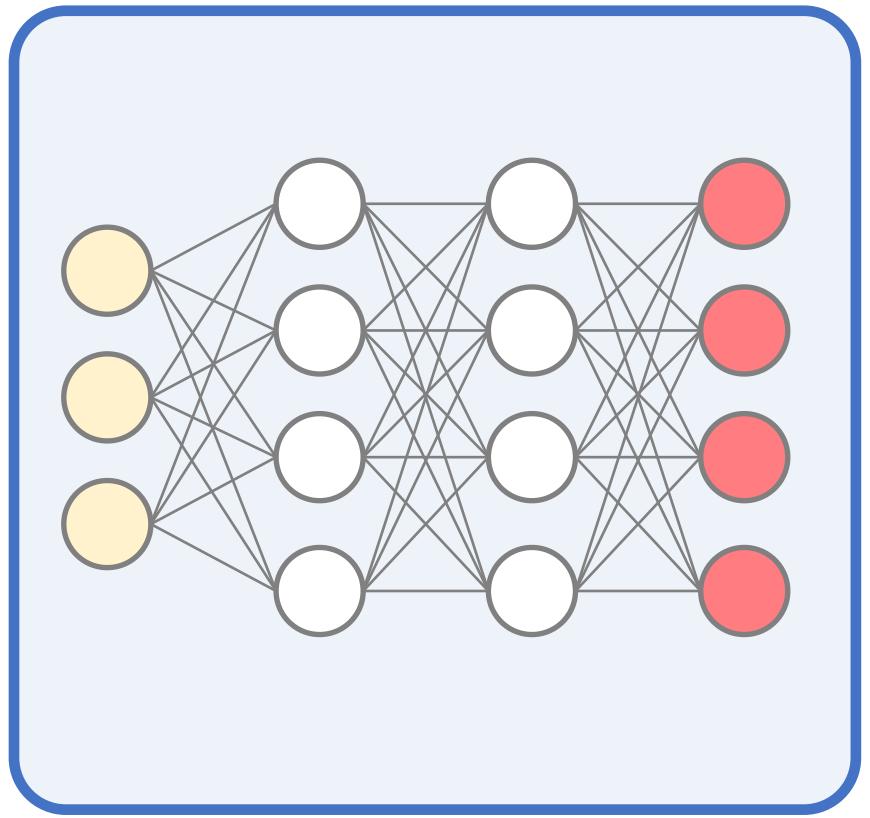


Render_θ

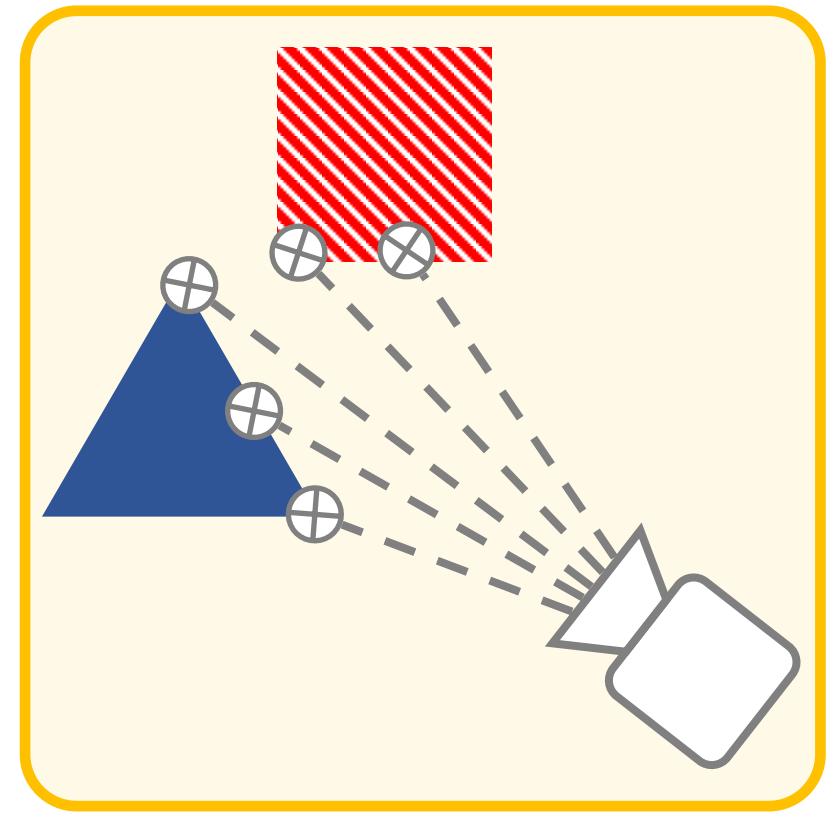


$$\underset{\phi, \theta}{\operatorname{argmin}} \| \text{Render}_\theta(\text{Srn}_\phi, \xi) - \mathcal{I} \|$$

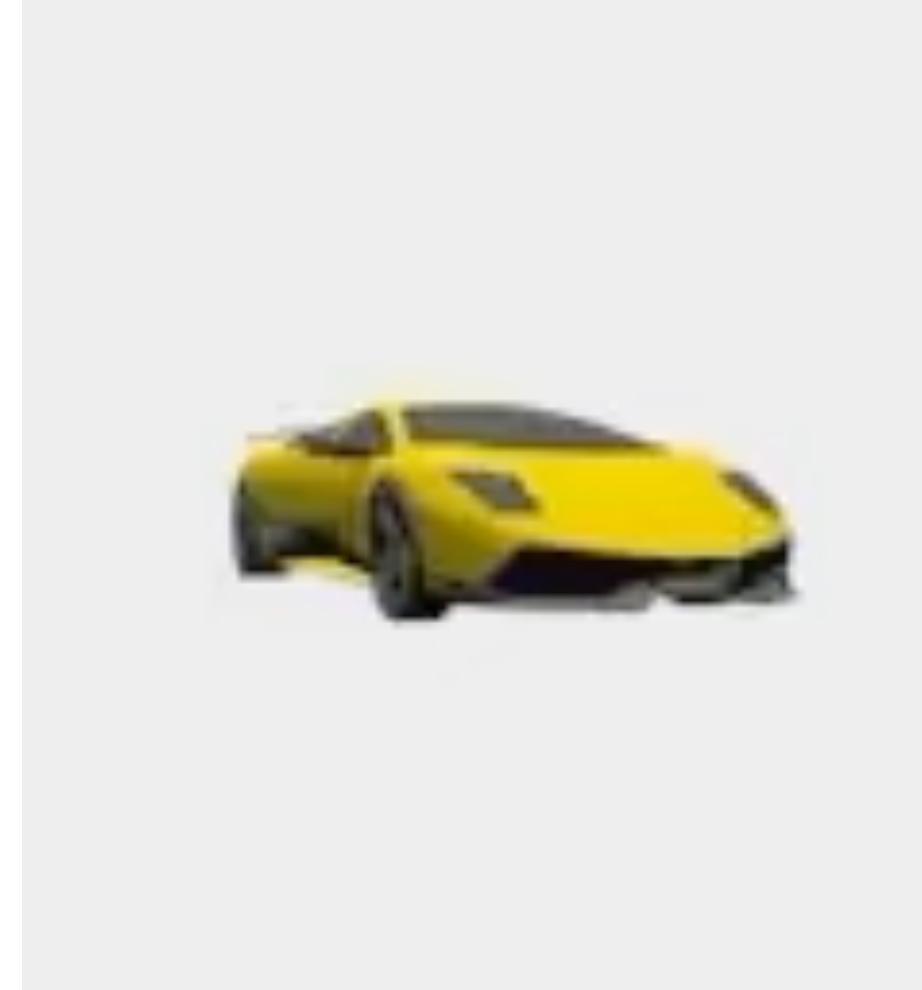
What if observations don't constrain scene representation?



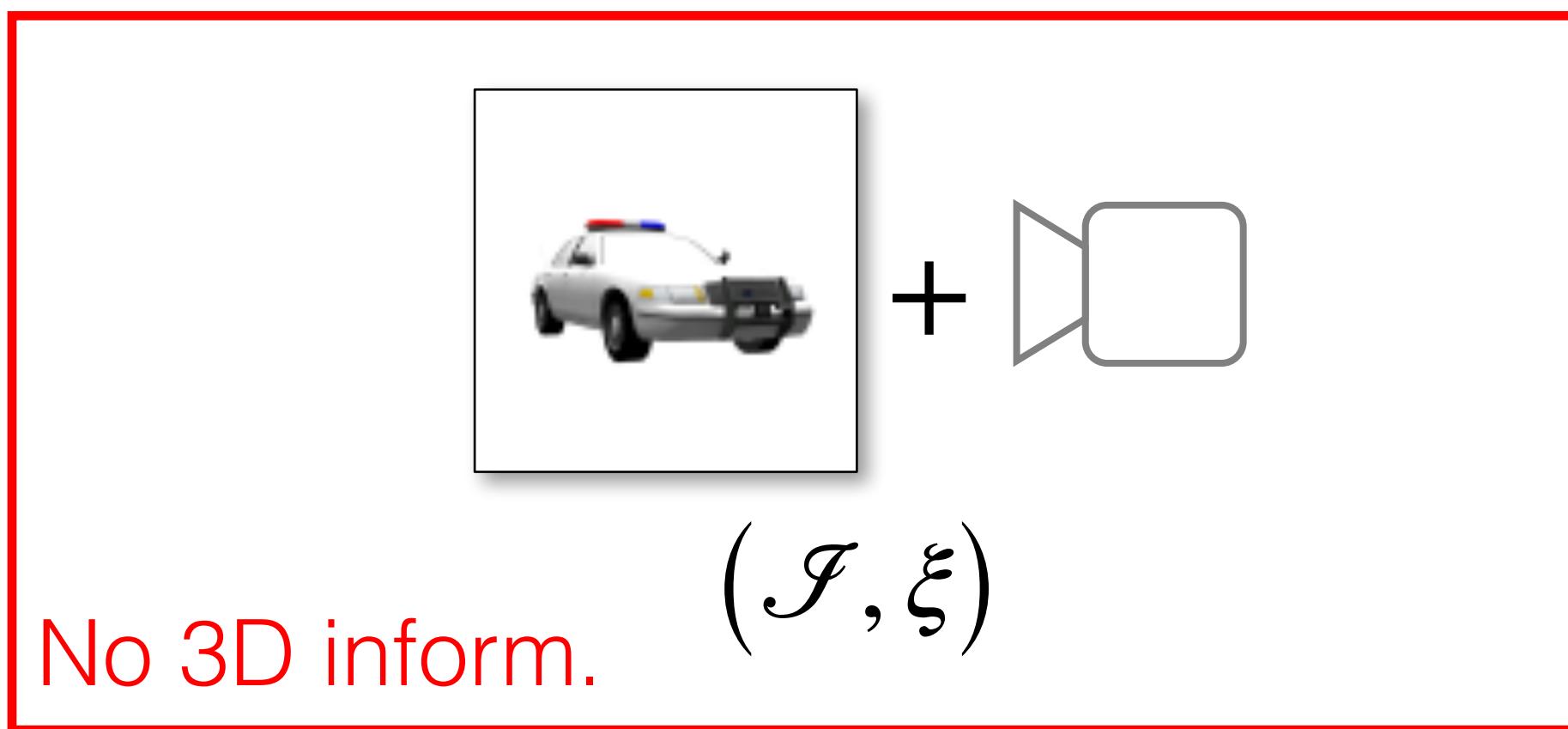
Srn_ϕ



Render_θ

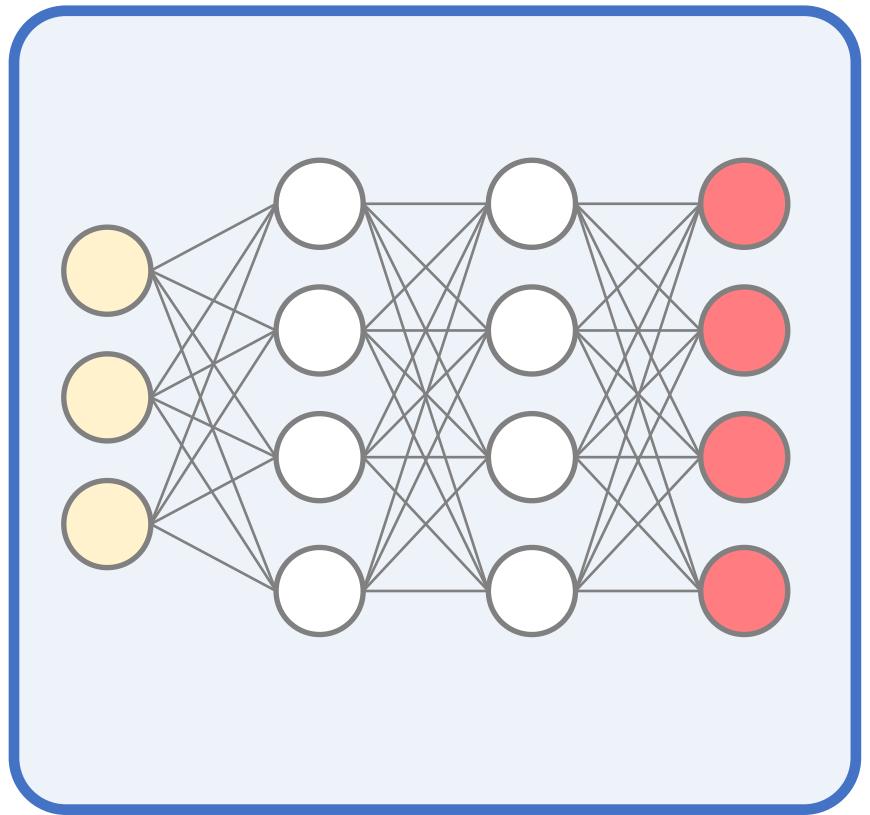


Input

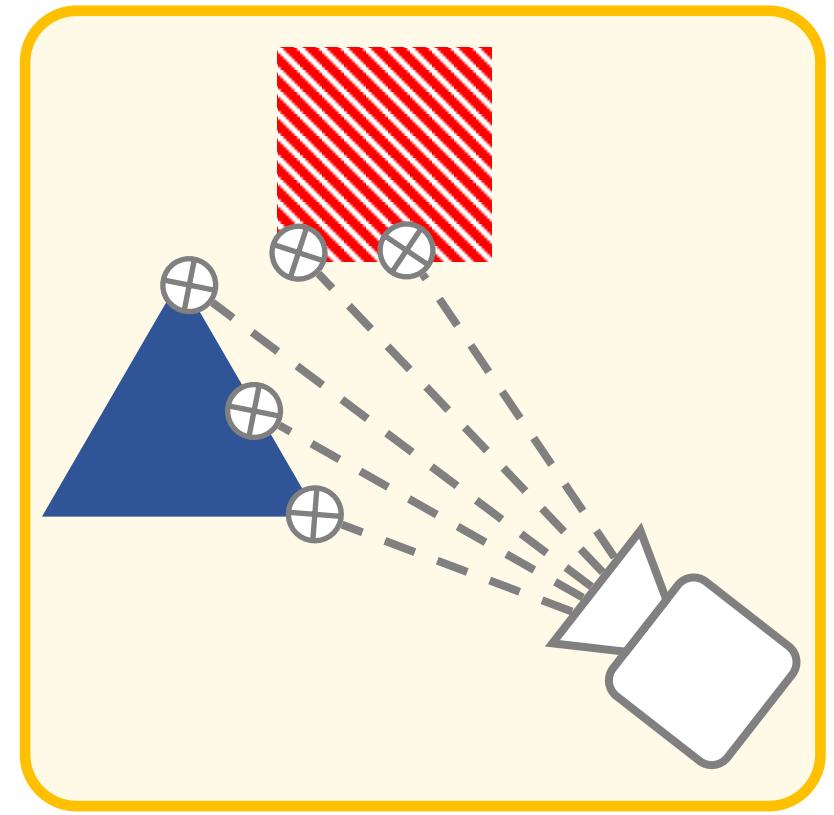


$$\operatorname{argmin}_{\phi, \theta} \| \text{Render}_\theta(\text{Srn}_\phi, \xi) - \mathcal{I} \|$$

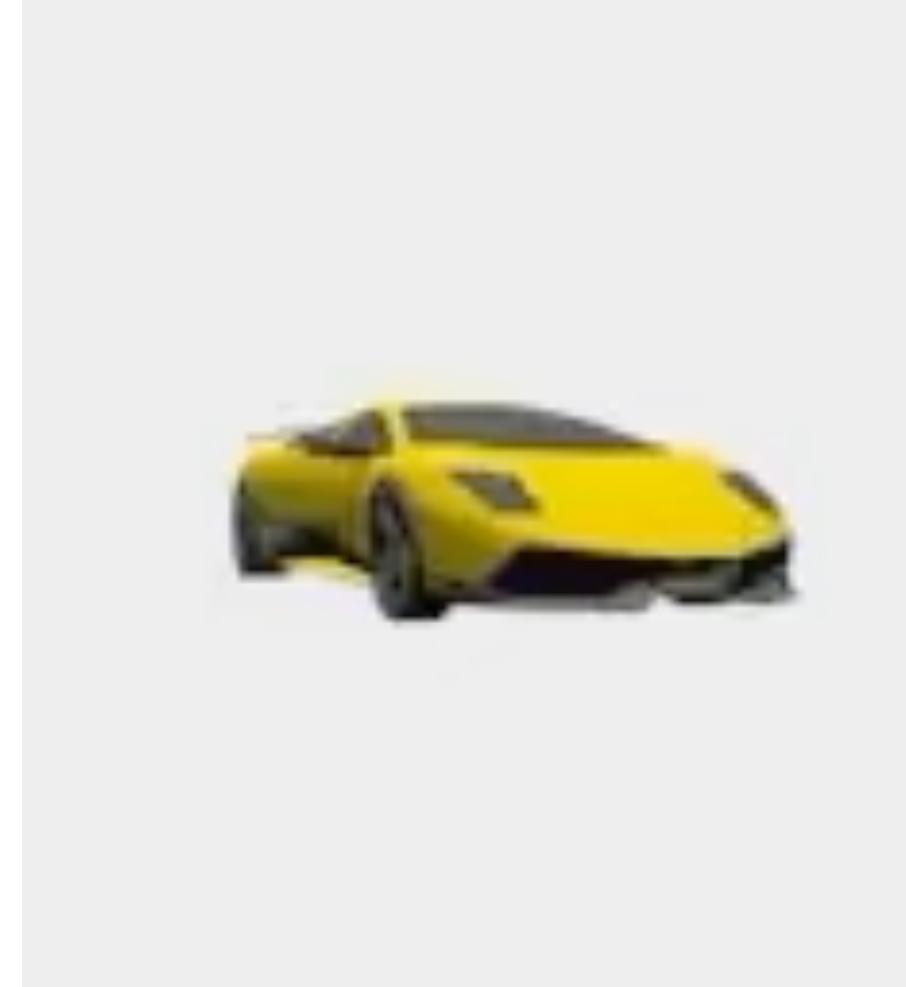
What if observations don't constrain scene representation?



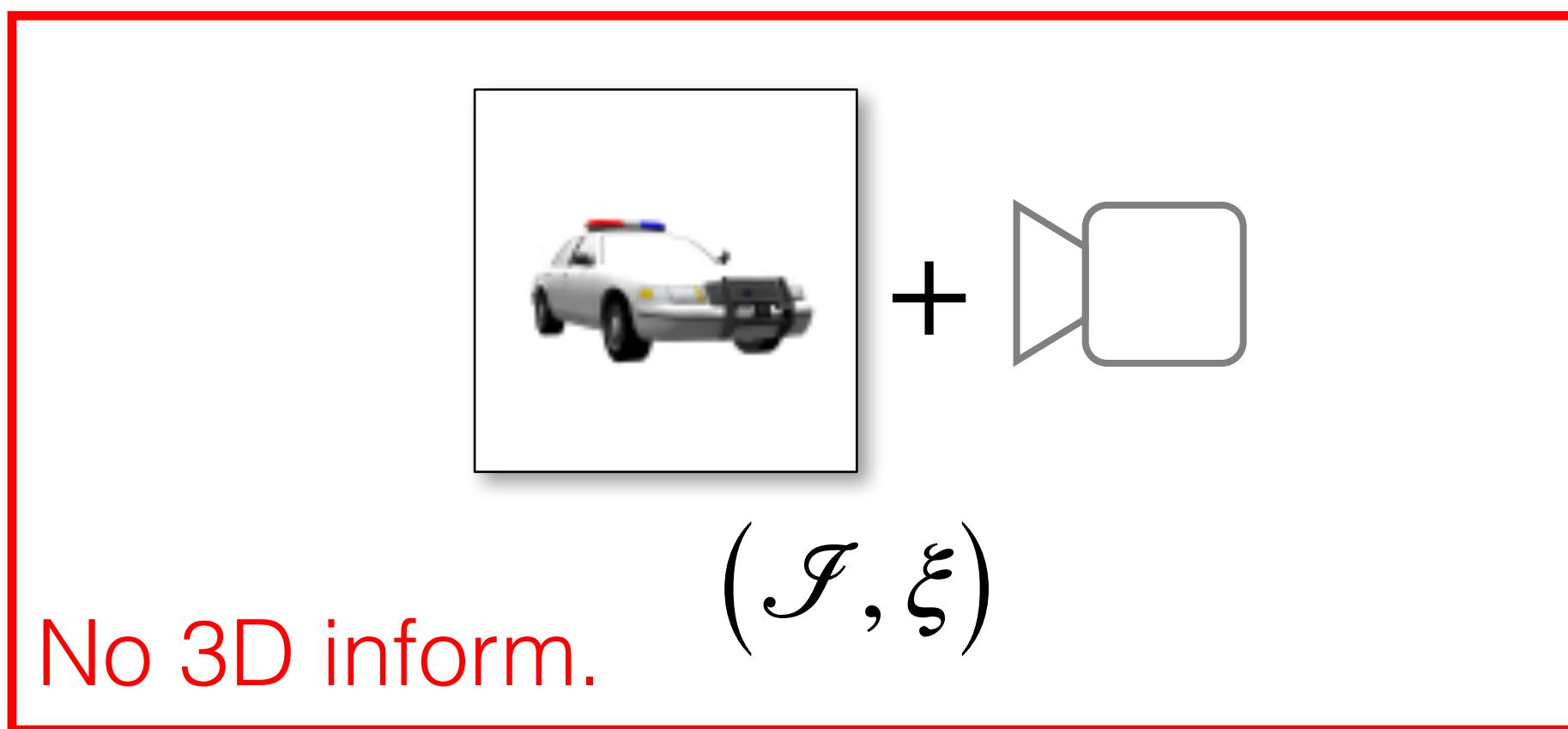
Srn_ϕ



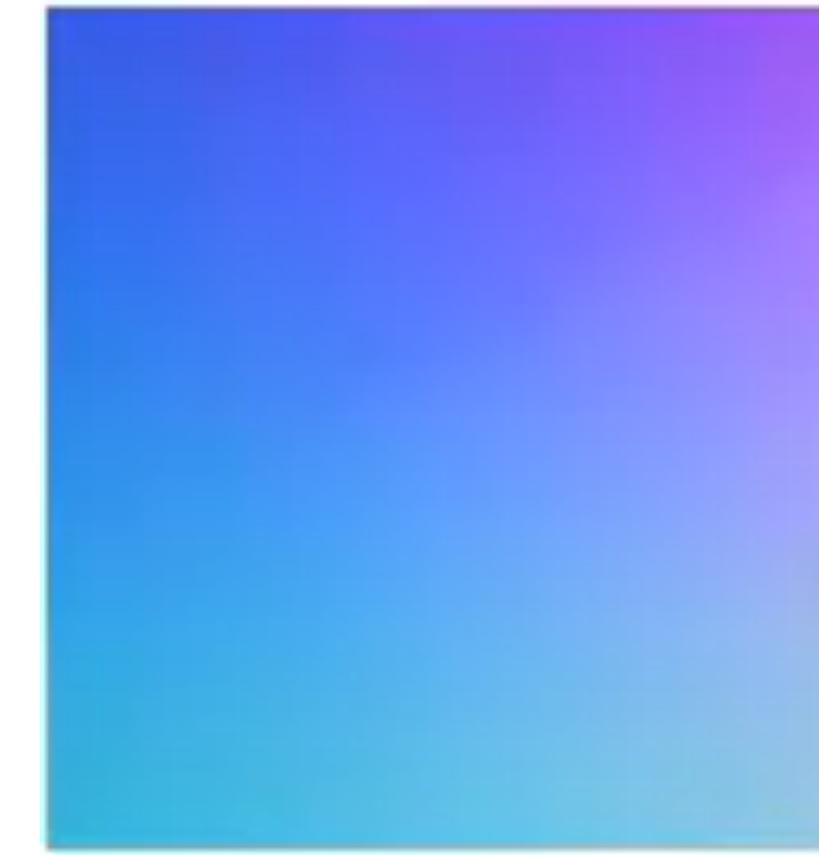
Render_θ



Input



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



Normal map

RGB



GT

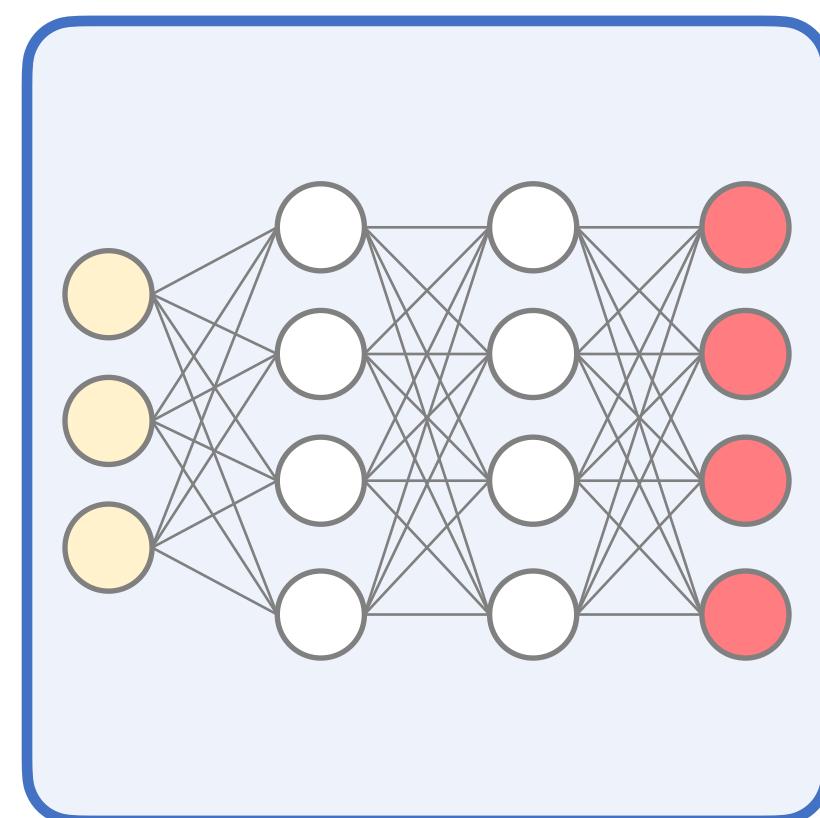
Today: *Learned* inference algorithms!

Observations

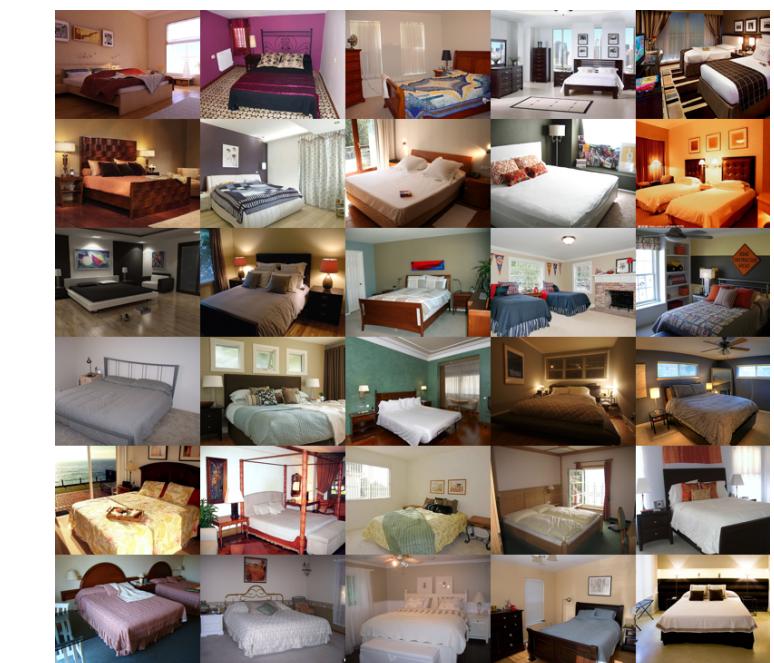


Inference

?



Renderings



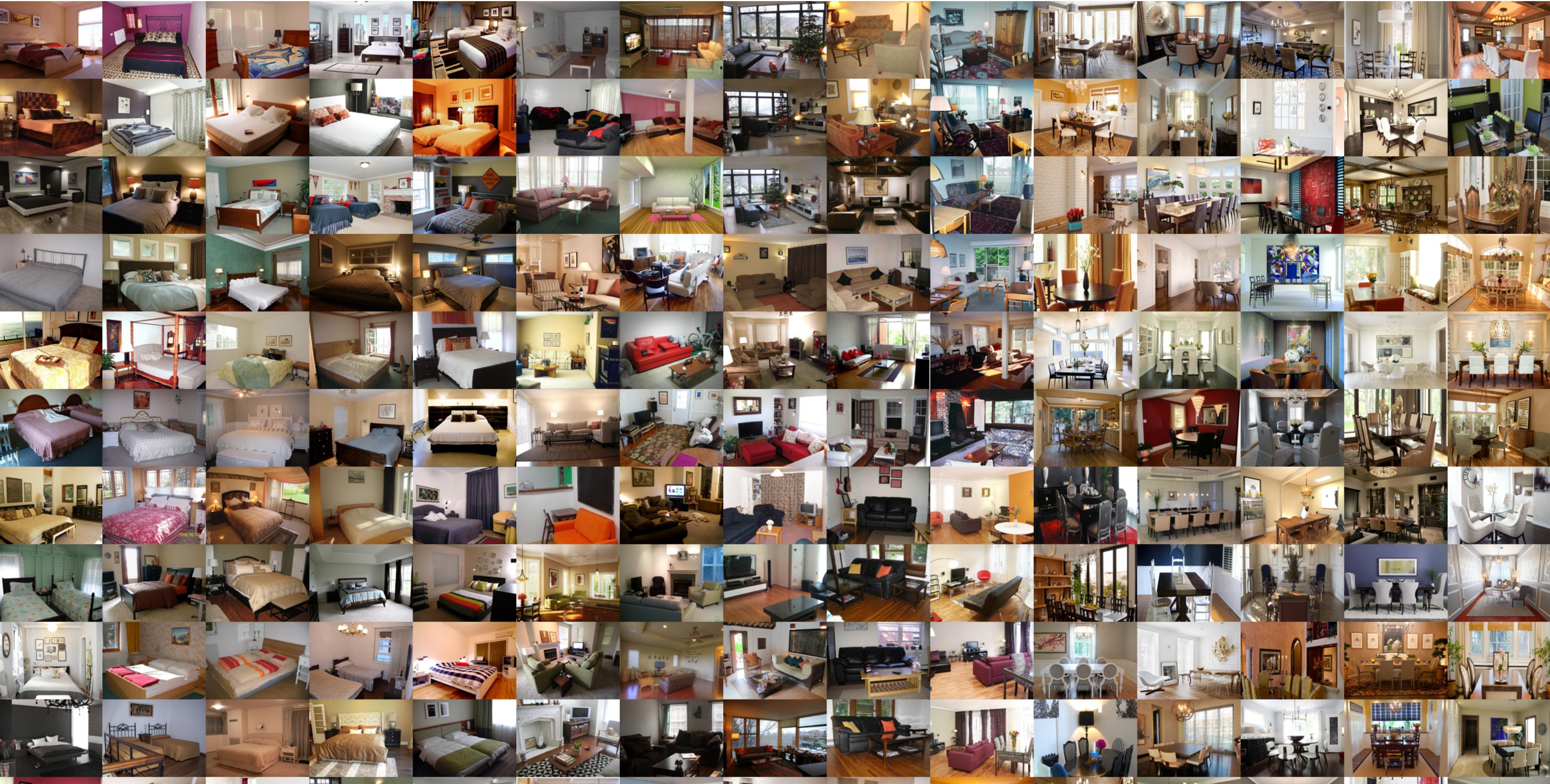
Why?

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*?



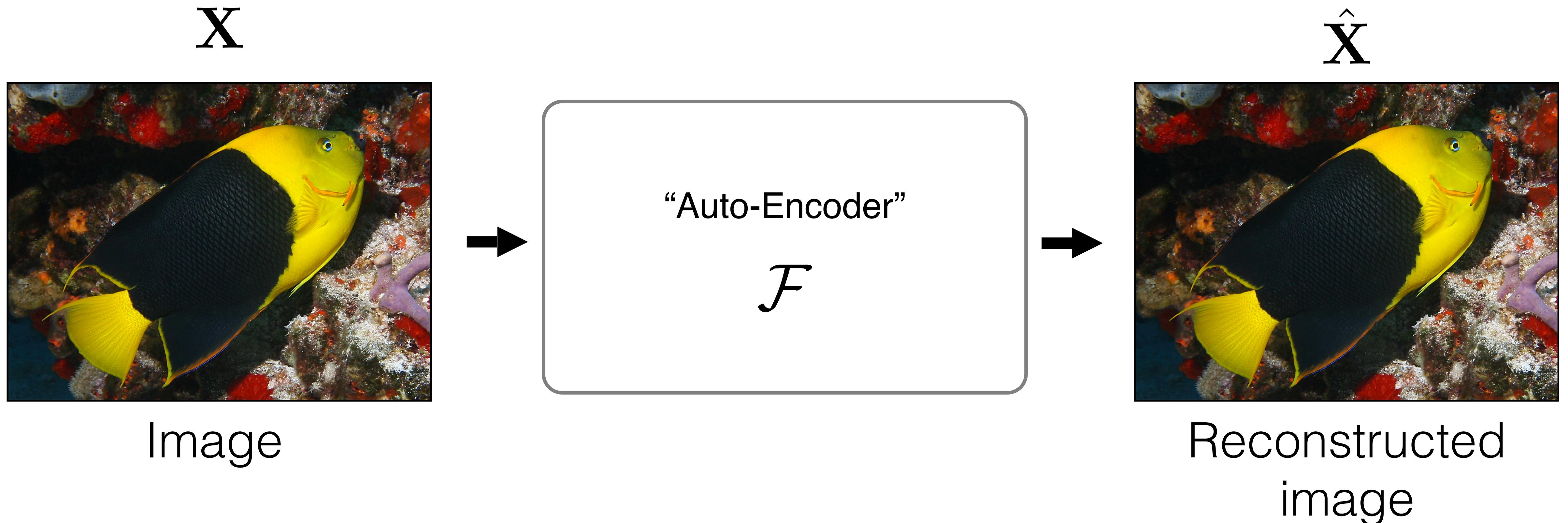
Auto-Encoding

X

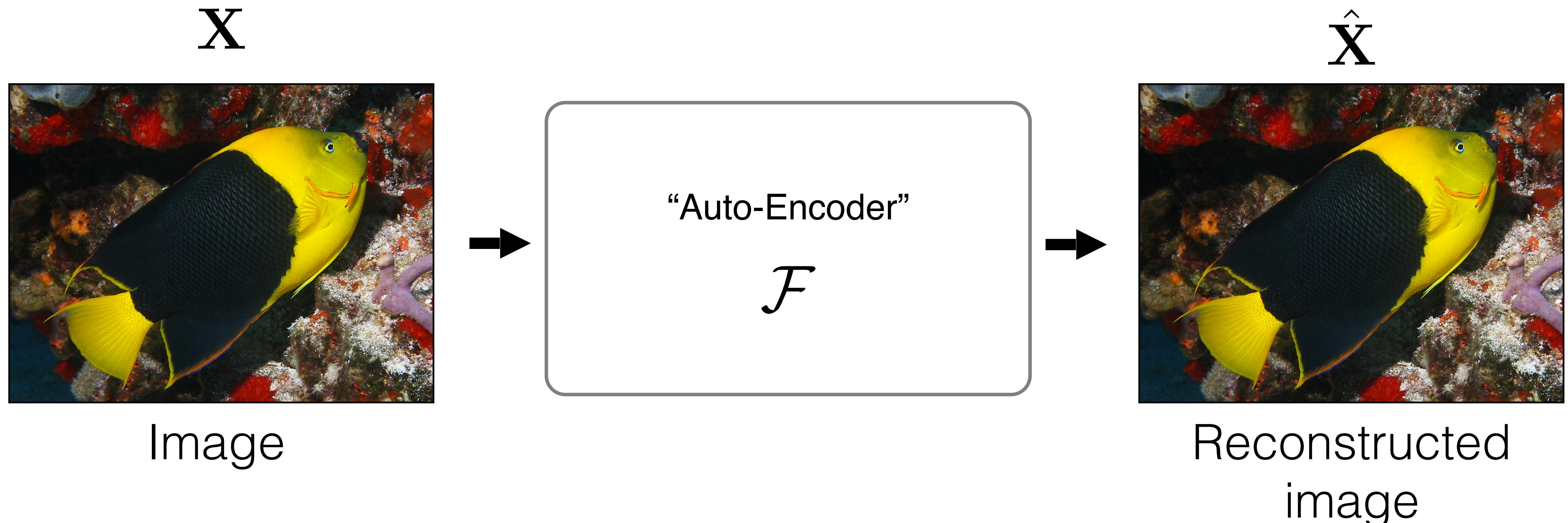


Image

Auto-Encoding



Auto-Encoding



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

Auto-Encoding

X



Image



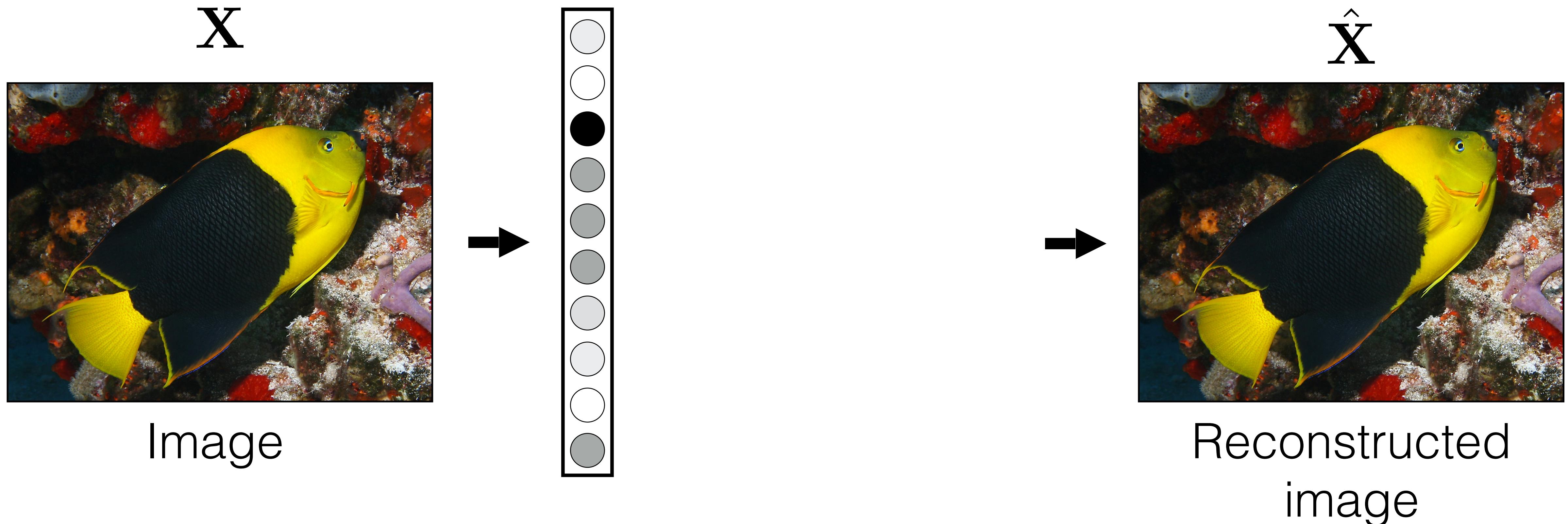
\hat{X}



Reconstructed
image

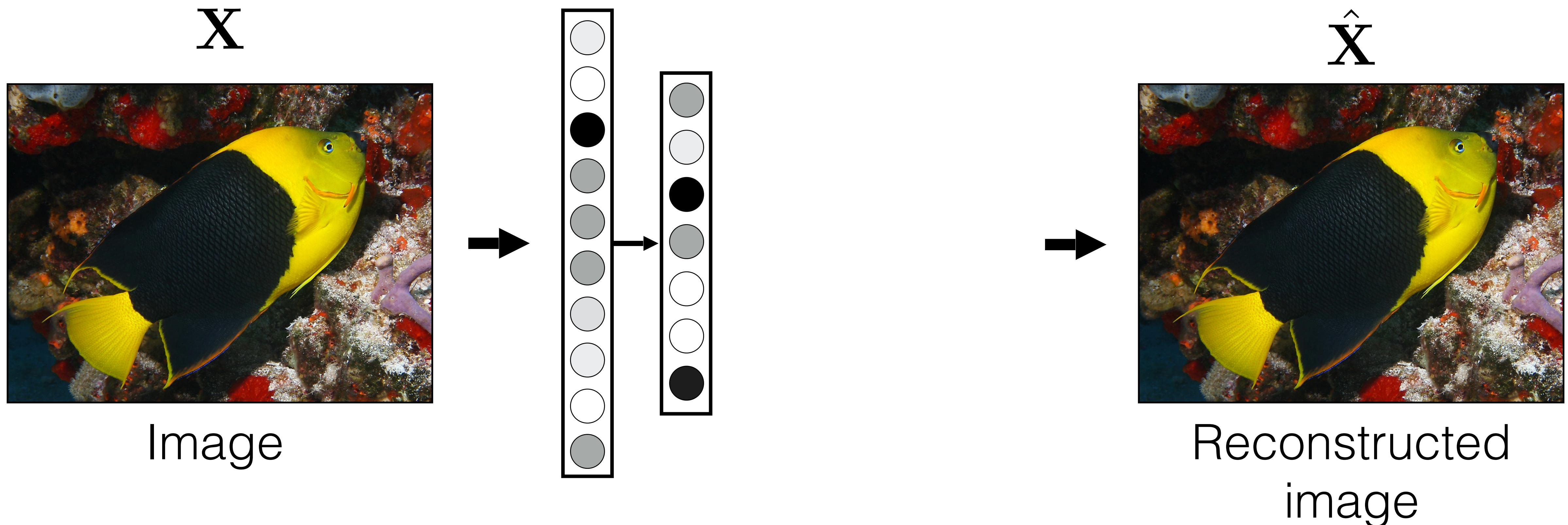
[e.g., Hinton & Salakhutdinov, Science 2006]

Auto-Encoding



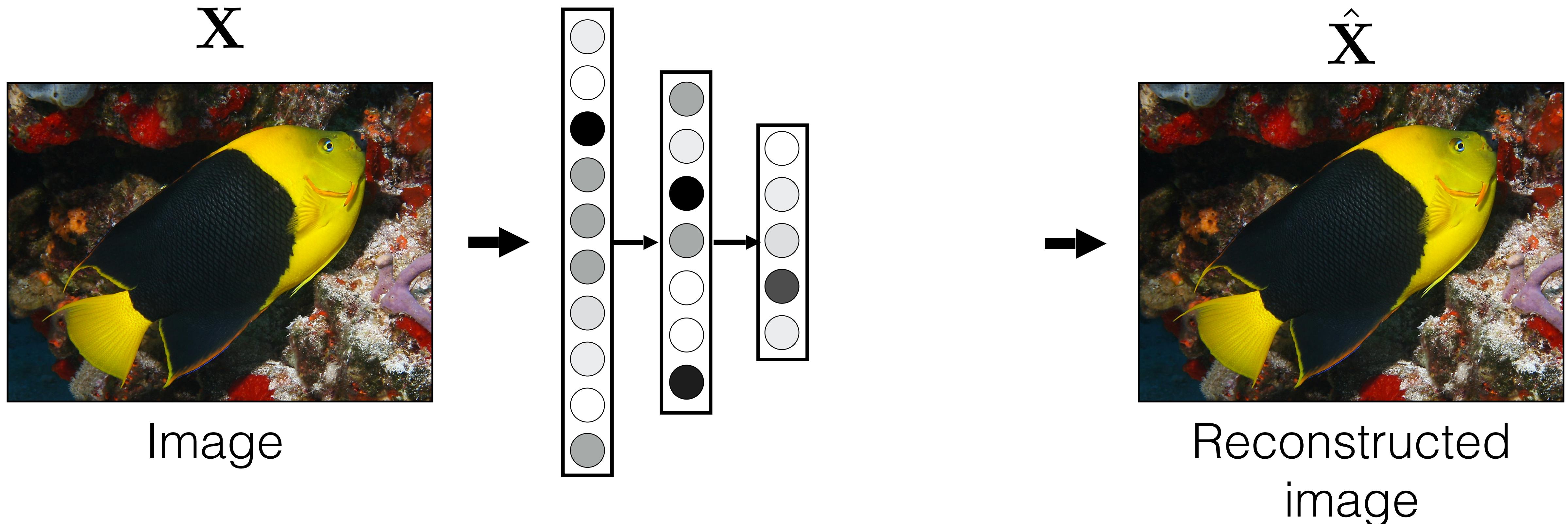
[e.g., Hinton & Salakhutdinov, Science 2006]

Auto-Encoding



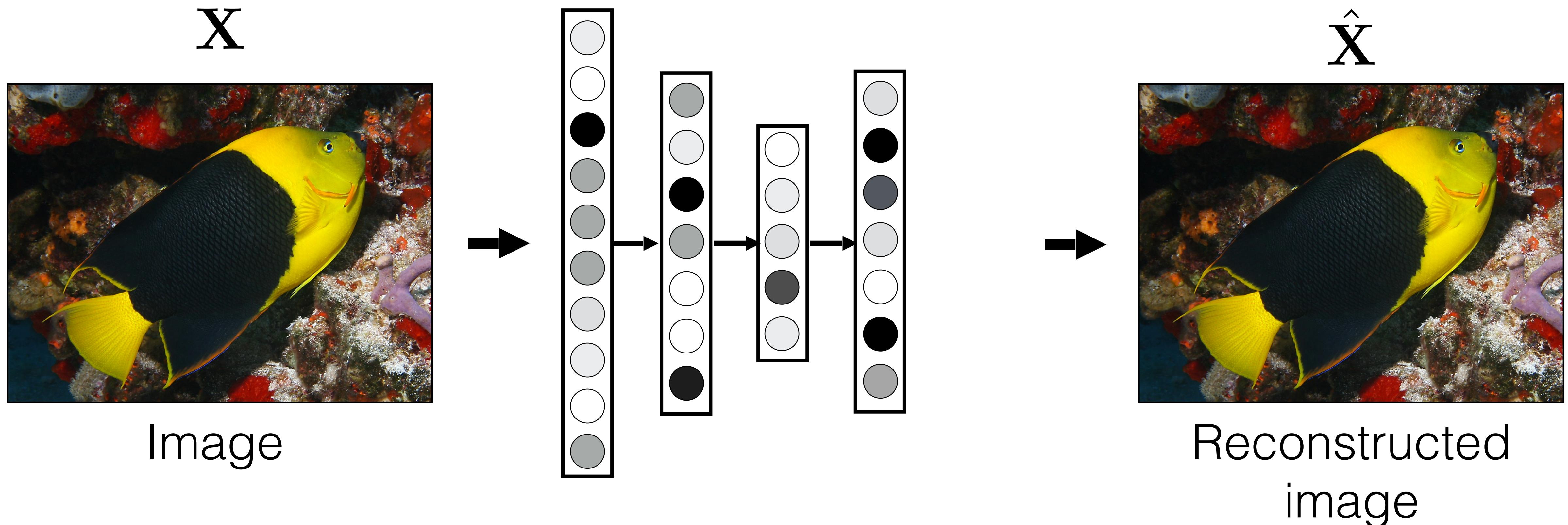
[e.g., Hinton & Salakhutdinov, Science 2006]

Auto-Encoding



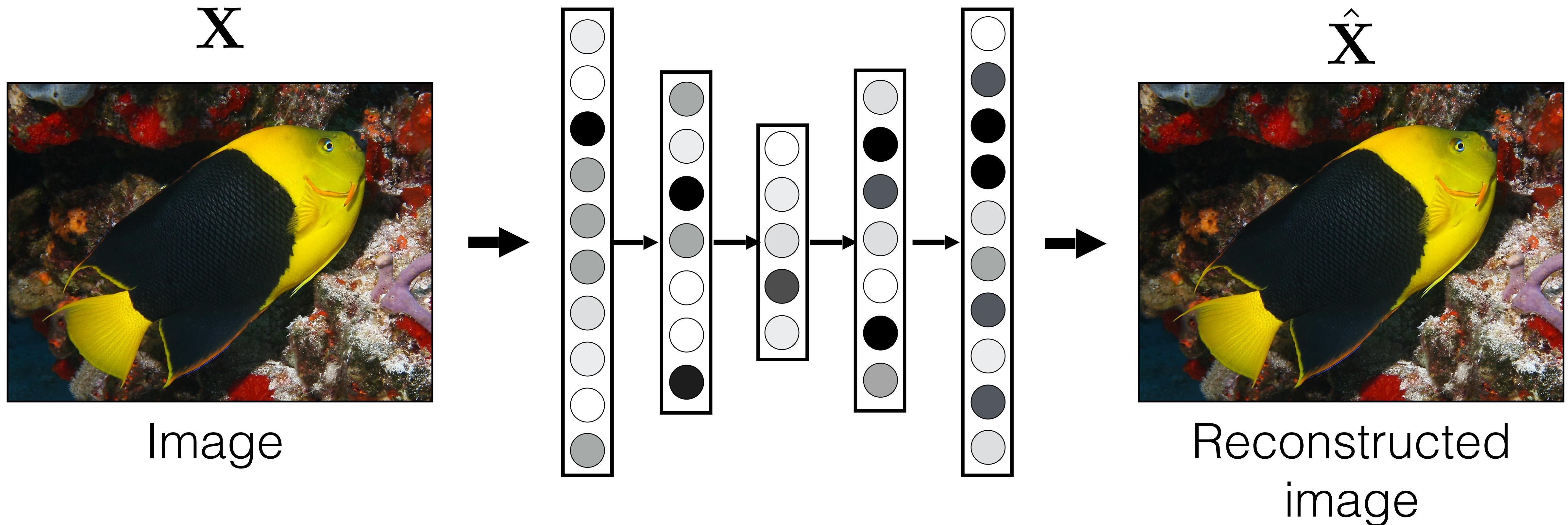
[e.g., Hinton & Salakhutdinov, Science 2006]

Auto-Encoding



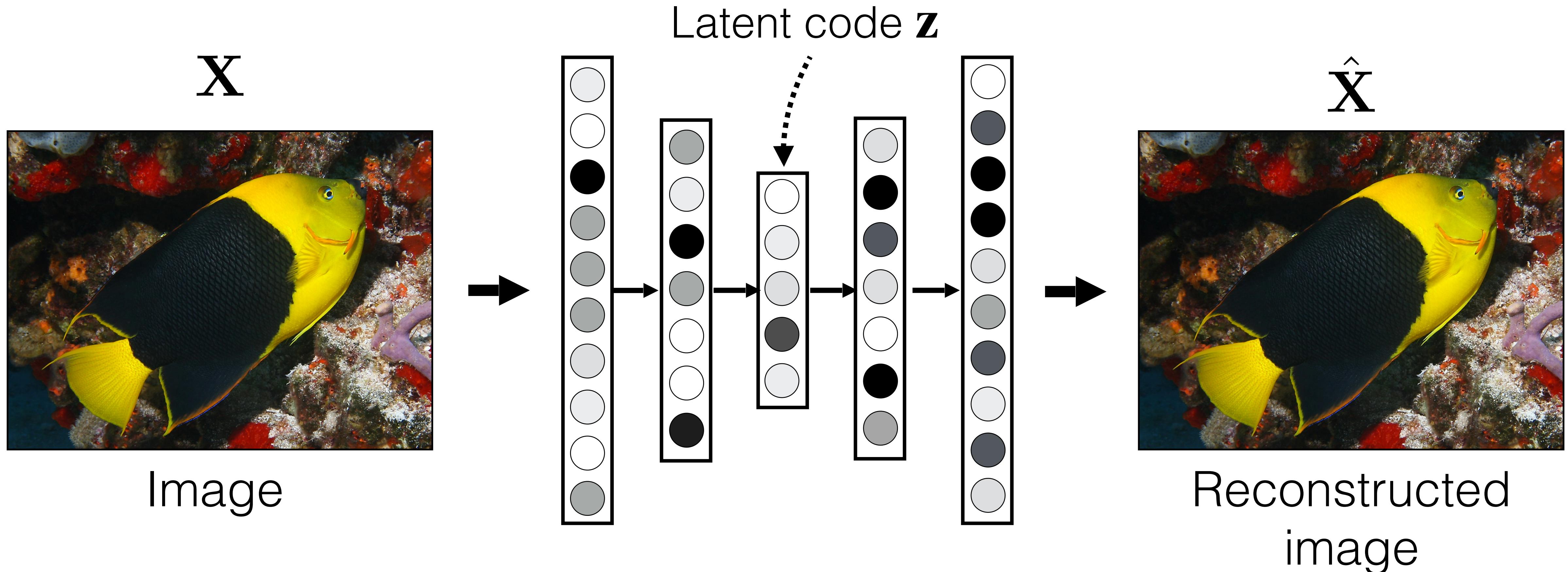
[e.g., Hinton & Salakhutdinov, Science 2006]

Auto-Encoding



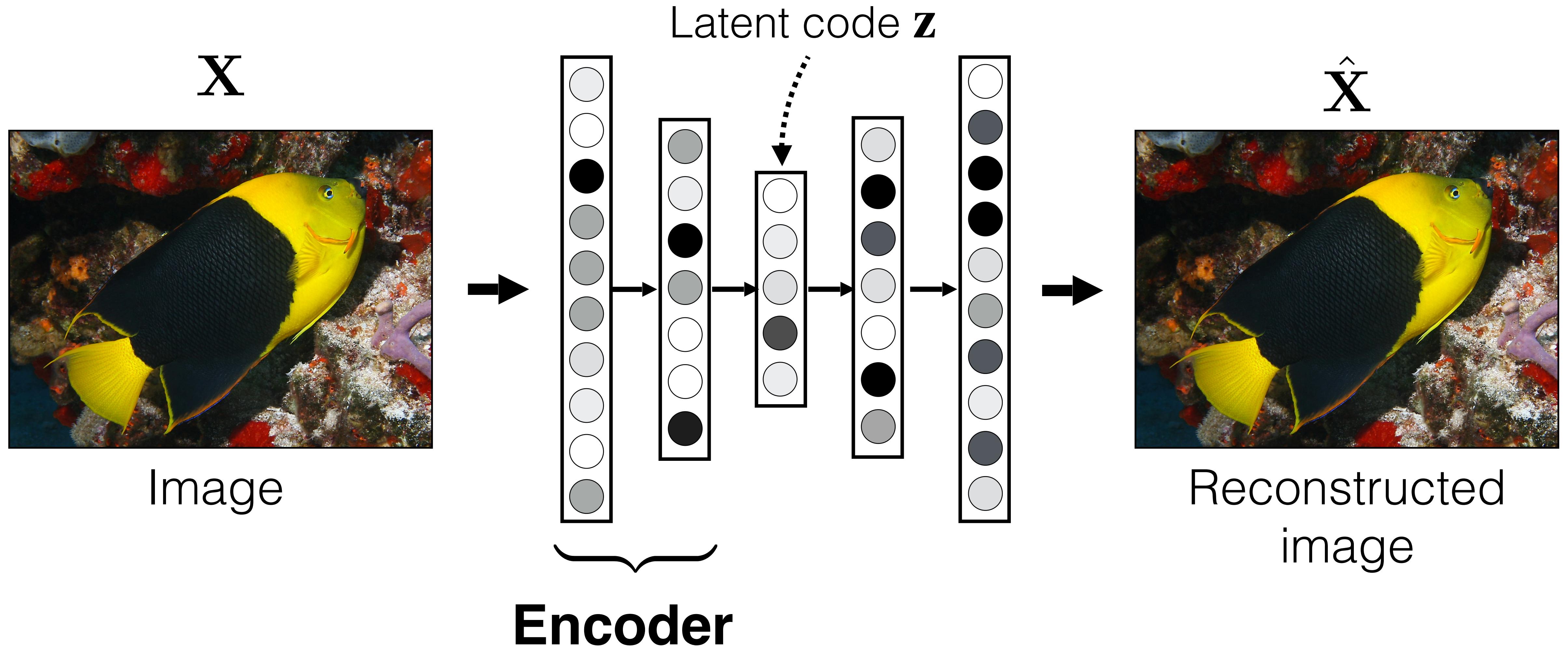
[e.g., Hinton & Salakhutdinov, Science 2006]

Auto-Encoding



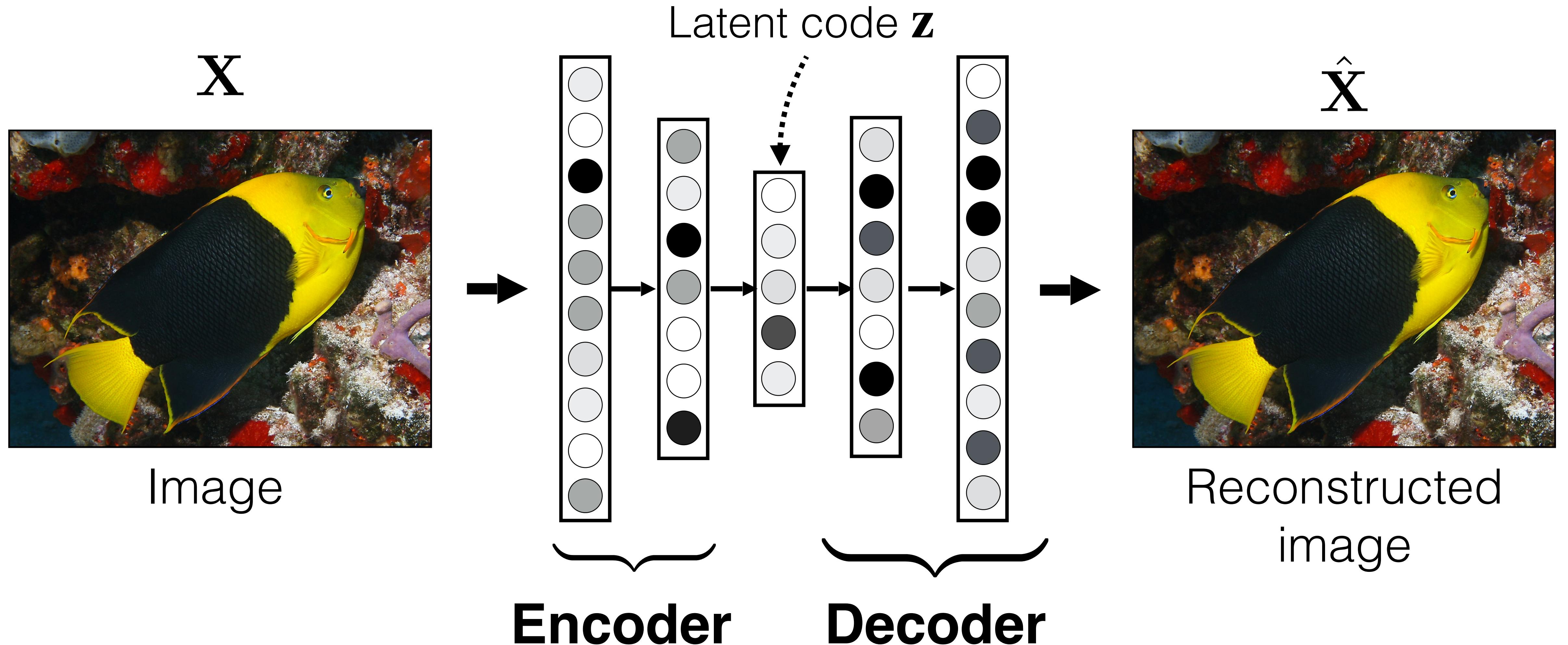
[e.g., Hinton & Salakhutdinov, Science 2006]

Auto-Encoding



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

Auto-Encoding



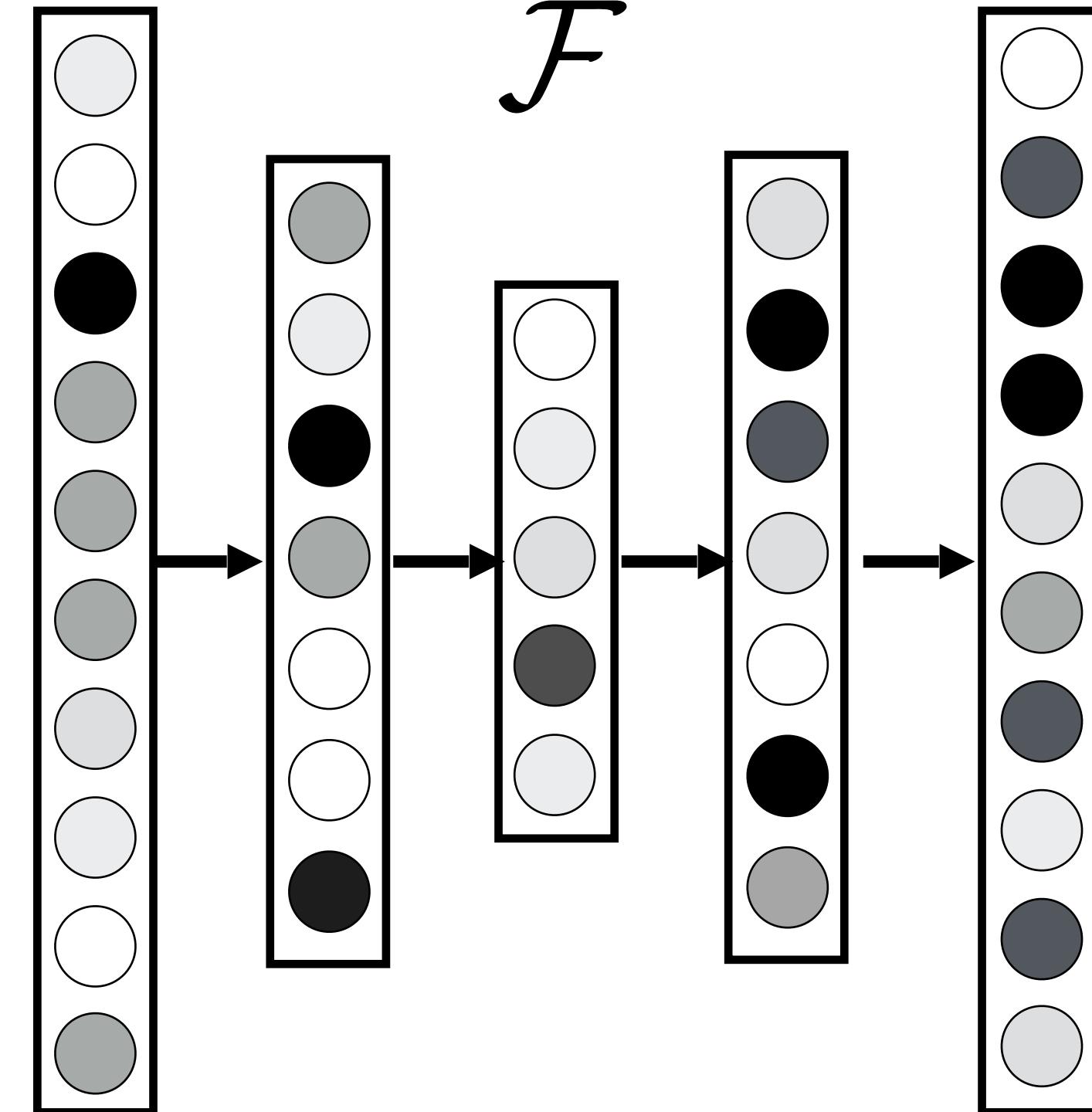
[e.g., Hinton & Salakhutdinov, Science 2006]

\mathbf{X}



Image

\mathcal{F}



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



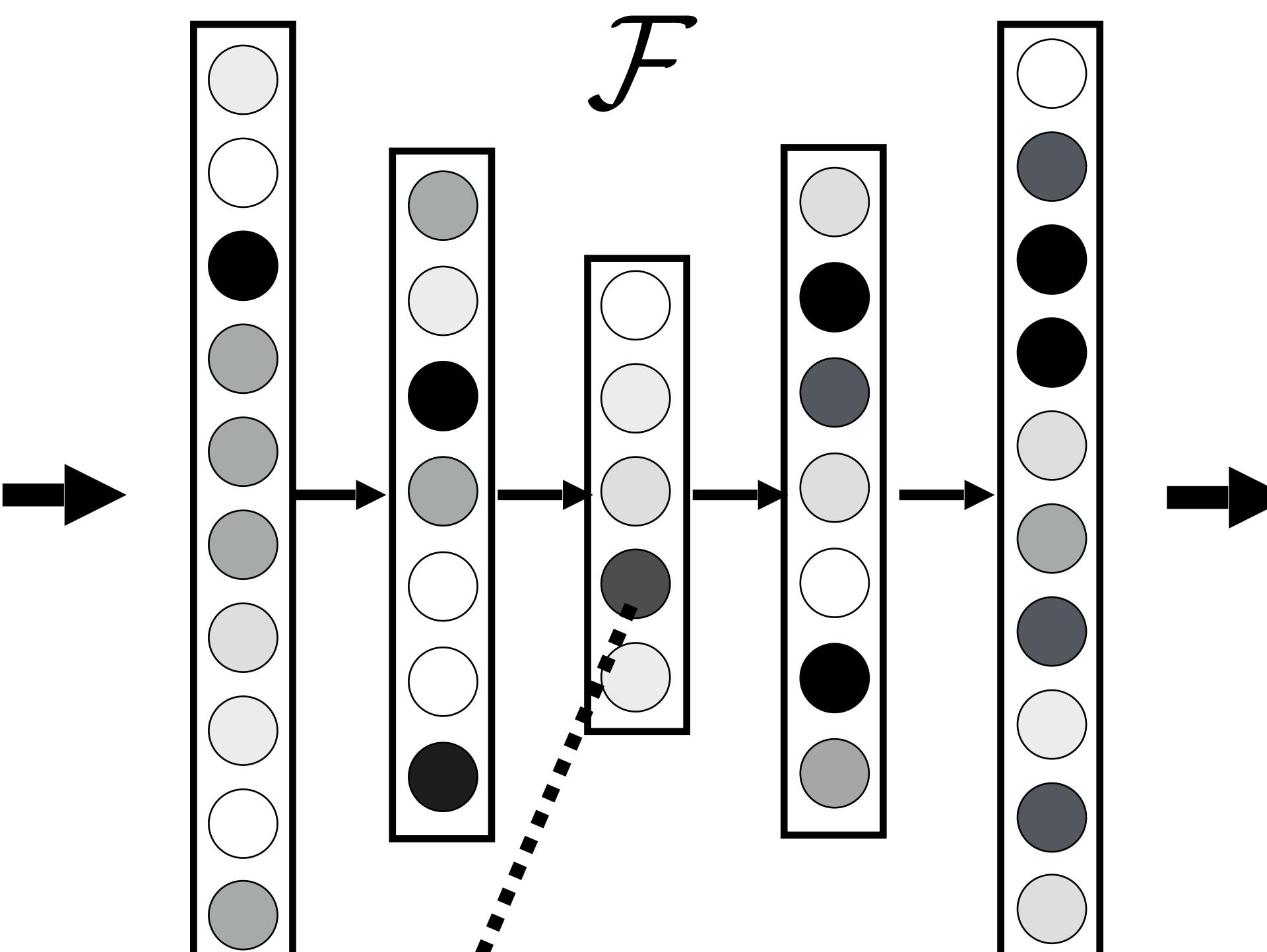
Reconstructed
image

Learning Signal: **Forced Compression**

\mathbf{X}



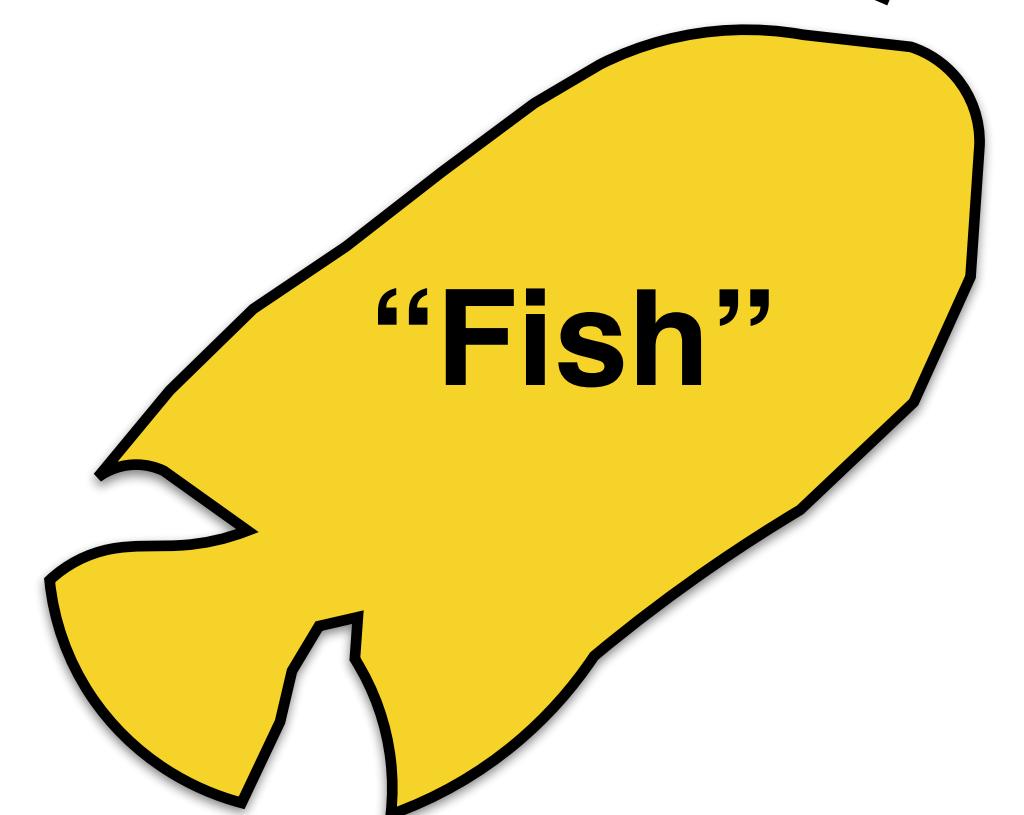
Image



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



Reconstructed
image



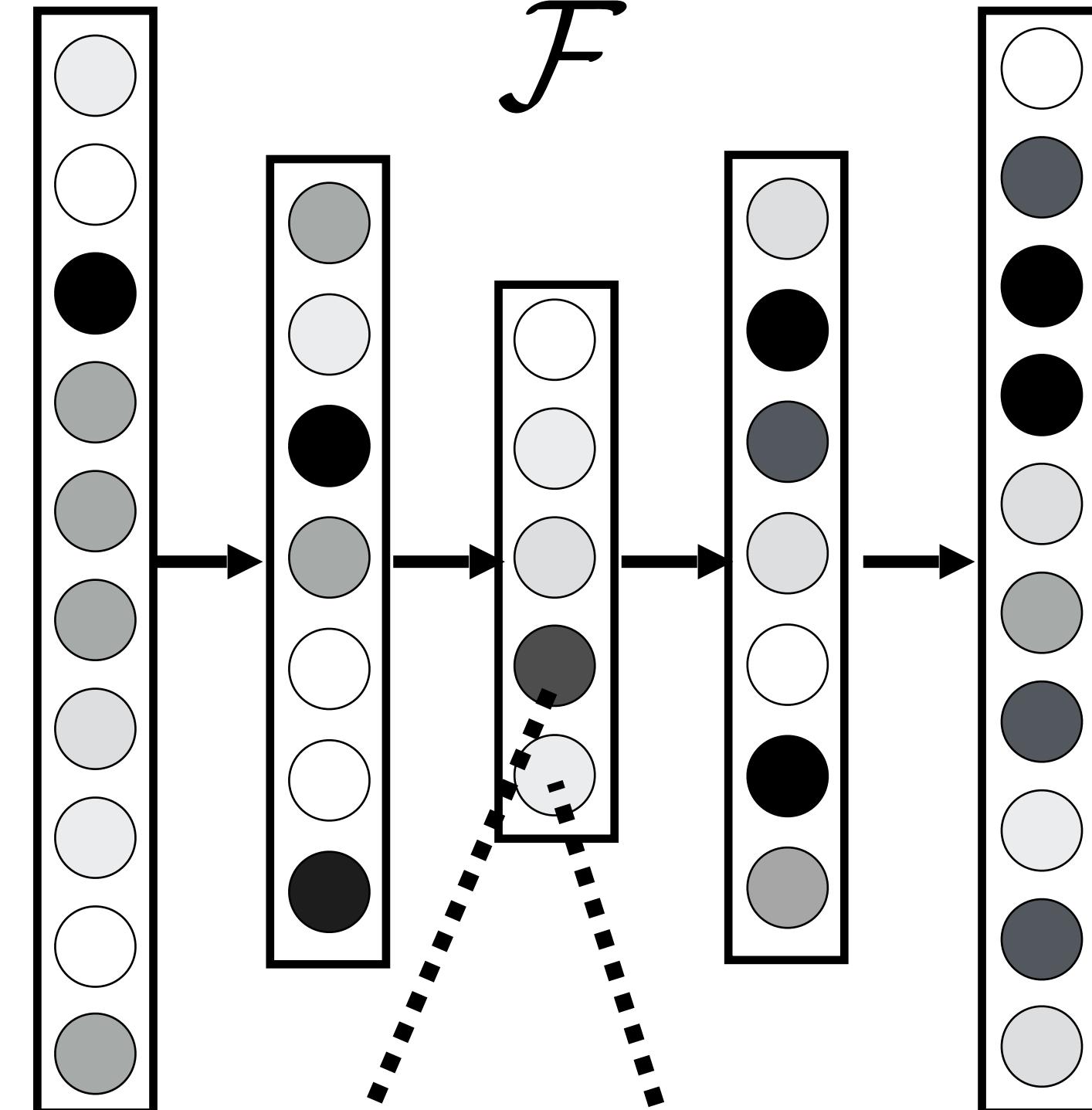
Learning Signal: **Forced Compression**

\mathbf{X}



Image

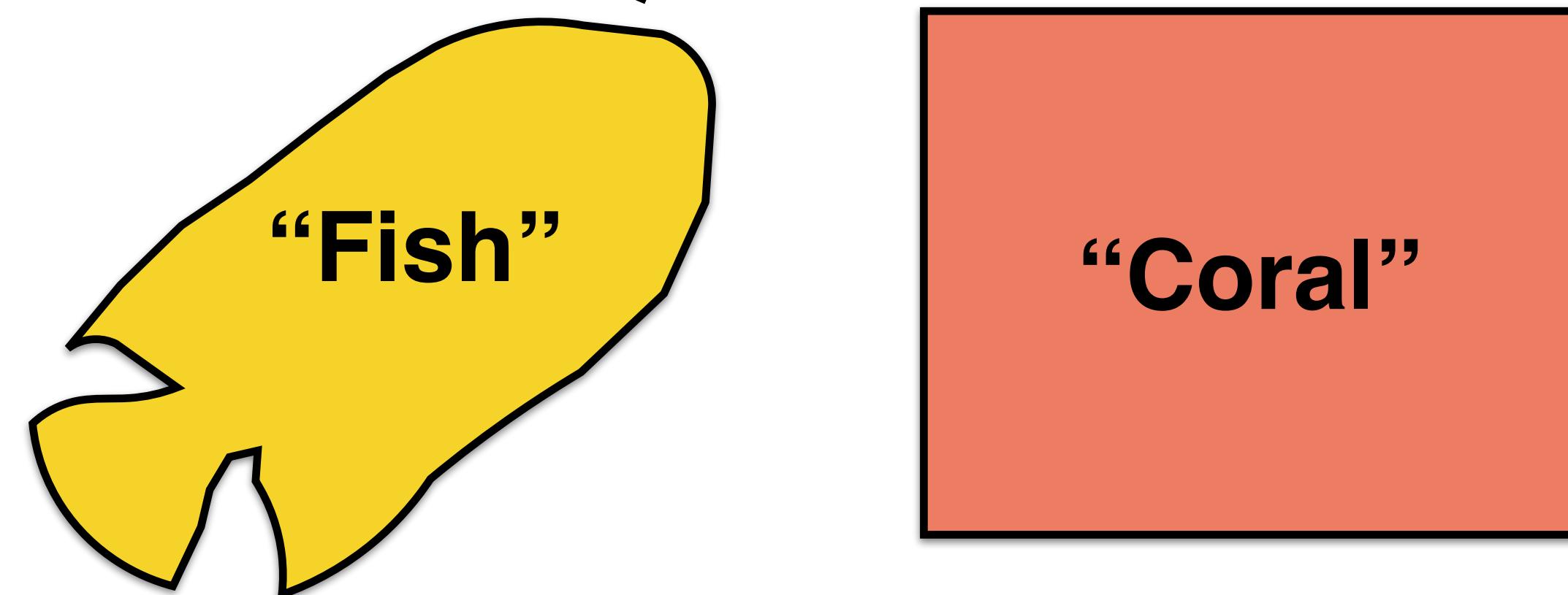
\mathcal{F}



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

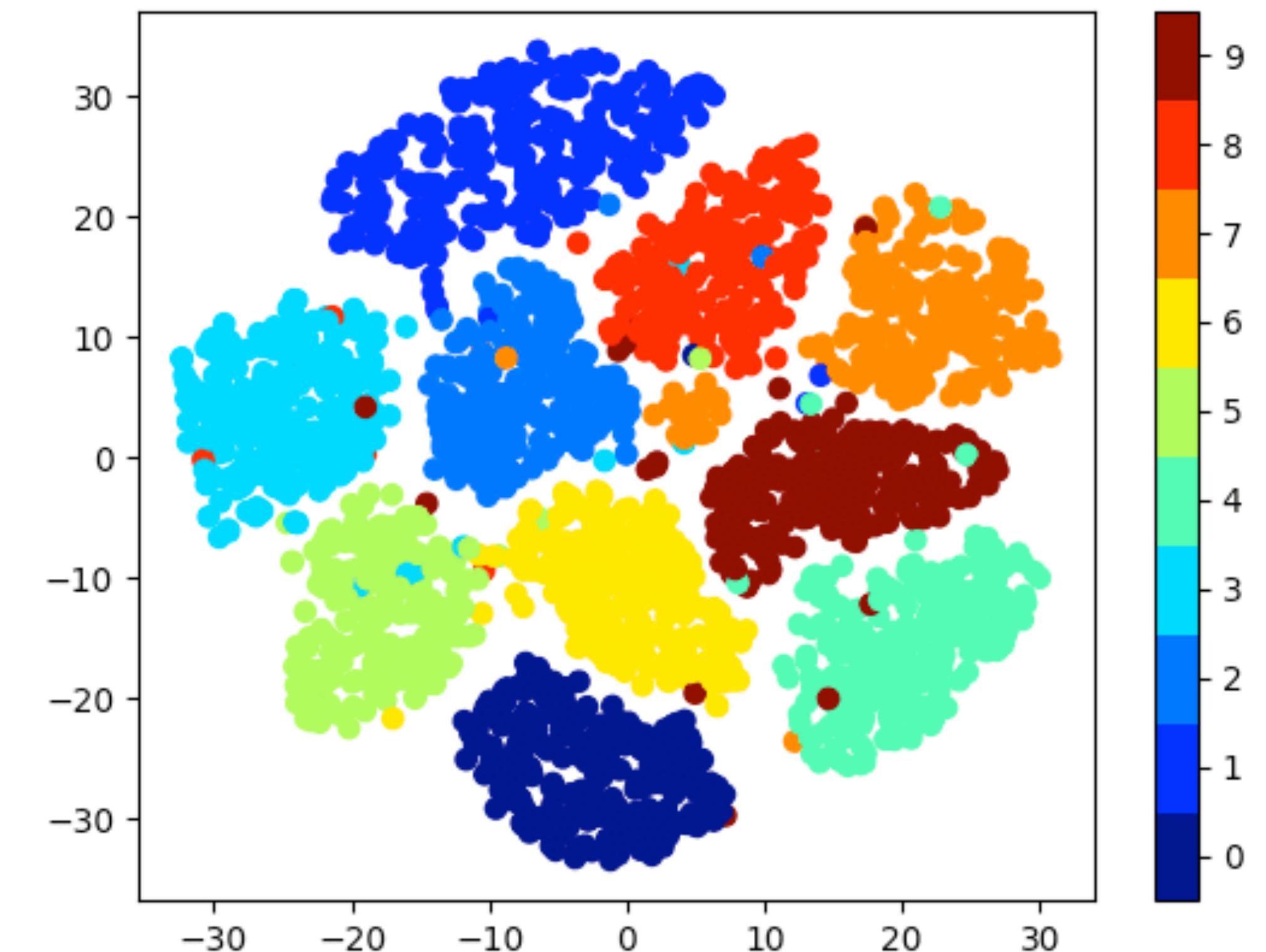


Reconstructed
image



Learning Signal: **Forced Compression**

On MNIST Digits: Clustering Latent Variables z



Self-supervised Scene Representation Learning

Latent 3D Scenes



Self-supervised Scene Representation Learning

Latent 3D Scenes

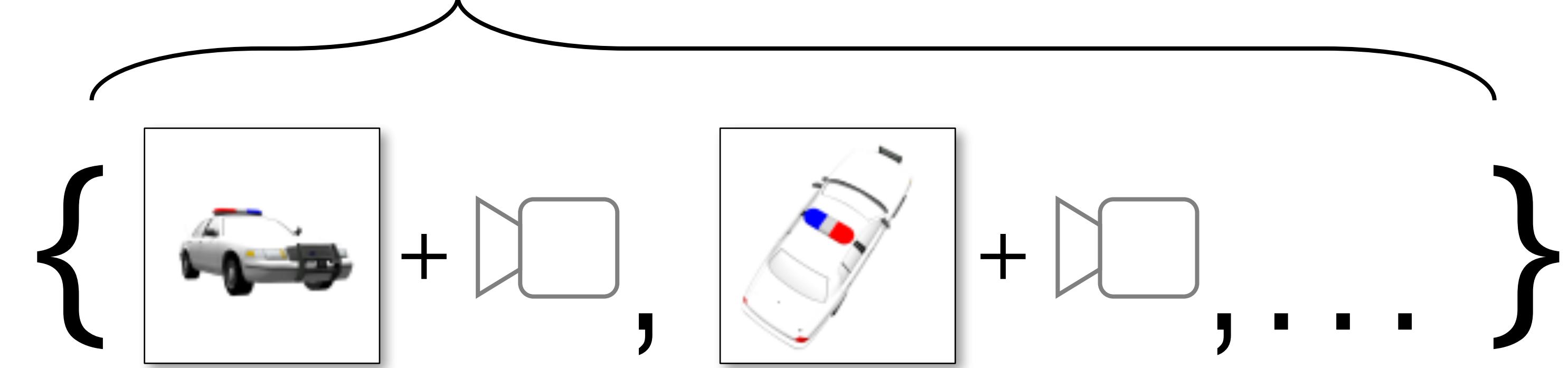


Self-supervised Scene Representation Learning

Latent 3D Scenes



Observations
Image + Pose & Intrinsics

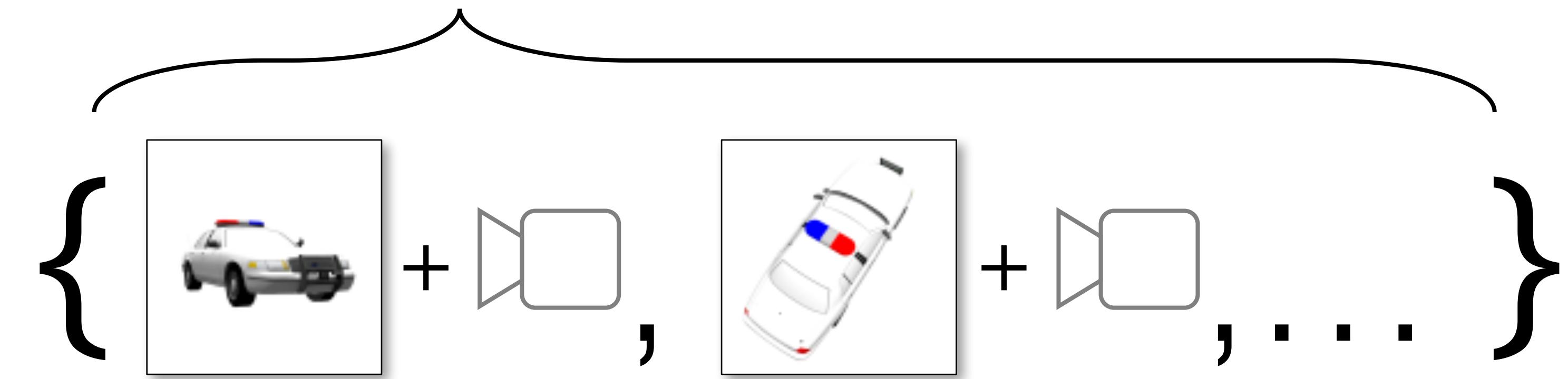


Self-supervised Scene Representation Learning

Latent 3D Scenes



Observations
Image + Pose & Intrinsics

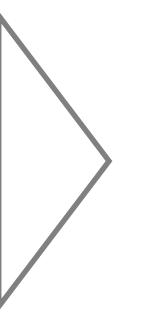


What can we learn about latent 3D scenes from observations?

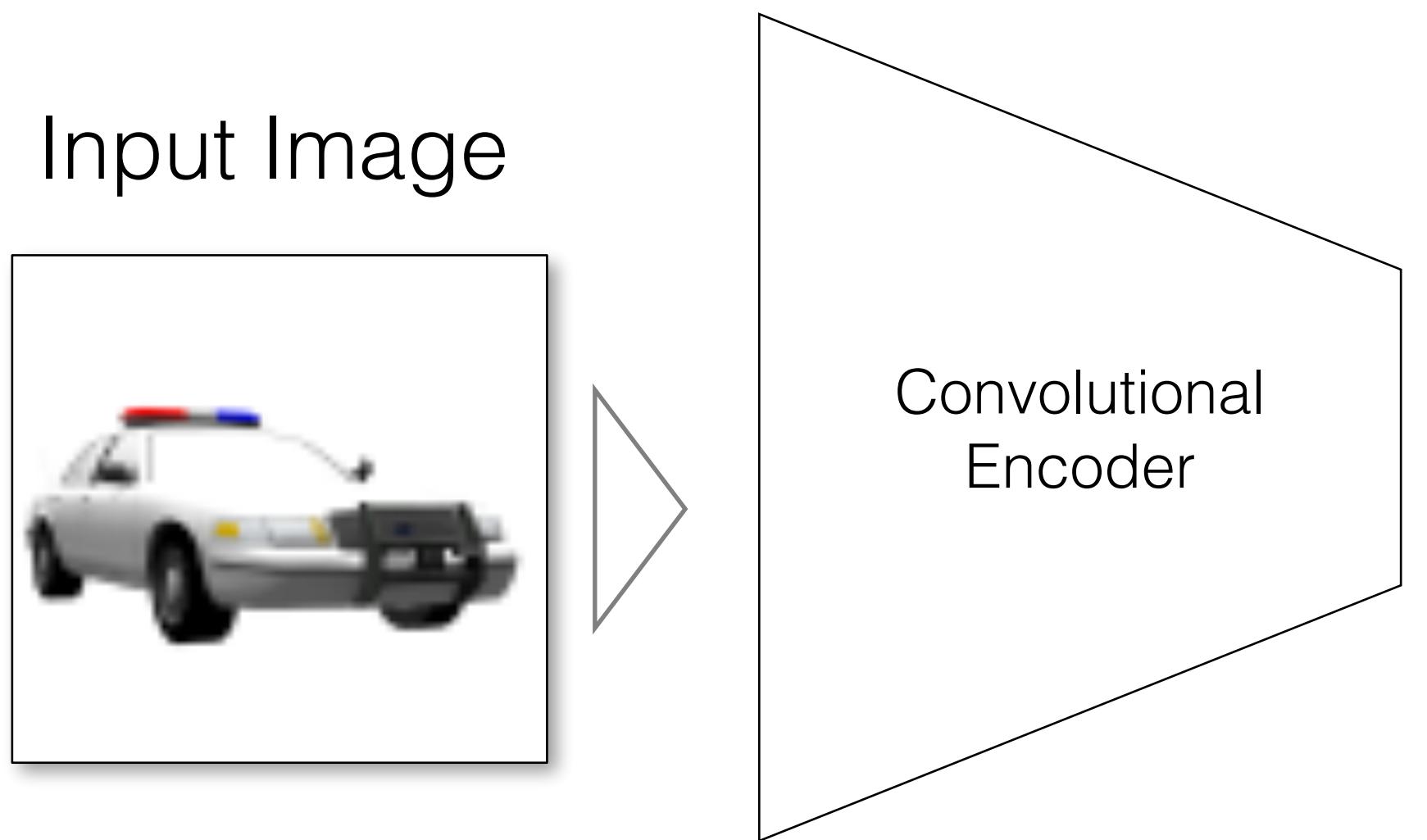
Vision: Learn rich representations just by watching video!

2D baseline: Auto-Encoder-Like model

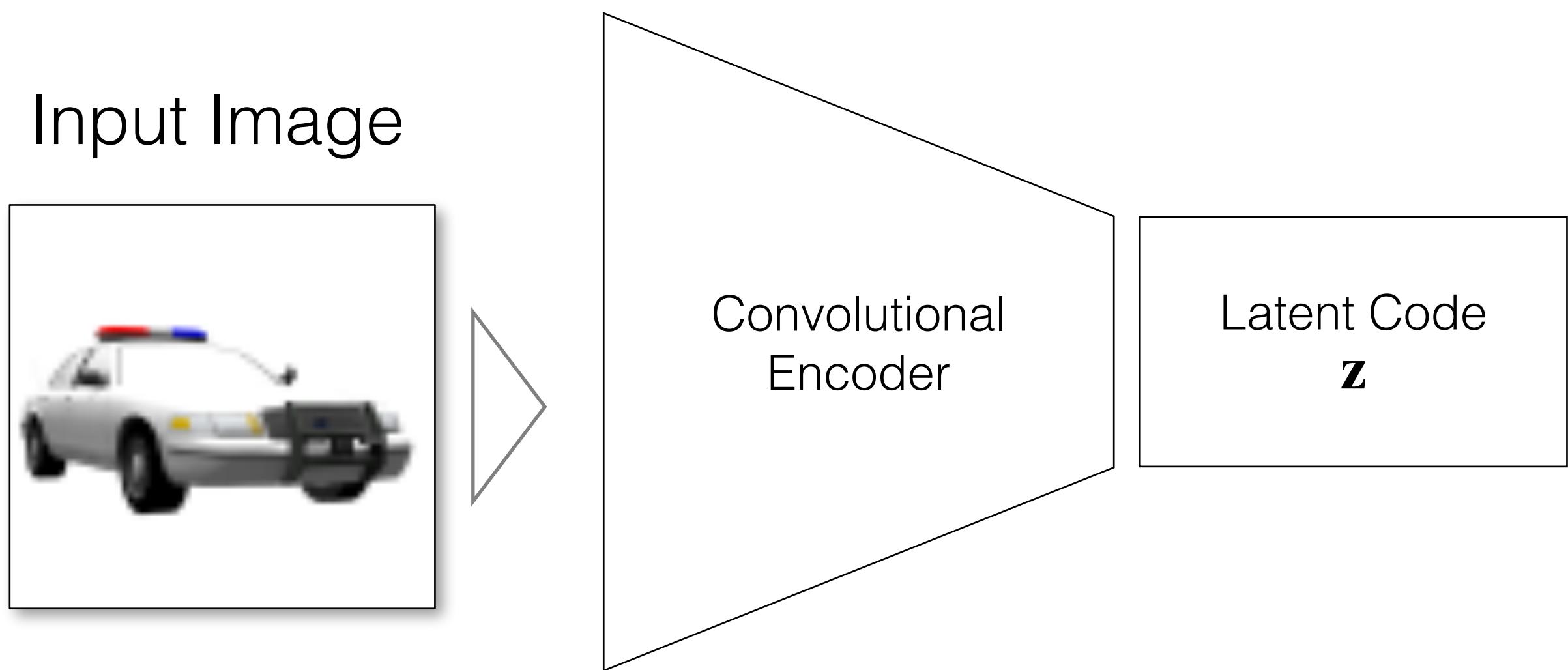
Input Image



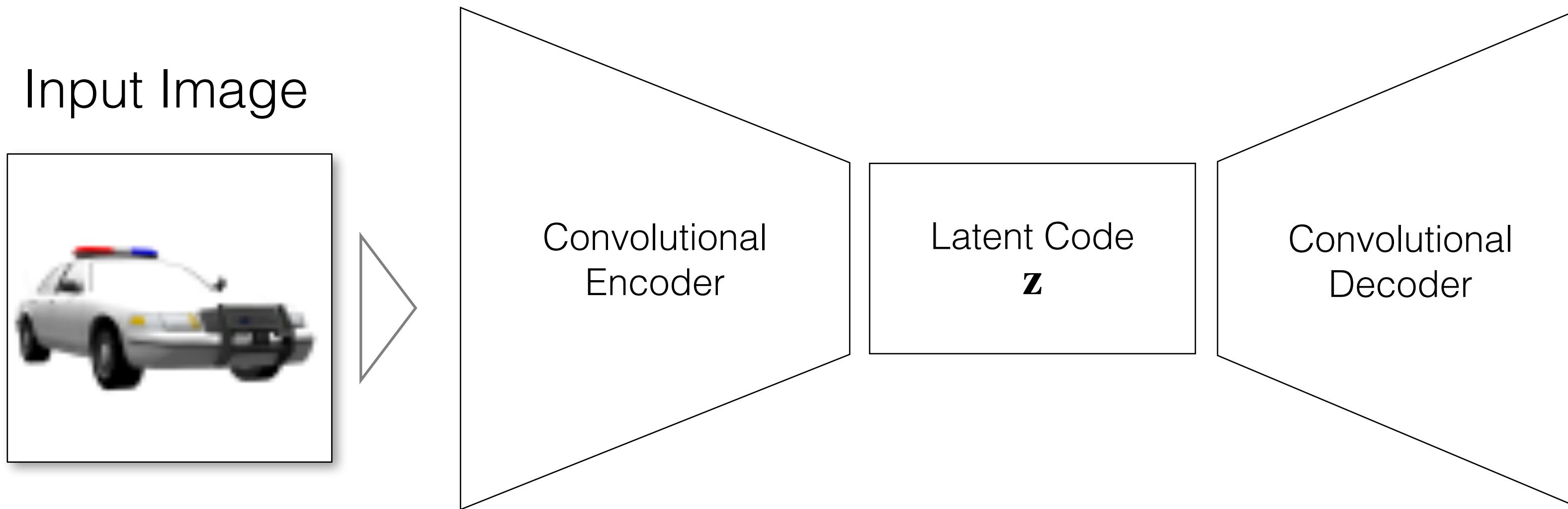
2D baseline: Auto-Encoder-Like model



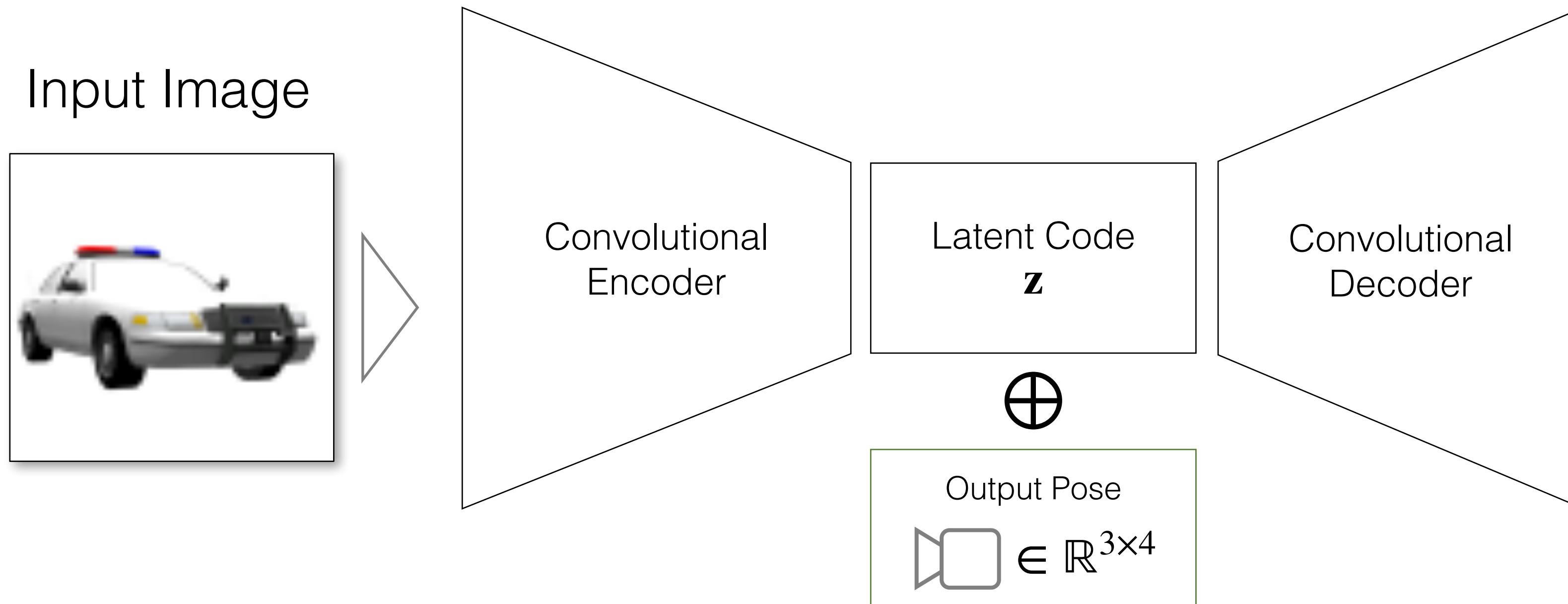
2D baseline: Auto-Encoder-Like model



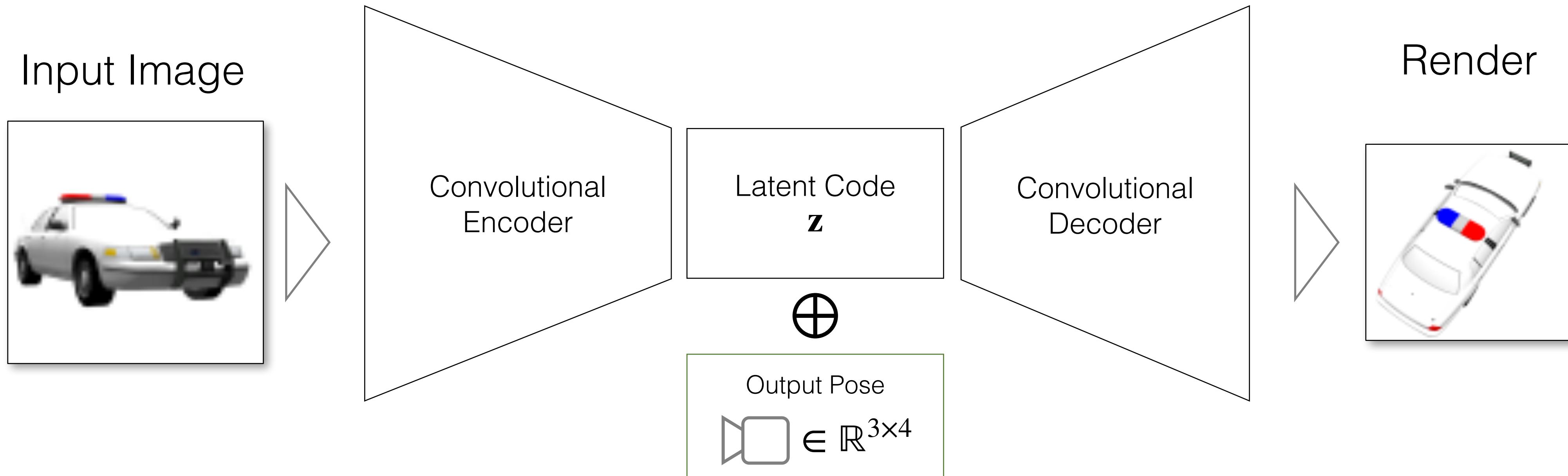
2D baseline: Auto-Encoder-Like model



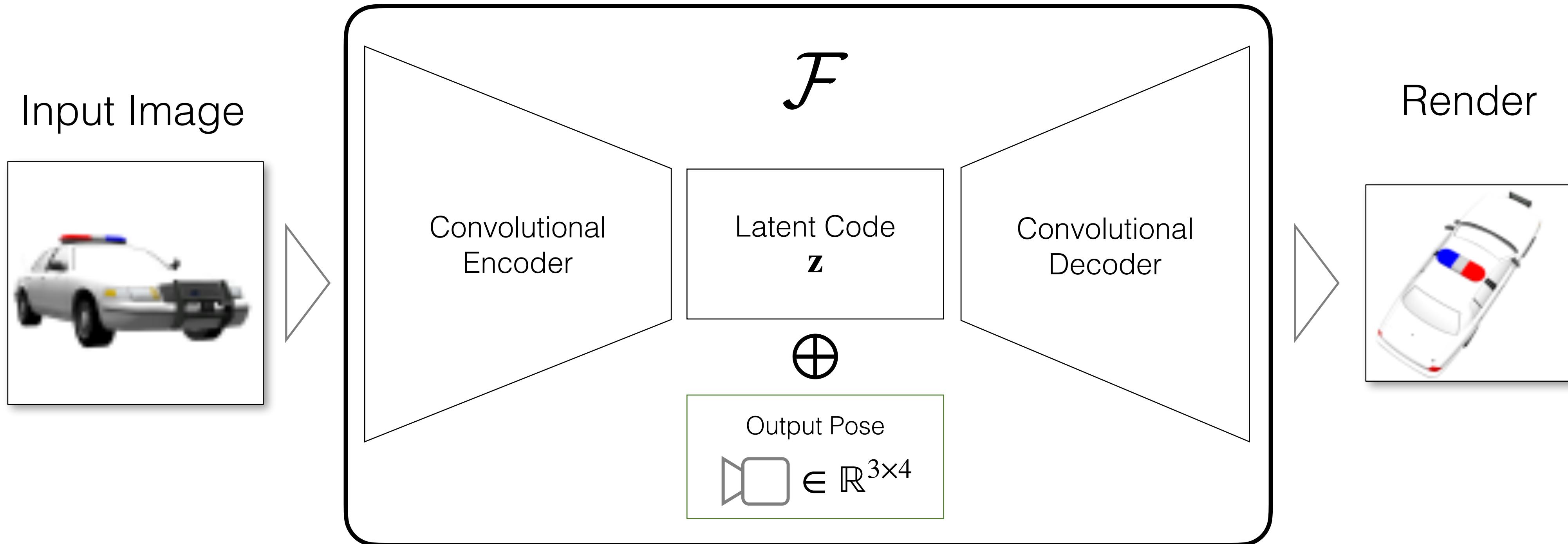
2D baseline: Auto-Encoder-Like model



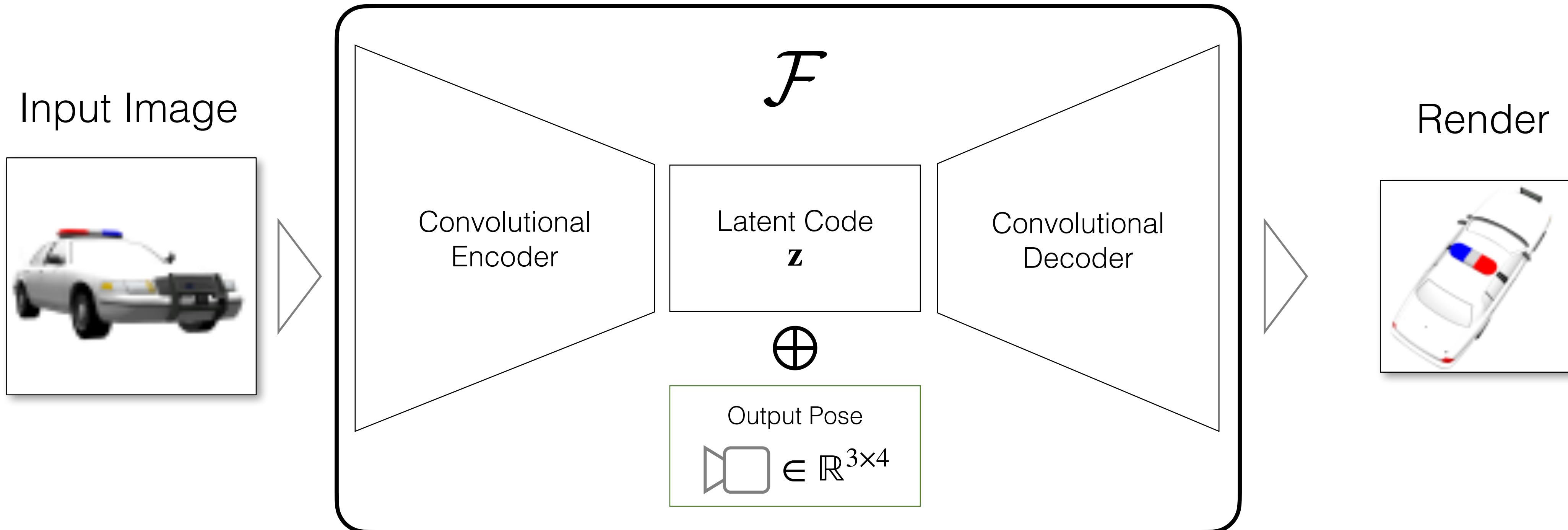
2D baseline: Auto-Encoder-Like model



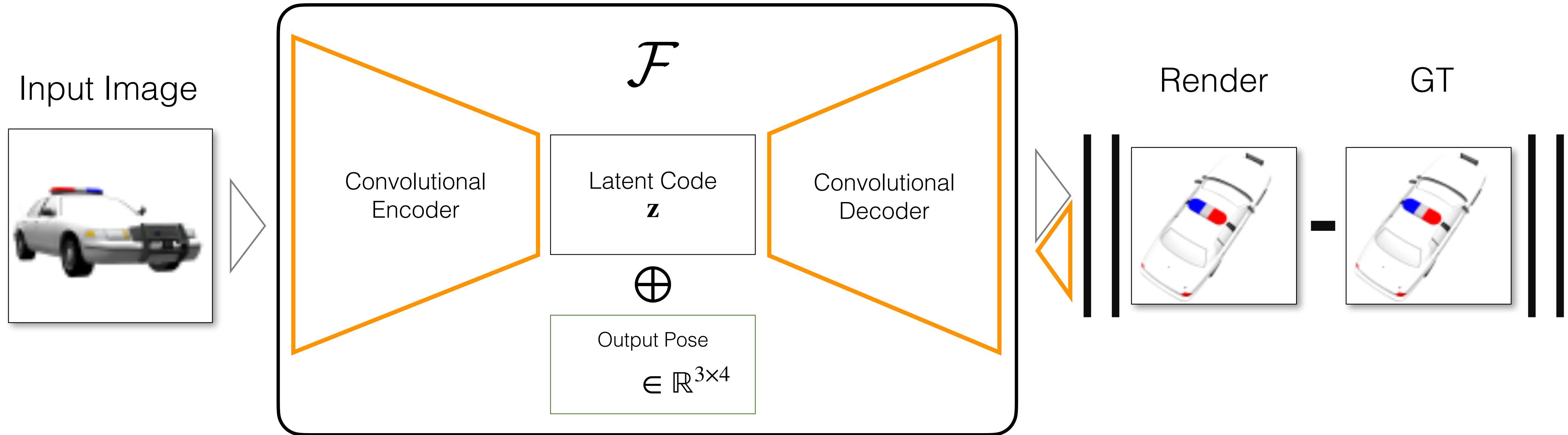
2D baseline: Auto-Encoder-Like model



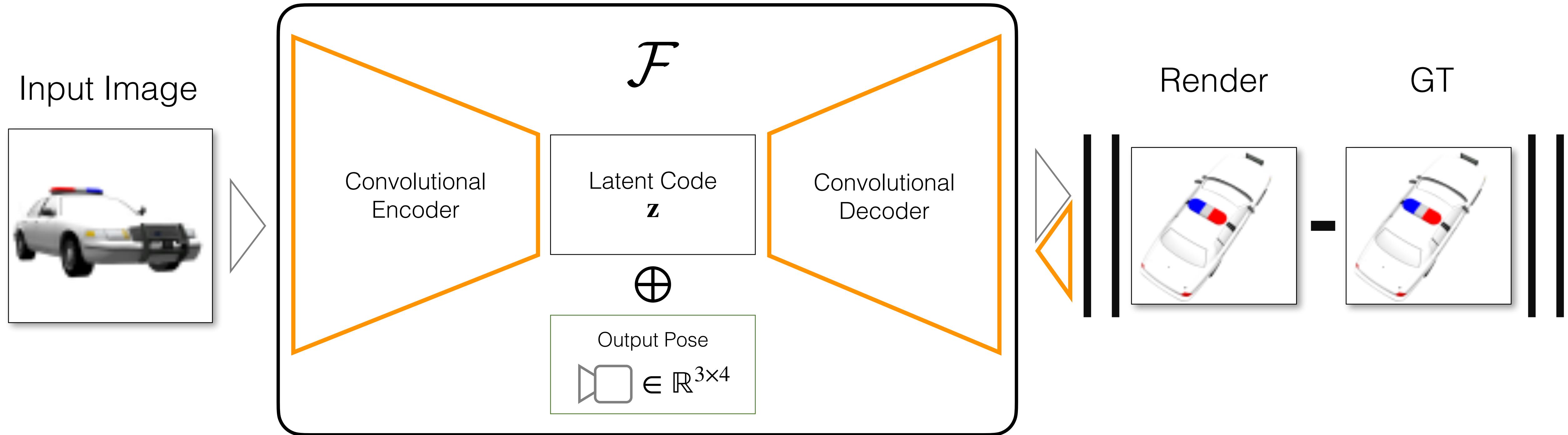
2D baseline: Auto-Encoder-Like model



2D baseline: Auto-Encoder-Like model



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.



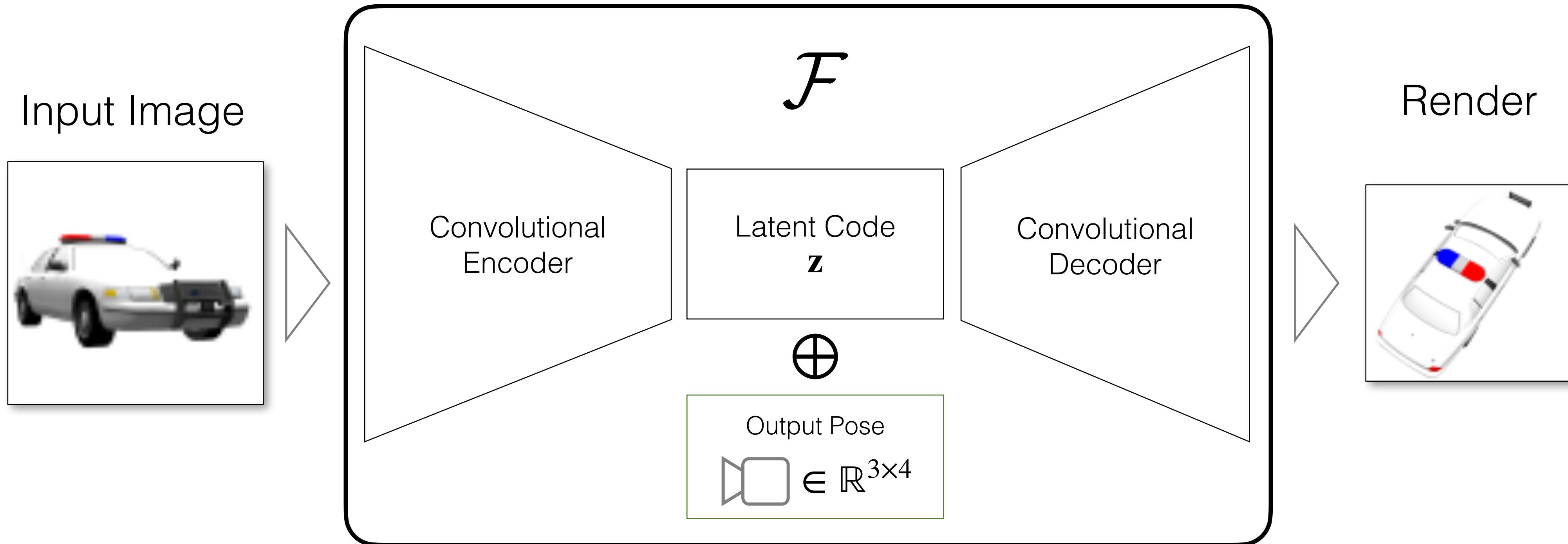
Infinite number of possible image-to-image functions to explain training set.

Only one of them is correct!

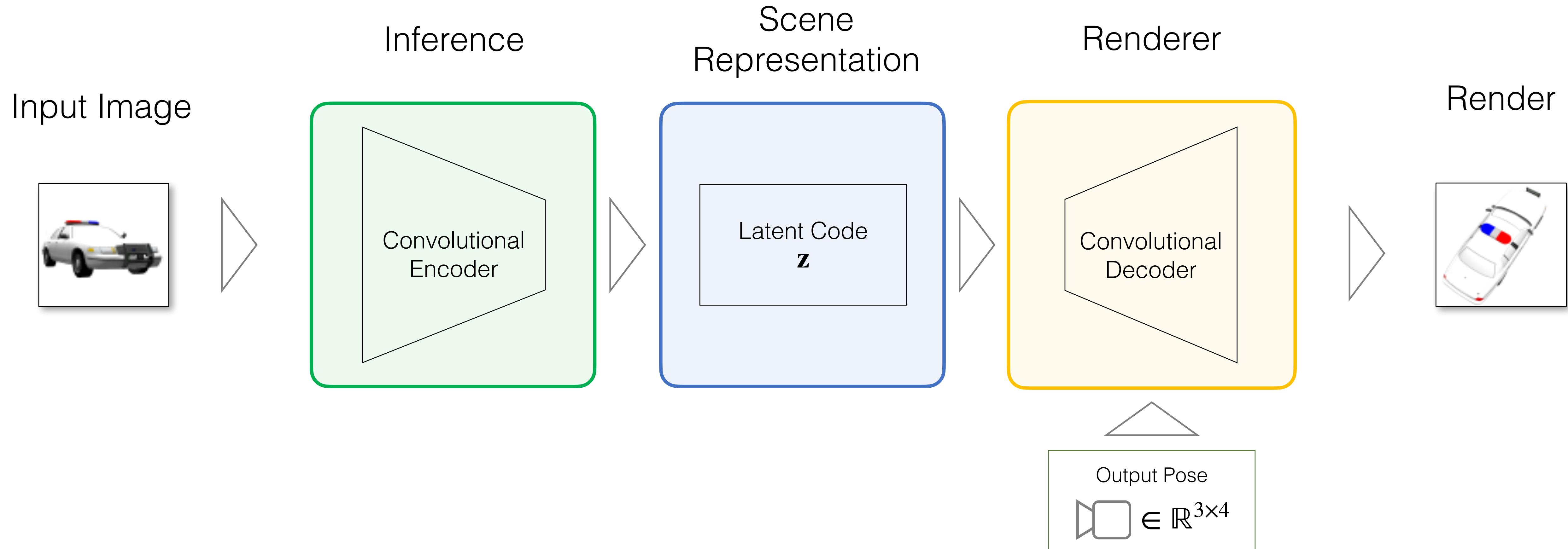
CNN does not have any 3D inductive bias - works by “matching” 2D features.

How to fix?

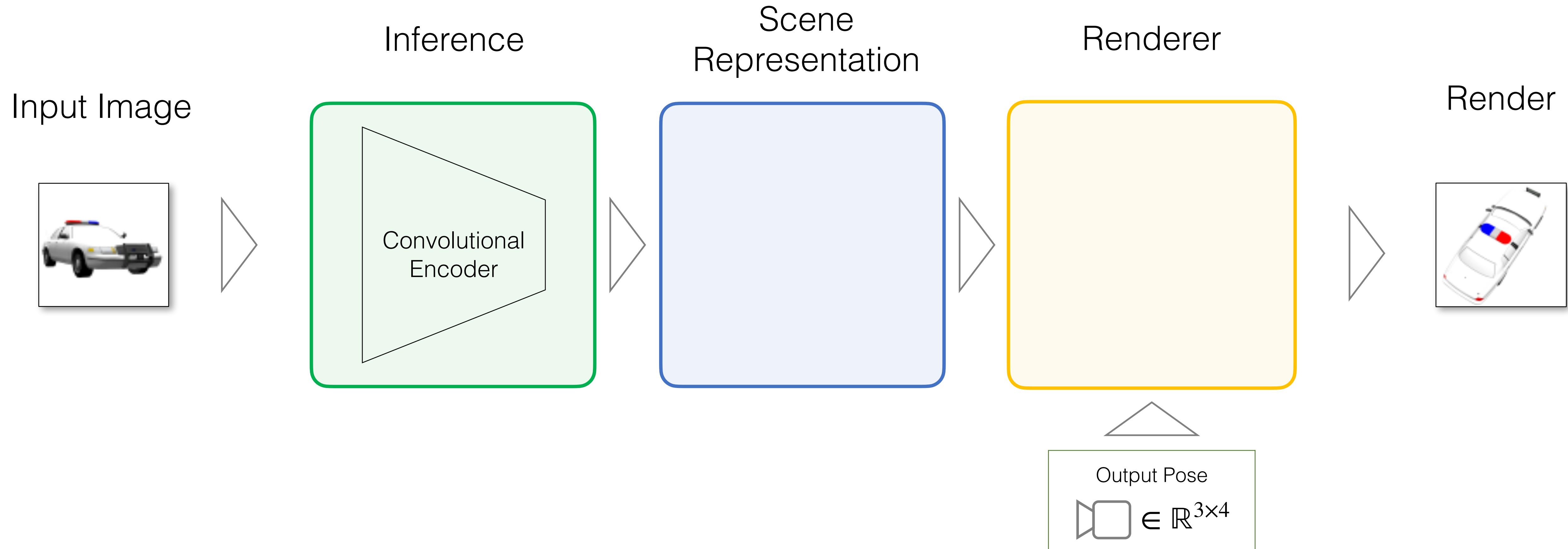
2D baseline: Auto-Encoder-Like model



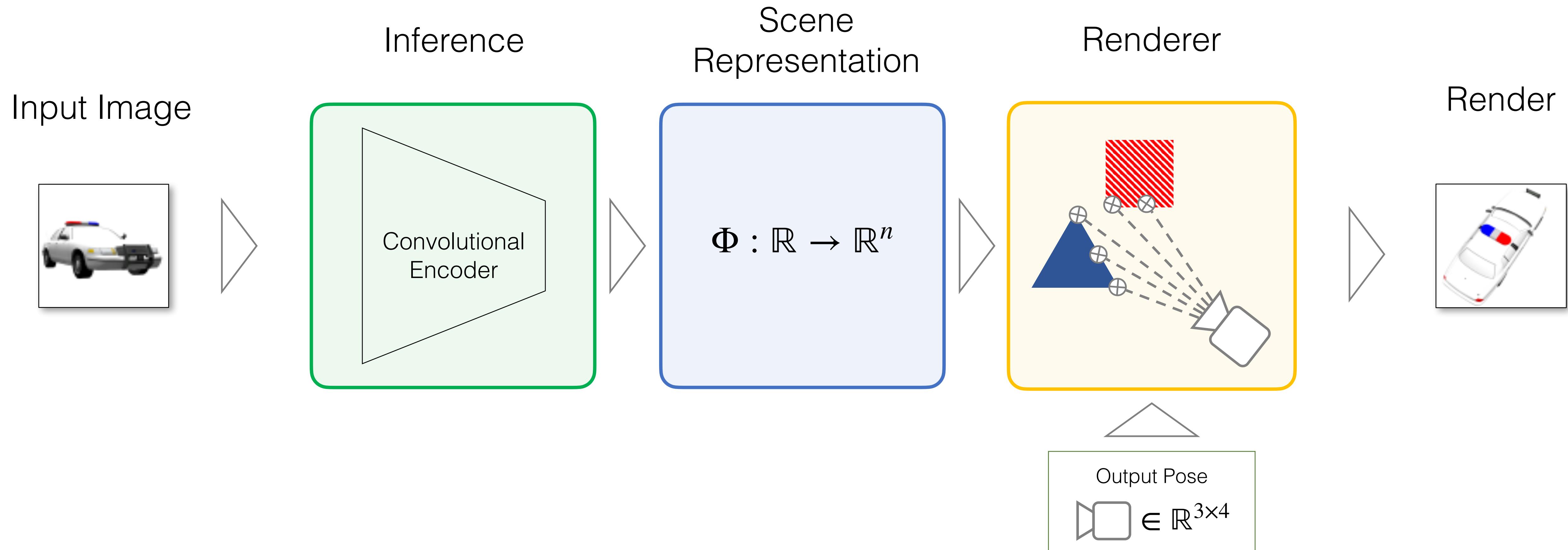
2D baseline: Auto-Encoder-Like model



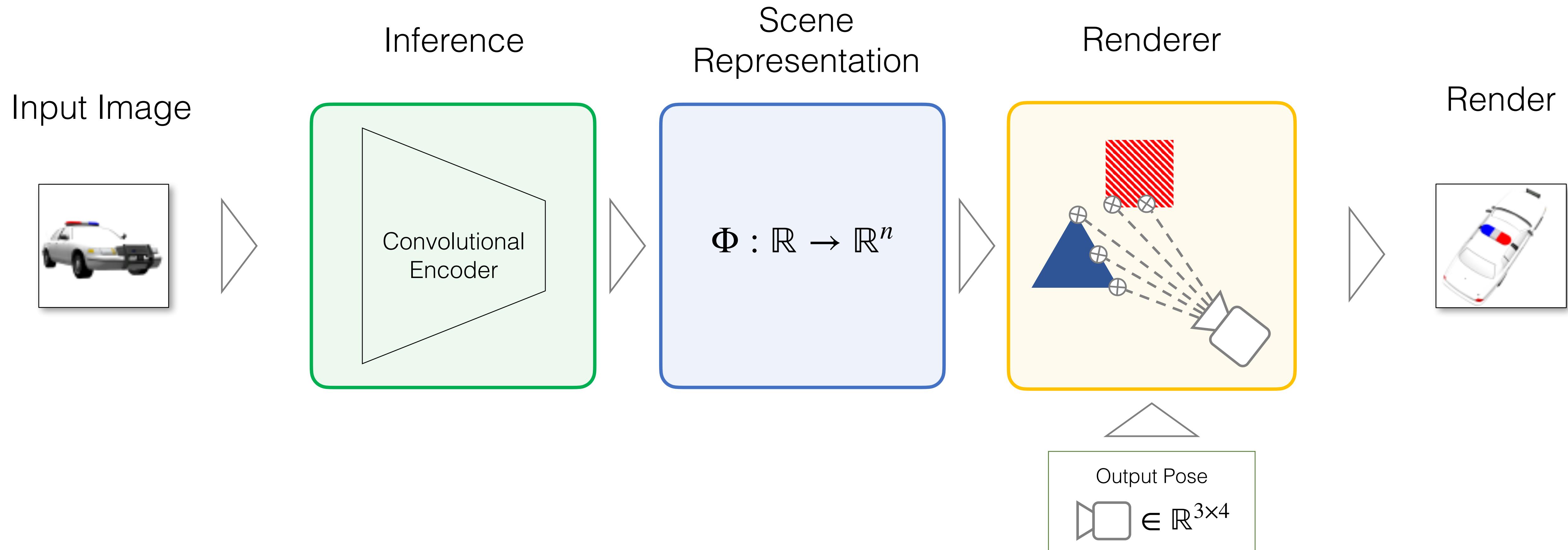
2D baseline: Auto-Encoder-Like model



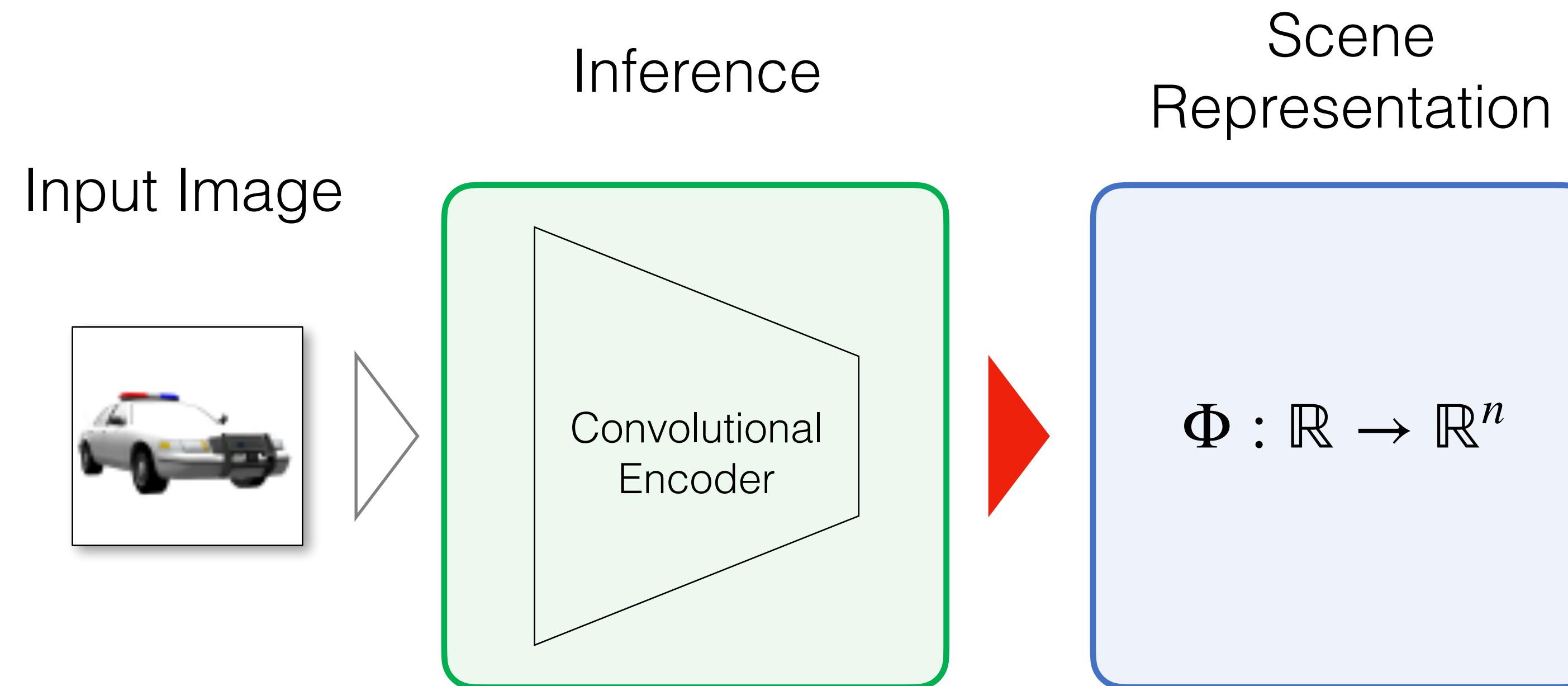
3D-Structured Decoder



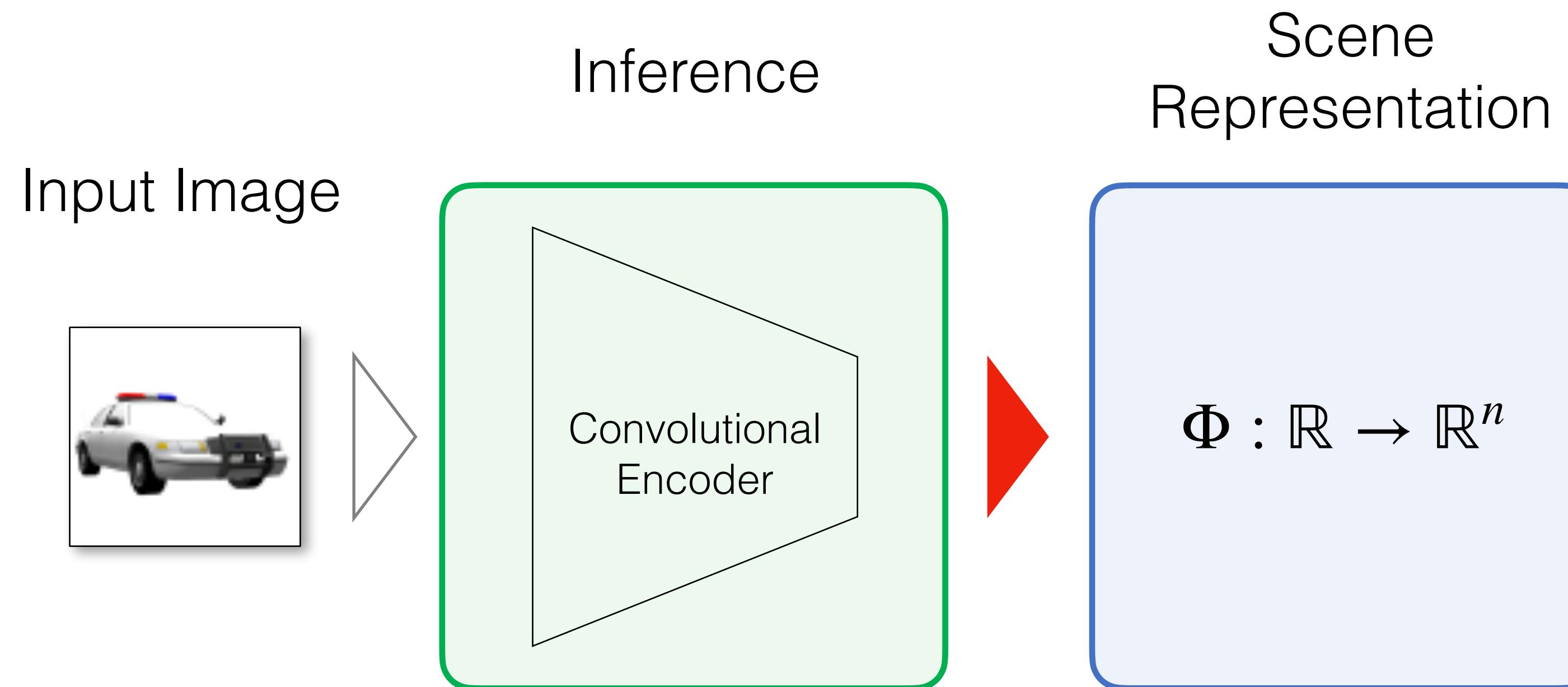
3D-Structured Decoder



3D-Structured Decoder

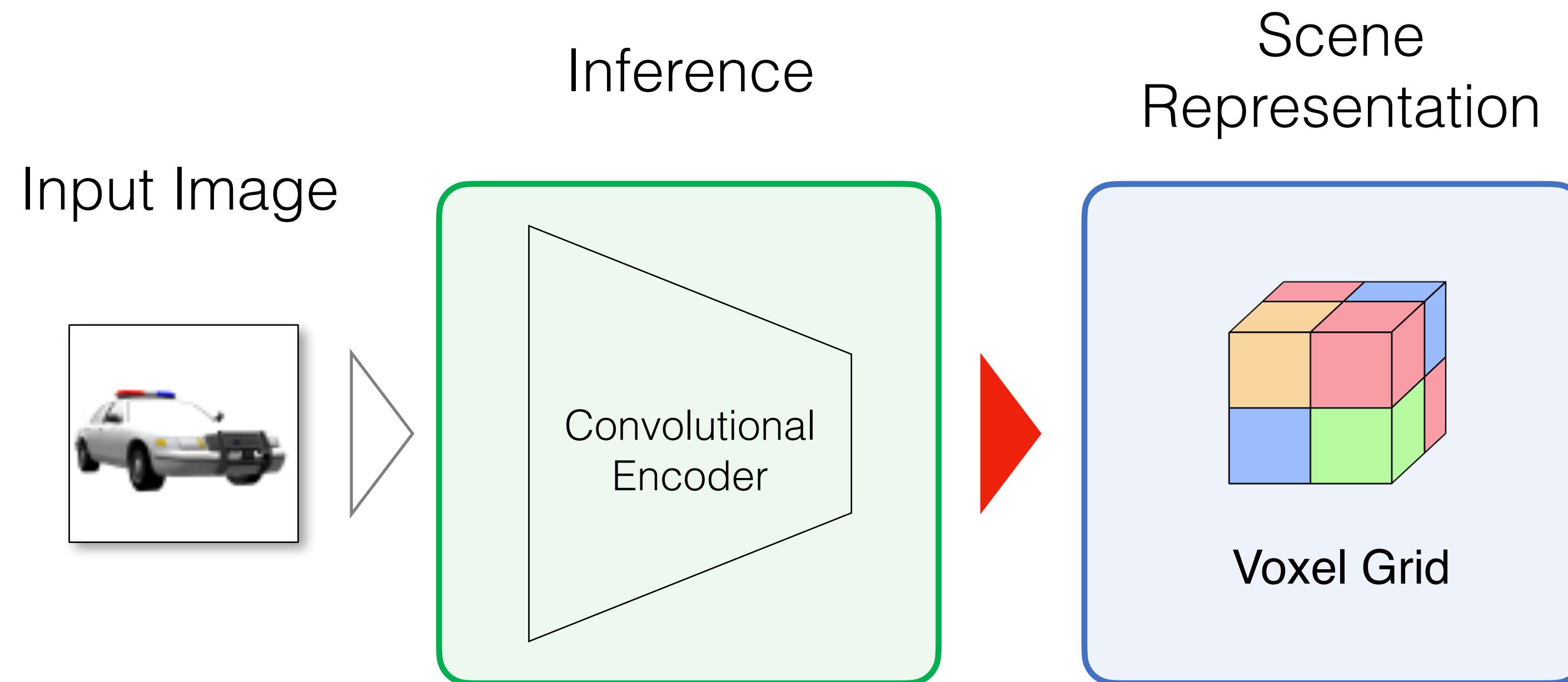


3D-Structured Decoder

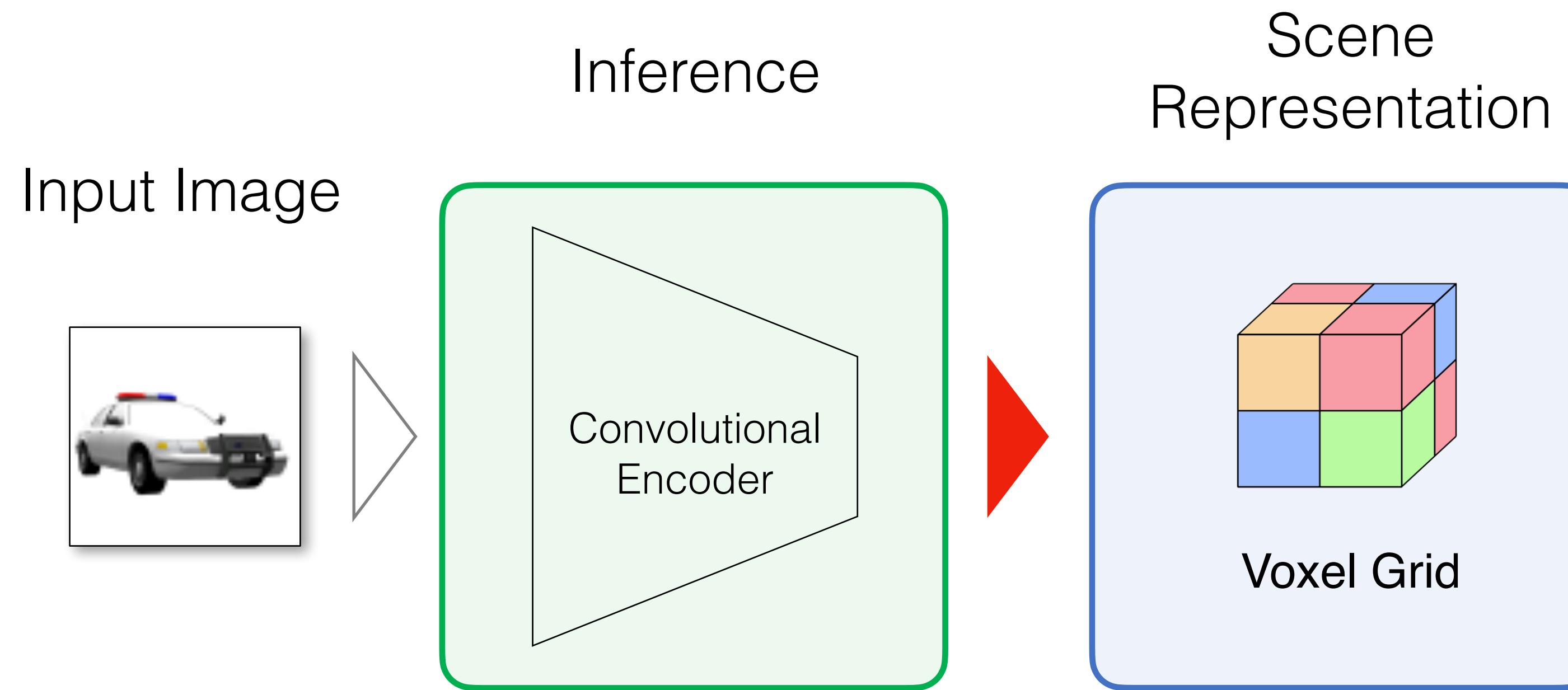


How can we have the convolution encoder “output” the Scene Representation?

3D-Structured Decoder

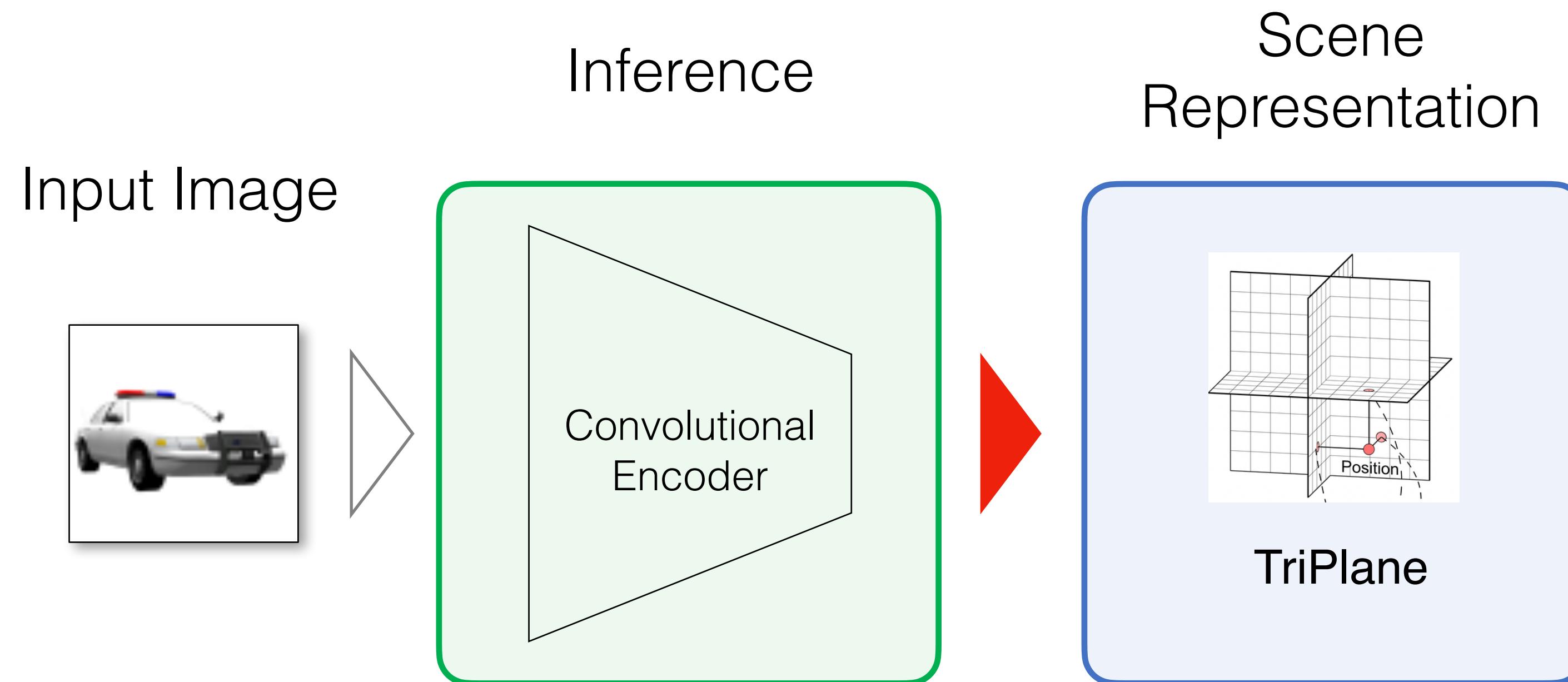


3D-Structured Decoder

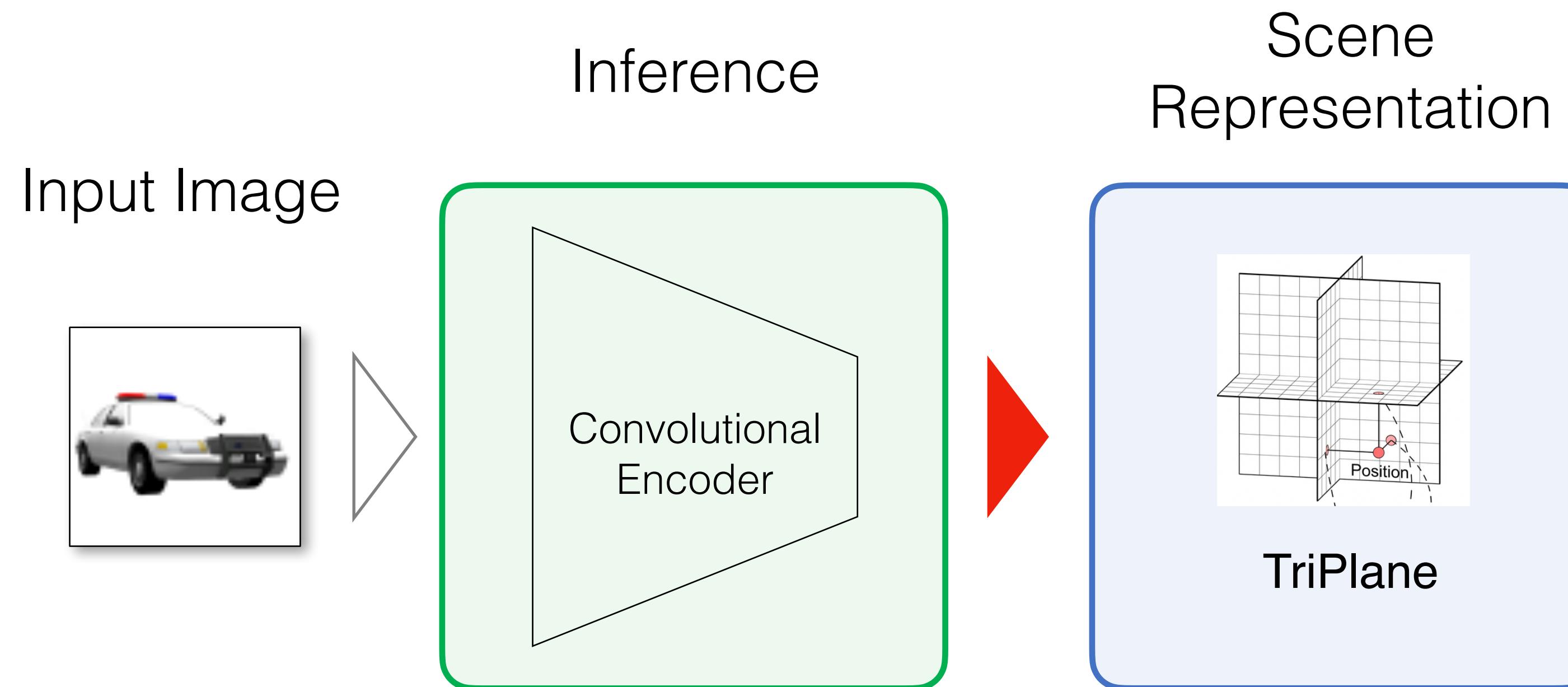


Just predict *all the parameters of that scene representation*.

3D-Structured Decoder

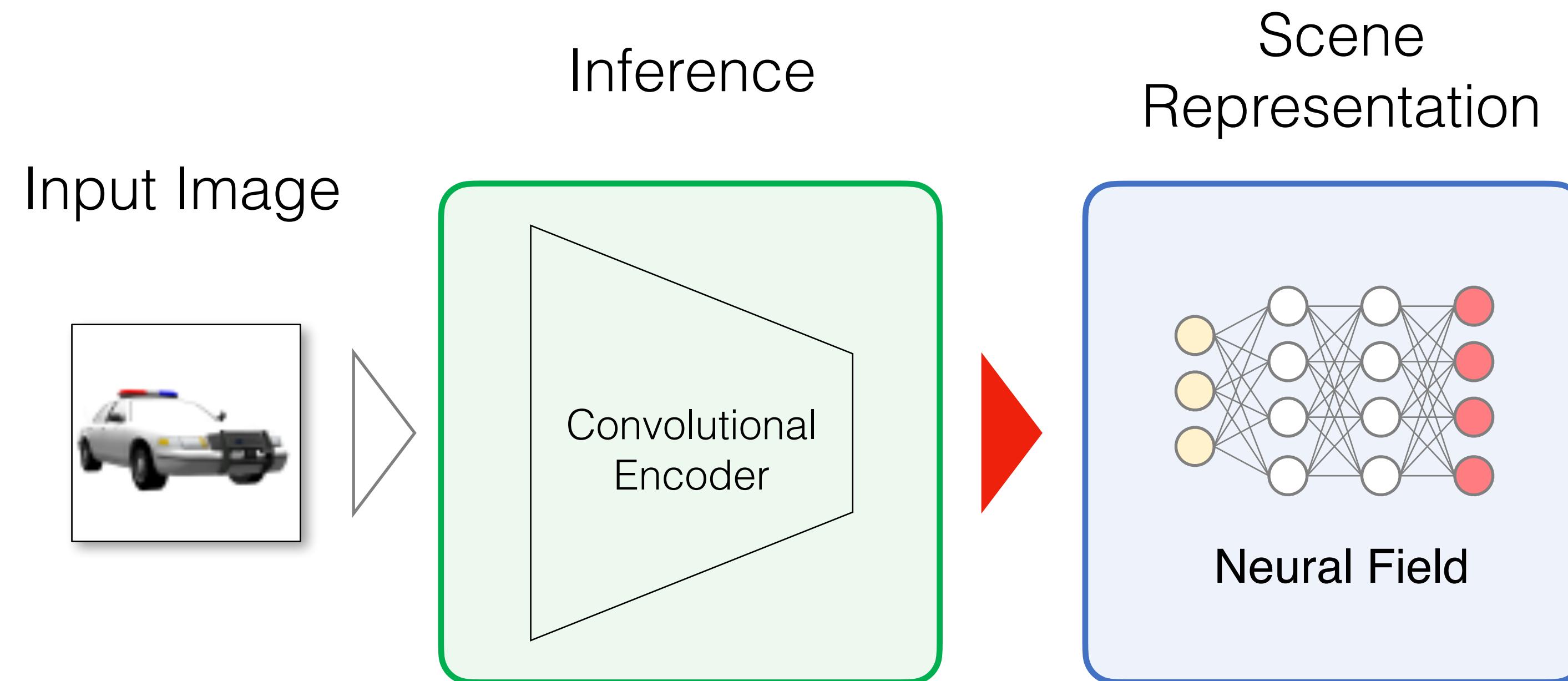


3D-Structured Decoder

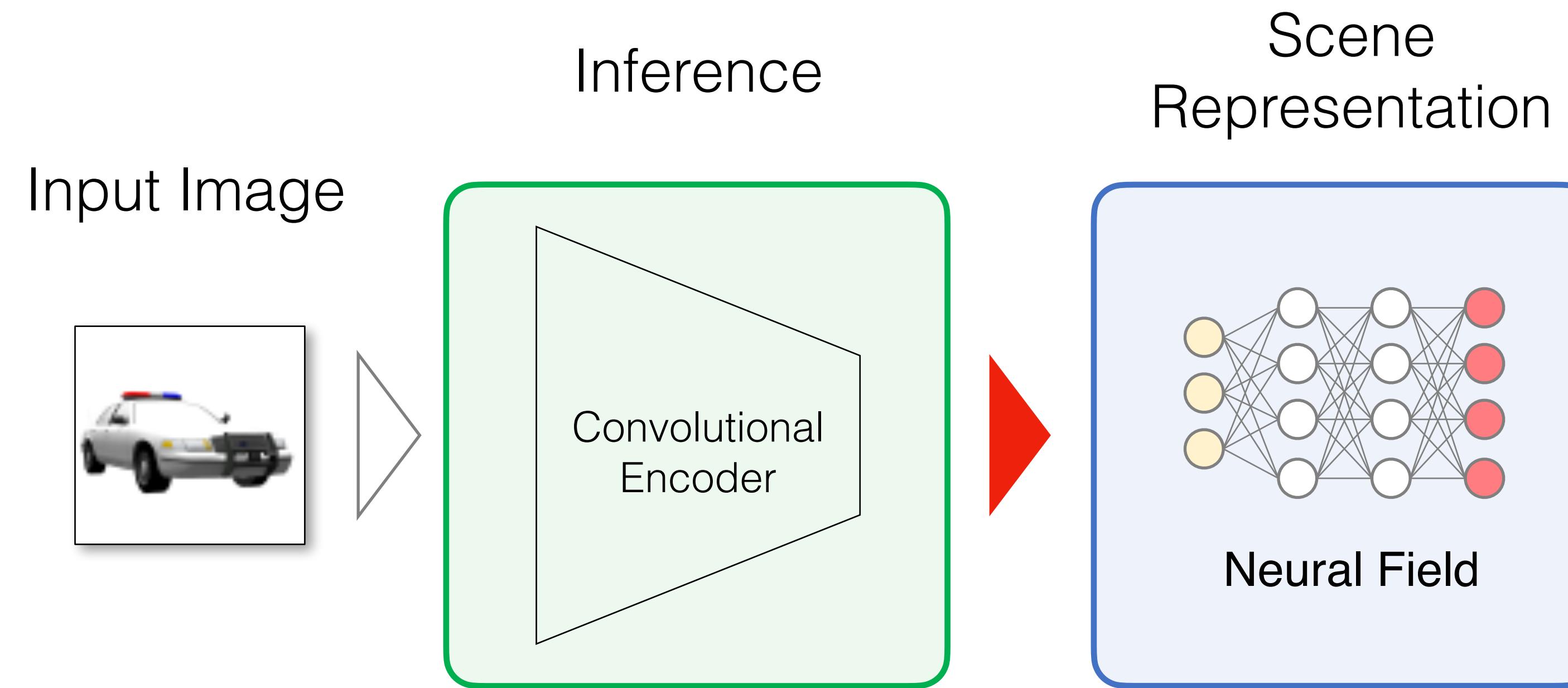


Just predict *all the parameters of that scene representation*.

3D-Structured Decoder

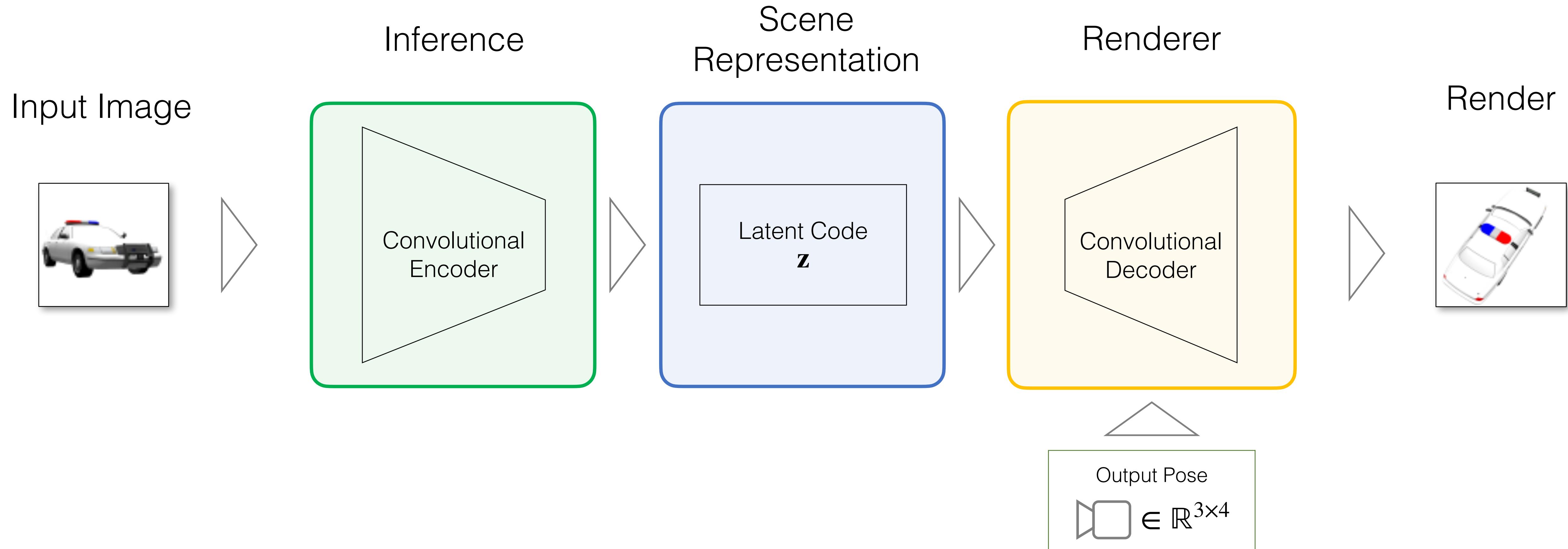


3D-Structured Decoder

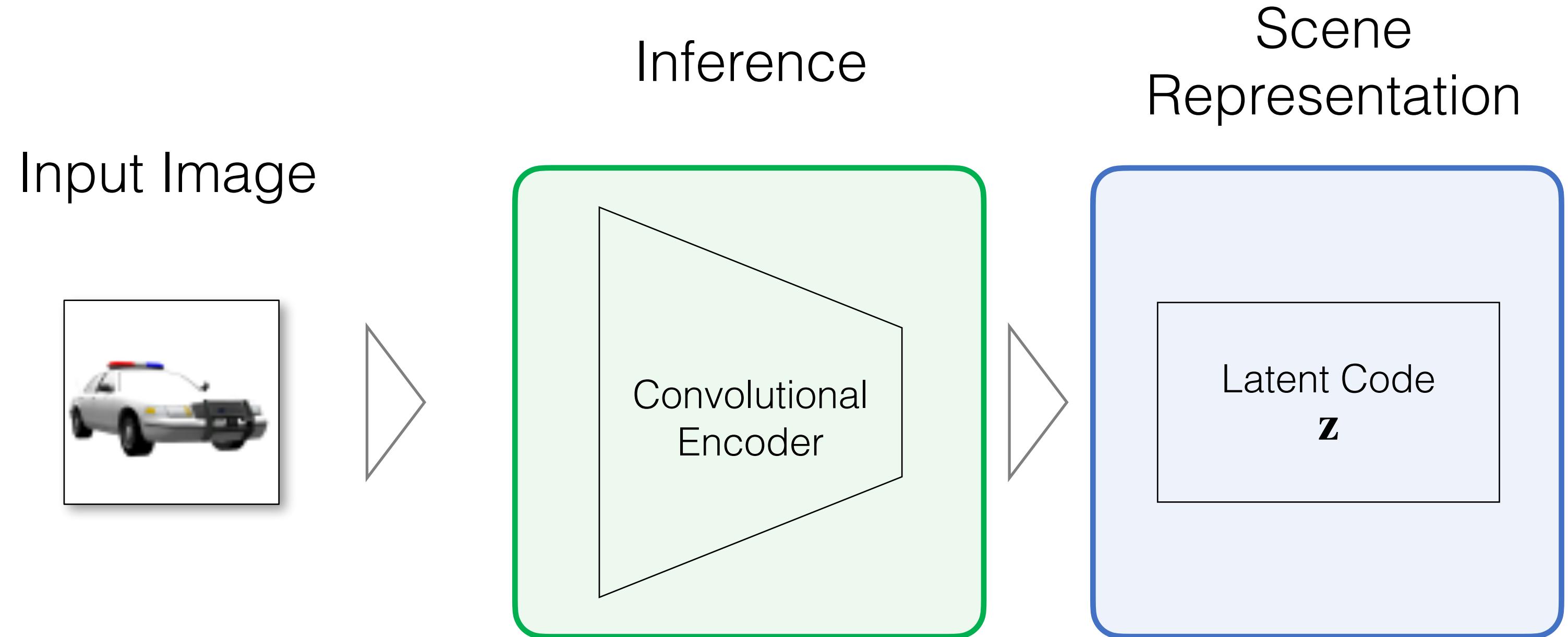


Just predict *all the parameters of that scene representation*.

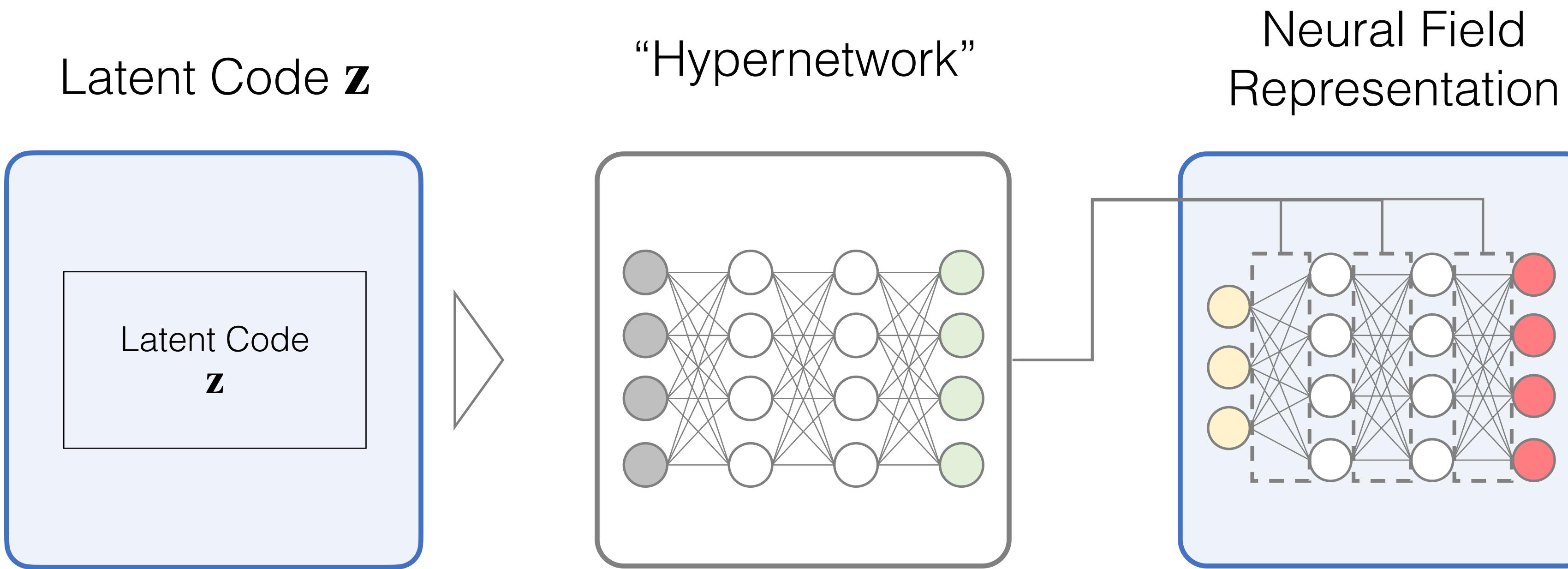
2D baseline: Auto-Encoder-Like model



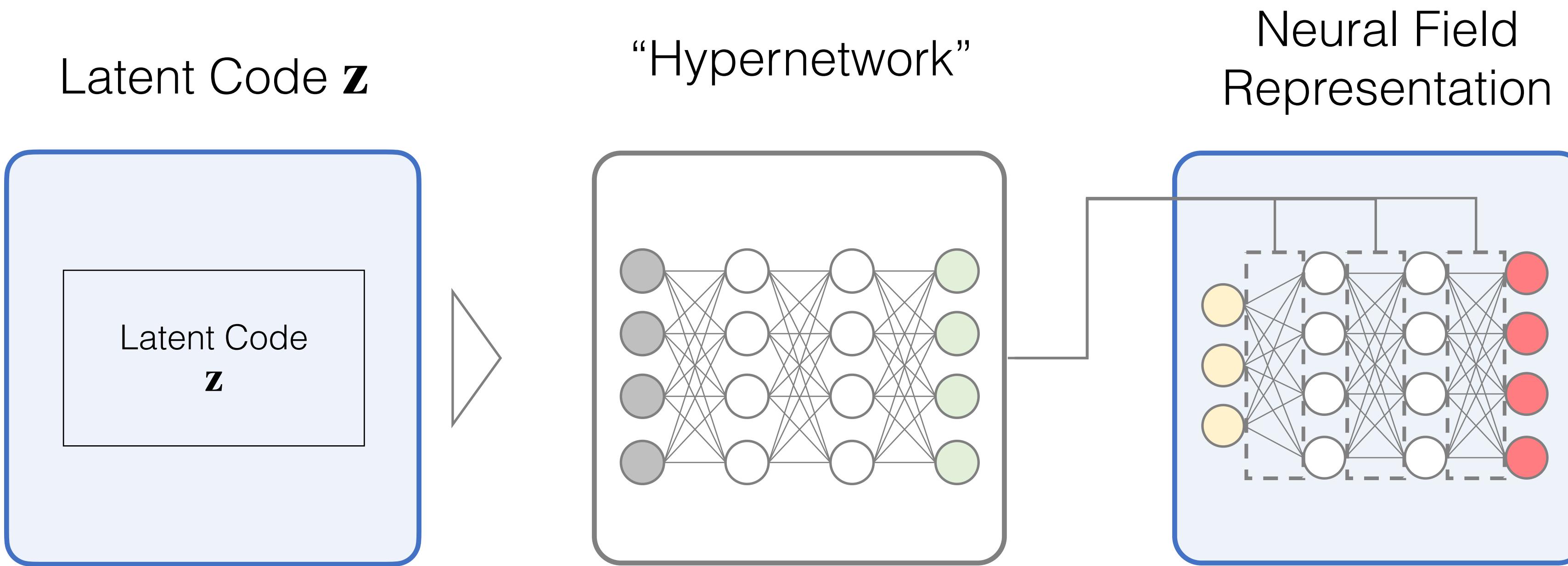
2D baseline: Auto-Encoder-Like model



Conditioning Neural Fields by predicting parameters



Conditioning Neural Fields by predicting parameters



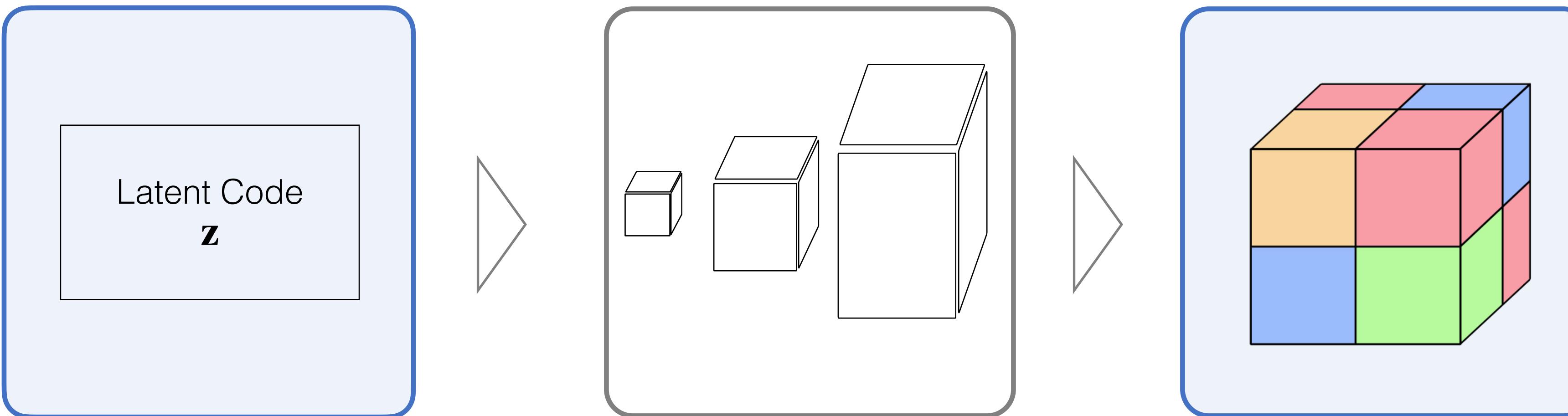
To condition neural field, can literally have MLP that takes as input \mathbf{z} and outputs *all* of the parameters (weights, biases) of the neural field.

Conditioning Voxel Grids by predicting parameters

Latent Code **z**

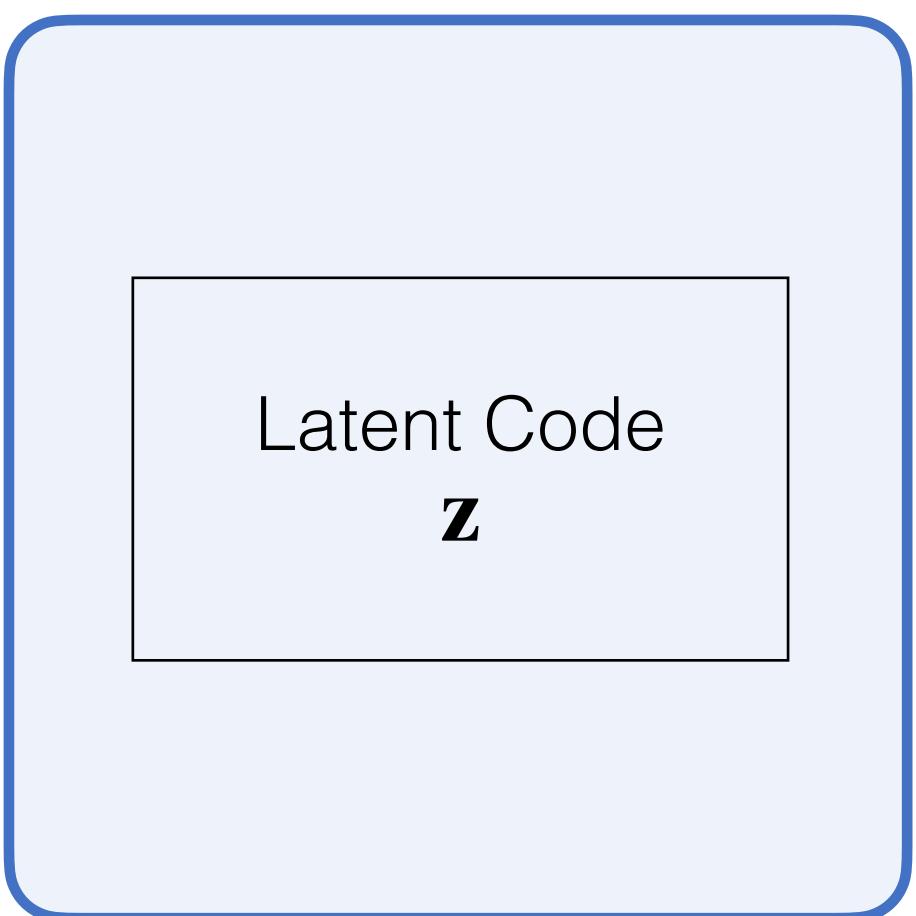
Transpose 3D
Convolutions

Voxel Grid
Representation

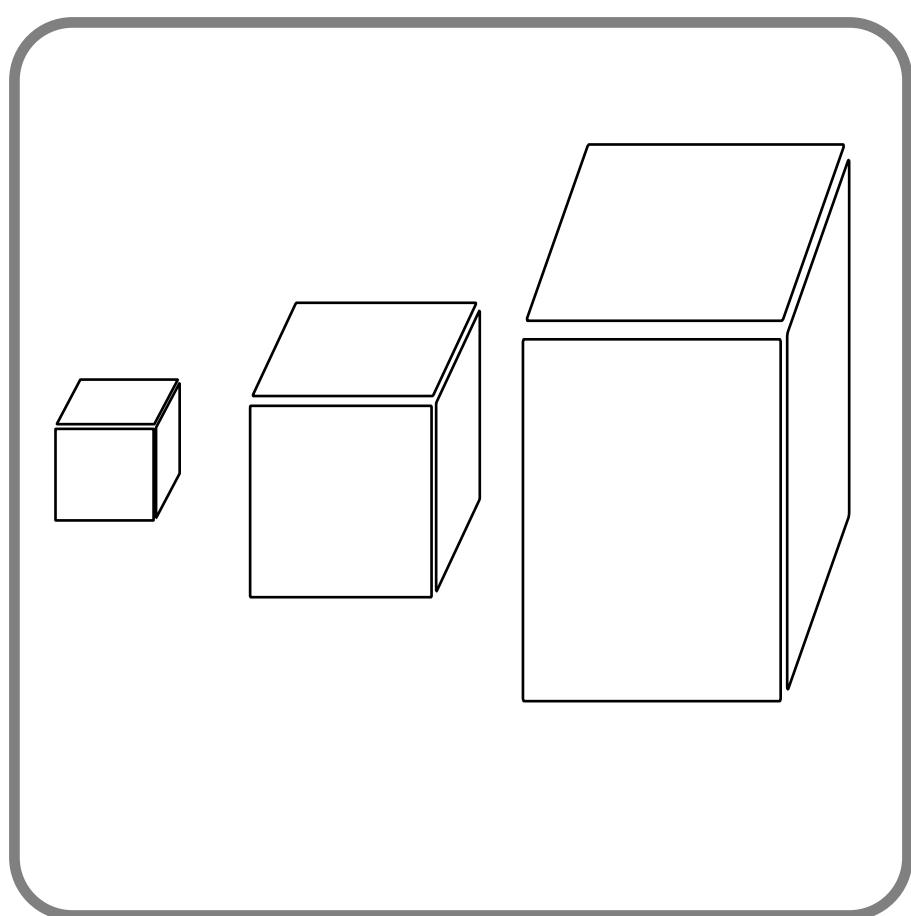


Conditioning Voxel Grids by predicting parameters

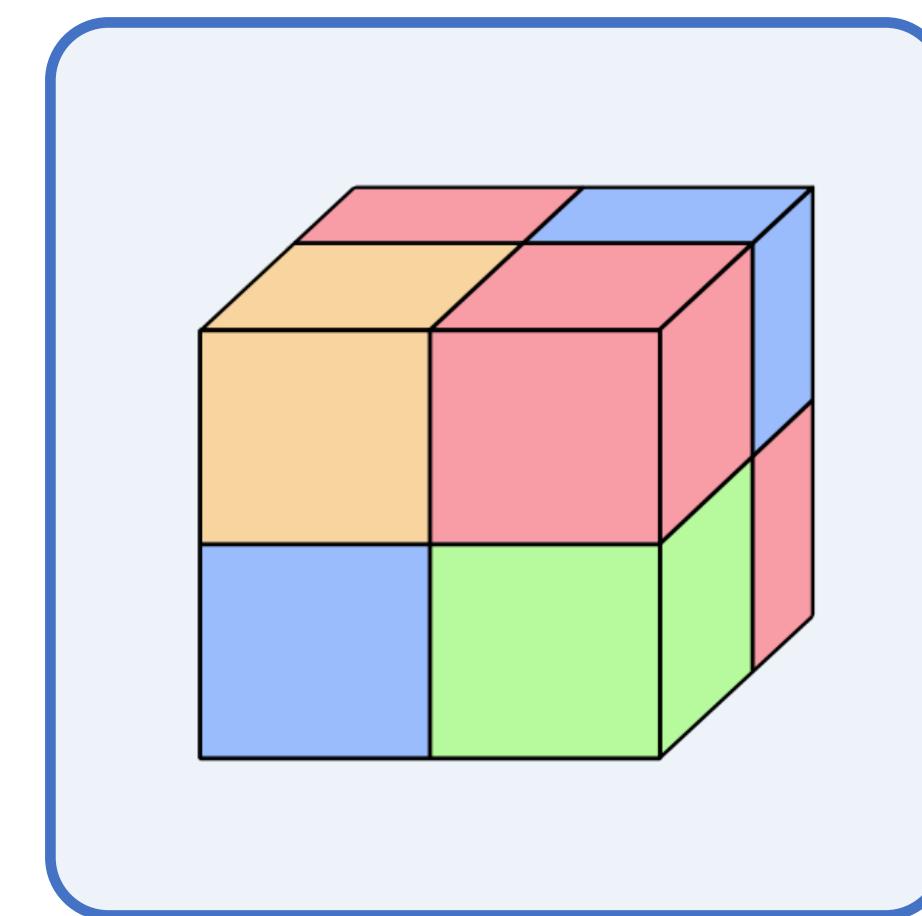
Latent Code **z**



Transpose 3D
Convolutions

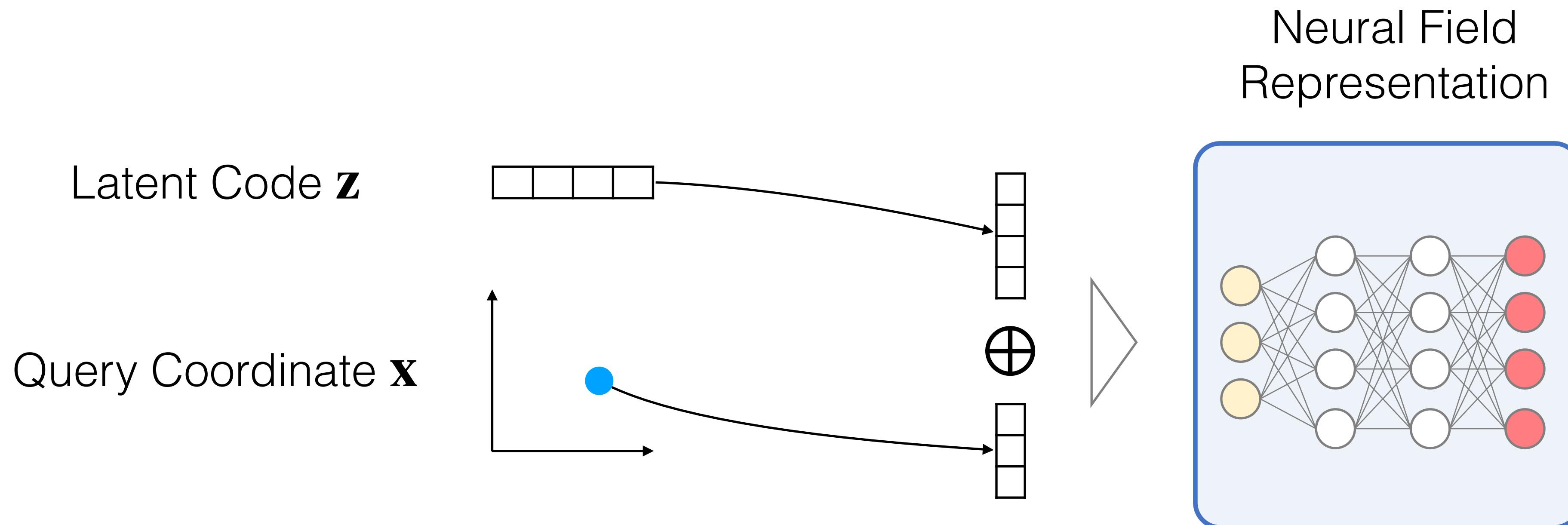


Voxel Grid
Representation

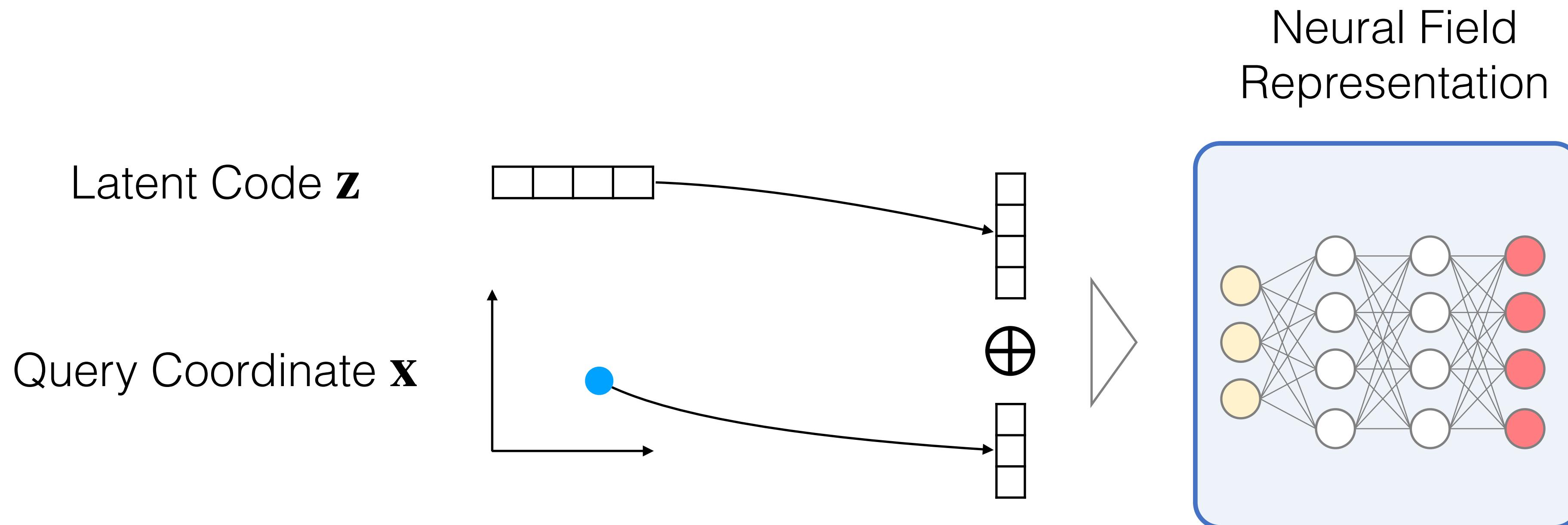


To condition voxel grid, can apply 3D de-convolutional neural network (aka transpose convolutions) to upsample $1 \times 1 \times 1 \times ch$ latent code to $N \times N \times N \times ch$.

Conditioning Neural Fields: Conditioning via Concatenation



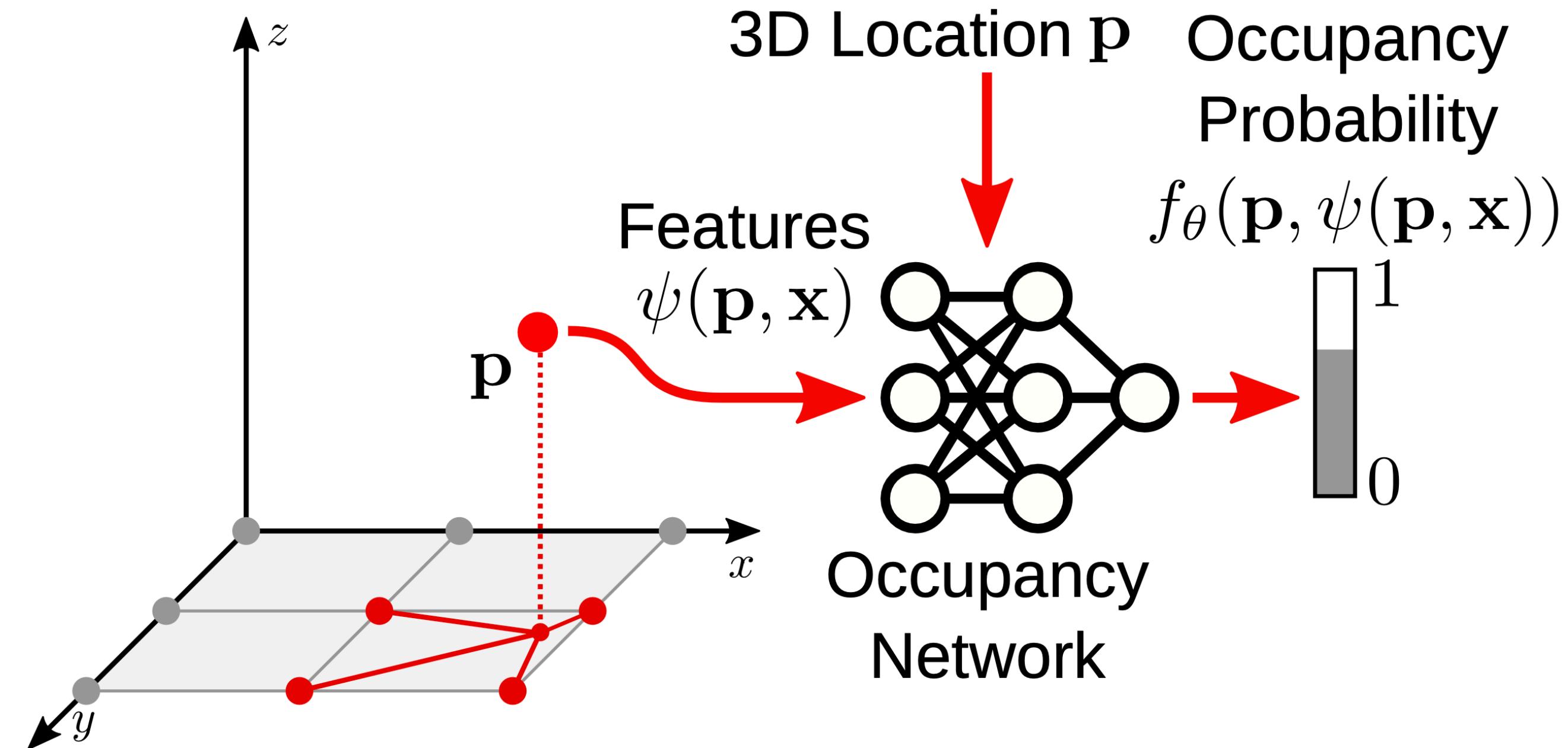
Conditioning Neural Fields: Conditioning via Concatenation



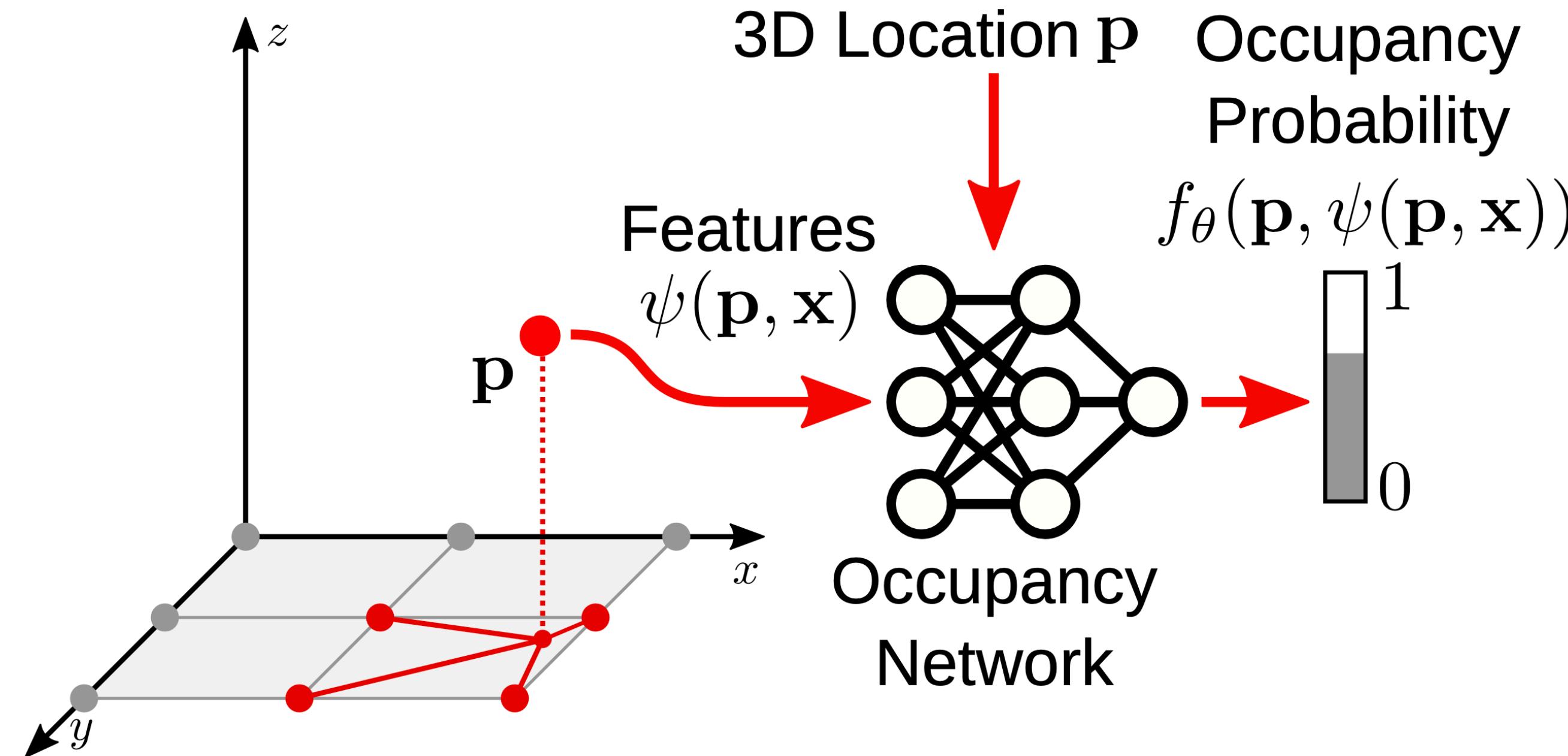
For neural fields, can concatenate latent code and query coordinate as input.

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

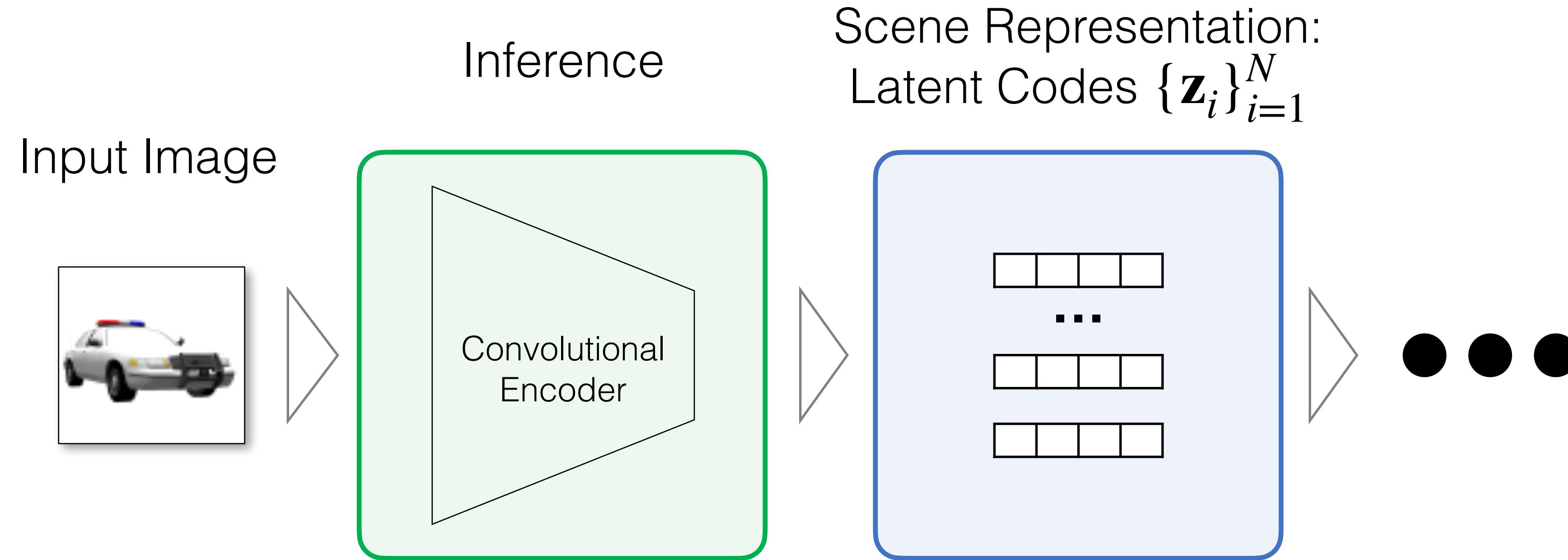


Conditioning Neural Fields: Conditioning via Concatenation

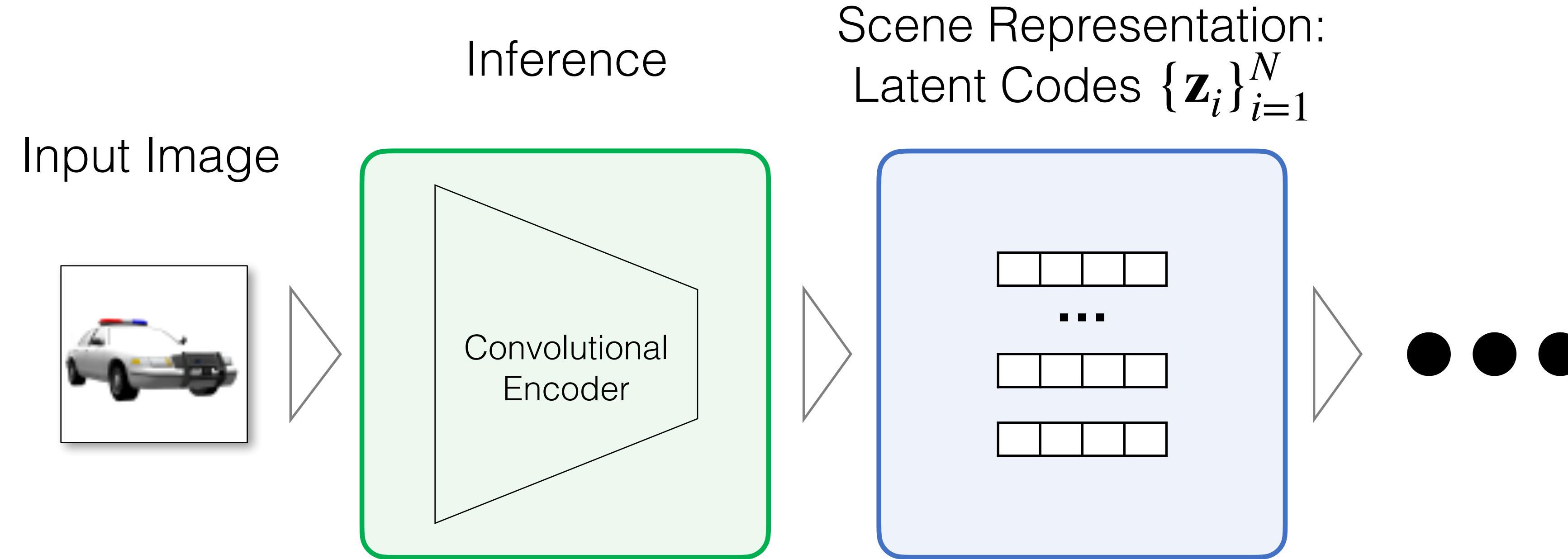


You already know this: That's what's usually done in hybrid discrete-continuous Representations!

Note on Terminology



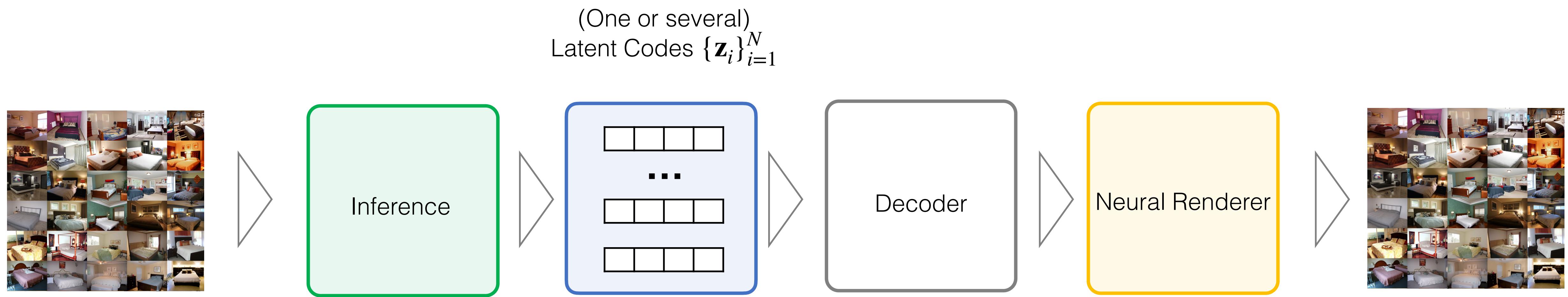
Note on Terminology



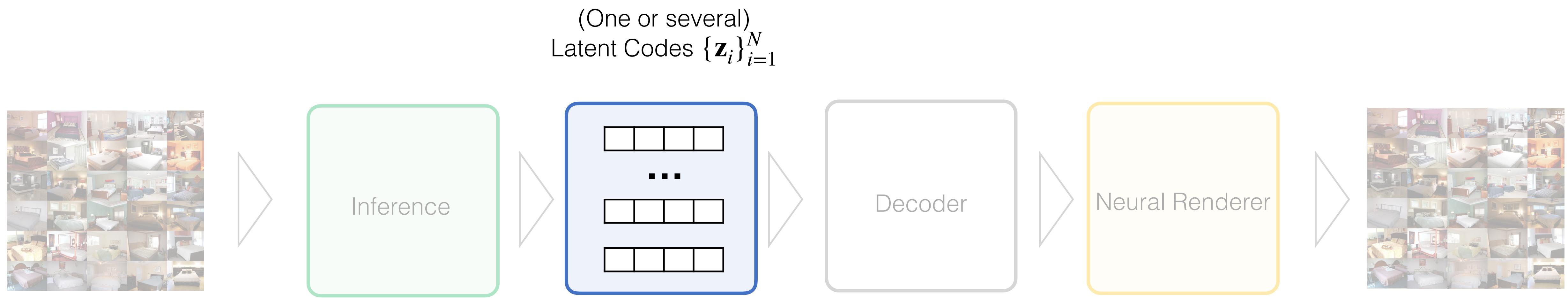
In fact, the **Latent Codes** are what's actually encoding the scene!

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

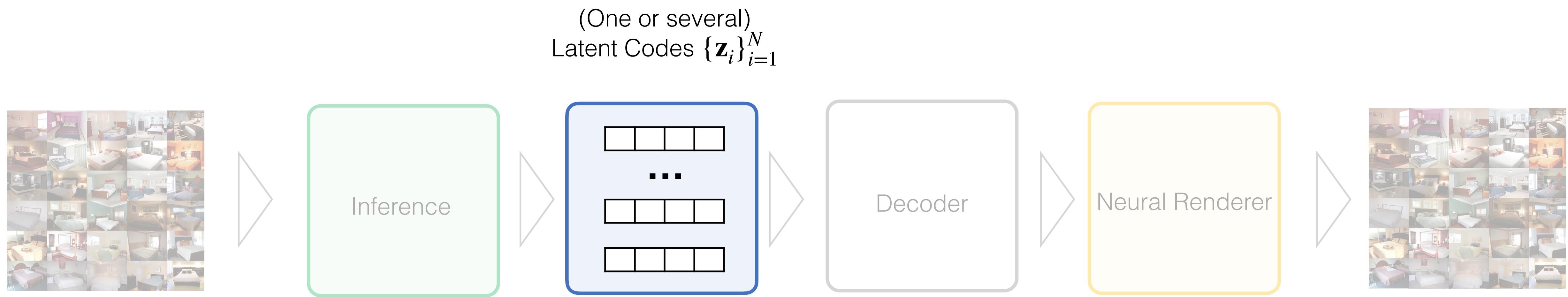
Summary: General Framework



Summary: General Framework

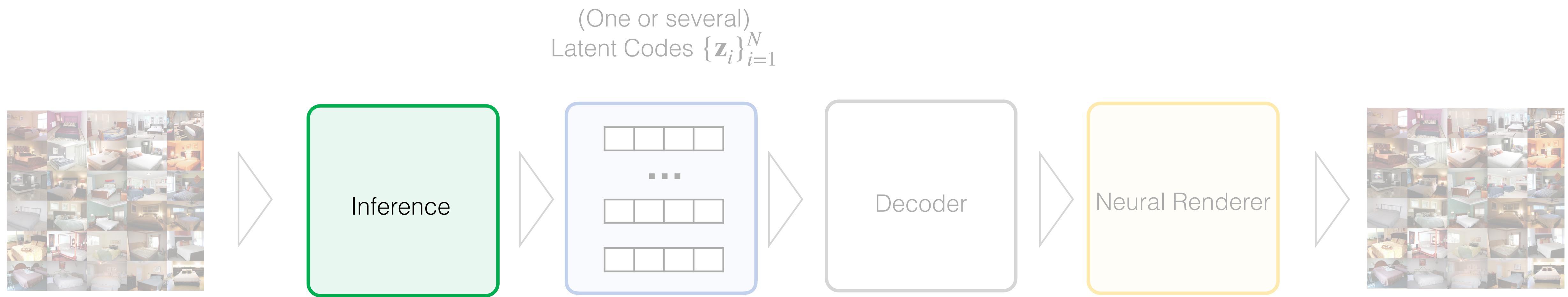


Summary: General Framework

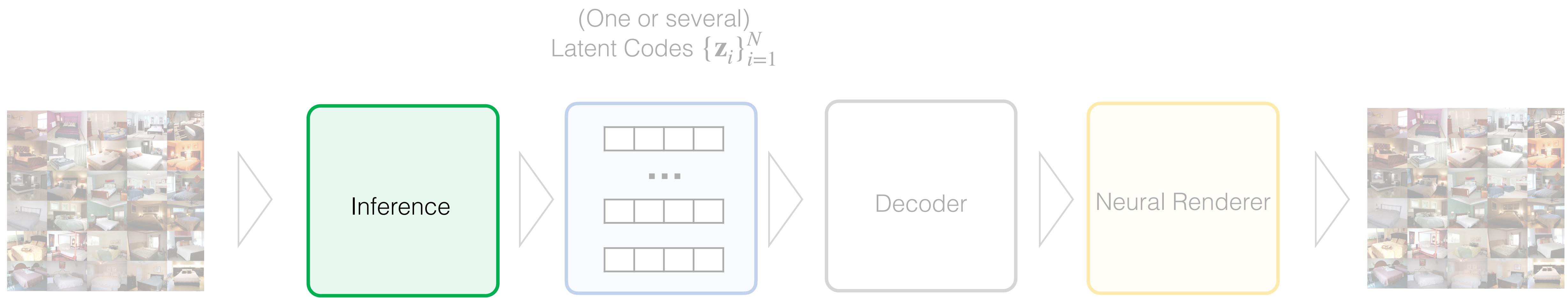


What are the **Latent Codes** and how many should there be?

Summary: General Framework



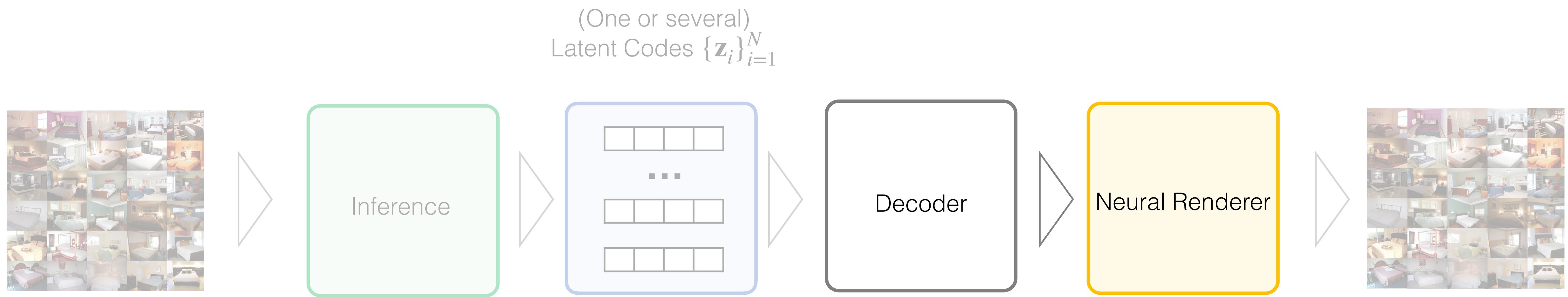
Summary: General Framework



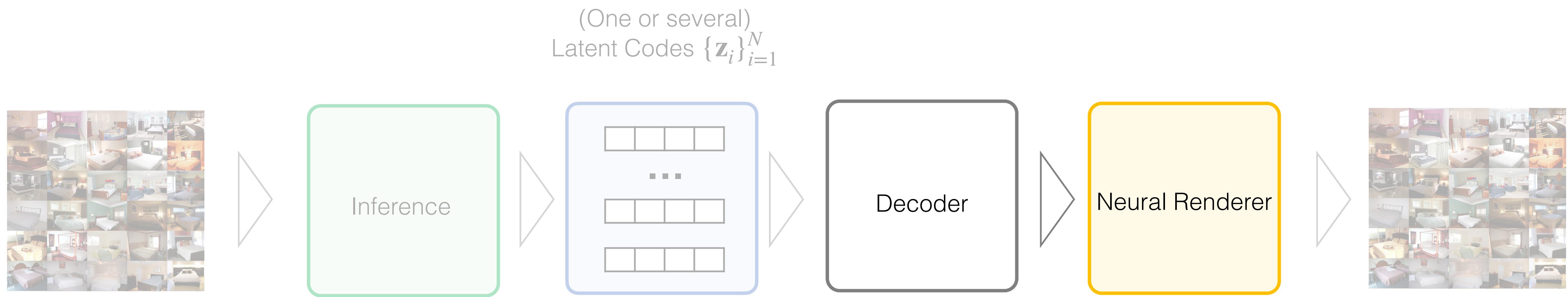
How to get the latent codes from images?

How to guarantee generalization?

Summary: General Framework



Summary: General Framework

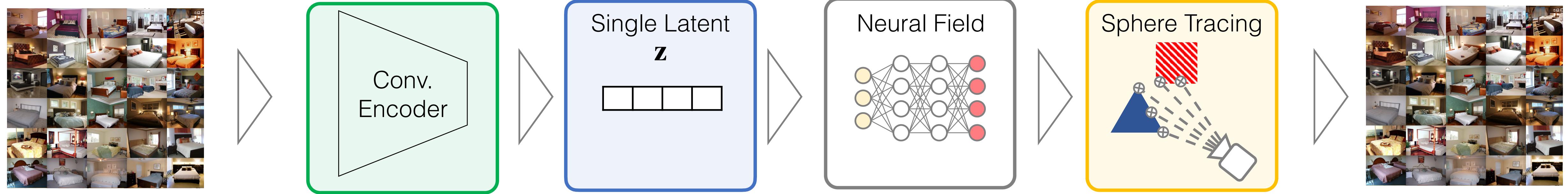


What forms & shapes can the decoder take?

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

Do we need 3D volumetric renderers?

Specific Model



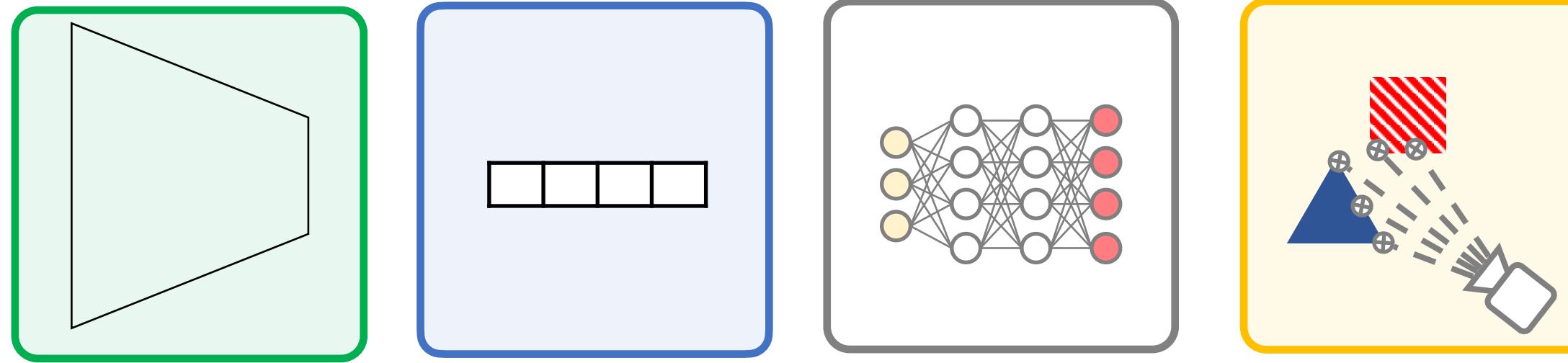
Learn inference model from many images of many scenes.

Learn inference model from many images of many scenes.

Model

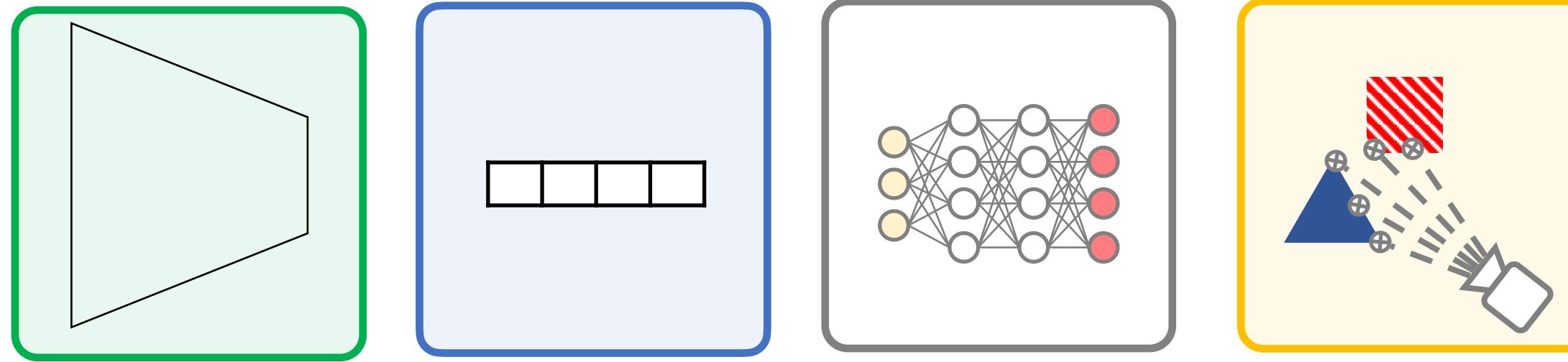
Learn inference model from many images of many scenes.

Model



Learn inference model from many images of many scenes.

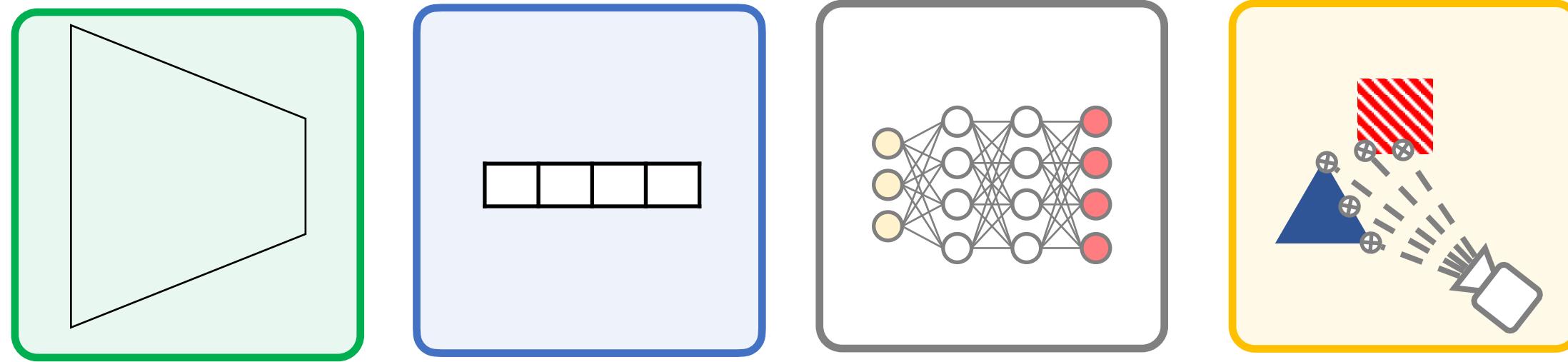
Model



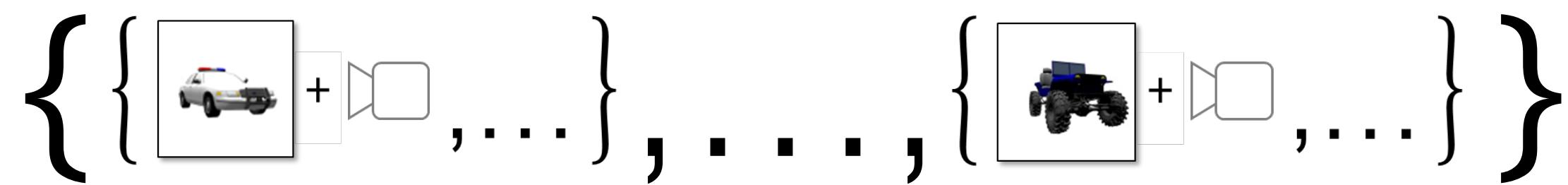
Dataset

Learn inference model from many images of many scenes.

Model

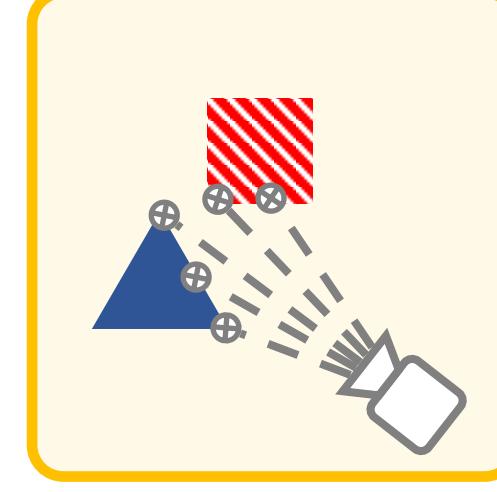
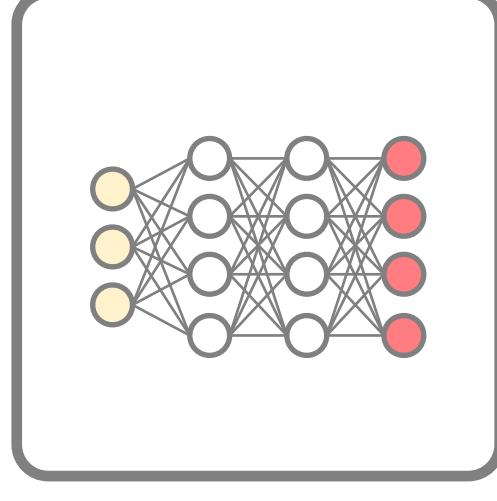
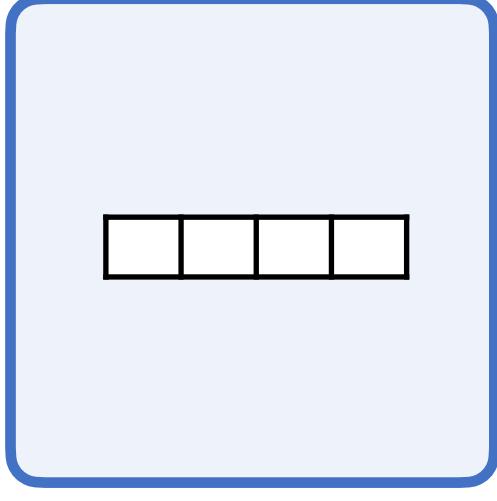
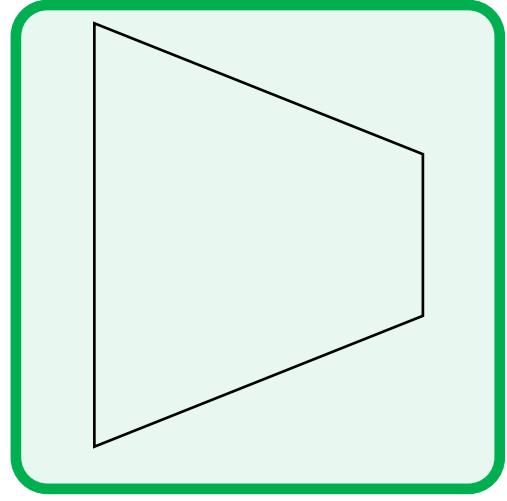


Dataset



Learn inference model from many images of many scenes.

Model



Input view



Normal map

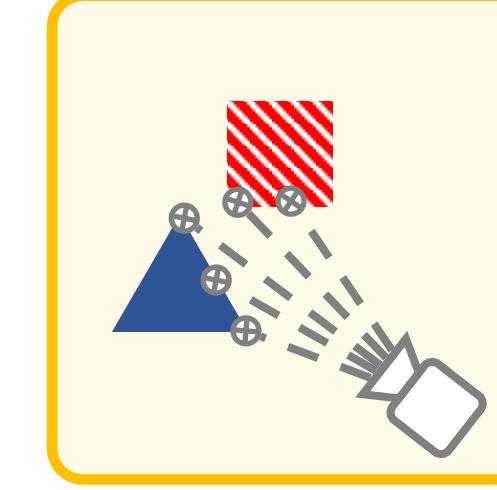
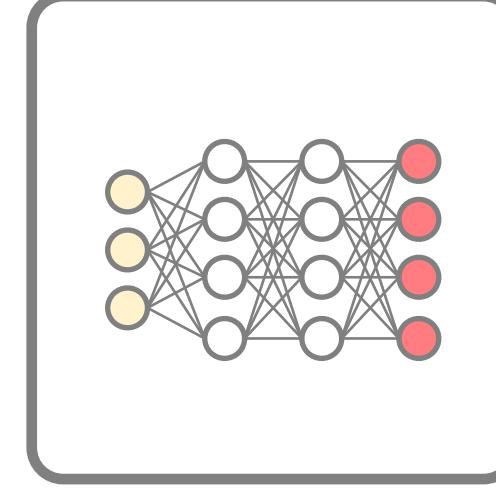
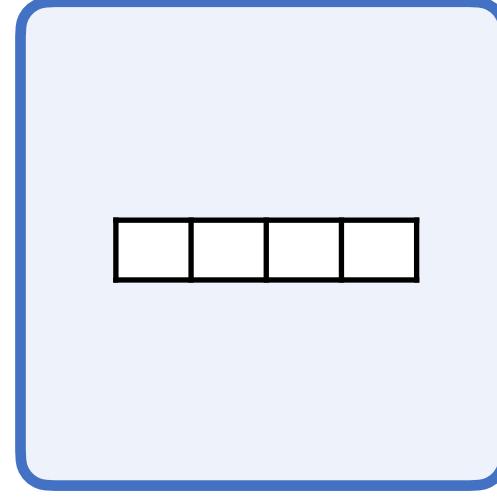
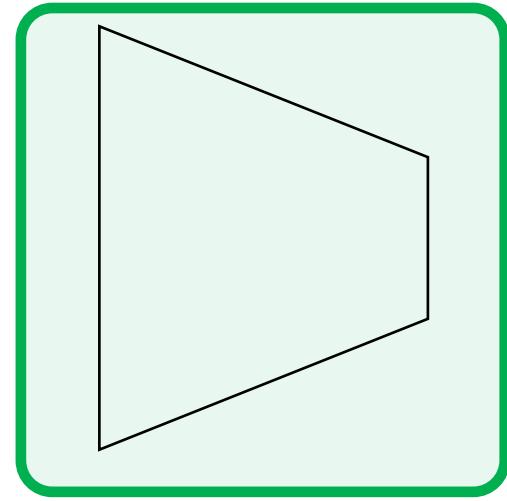
RGB

Dataset

$$\left\{ \left\{ \begin{array}{c} \text{car icon} \\ + \\ \text{camera icon} \end{array}, \dots \right\}, \dots, \left\{ \begin{array}{c} \text{truck icon} \\ + \\ \text{camera icon} \end{array}, \dots \right\} \right\}$$

Learn inference model from many images of many scenes.

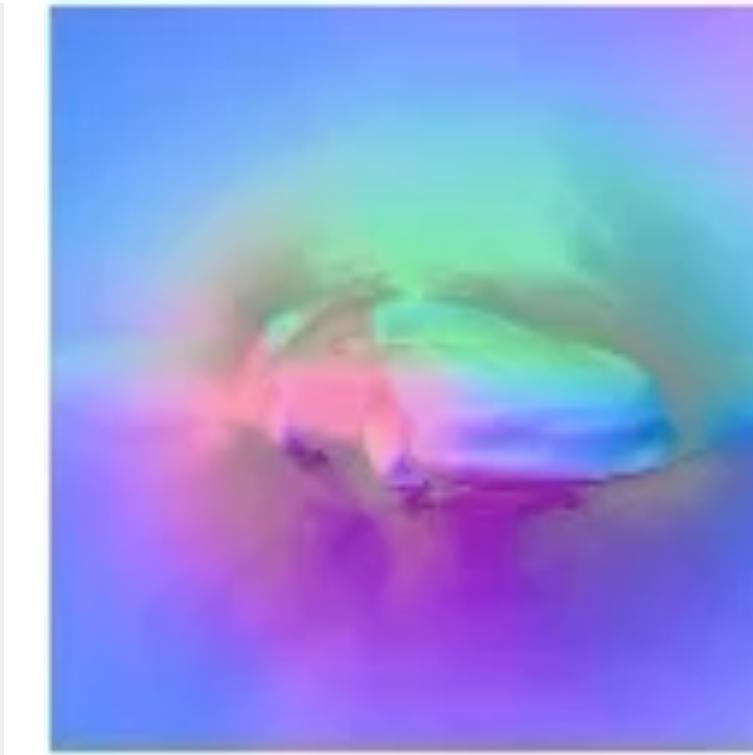
Model



Input view



Normal map



RGB

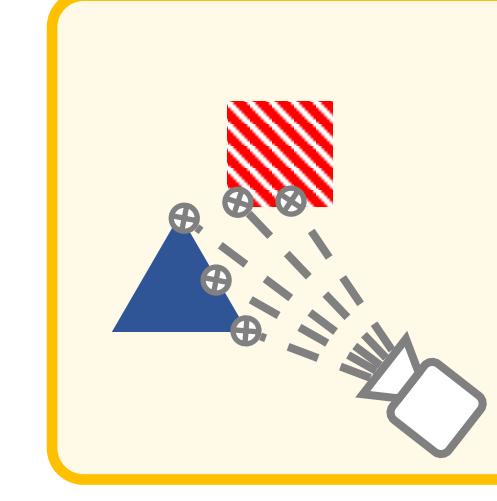
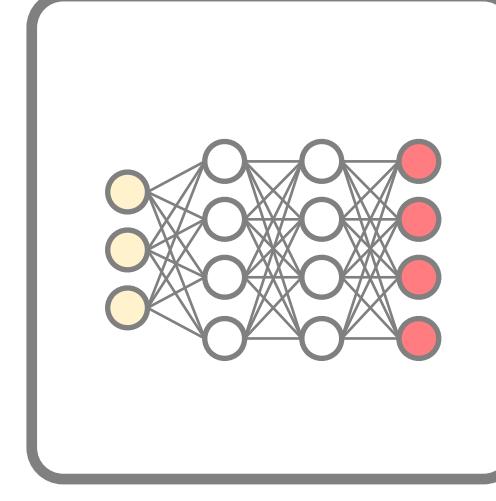
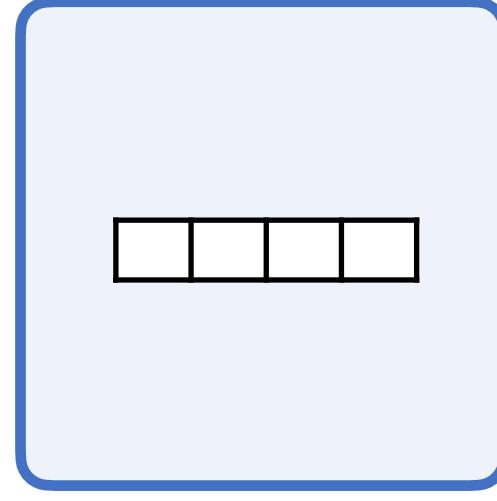
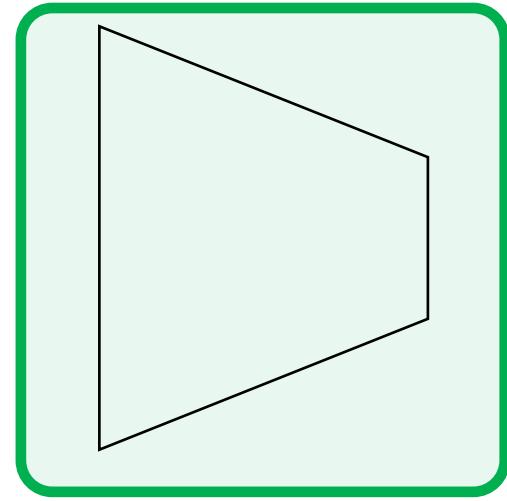


Dataset

$$\left\{ \left\{ \begin{array}{c} \text{car icon} \\ + \\ \text{camera icon} \end{array}, \dots \right\}, \dots, \left\{ \begin{array}{c} \text{truck icon} \\ + \\ \text{camera icon} \end{array}, \dots \right\} \right\}$$

Learn inference model from many images of many scenes.

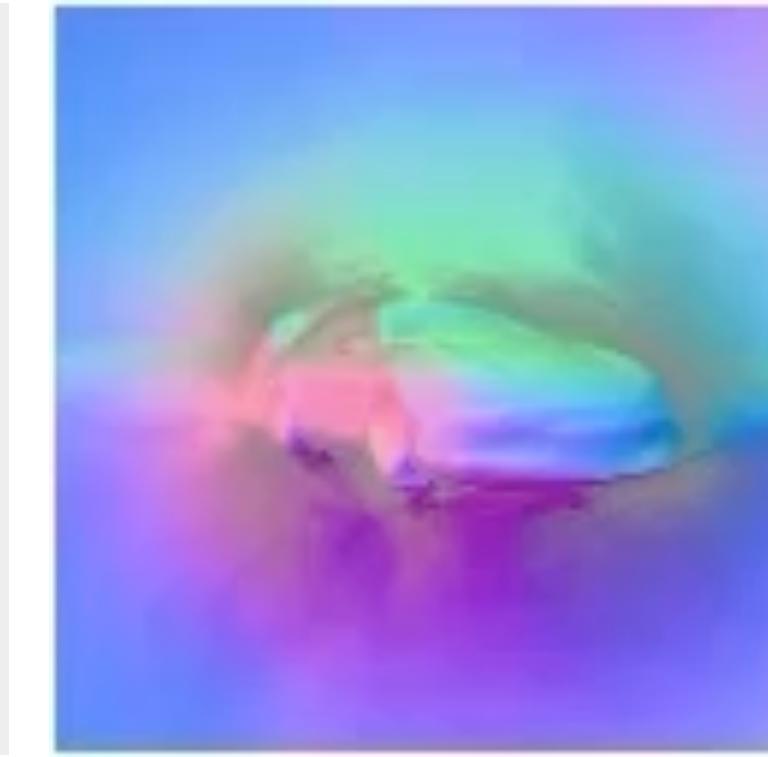
Model



Input view



Normal map



RGB



Dataset

$$\left\{ \left\{ \begin{array}{c} \text{car icon} \\ + \\ \text{camera icon} \end{array}, \dots \right\}, \dots, \left\{ \begin{array}{c} \text{truck icon} \\ + \\ \text{camera icon} \end{array}, \dots \right\} \right\}$$

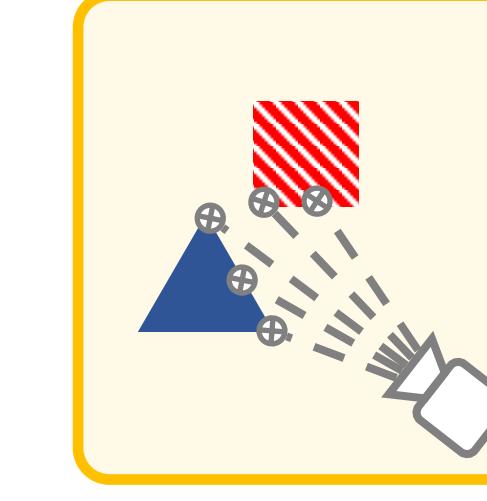
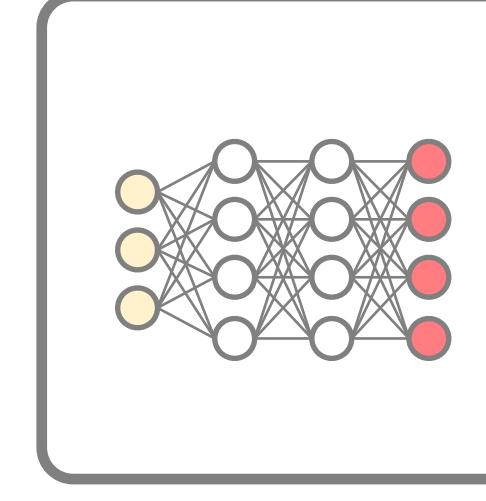
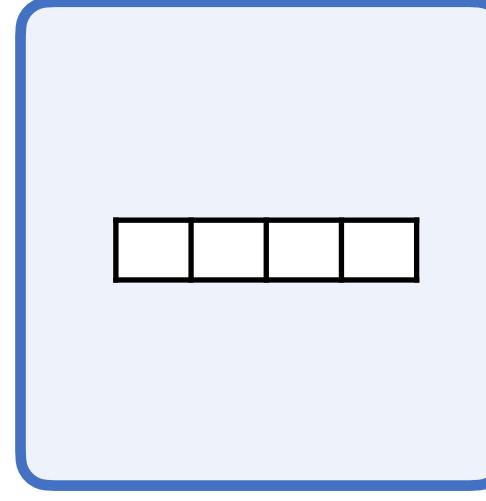
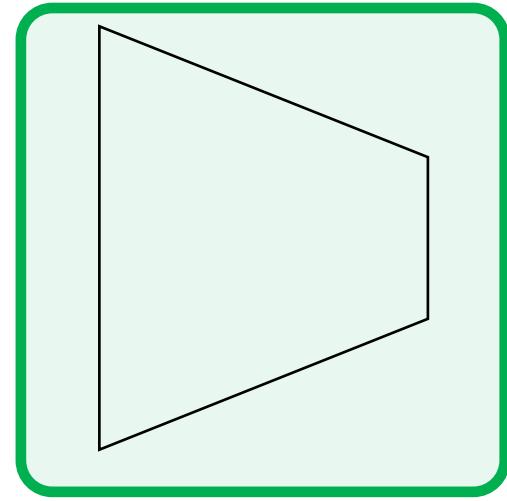
Out-of-distribution
Input view

Normal map

RGB

Learn inference model from many images of many scenes.

Model



Input view



Normal map



RGB



Dataset

$$\left\{ \left\{ \begin{array}{c} \text{car icon} \\ + \\ \text{camera icon} \end{array}, \dots \right\}, \dots, \left\{ \begin{array}{c} \text{truck icon} \\ + \\ \text{camera icon} \end{array}, \dots \right\} \right\}$$

Out-of-distribution
Input view

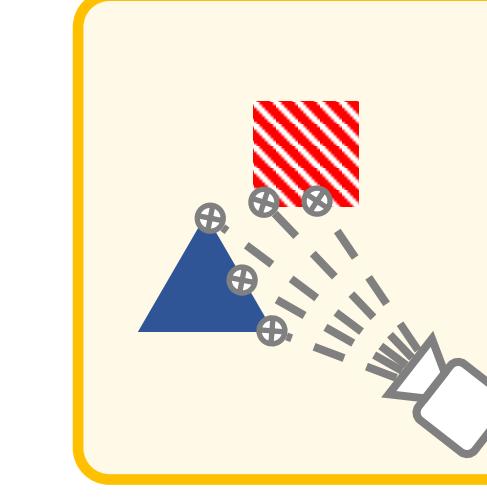
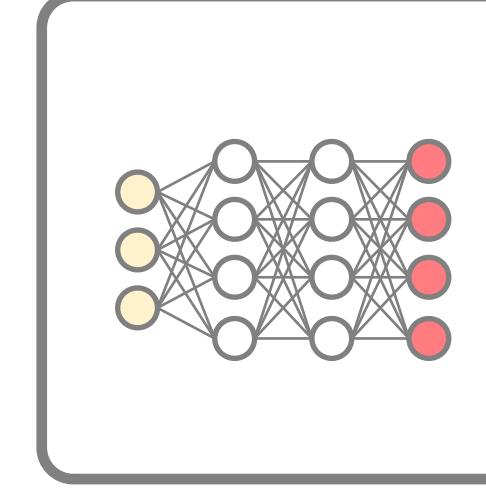
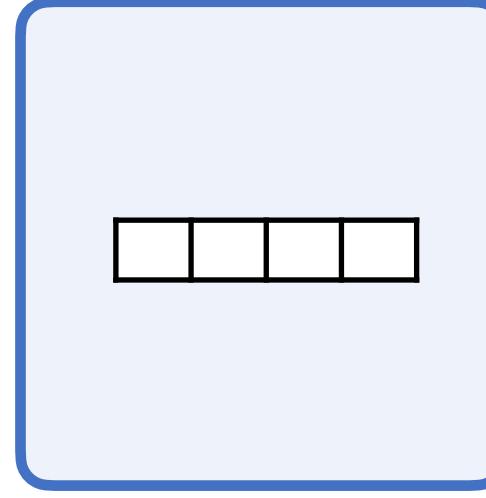
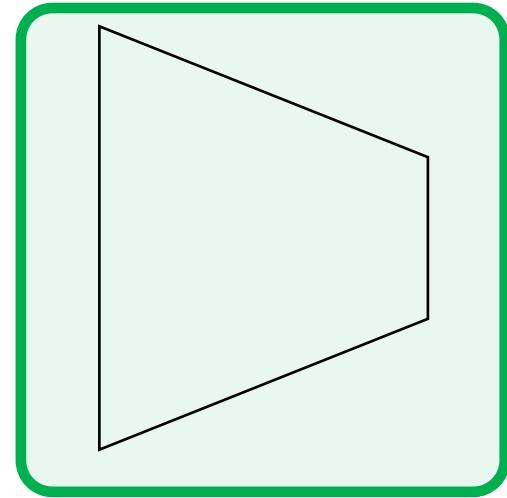


Normal map

RGB

Learn inference model from many images of many scenes.

Model



Input view



Normal map



RGB



Dataset

$$\left\{ \left\{ \begin{array}{c} \text{car icon} \\ + \end{array} \right. \text{camera icon}, \dots \right\}, \dots, \left\{ \begin{array}{c} \text{truck icon} \\ + \end{array} \text{camera icon}, \dots \right\} \right\}$$

Out-of-distribution
Input view



Normal map

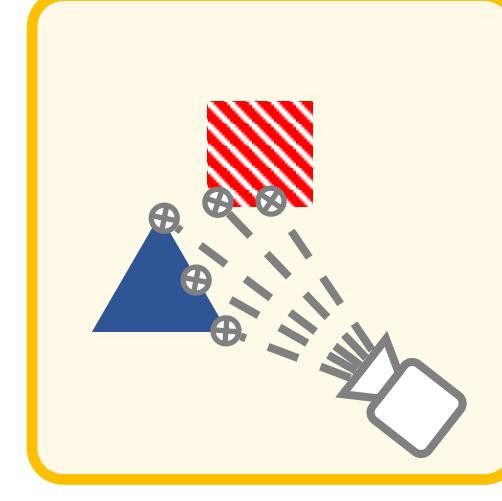
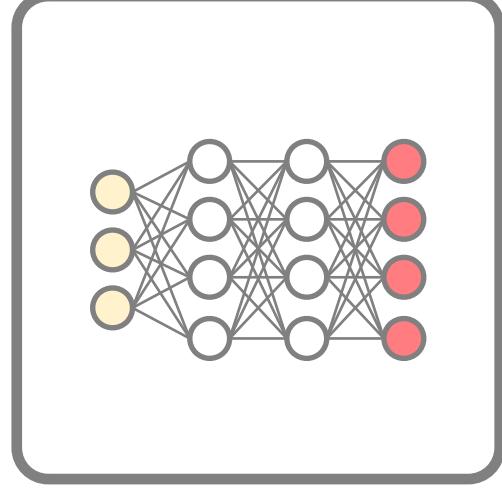
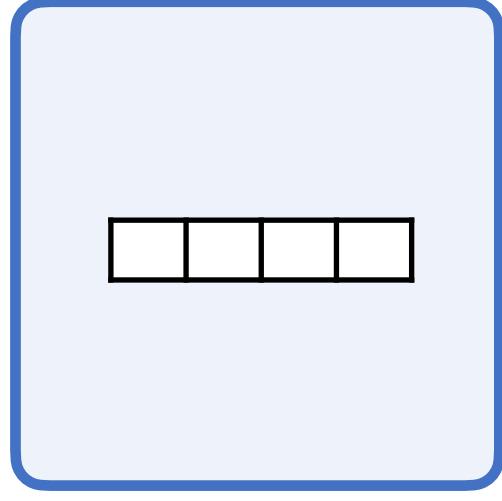
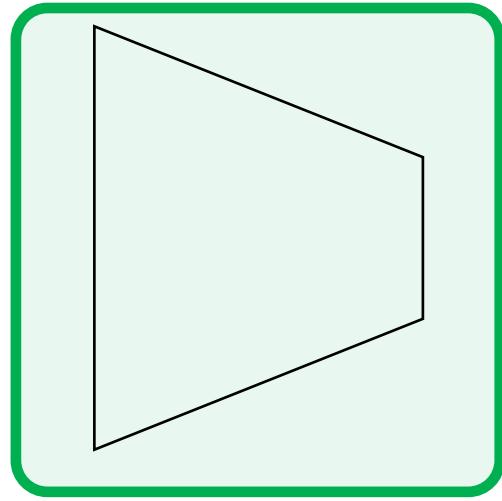


RGB



Learn inference model from many images of many scenes.

Model



Input view



Normal map



RGB



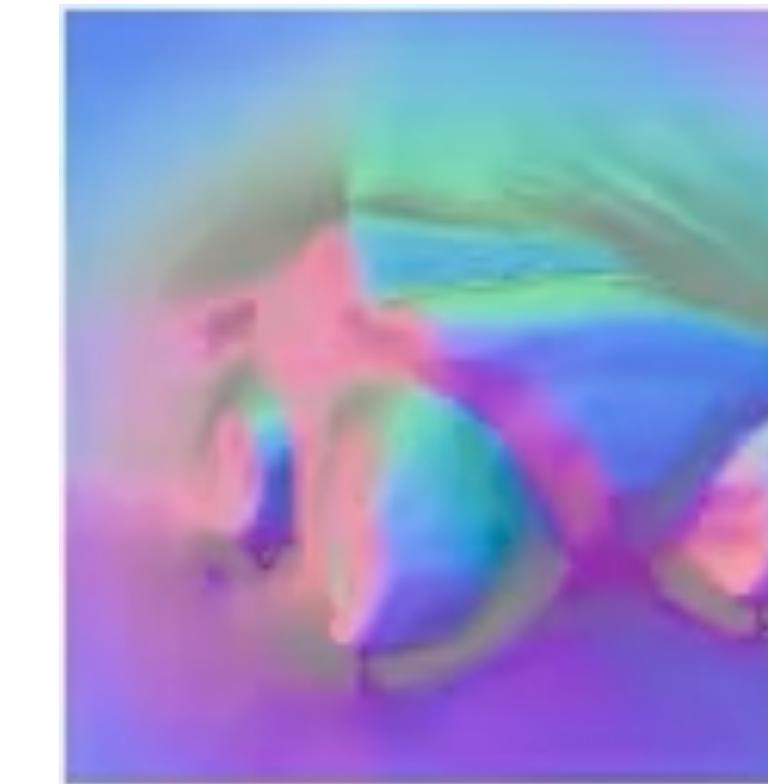
Dataset

$$\left\{ \left\{ \begin{array}{c} \text{car icon} \\ + \text{camera icon} \end{array}, \dots \right\}, \dots, \left\{ \begin{array}{c} \text{truck icon} \\ + \text{camera icon} \end{array}, \dots \right\} \right\}$$

Out-of-distribution
Input view



Normal map

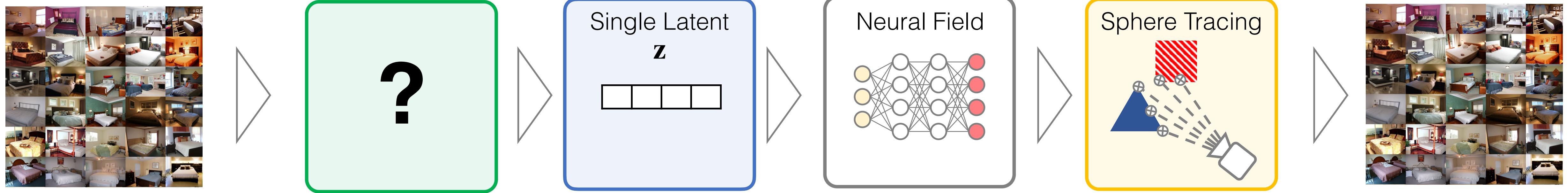


RGB

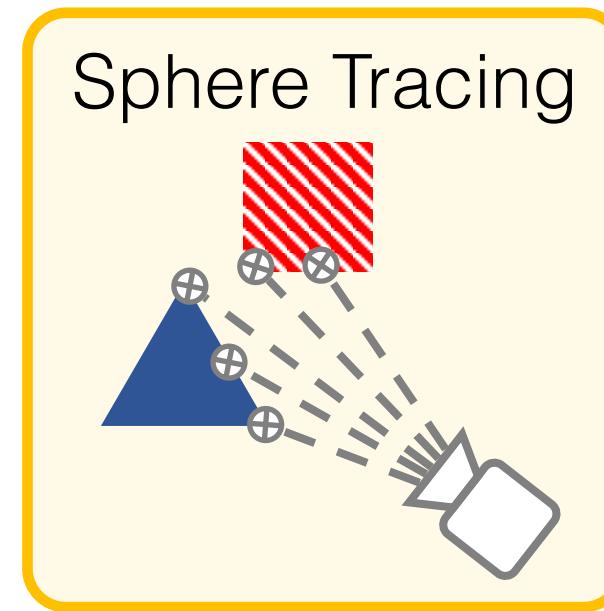
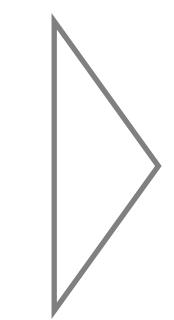
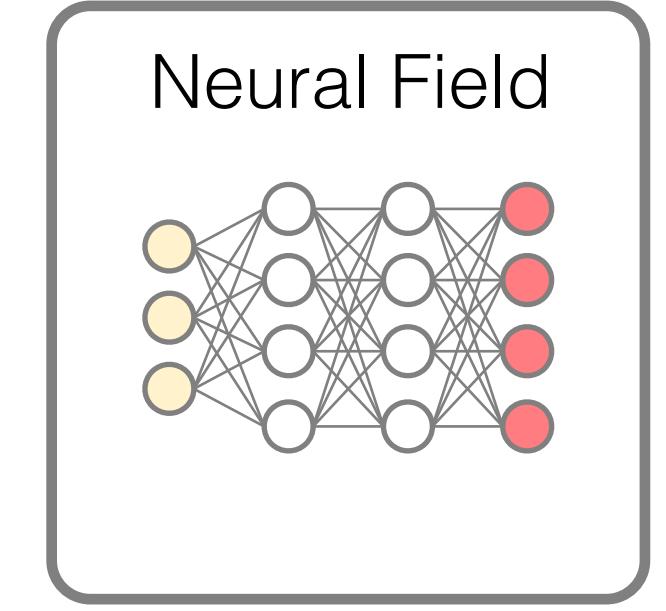
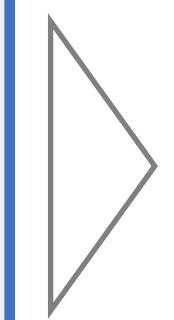
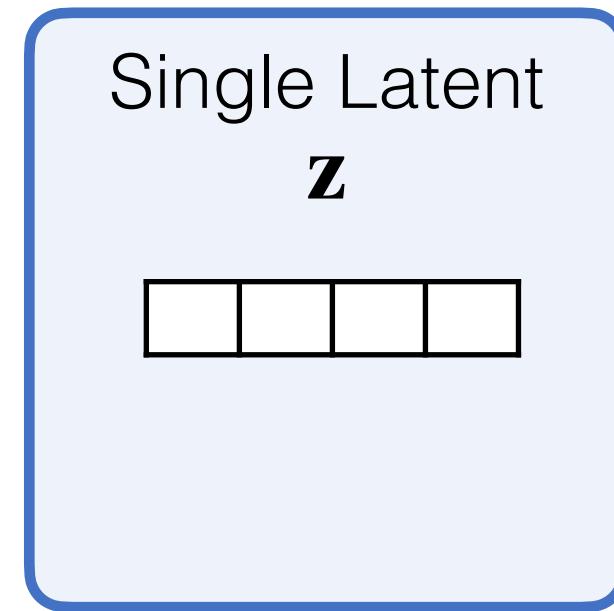
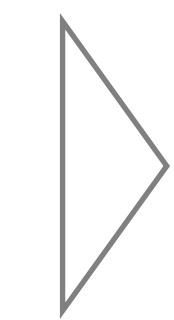
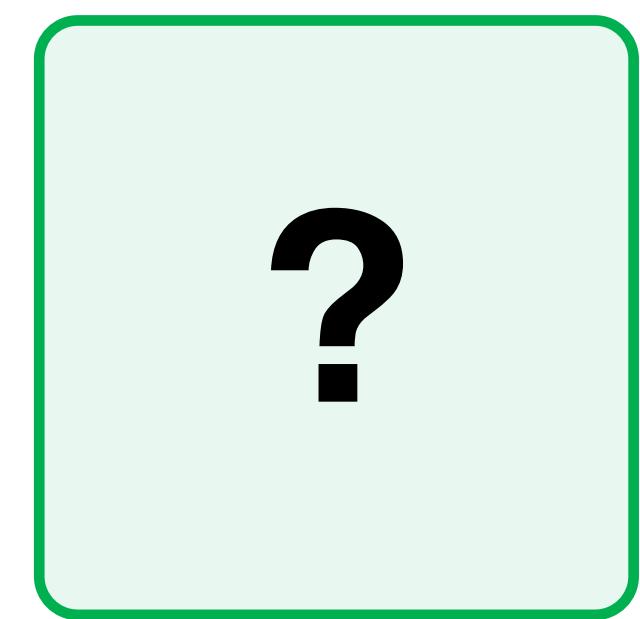


Wrong window color, wrong tires, wrong shape...

Specific Model

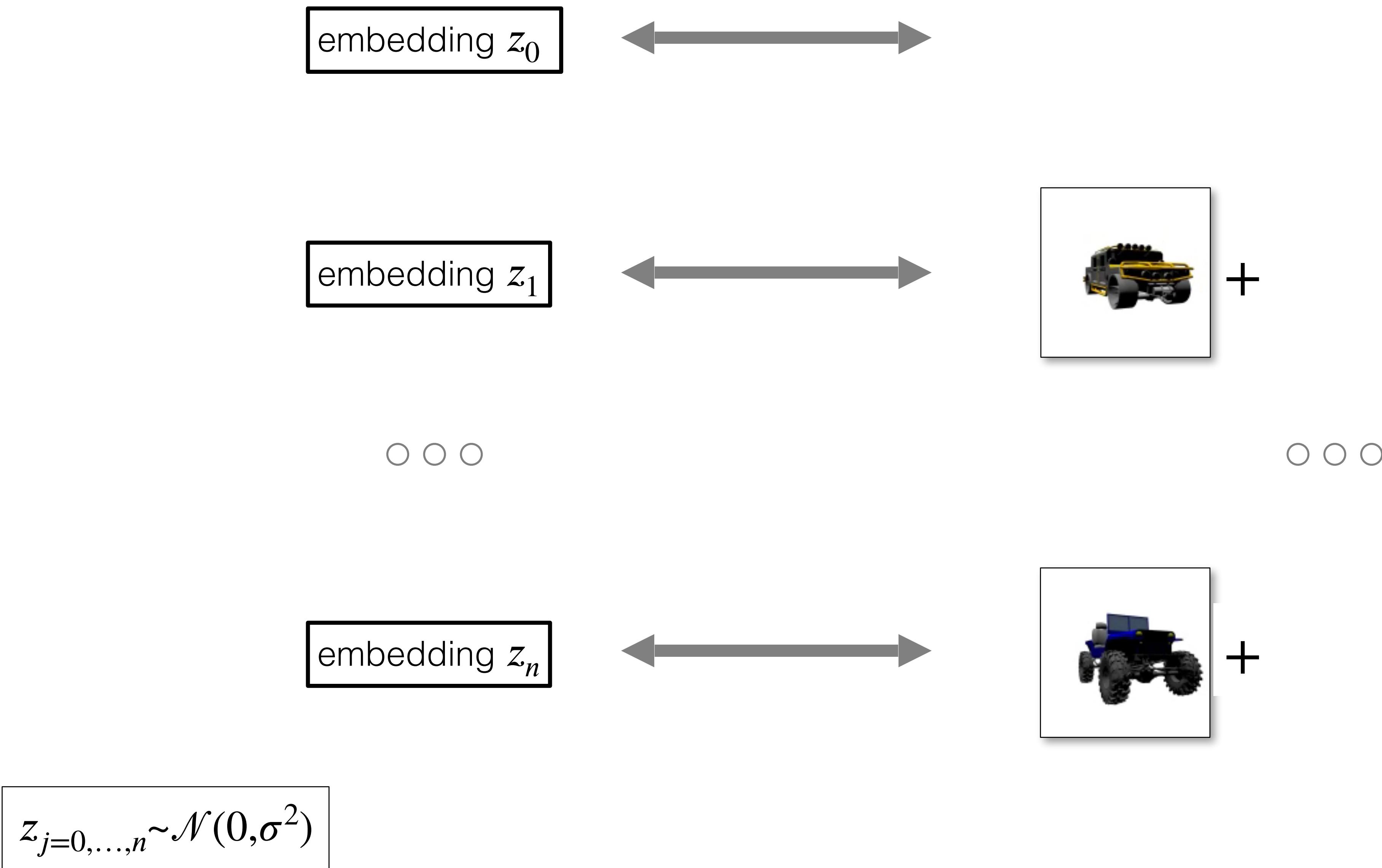


Specific Model

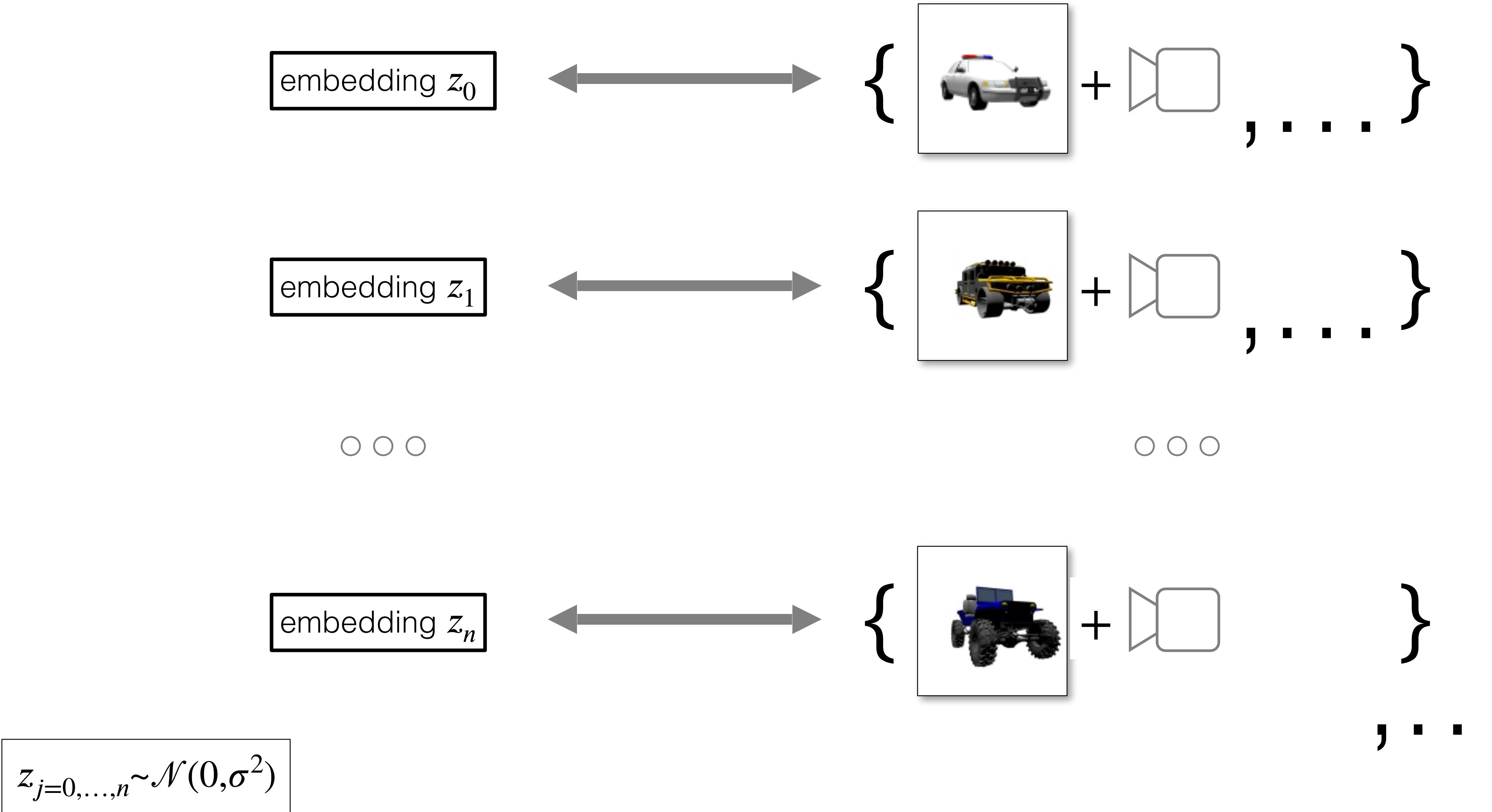


Can we think of an alternative inference method?

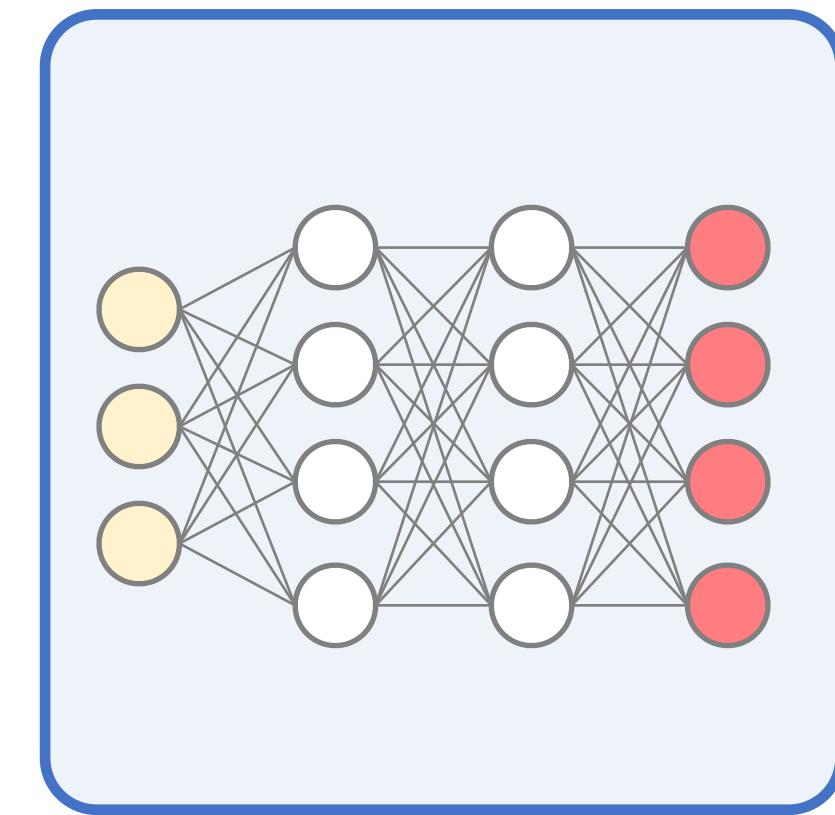
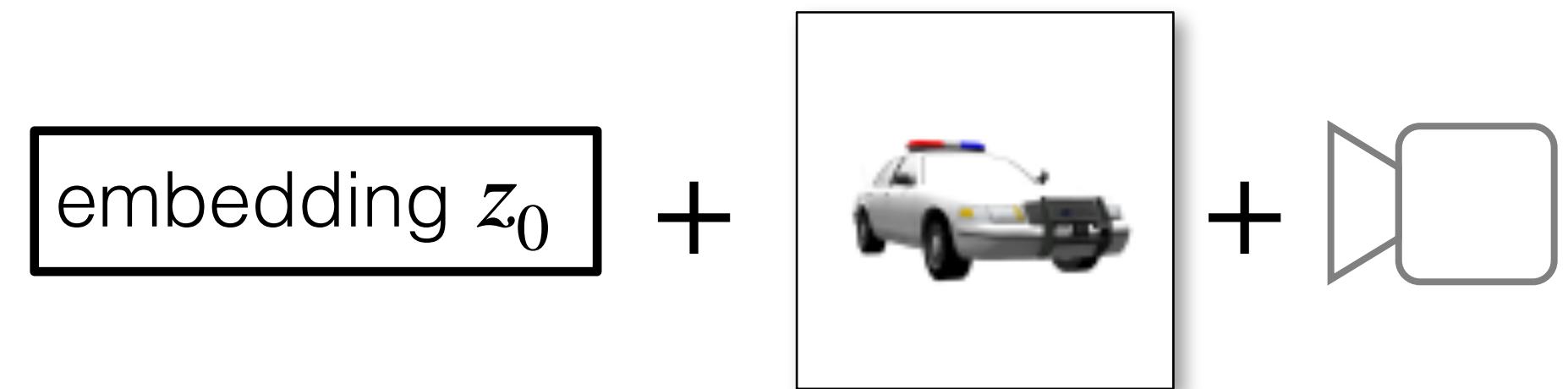
Training: Initialize one embedding per scene



Training: Initialize one embedding per scene

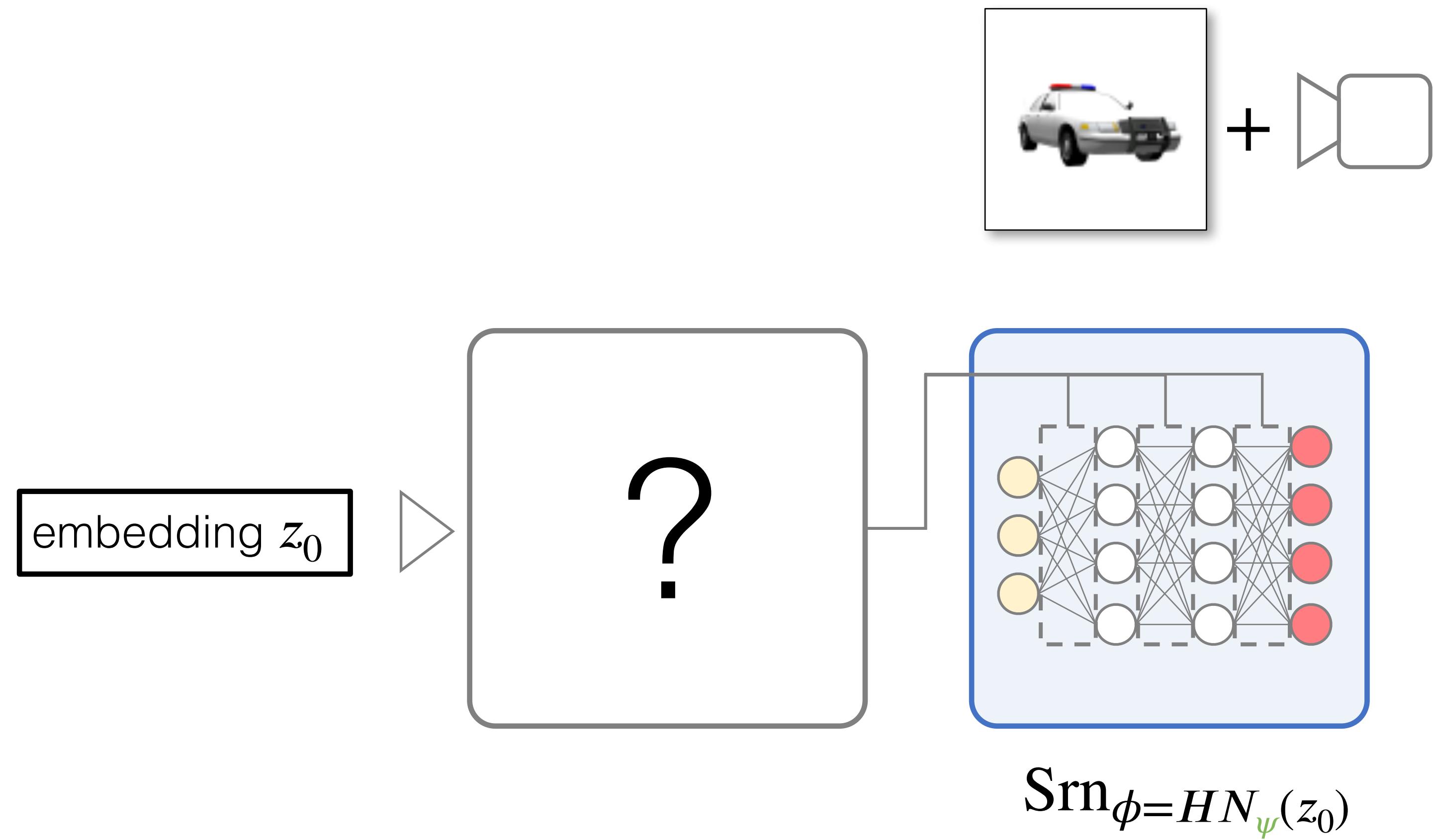


Decode embedding into scene representation

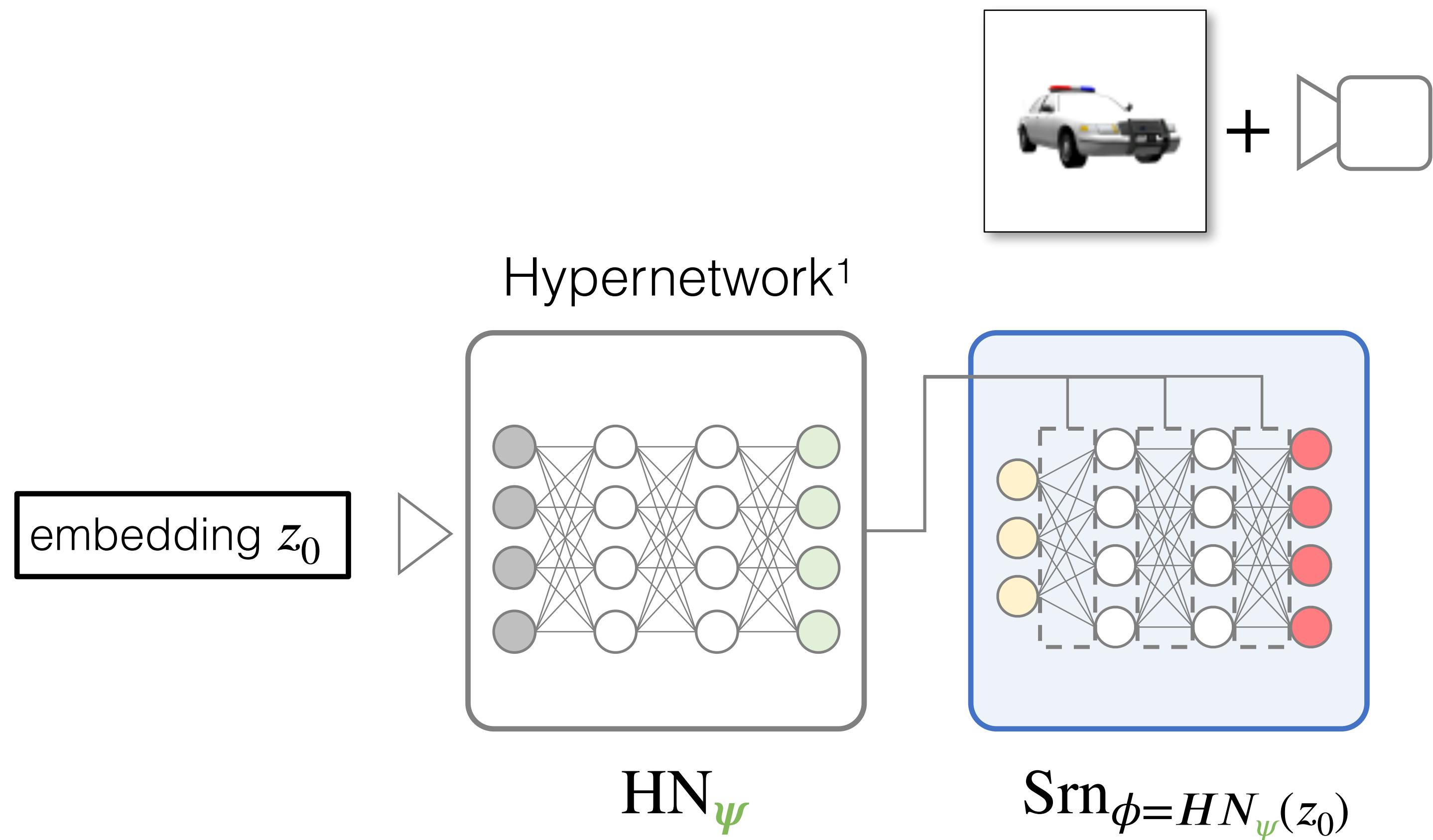


$$\text{Srn}_{\phi=HN_{\psi}(z_0)}$$

Decode embedding into scene representation

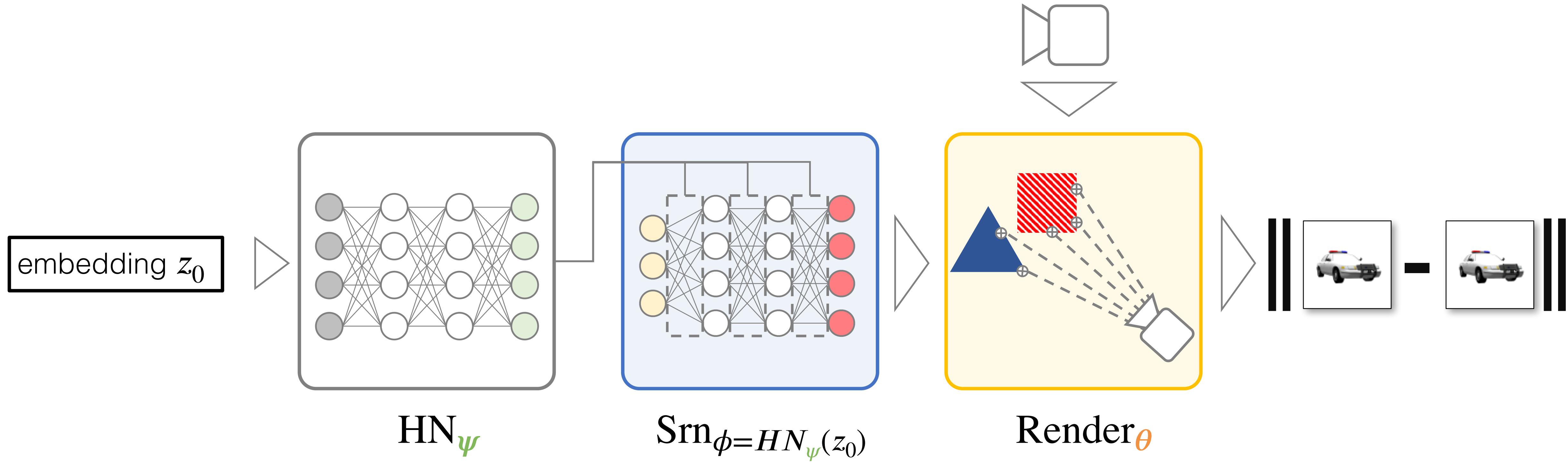


Decode embedding into scene representation

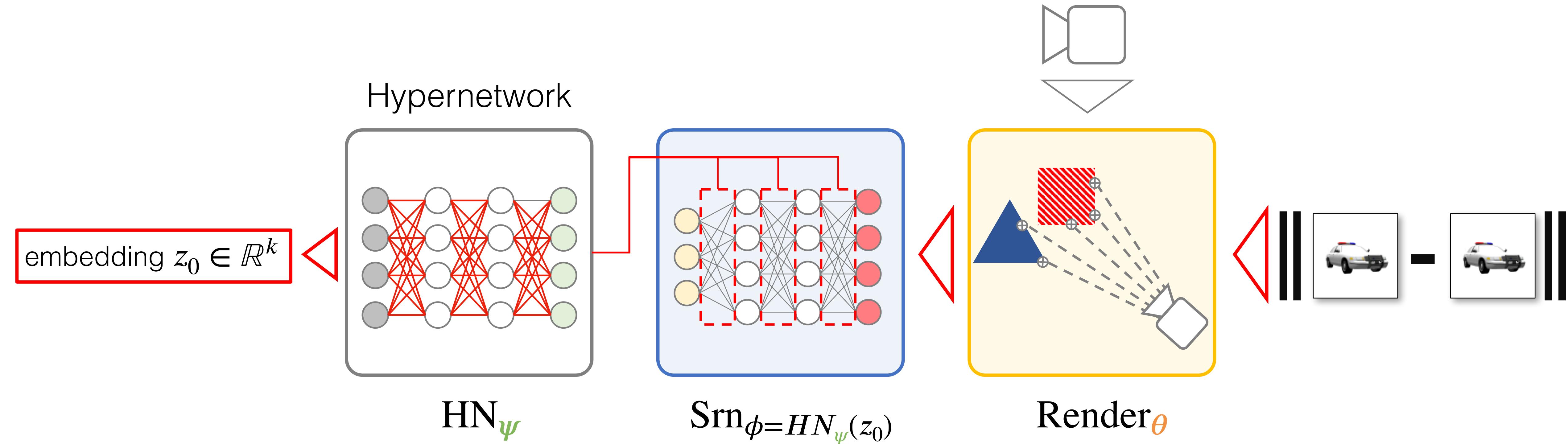


¹[Schmidhuber et al. 1992, Schmidhuber et al. 1993, Stanley et al. 2009, Ha et al., 2016]

Render training view

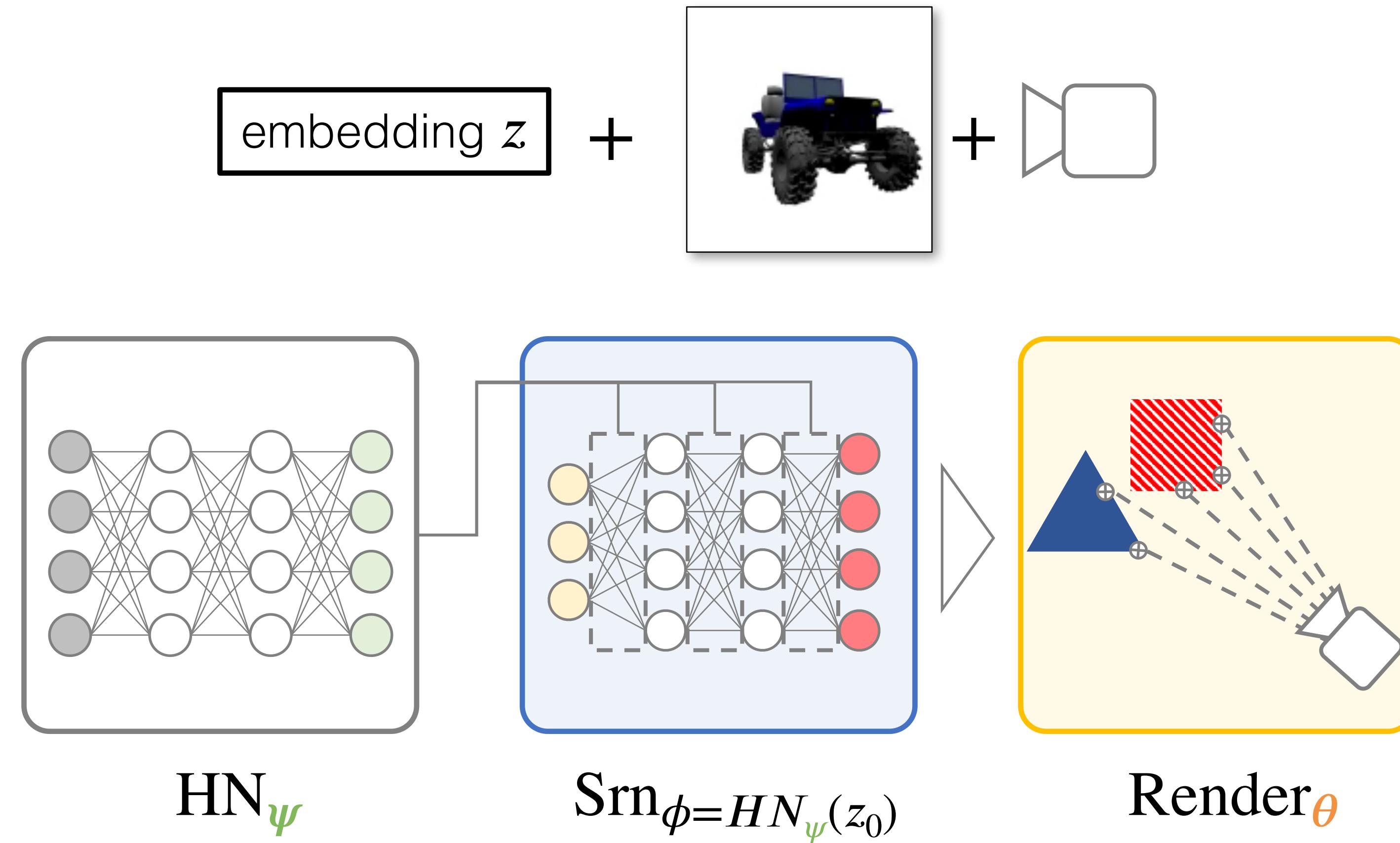


Co-optimize embeddings and model weights!

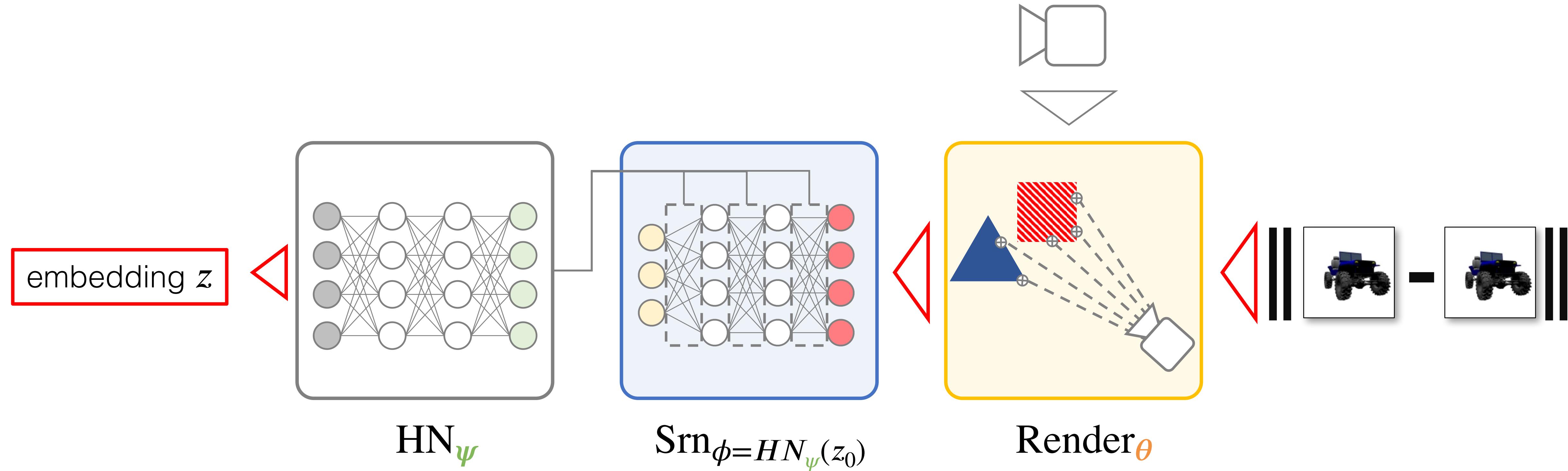


$$\arg \min_{\left\{z_j\right\}_{j=1}^M, \psi, \theta} \sum_j \sum_i \left\| \text{Render}_\theta(\text{Srn}_{\phi=H\text{N}_\psi(z_j)}, \xi_i) - \mathcal{I}_i^j \right\|$$

Test time: initialize new embedding



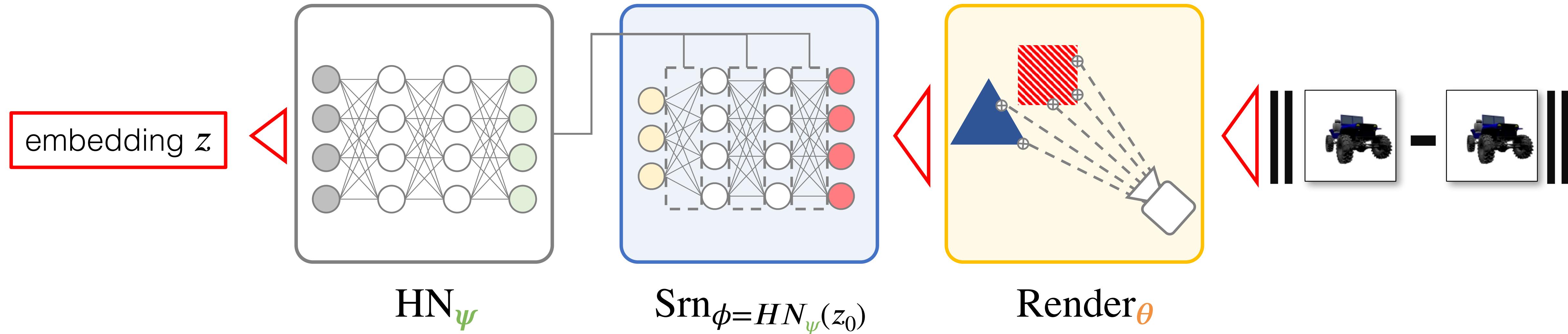
Inference: Freeze model weights, backprop into embedding only!



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

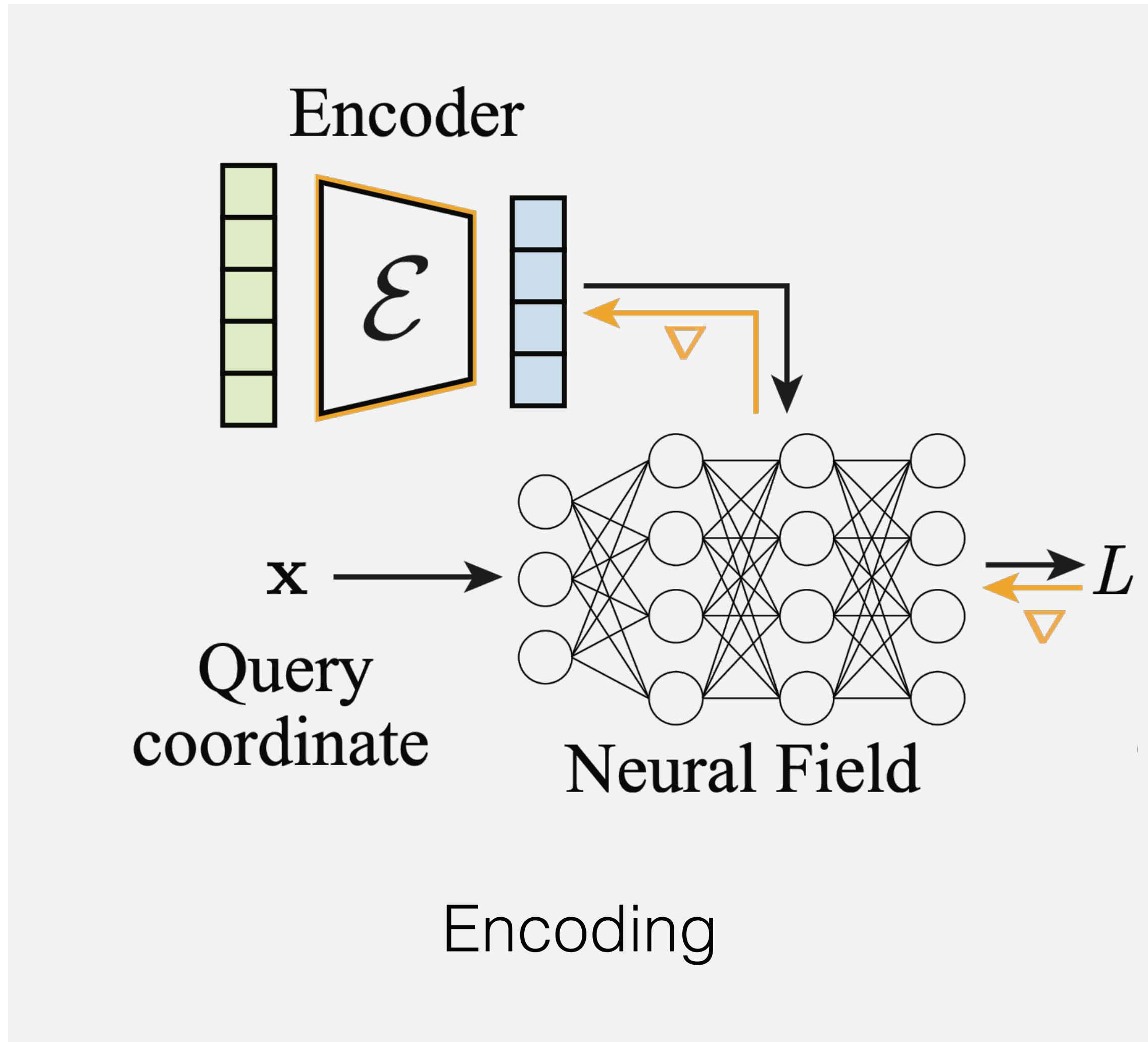
Inference: Freeze model weights, backprop into embedding only!

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

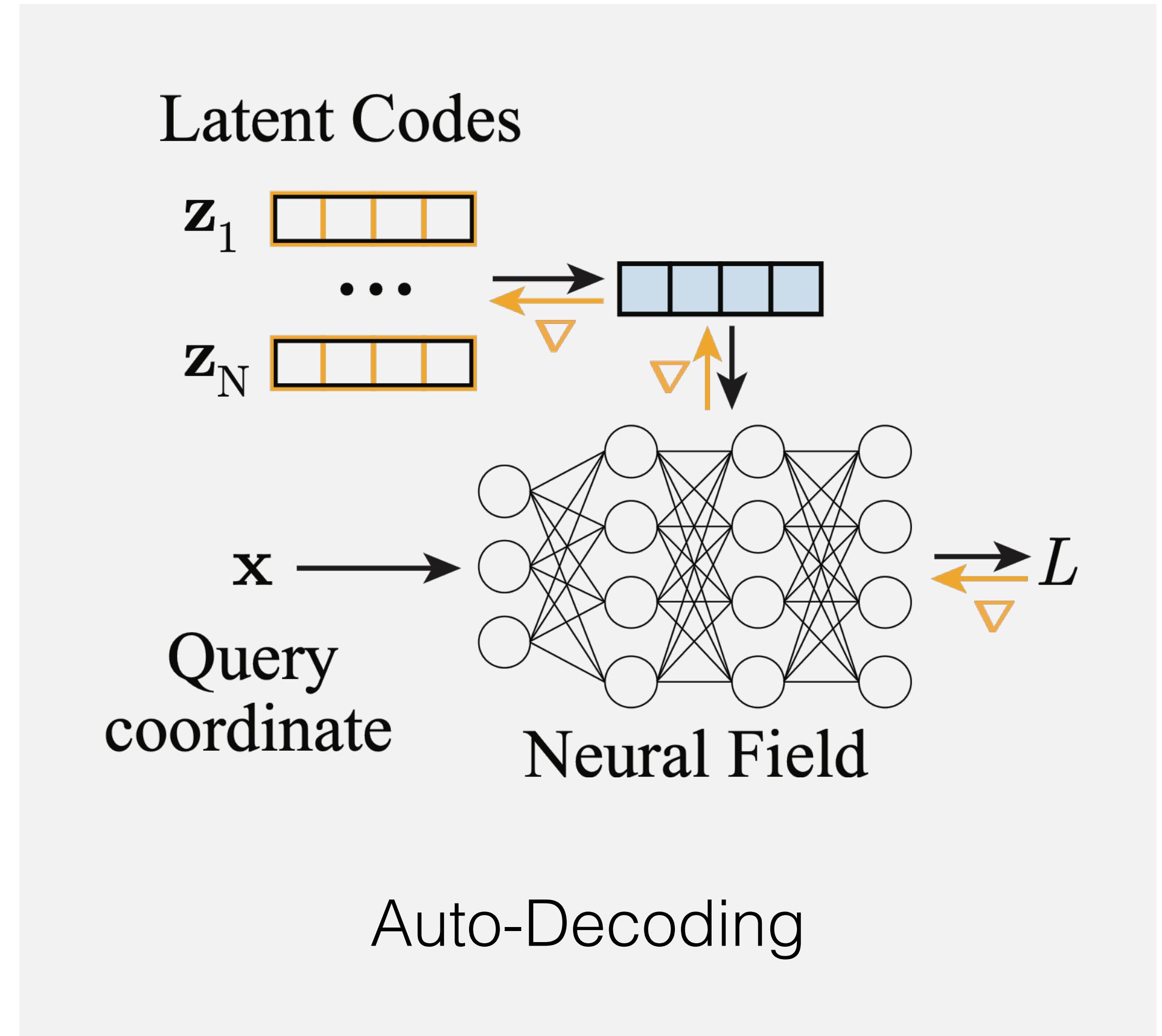
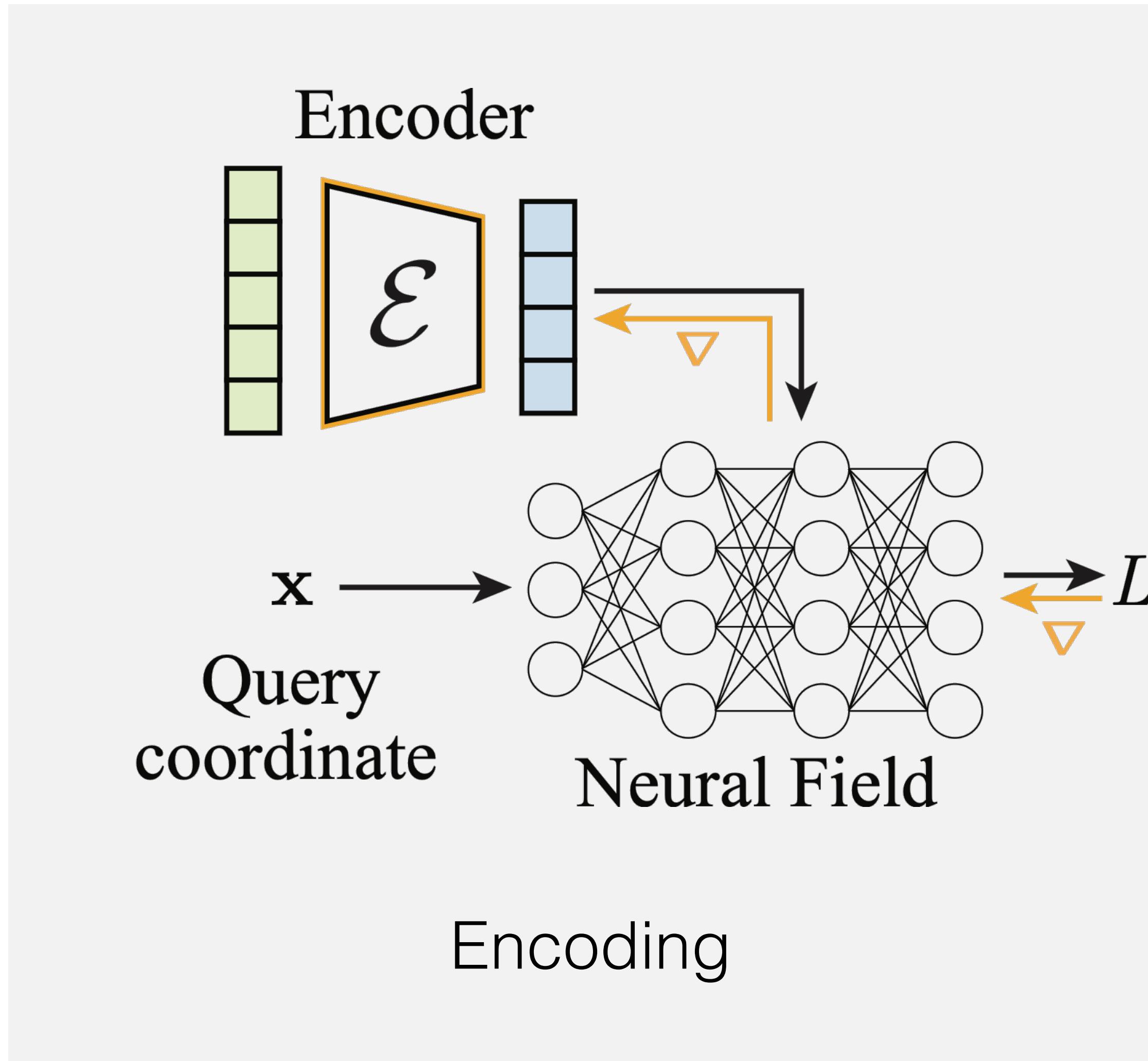


$$\hat{z} = \operatorname{argmin}_z \left\| \text{Render}_{\theta}(\text{Srн}_{\phi=HN_{\psi}(z)}, \xi) - \mathcal{I} \right\|$$

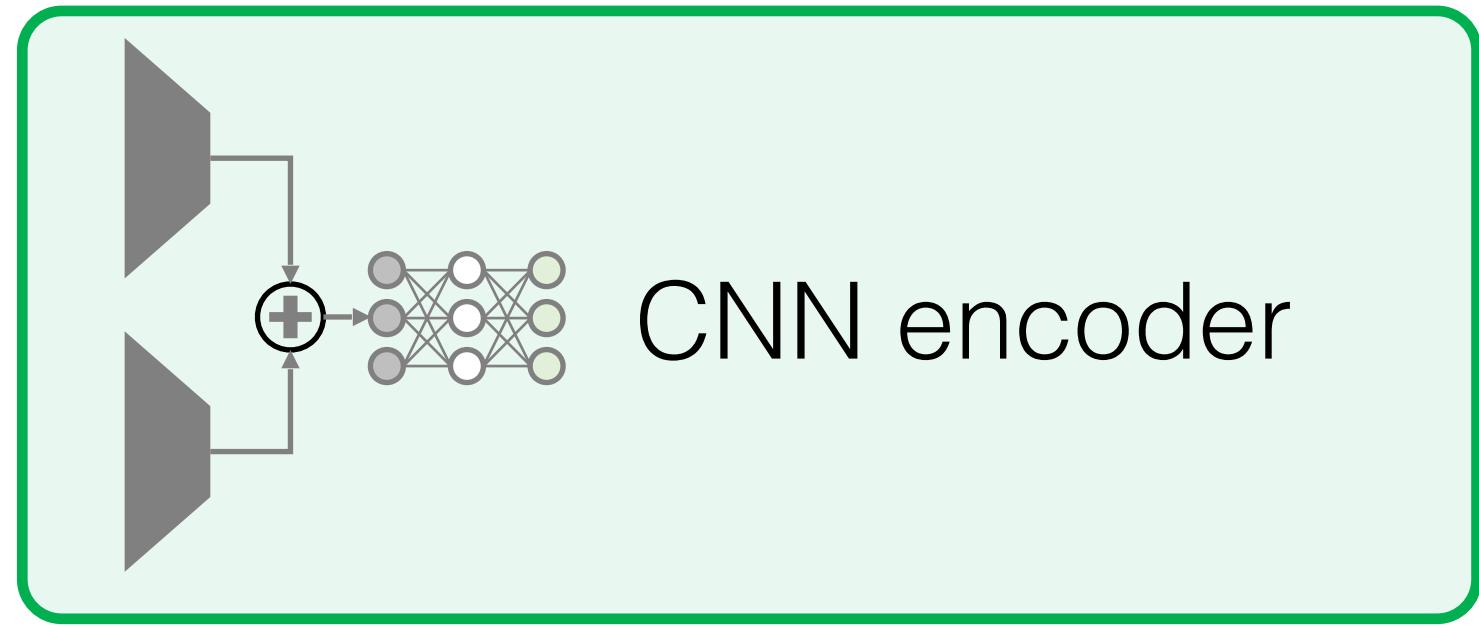
Encoding & Auto-Decoding



Encoding & Auto-Decoding



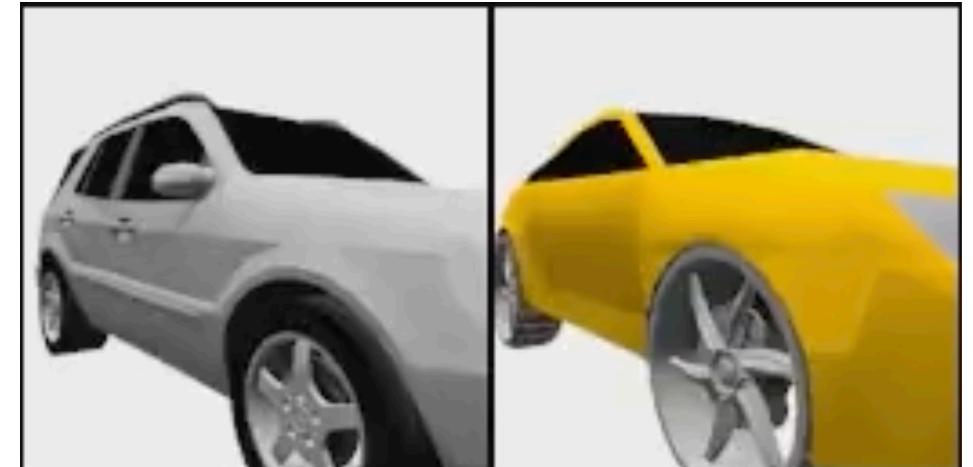
Out-of-distribution generalization



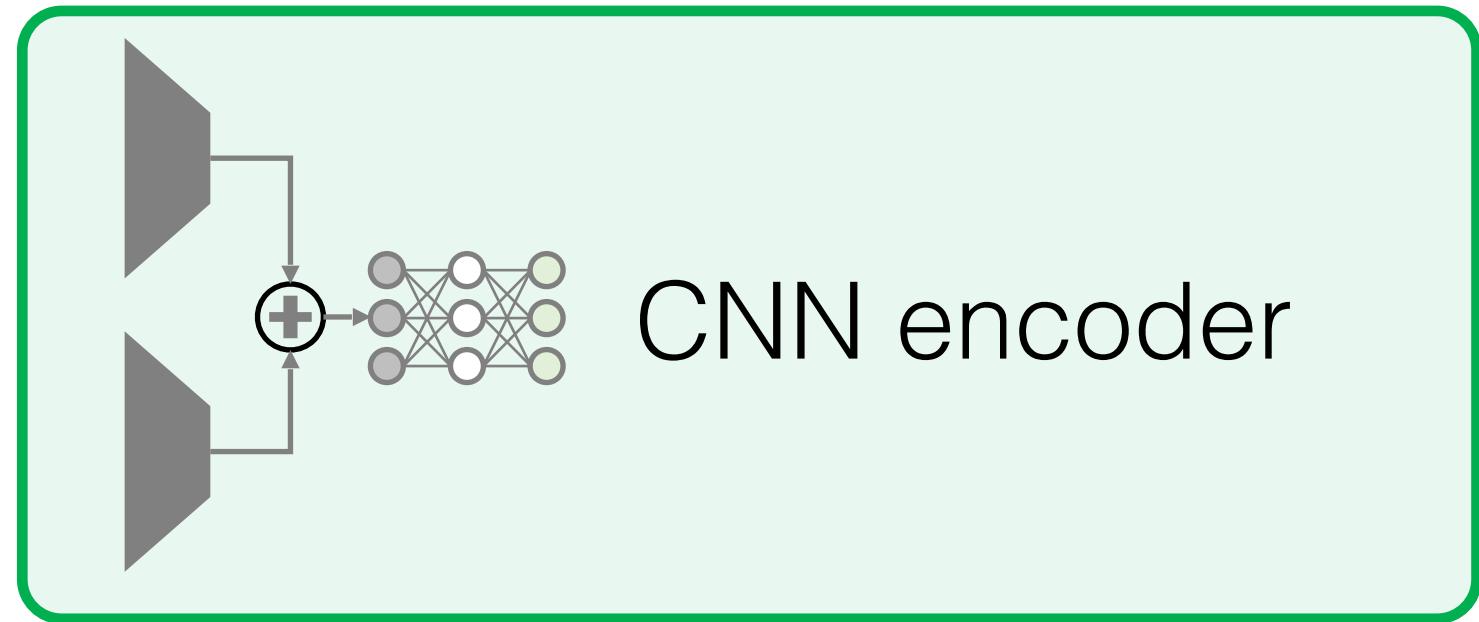
Input



Reconstructions



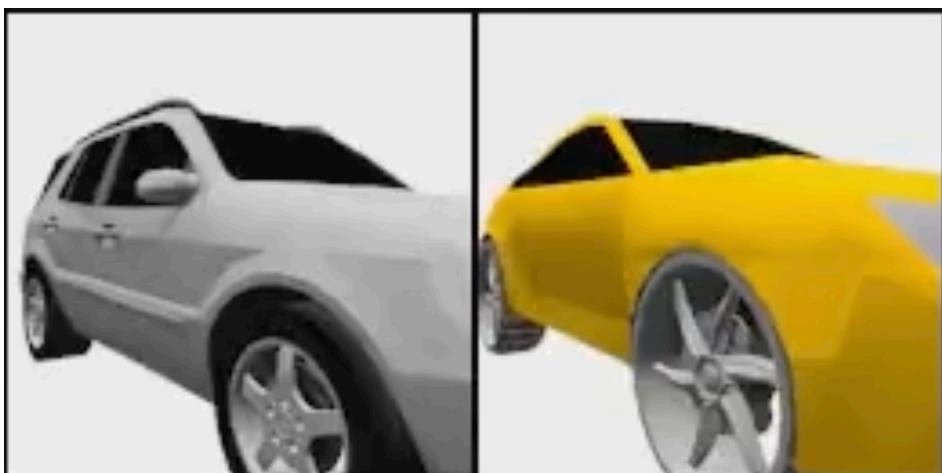
Out-of-distribution generalization



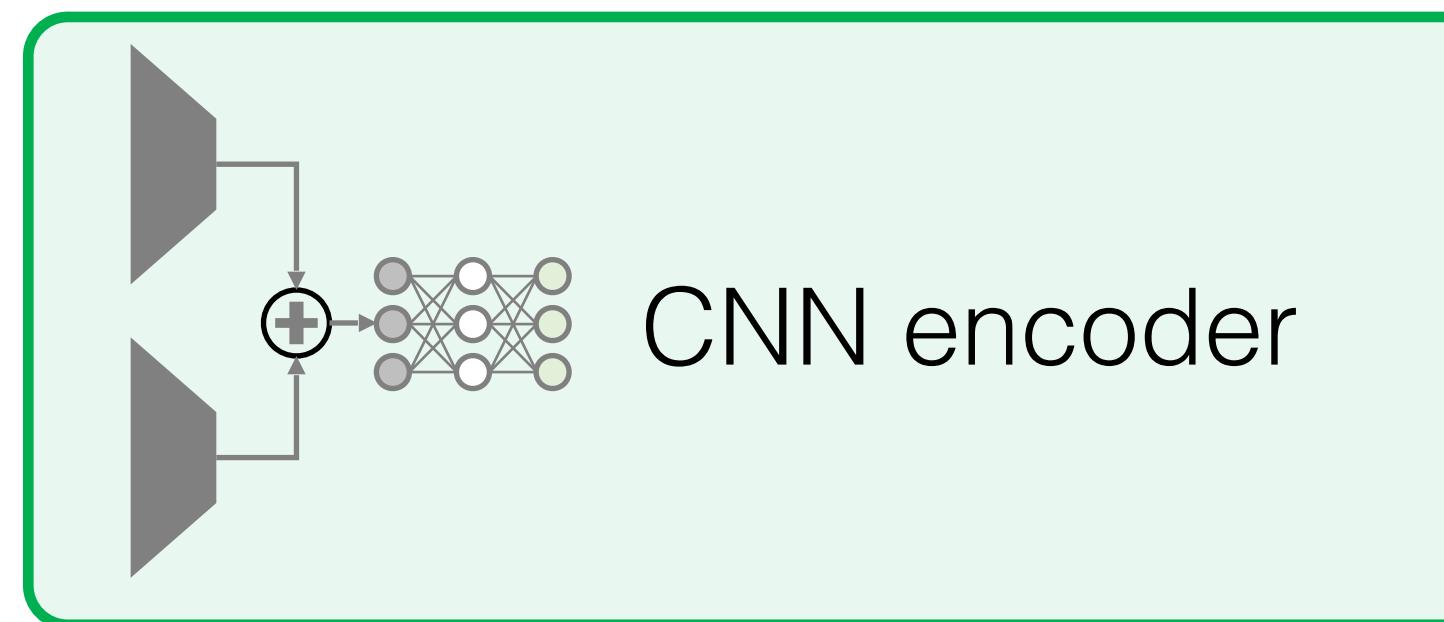
Input



Reconstructions



Out-of-distribution generalization

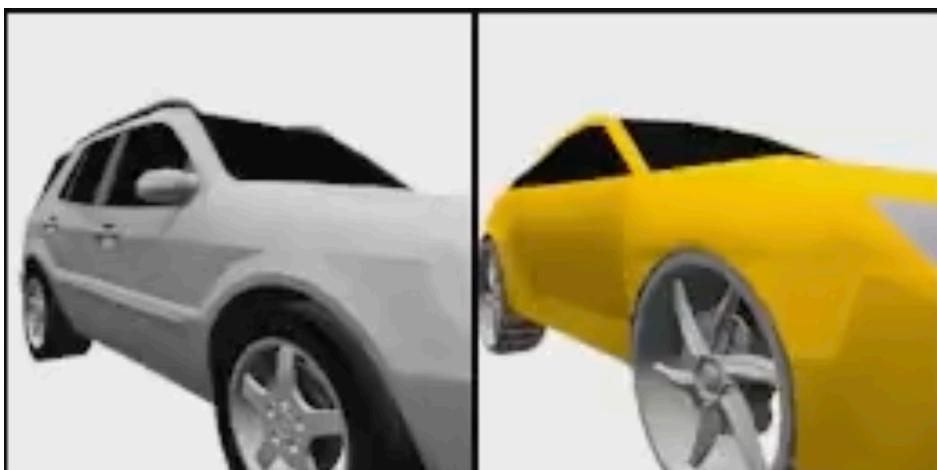


$$\operatorname{argmin}_z \| \operatorname{Render}(SRN, \xi) - \mathcal{I} \|$$

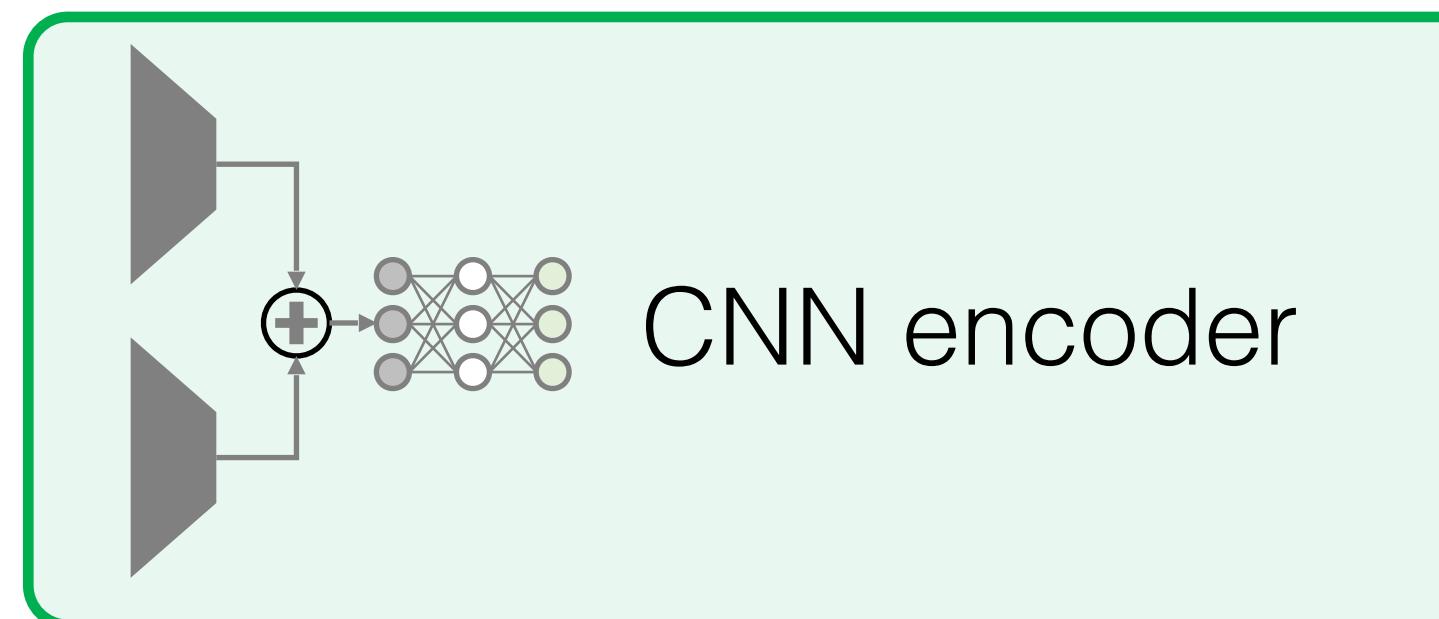
Input



Reconstructions

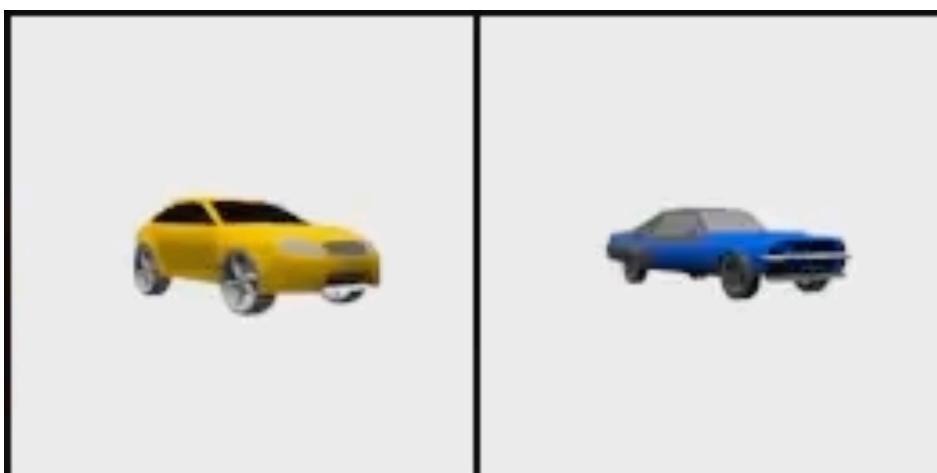


Out-of-distribution generalization

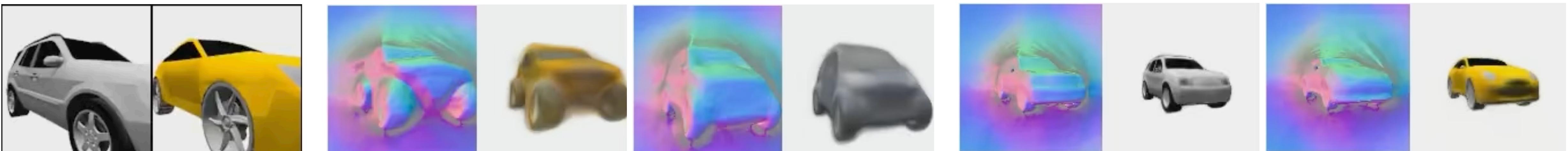
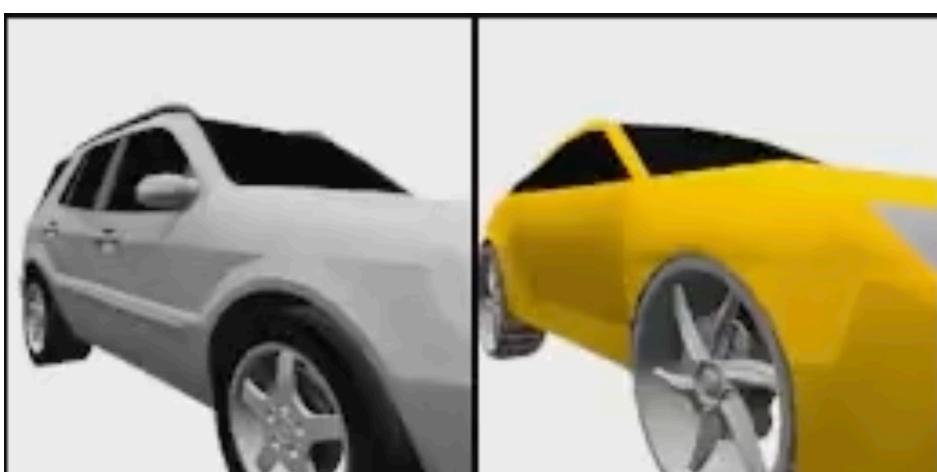


$$\operatorname{argmin}_z \| \operatorname{Render}(SRN, \xi) - \mathcal{I} \|$$

Input

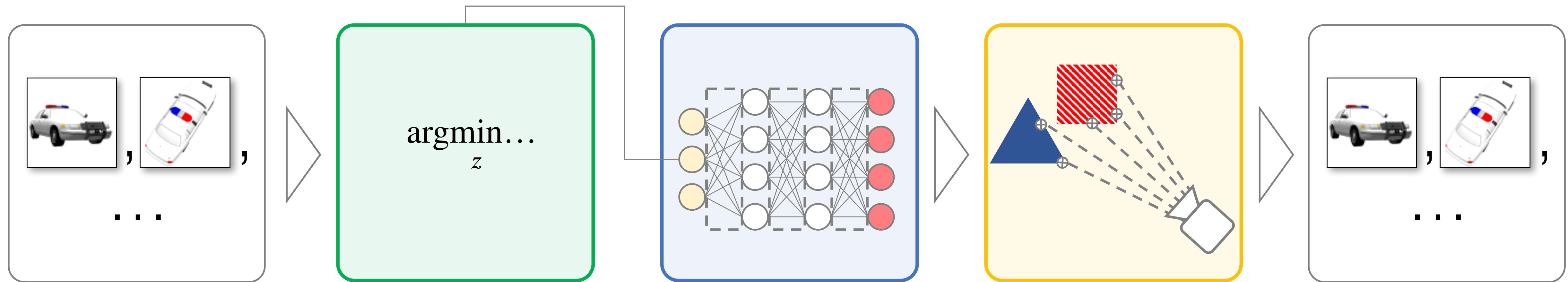


Reconstructions

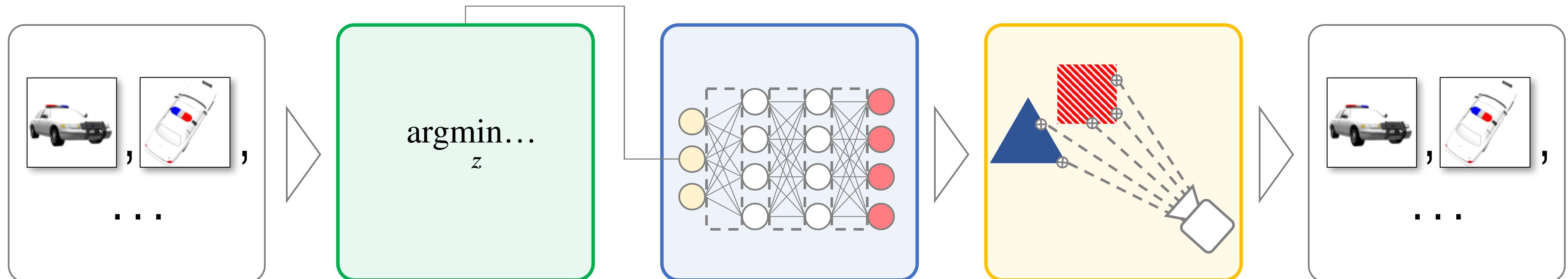


3D structure enables generalization to out-of-distribution camera poses!

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

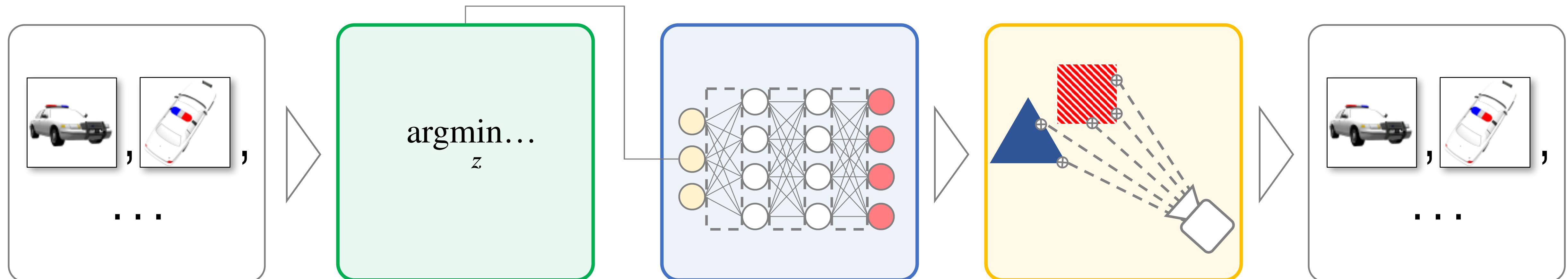


Scene Representation Networks (Sitzmann et al., Neurips 2019)



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



Single-shot reconstruction of held-out test objects

Input observation



SRNs (Ours)



Tatarchenko et al.
2015



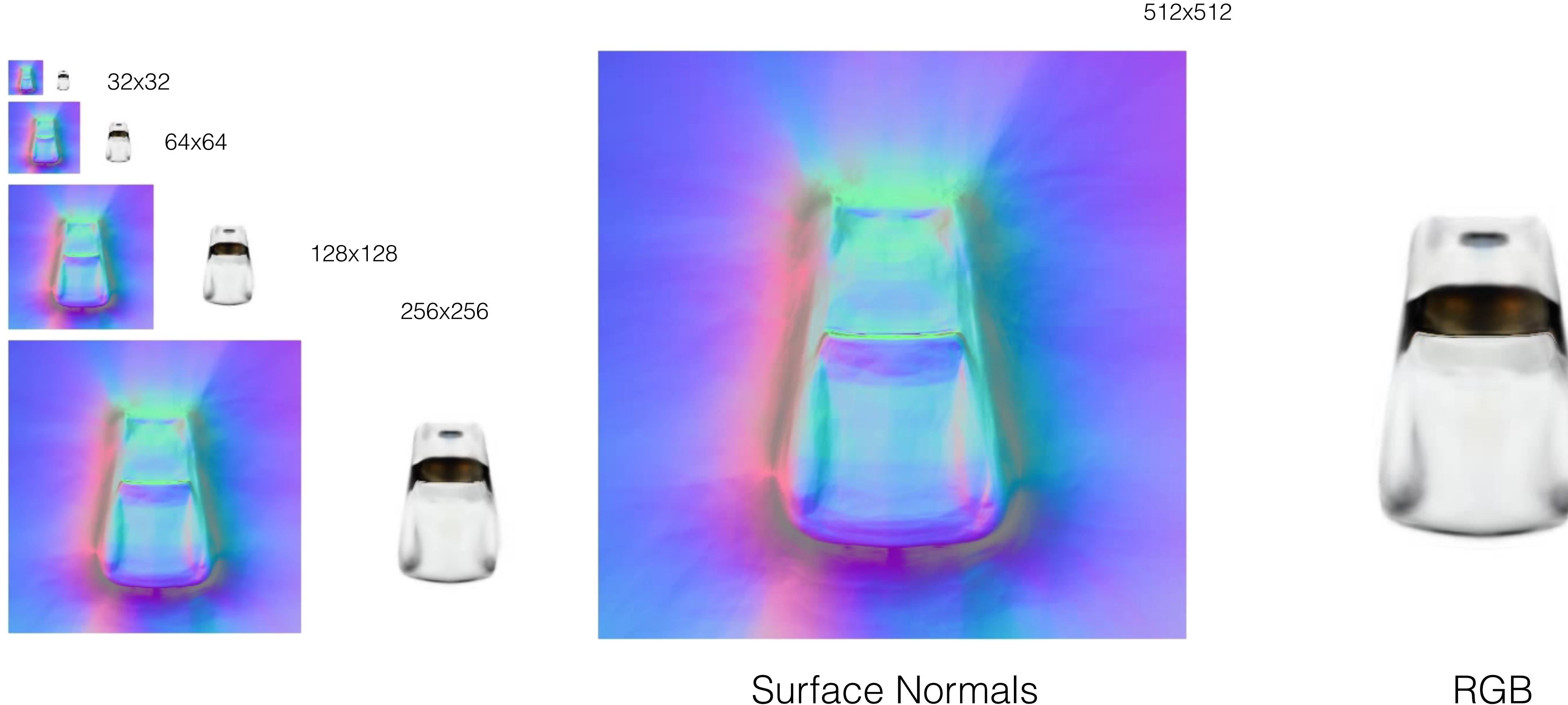
Worrall et al.
2017



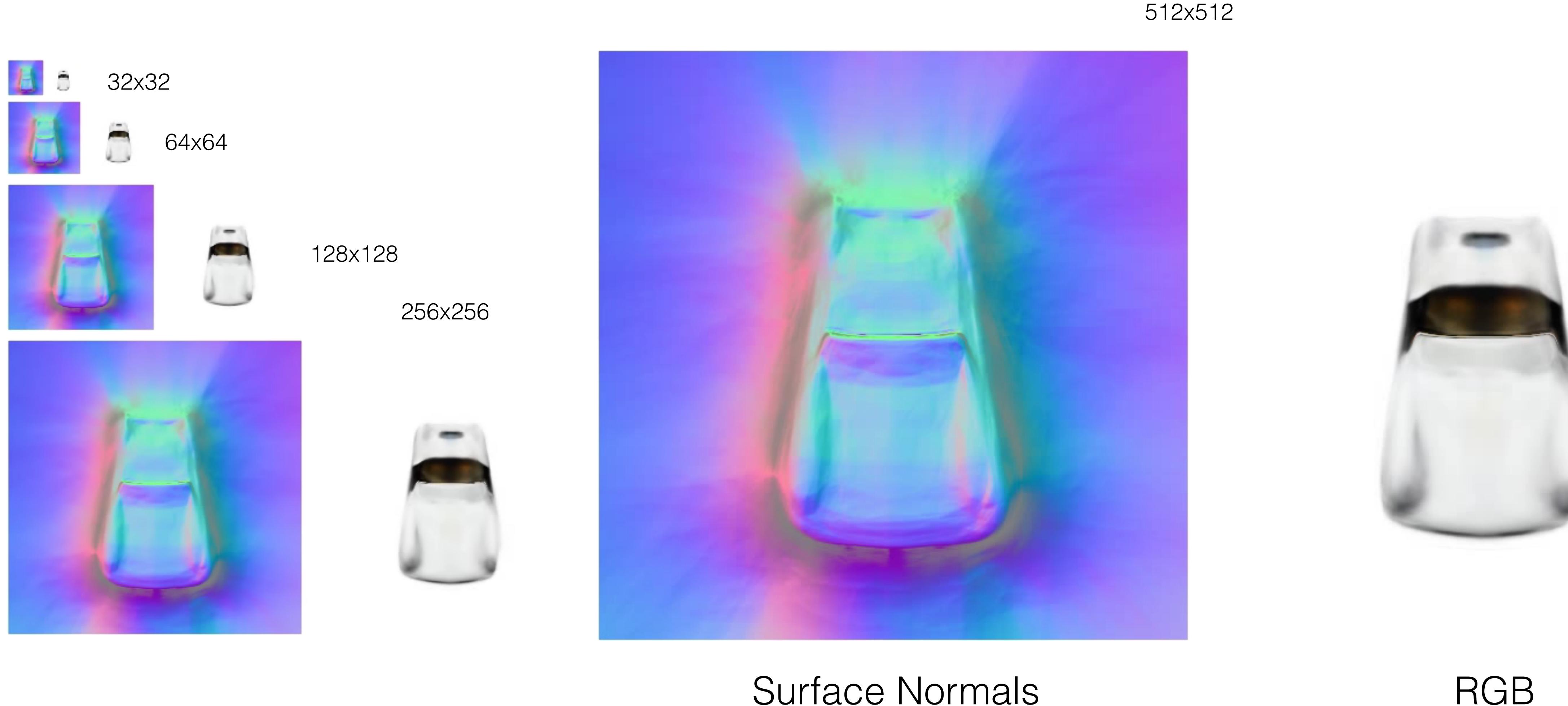
Deterministic
GQN, adapted
Eslami et al.
2018



Sampling at arbitrary resolutions



Sampling at arbitrary resolutions



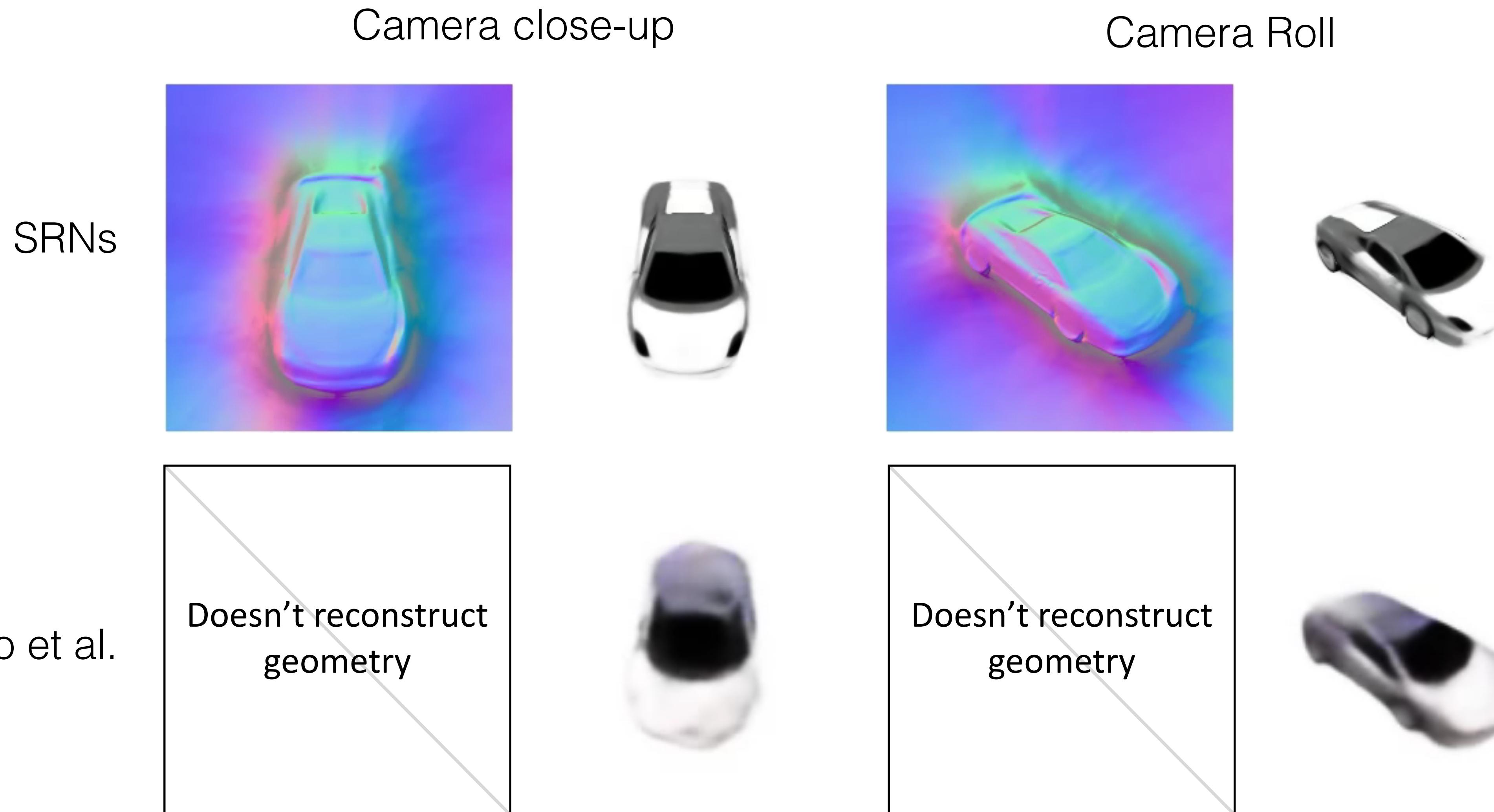
Generalization to unseen camera poses



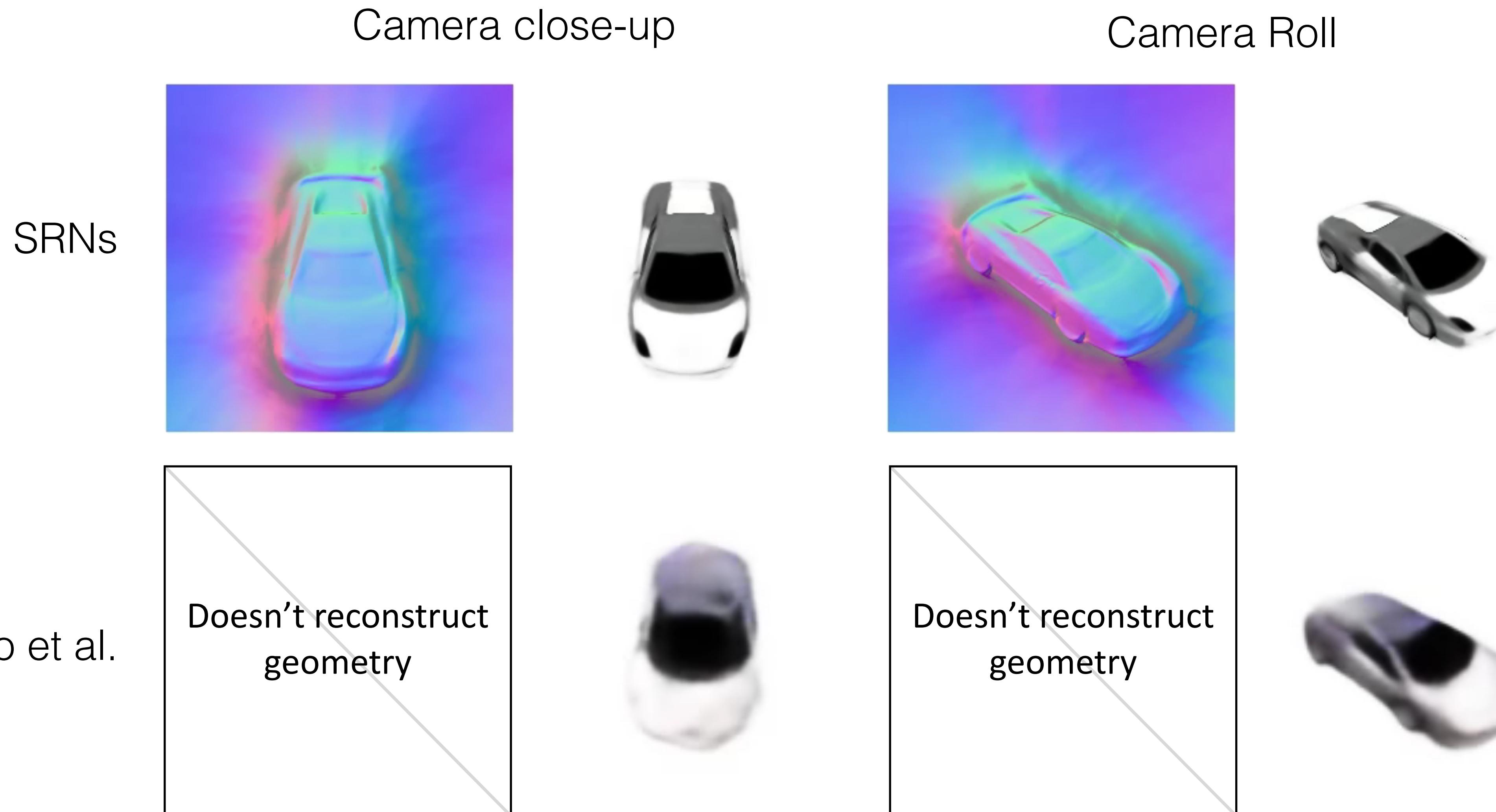
Generalization to unseen camera poses



Generalization to unseen camera poses



Generalization to unseen camera poses



Latent code interpolation



Surface Normals



RGB

Latent code interpolation

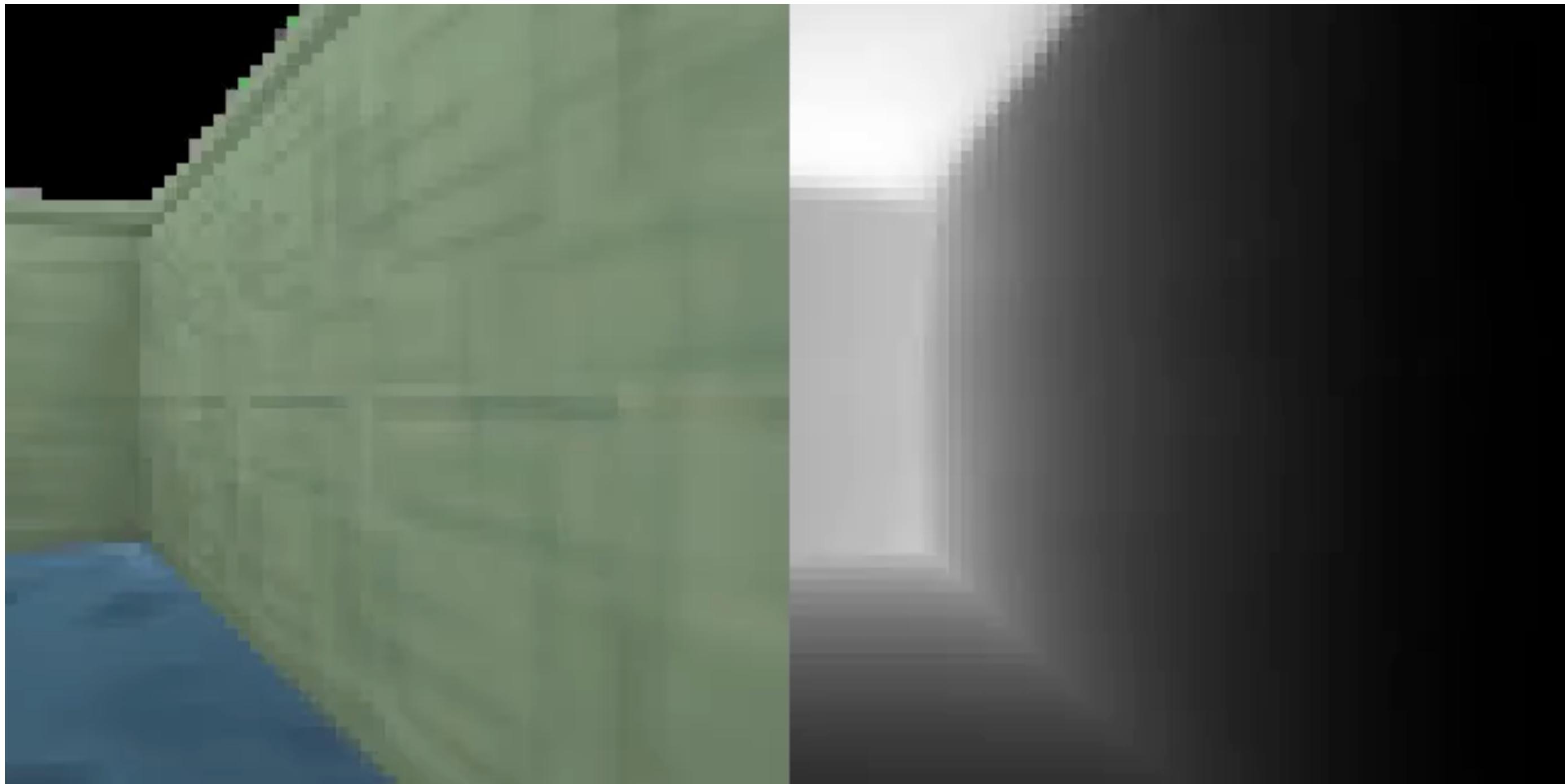


Surface Normals

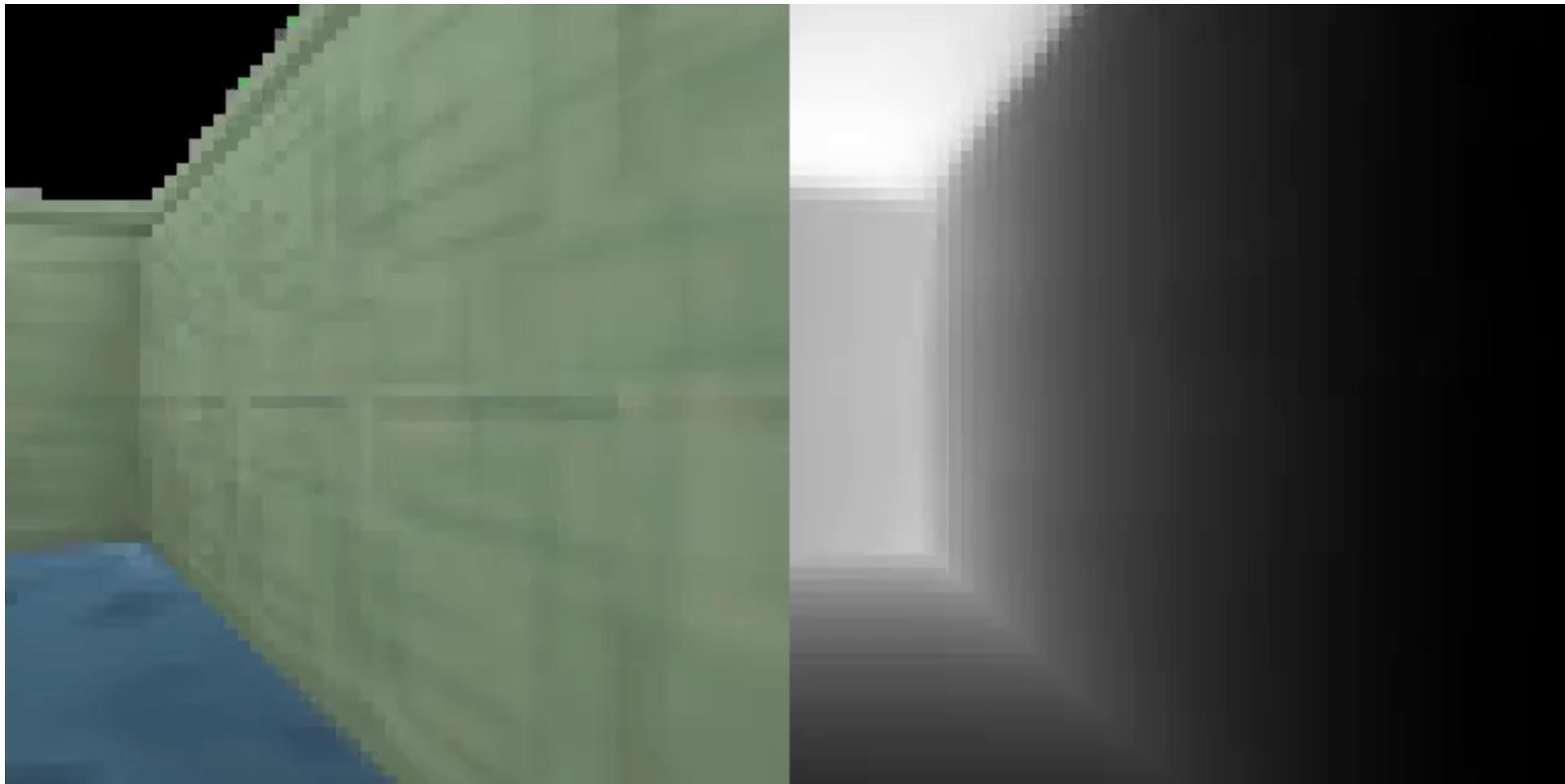


RGB

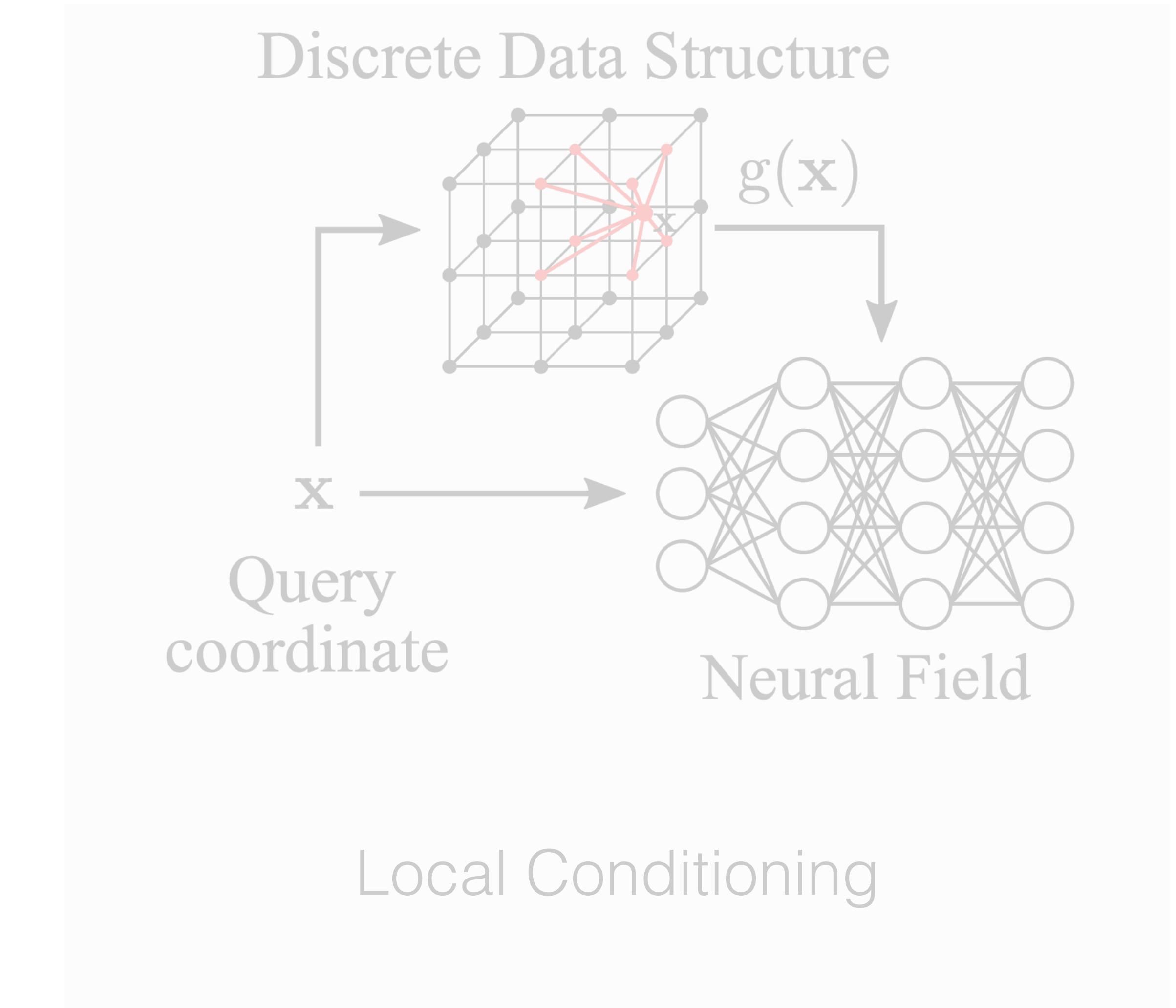
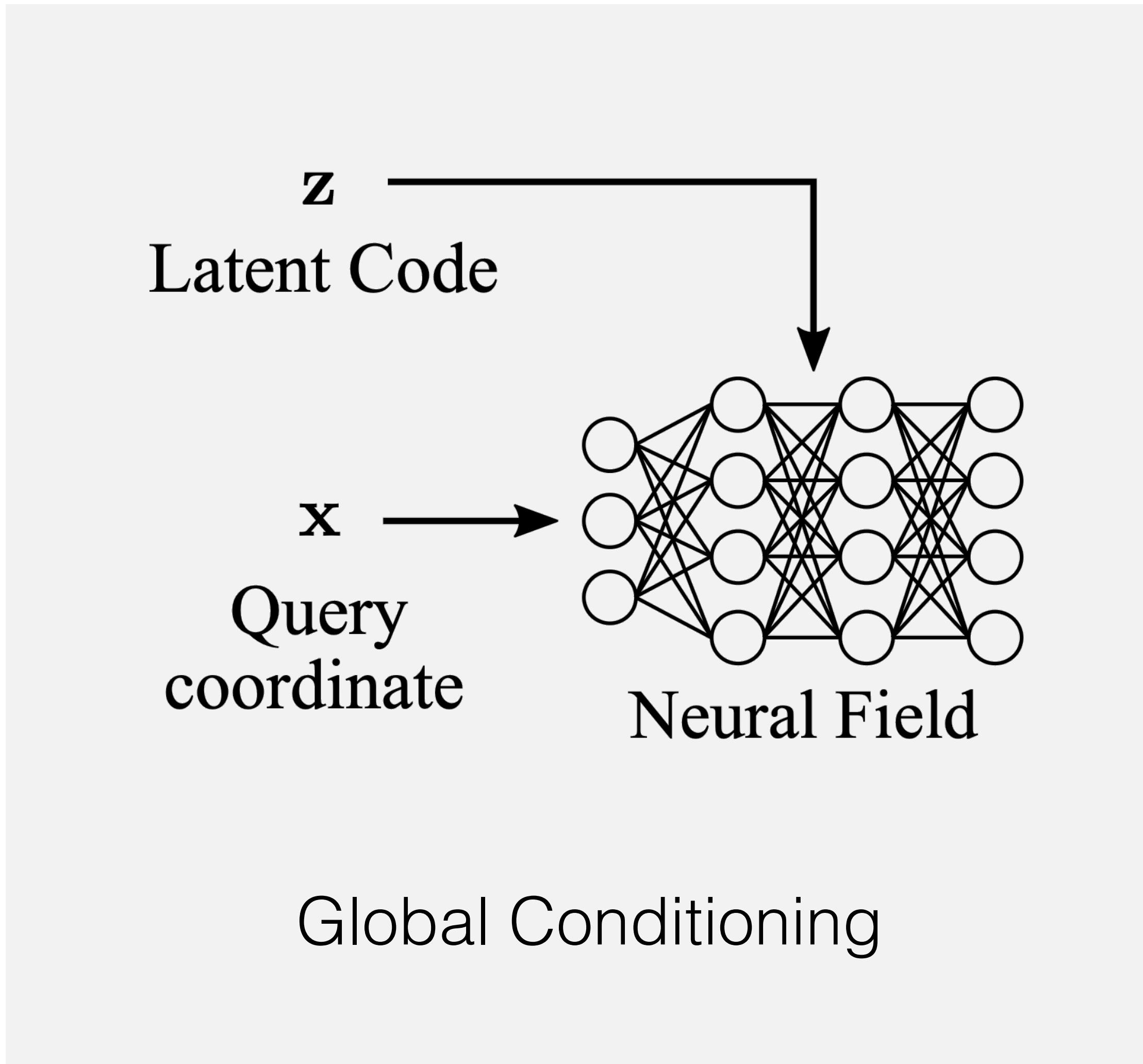
Can generalize across simple room-scale scenes.



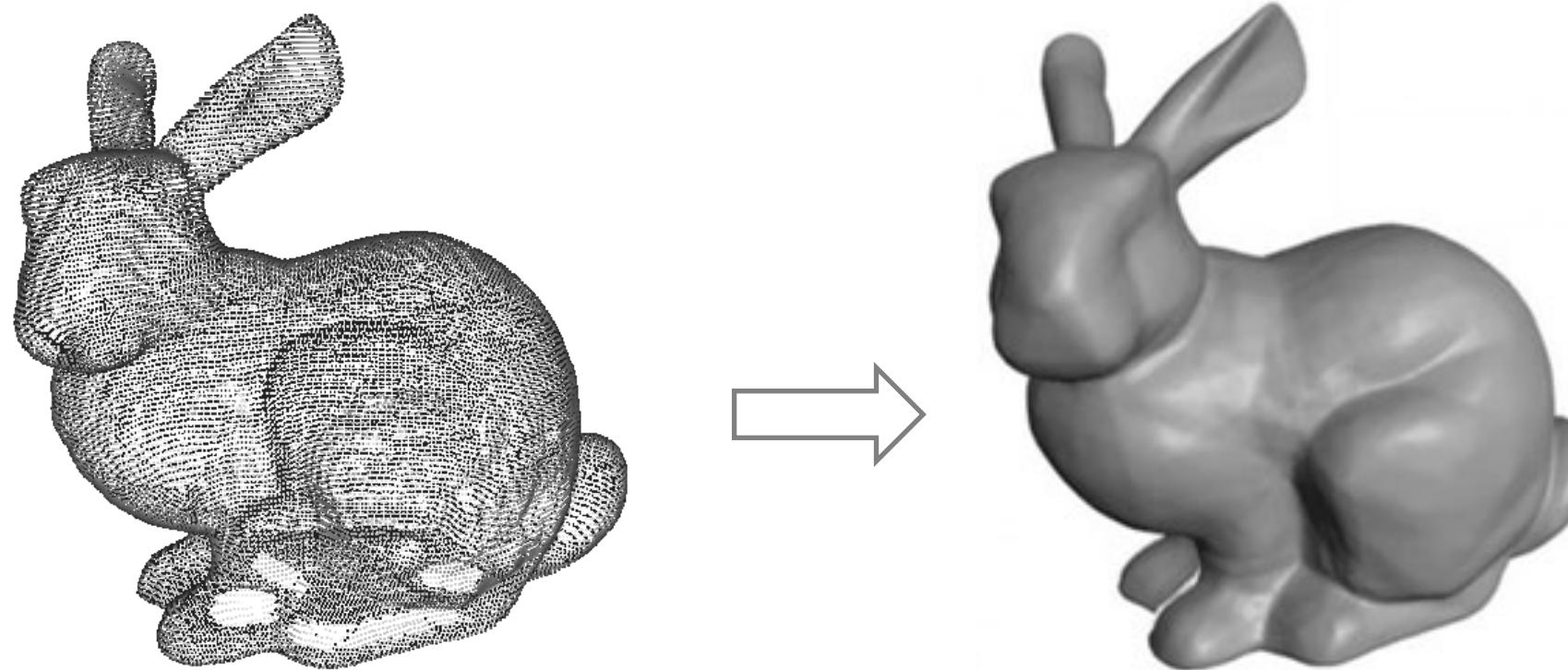
Can generalize across simple room-scale scenes.



Global Conditioning: Single Latent Code for whole 3D Scene



Global Latent Codes: Enables reconstruction from partial observations!



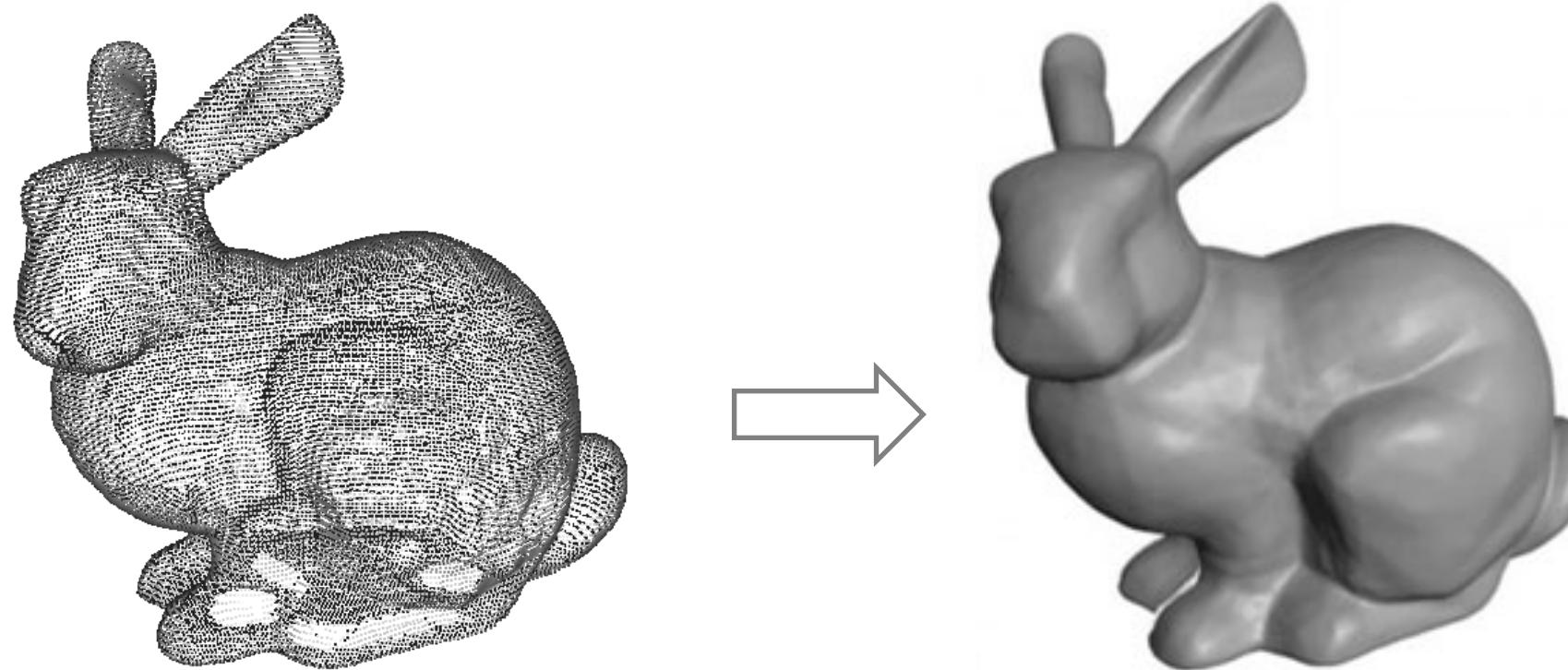
DeepSDF, Occupancy Networks, IM-Net



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!



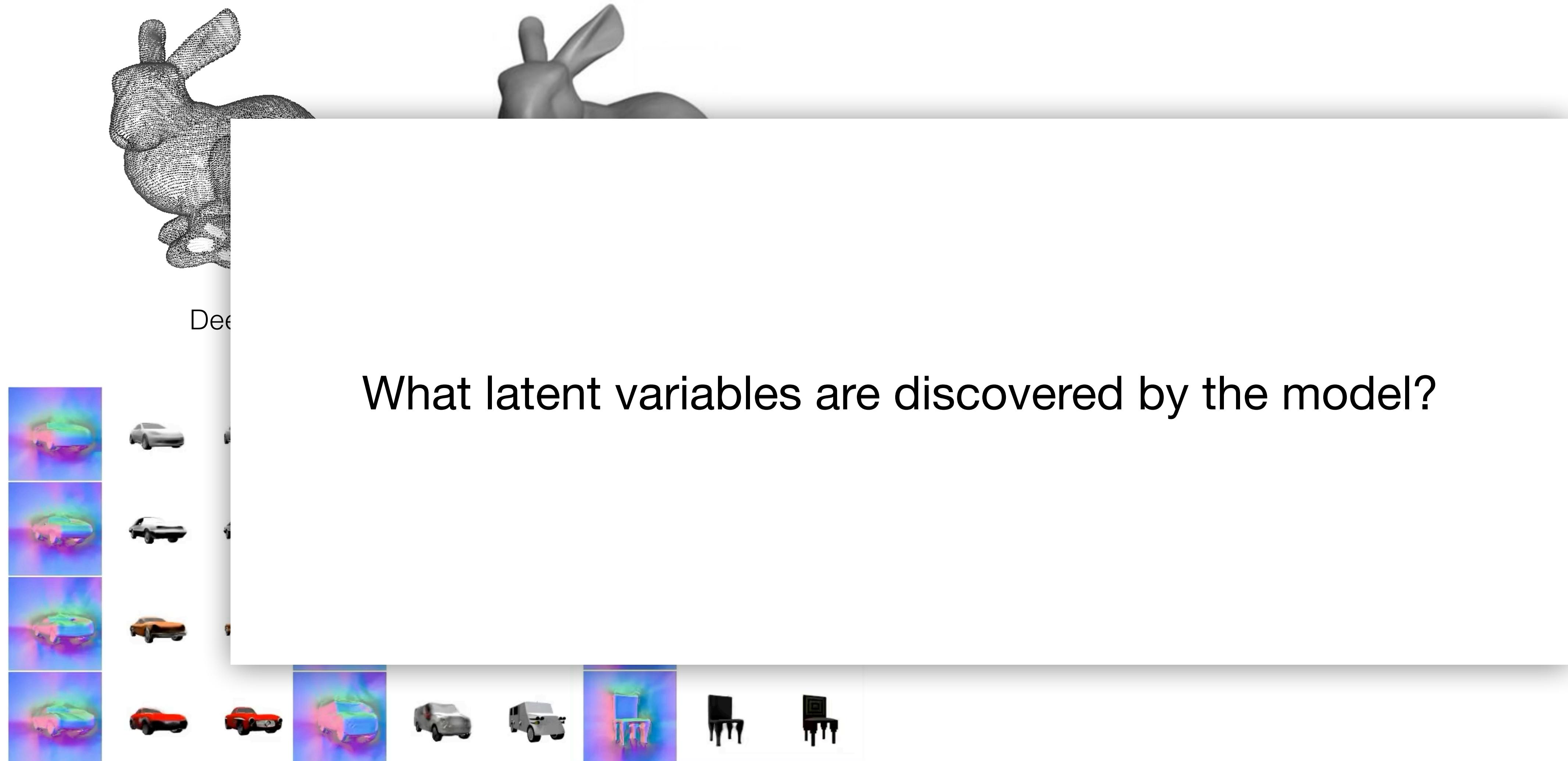
DeepSDF, Occupancy Networks, IM-Net



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

Differential Volumetric Rendering,
Niemeyer et al., CVPR 2020

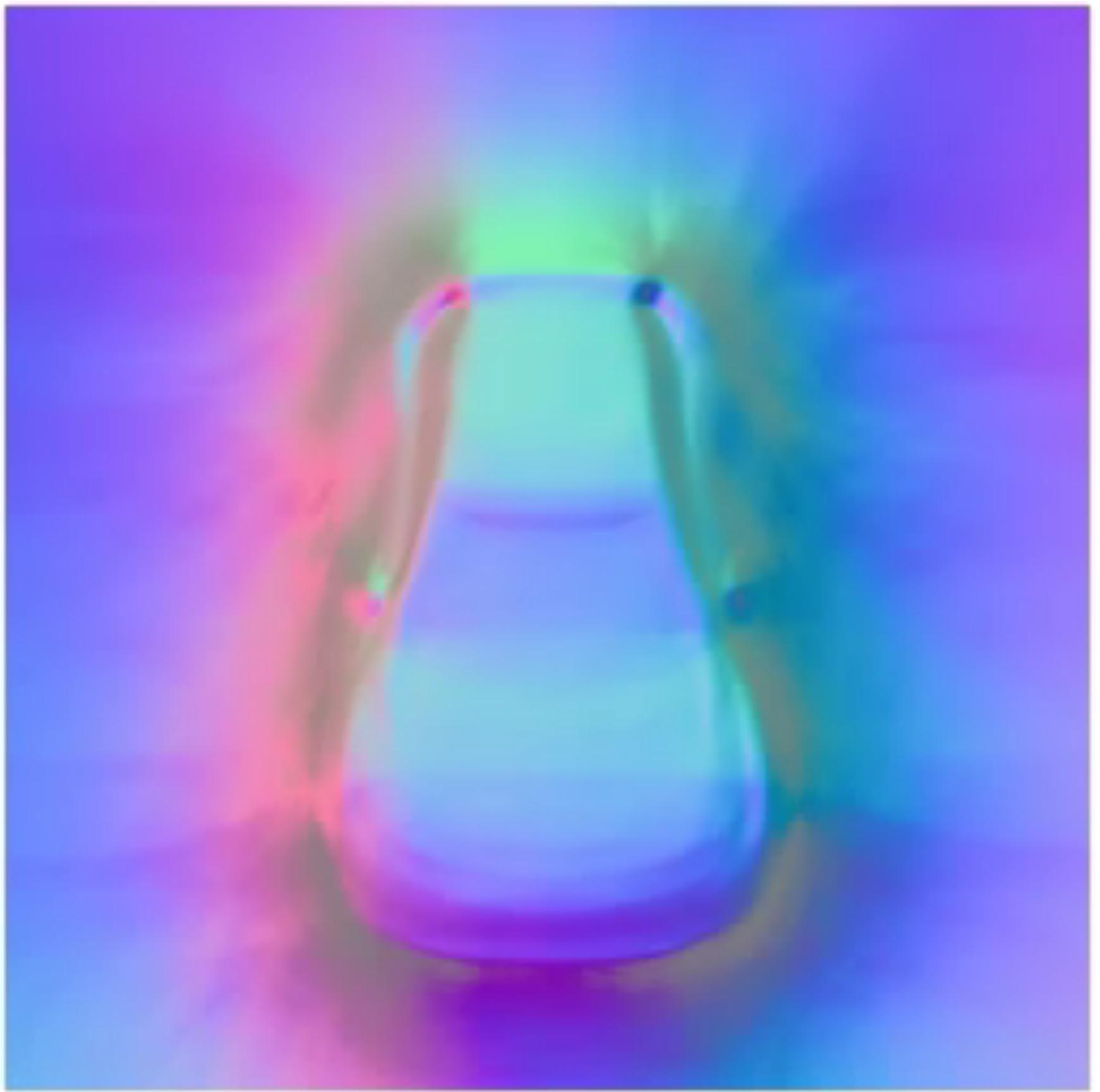
Global Latent Codes: Enables reconstruction from partial observations!



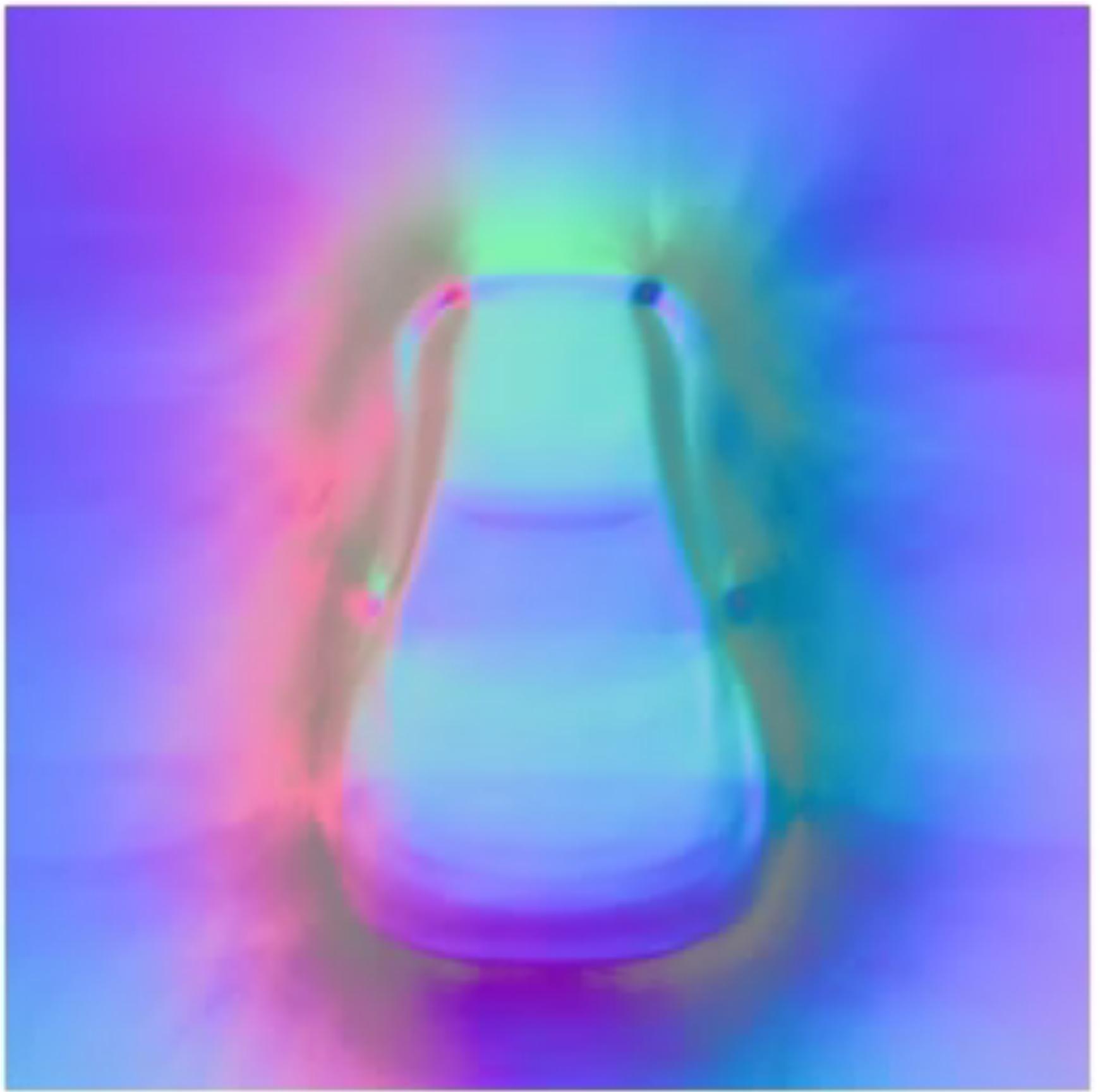
Scene Representation Networks: Continuous
3D-Structure-Aware Neural Scene Representations, NeurIPS 2019.

Differential Volumetric Rendering,
Niemeyer et al., CVPR 2020

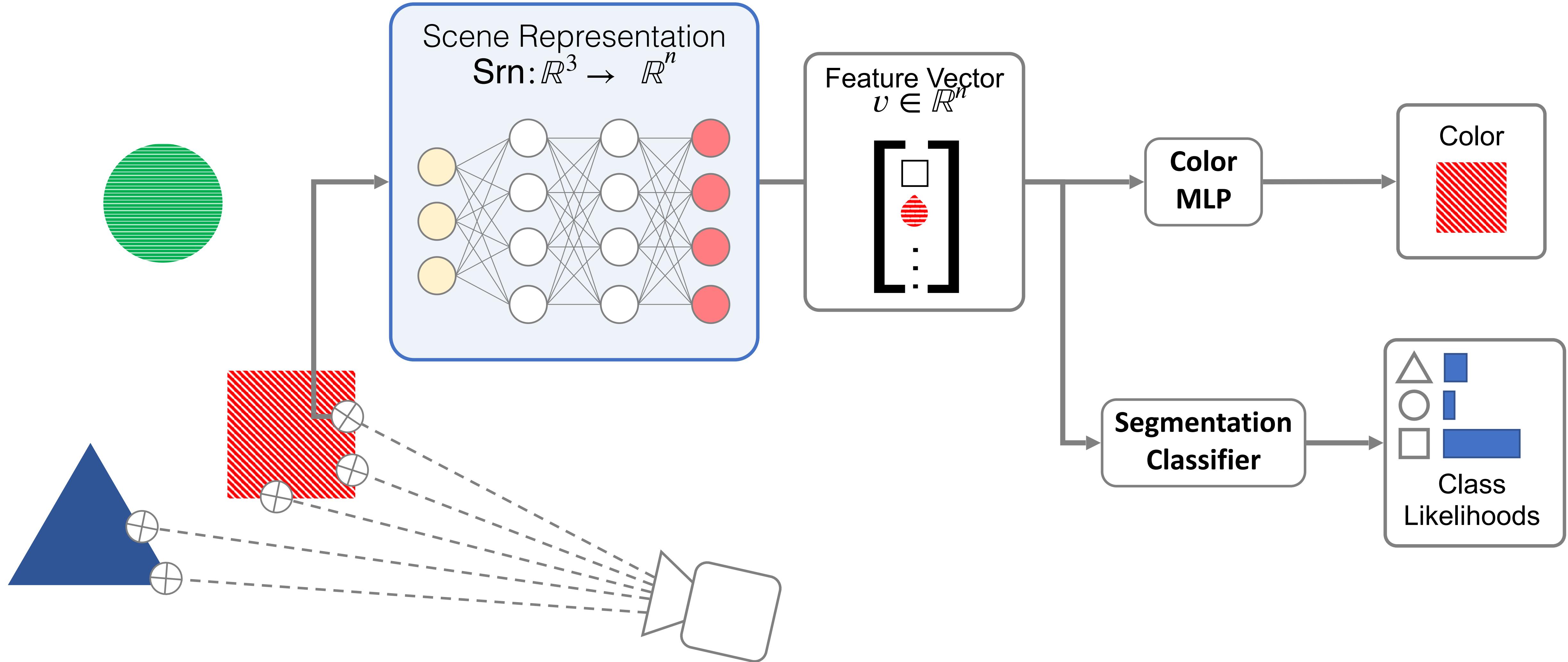
Latent Space Interpolation



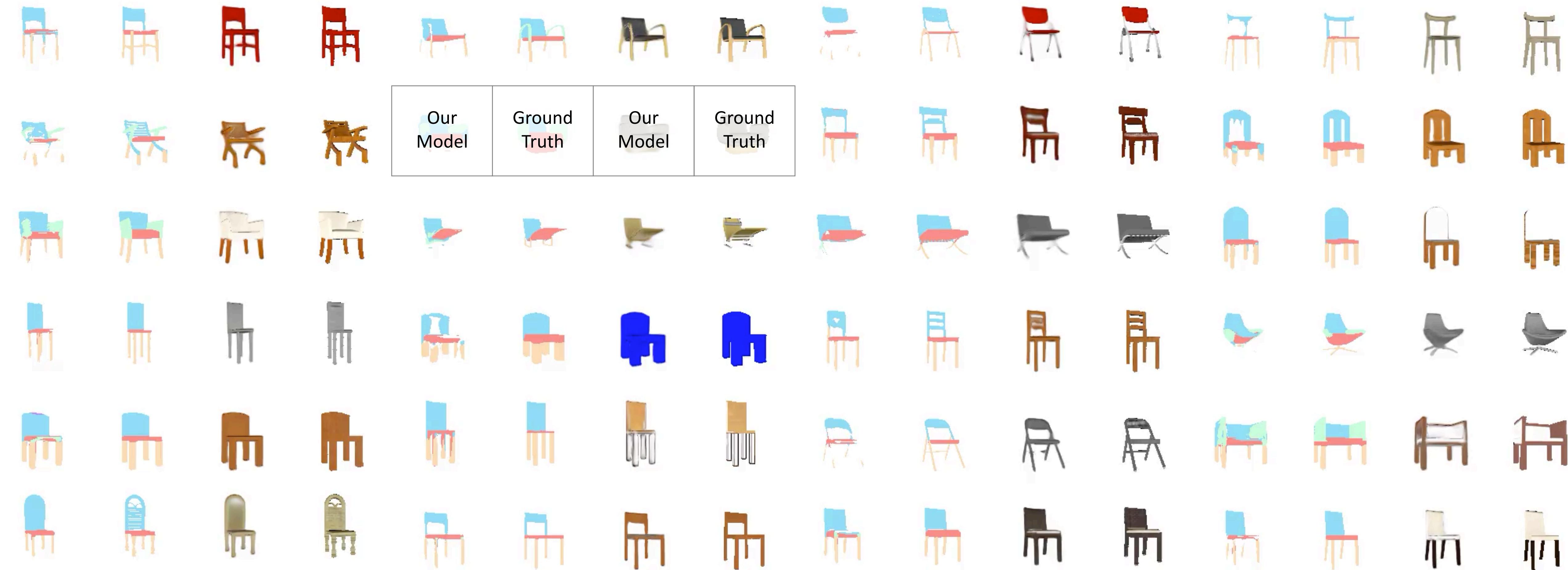
Latent Space Interpolation



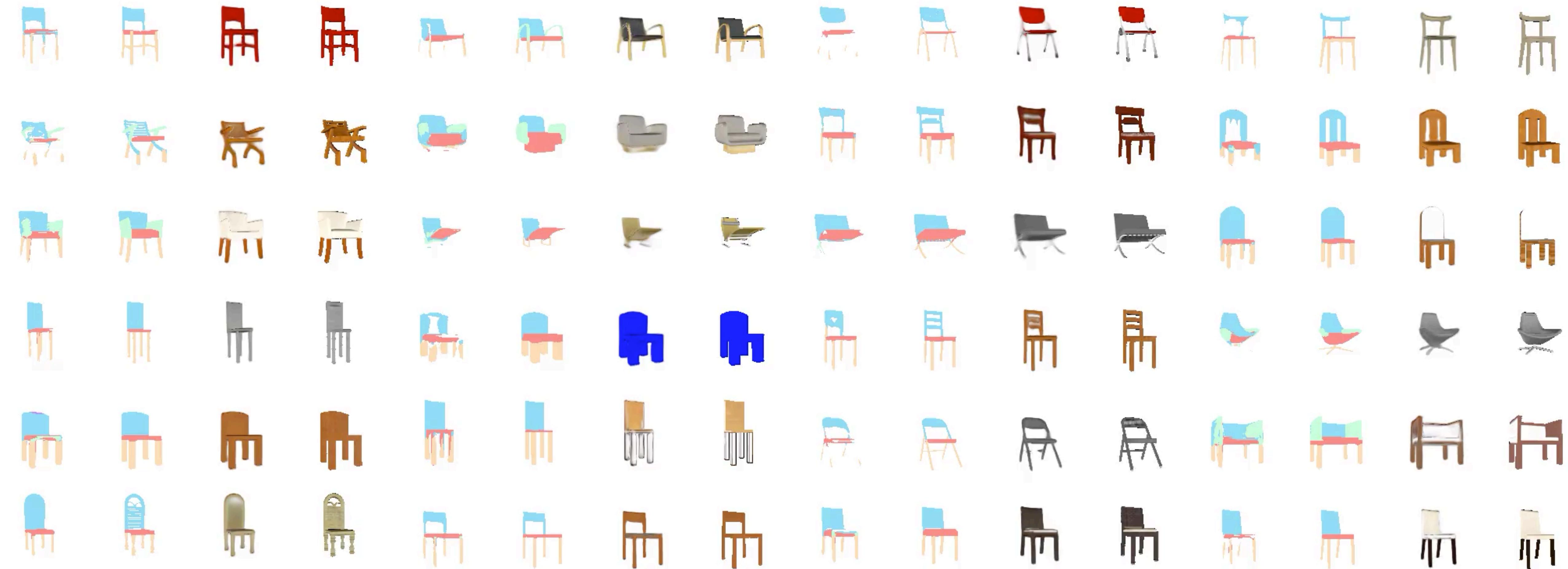
Learned features enable semantic segmentation from few labels.



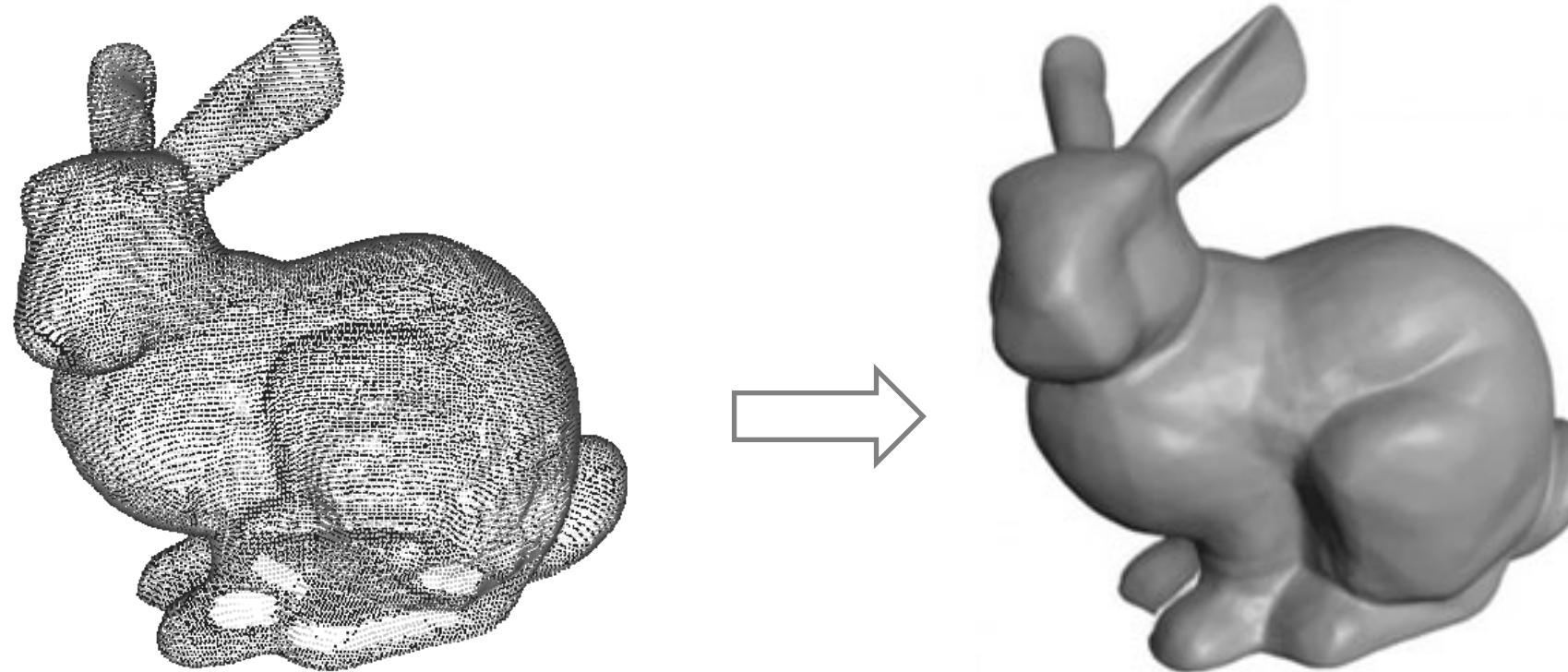
Learned features enable semantic segmentation from few labels.



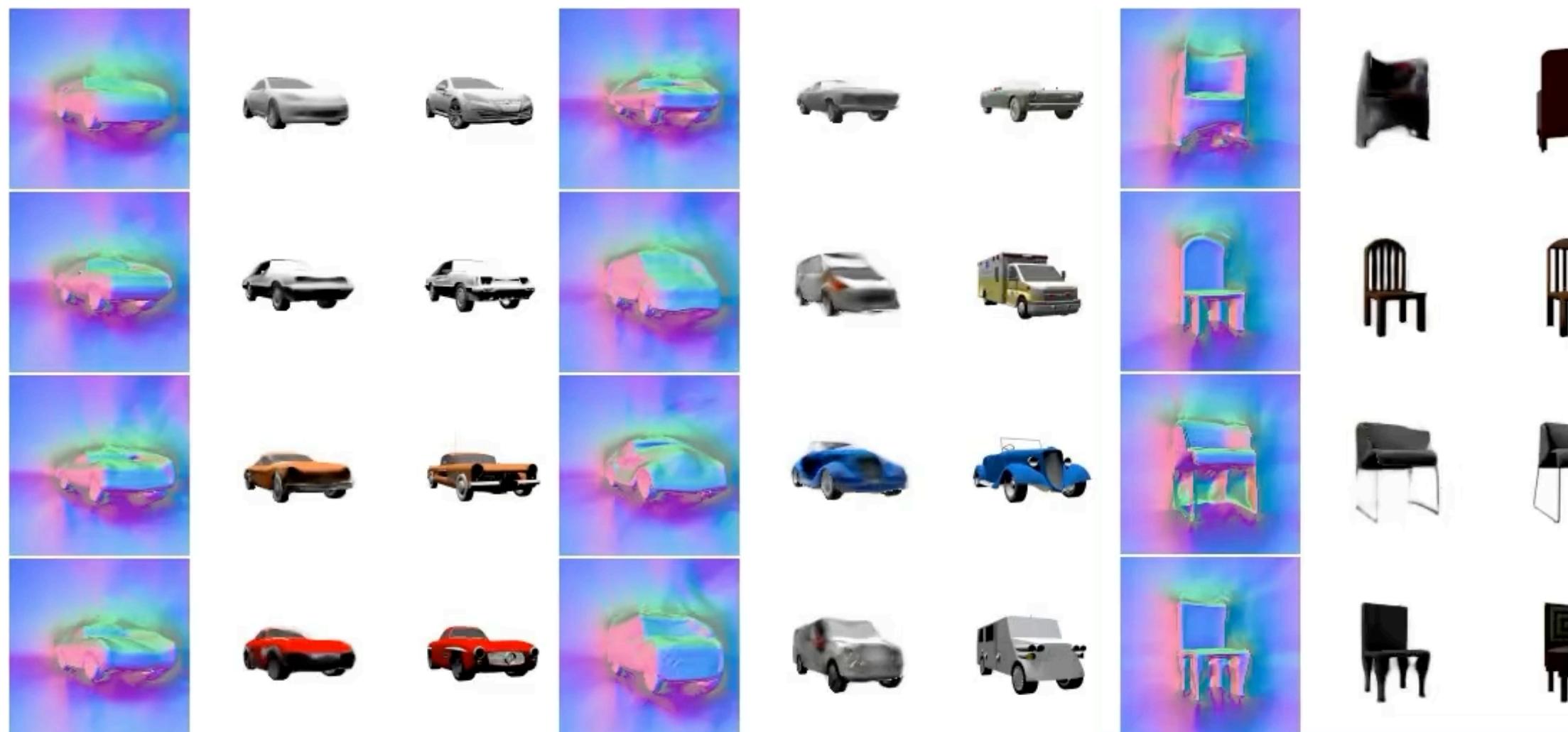
Learned features enable semantic segmentation from few labels.



Global Latent Codes: Enables reconstruction from partial observations!



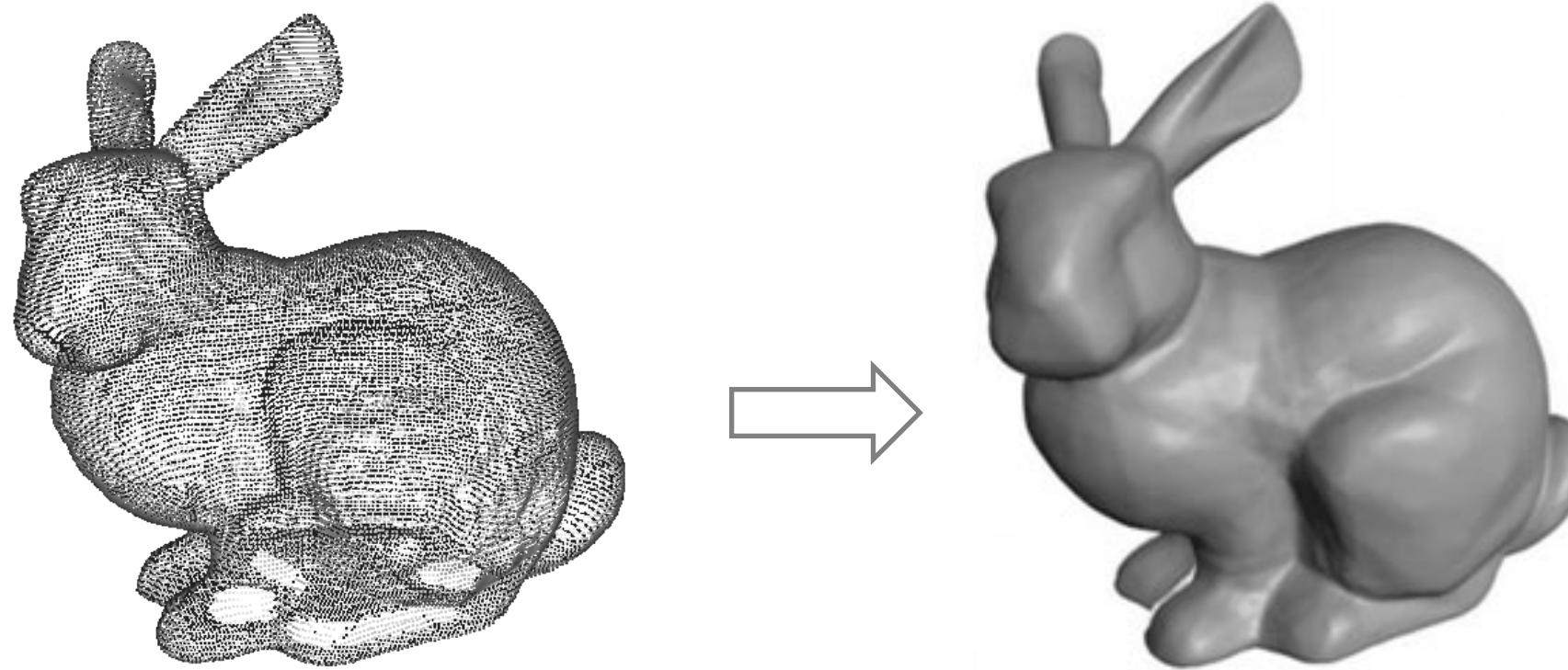
DeepSDF, Occupancy Networks, IM-Net



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!



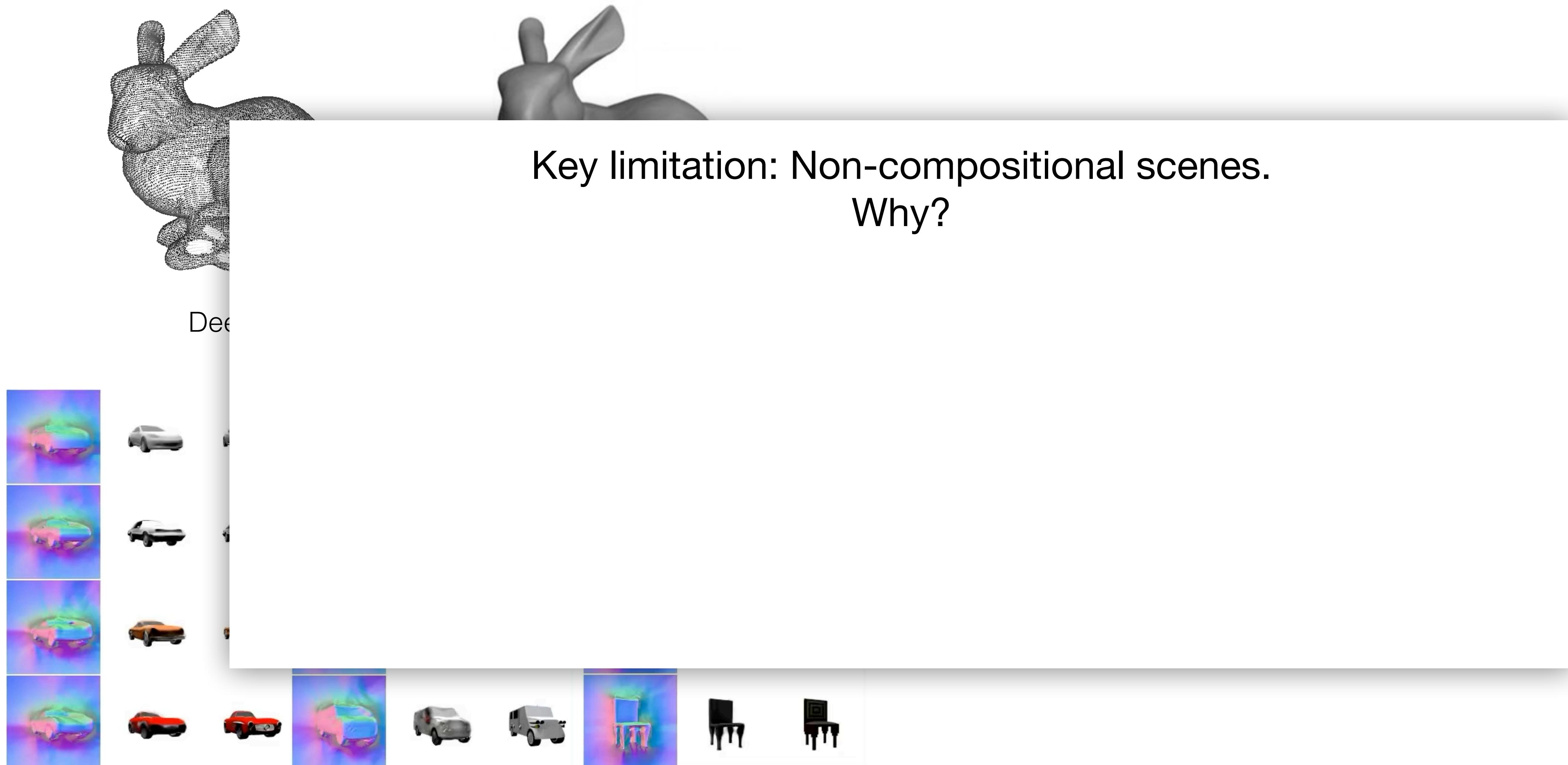
DeepSDF, Occupancy Networks, IM-Net



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

Differential Volumetric Rendering,
Niemeyer et al., CVPR 2020

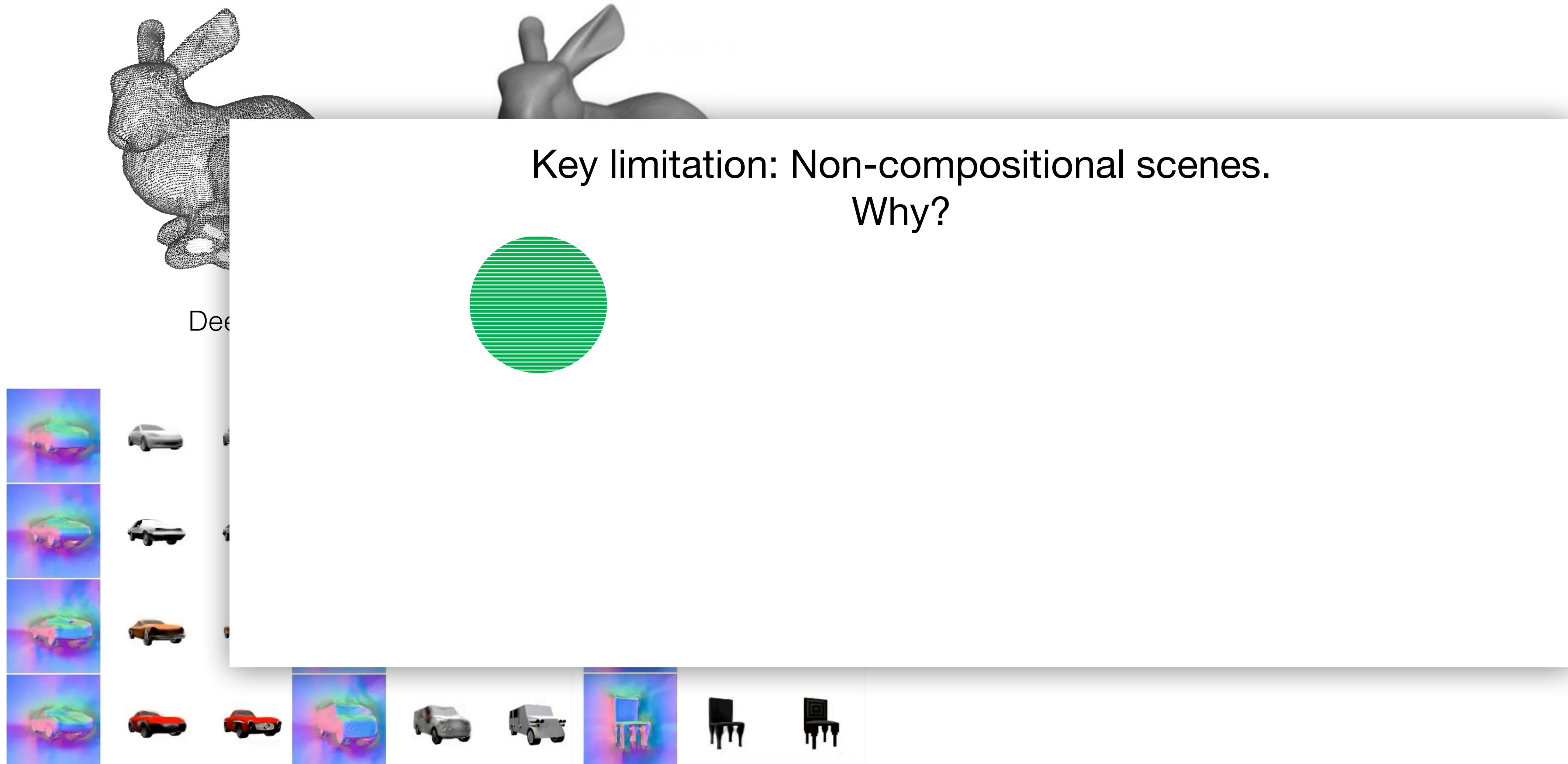
Global Latent Codes: Enables reconstruction from partial observations!



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

Differential Volumetric Rendering, Niemeyer et al., CVPR 2020

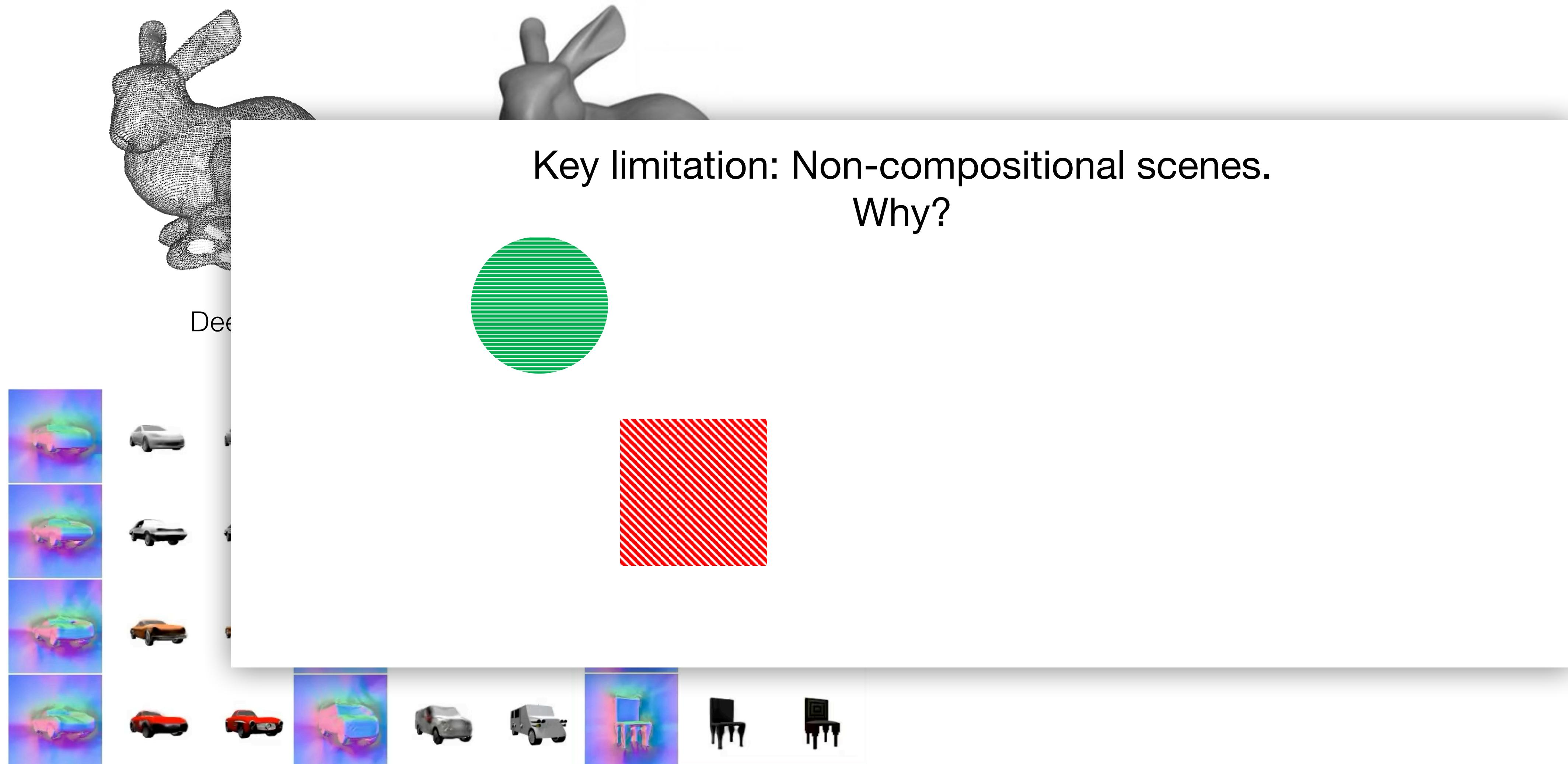
Global Latent Codes: Enables reconstruction from partial observations!



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

Differential Volumetric Rendering,
Niemeyer et al., CVPR 2020

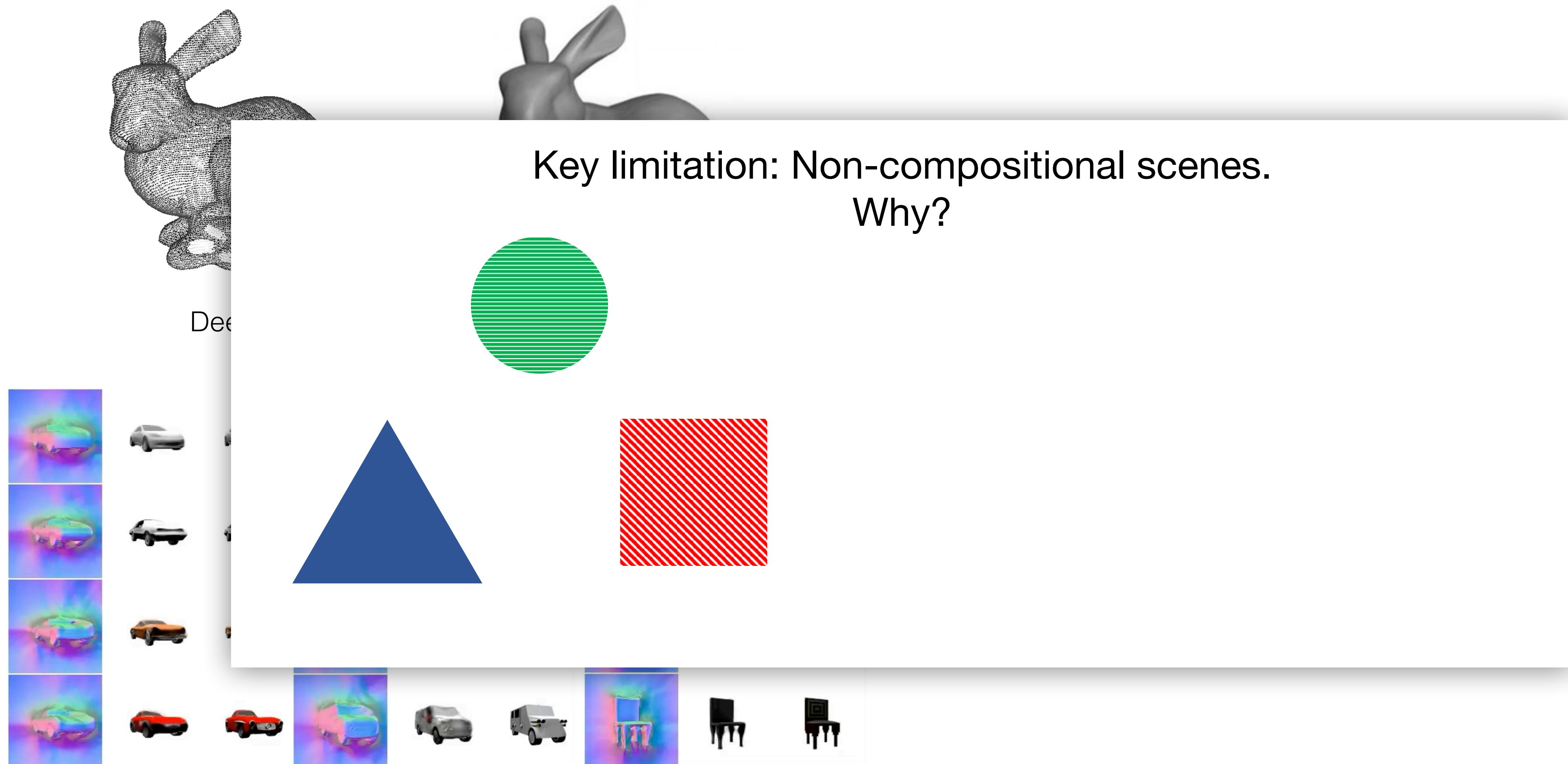
Global Latent Codes: Enables reconstruction from partial observations!



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!



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

Differential Volumetric Rendering, Niemeyer et al., CVPR 2020

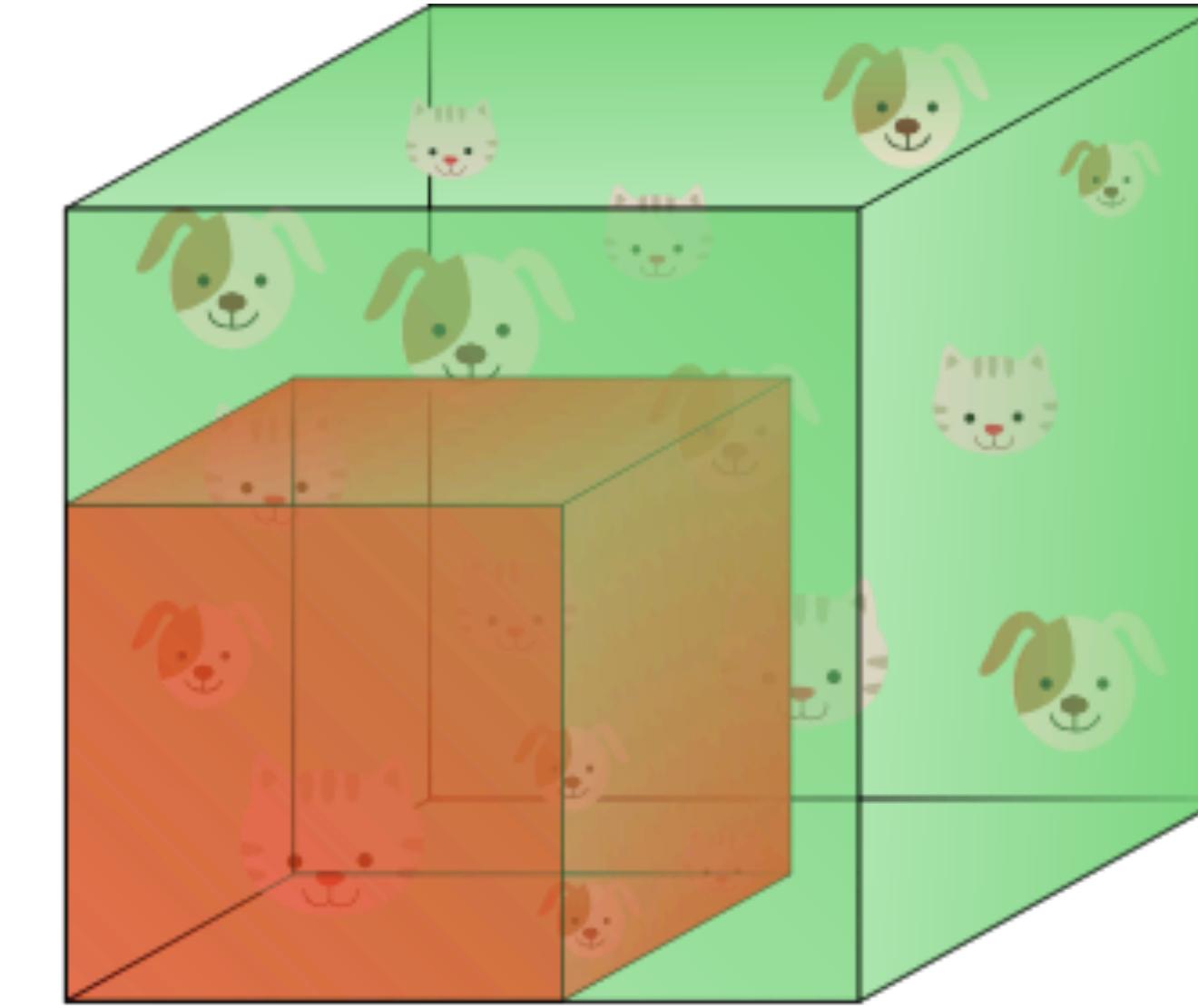
Reminder: Curse of Dimensionality



1D



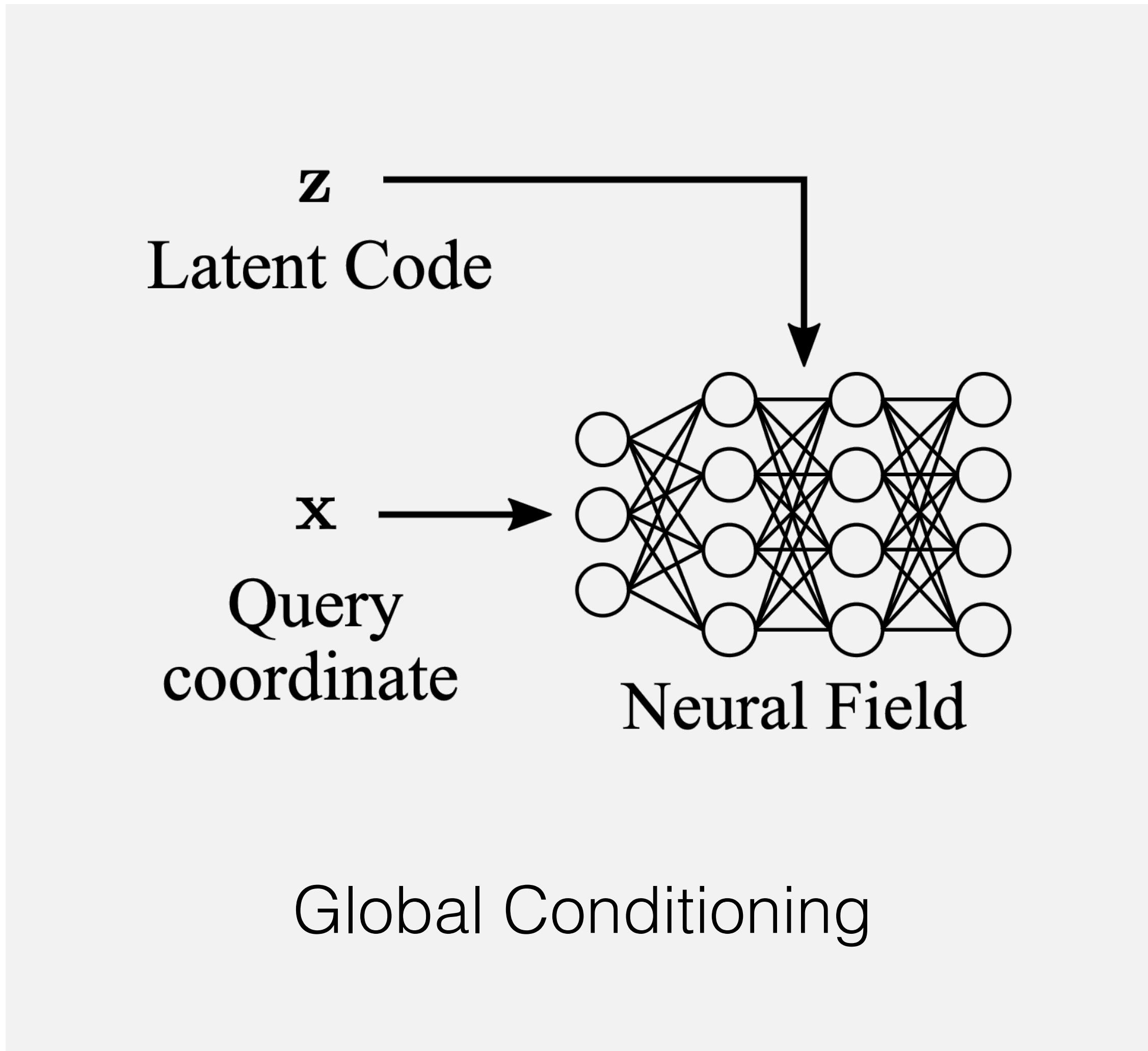
2D



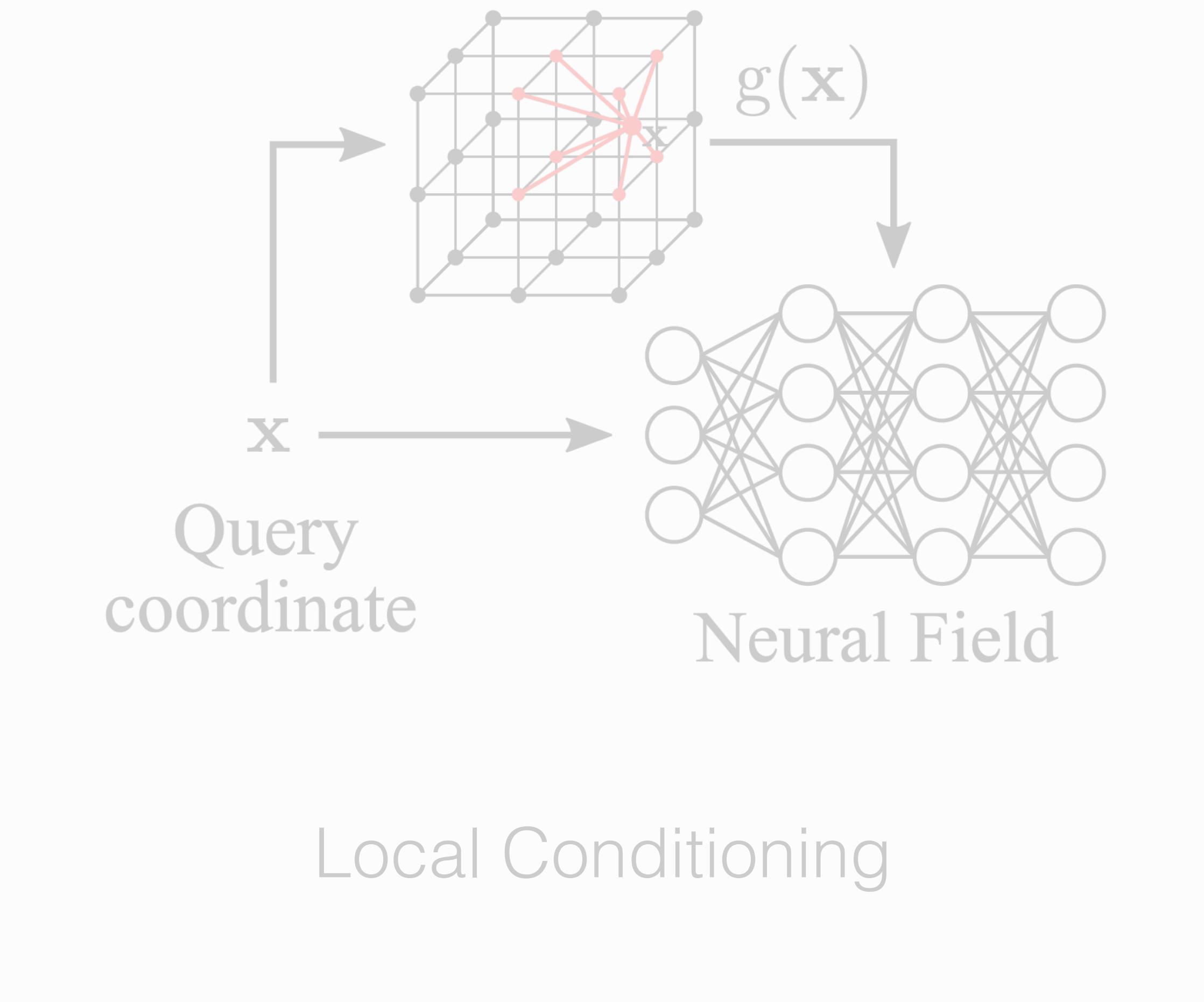
3D

...

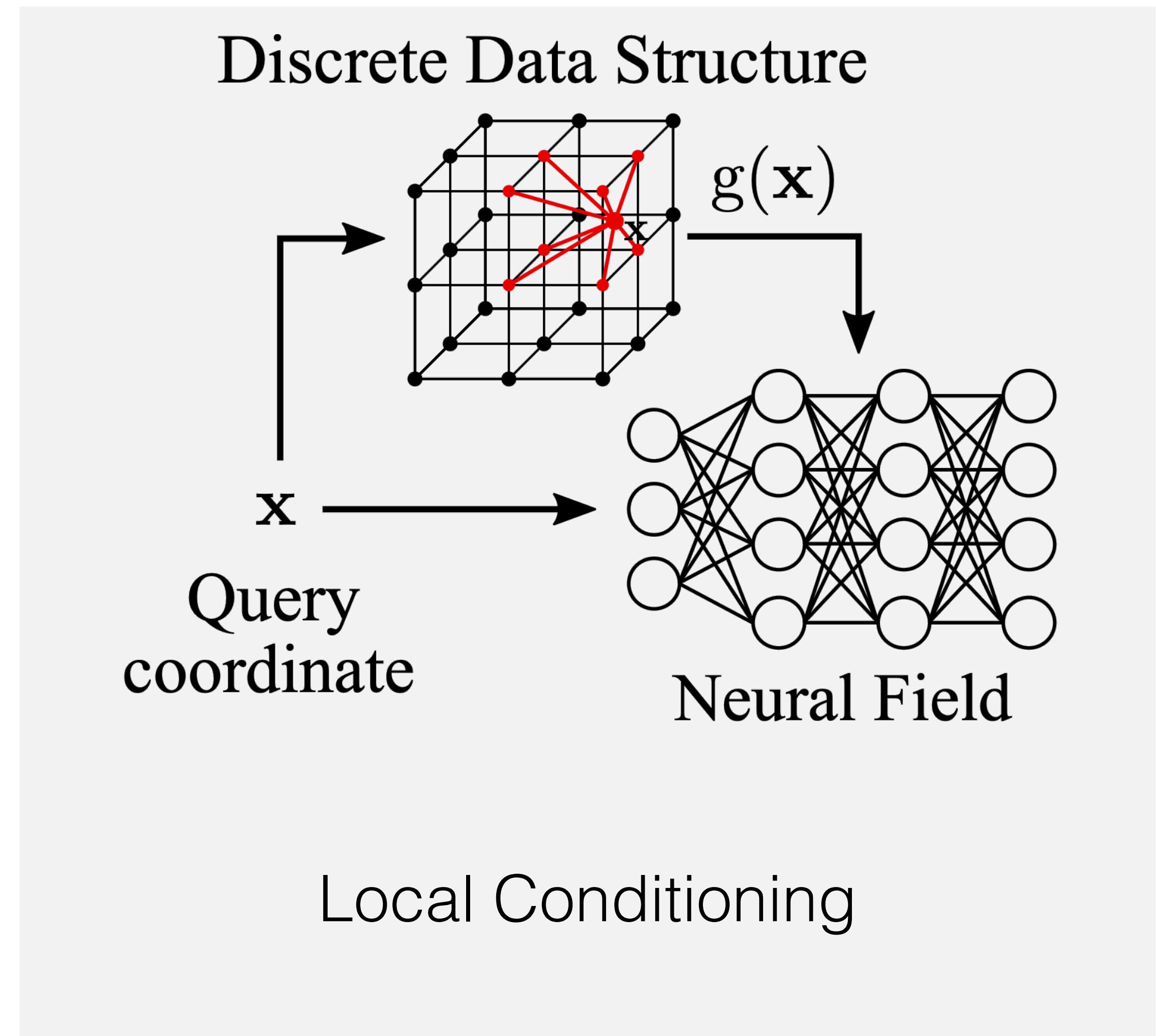
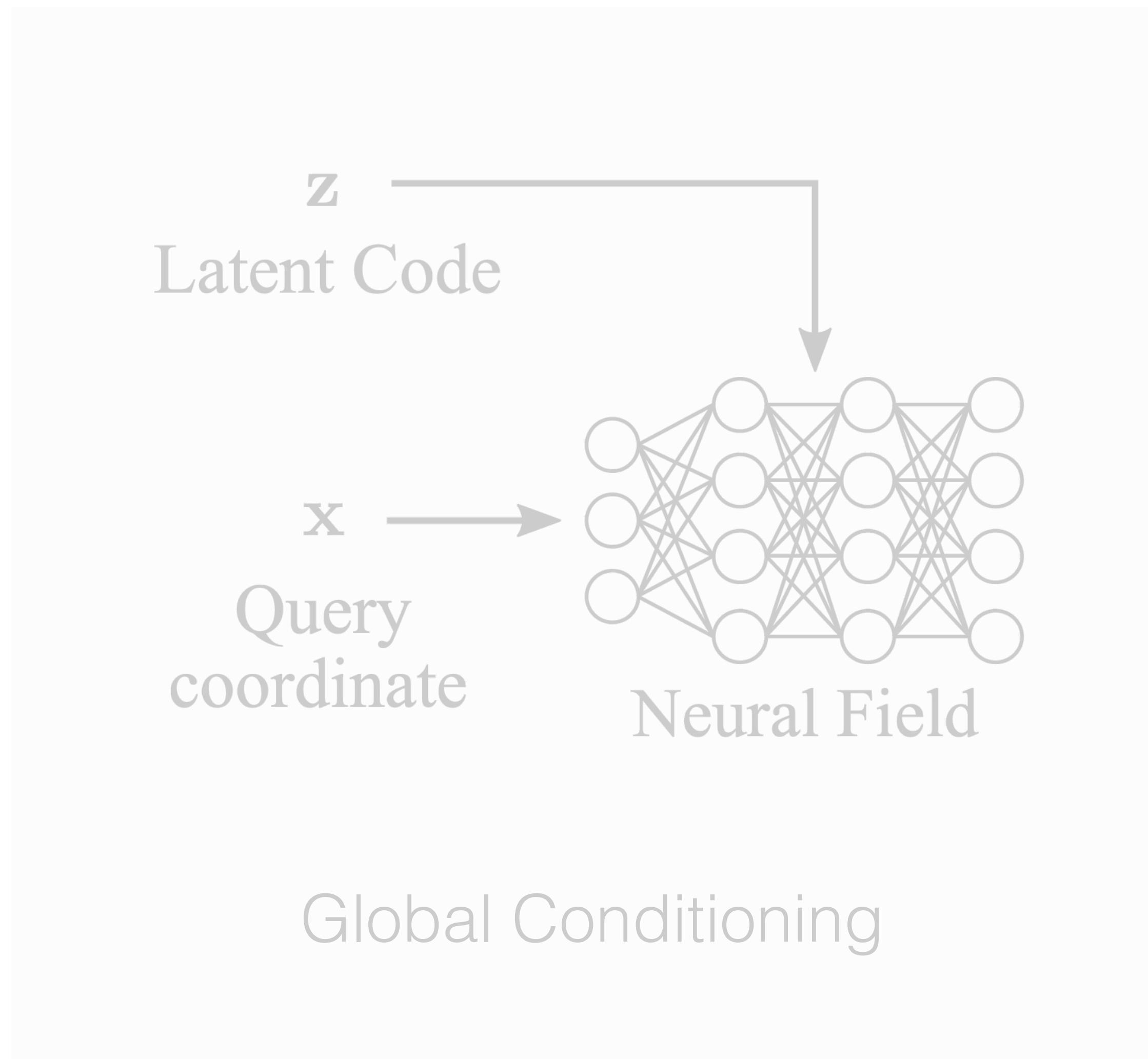
Global Latent Codes



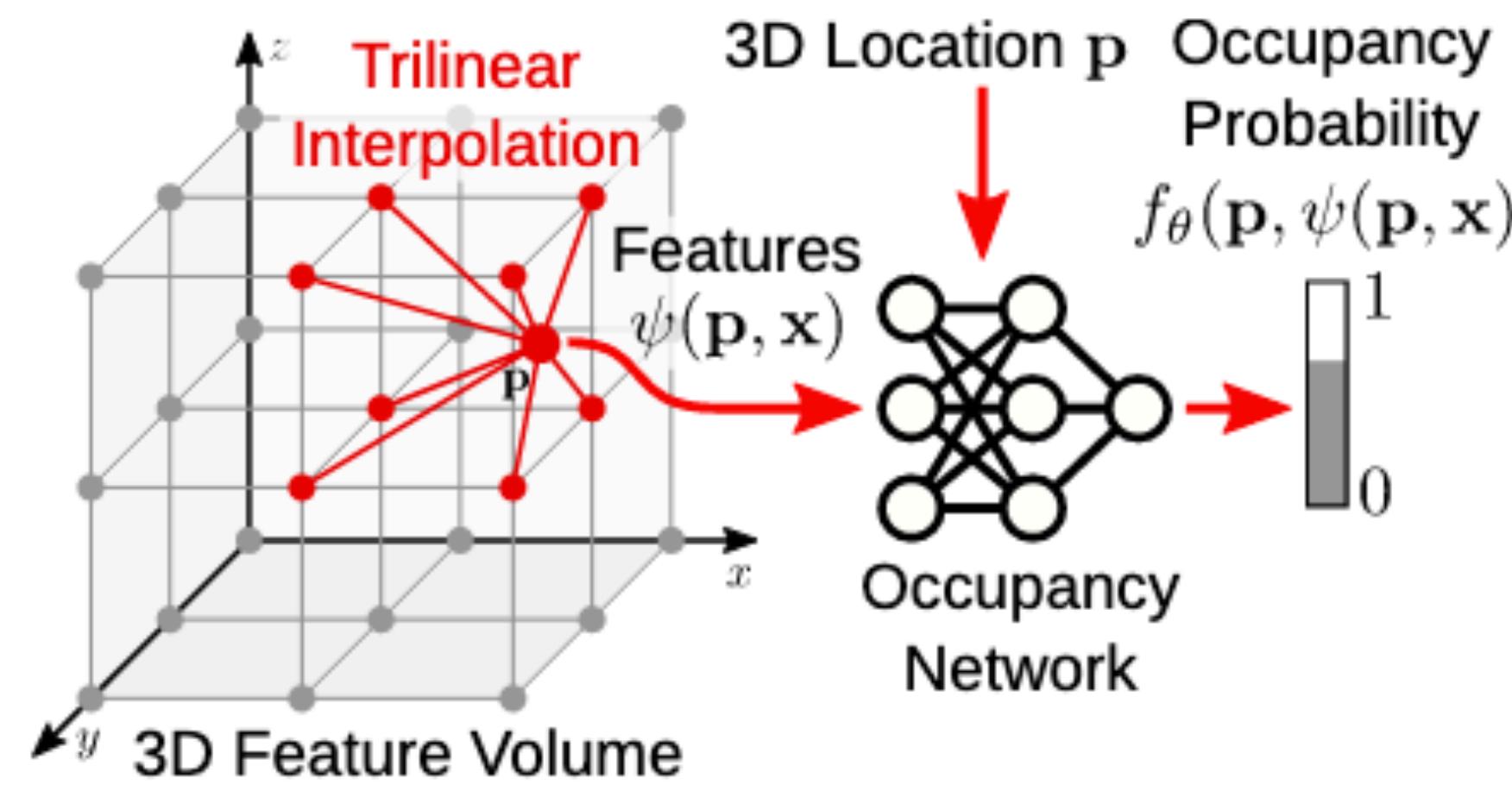
Discrete Data Structure



Local Latent Codes



From point clouds: Conditioning on Feature Voxel grids



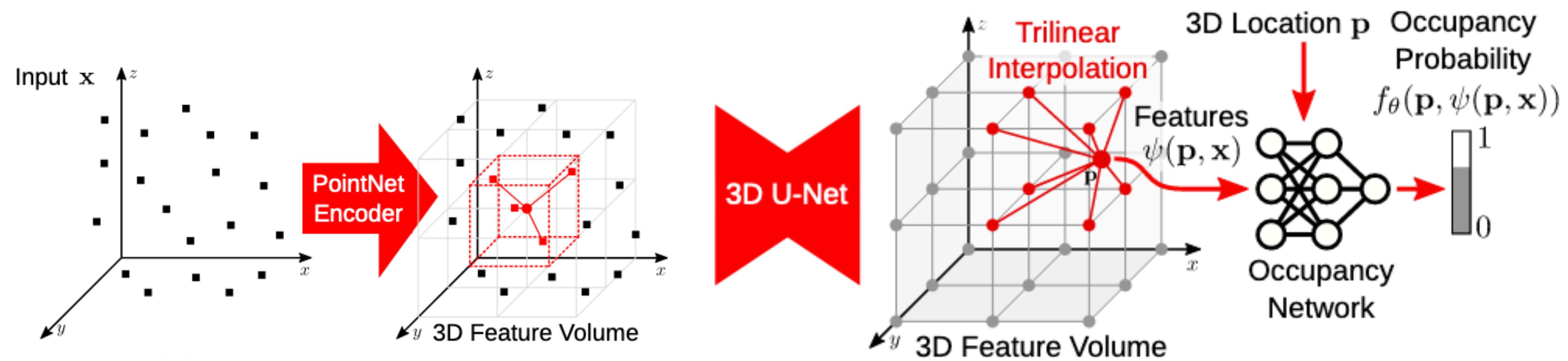
Convolutional Occupancy Networks [Peng et al. 2020]

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



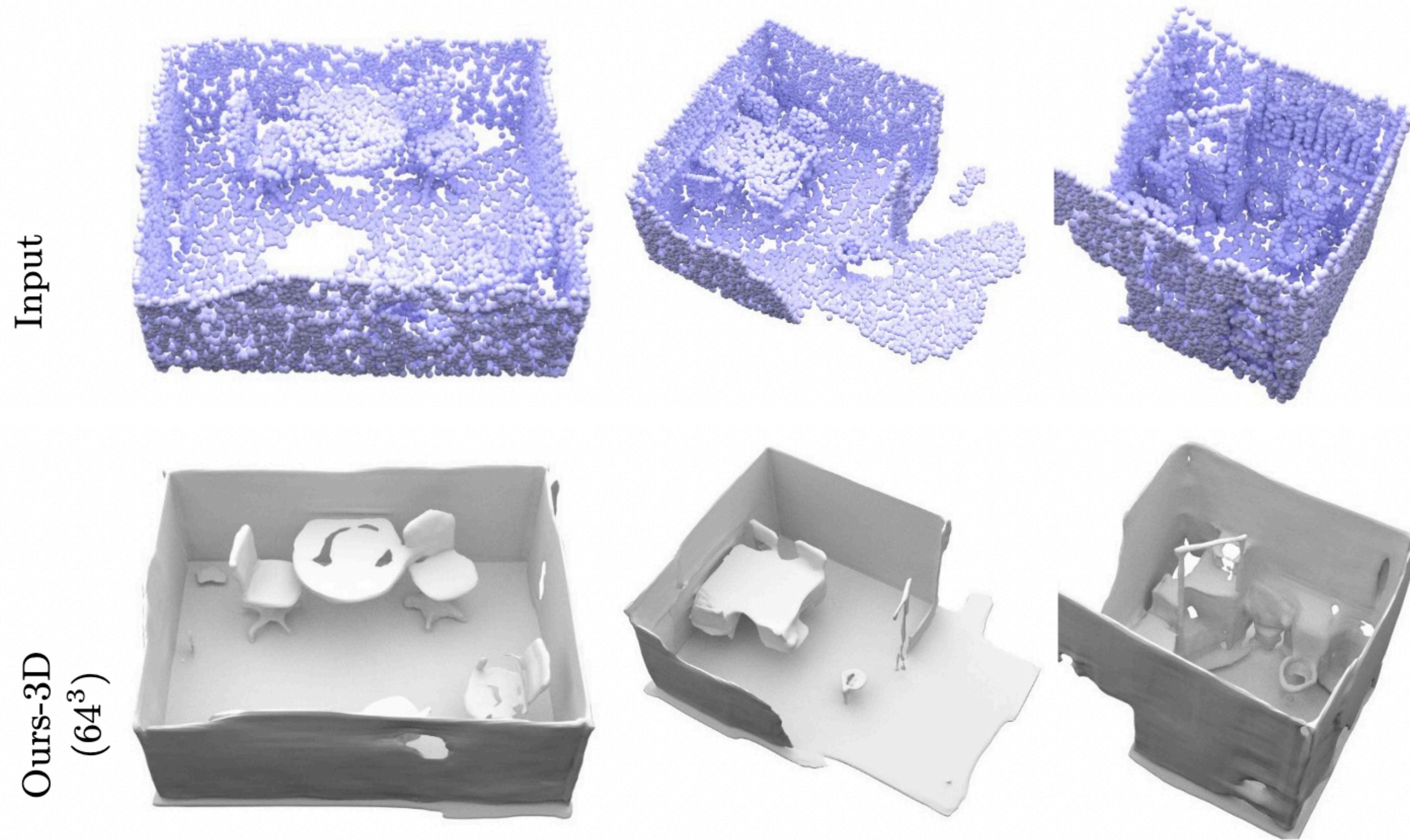
Convolutional Occupancy Networks [Peng et al. 2020]

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

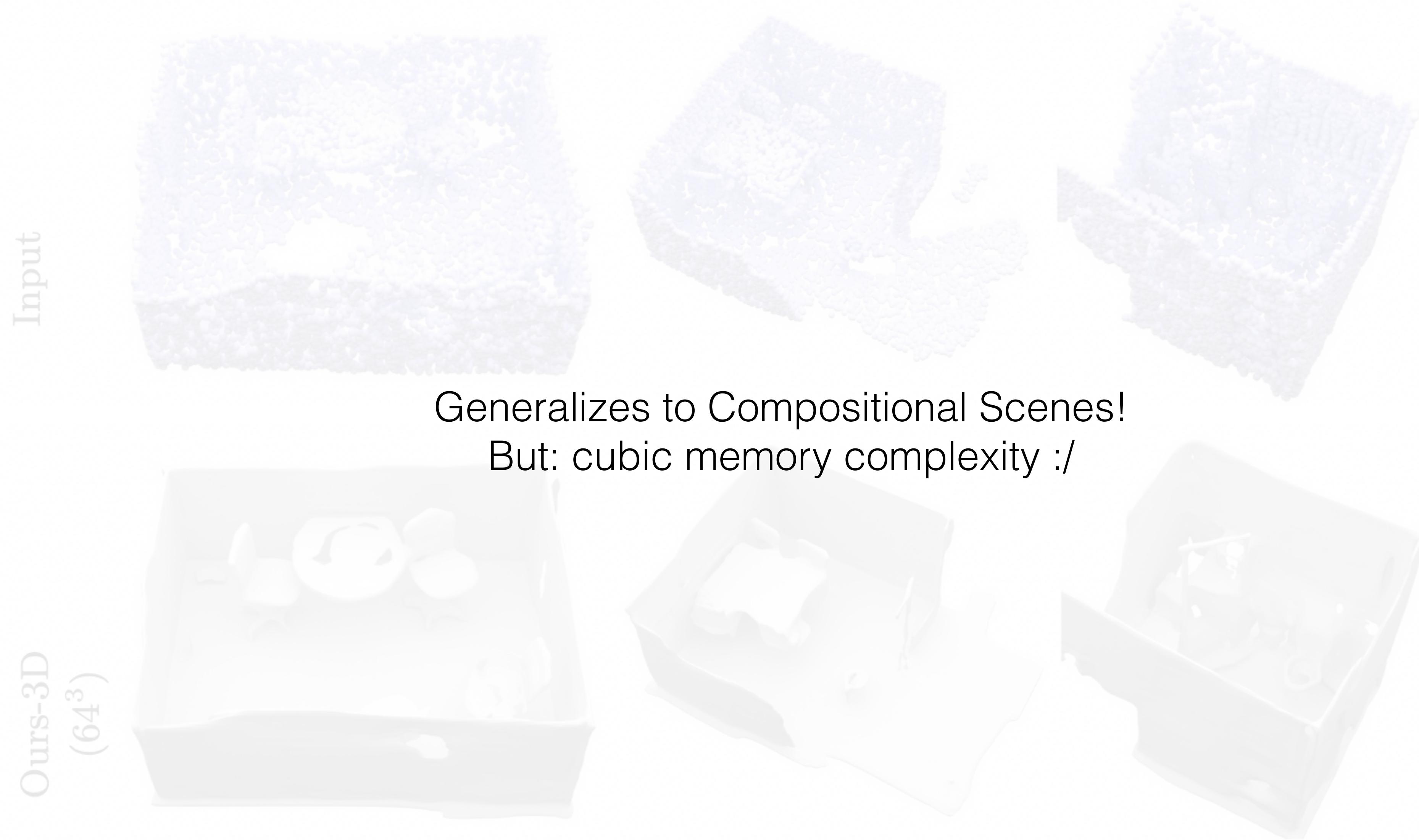
Implicit Functions in Feature Space for 3D Shape Reconstruction and Completion [Chabra et al. 2020]

Deep Local Shapes: Learning Local SDF Priors for Detailed 3D Reconstruction [Chibane et al. 2020]

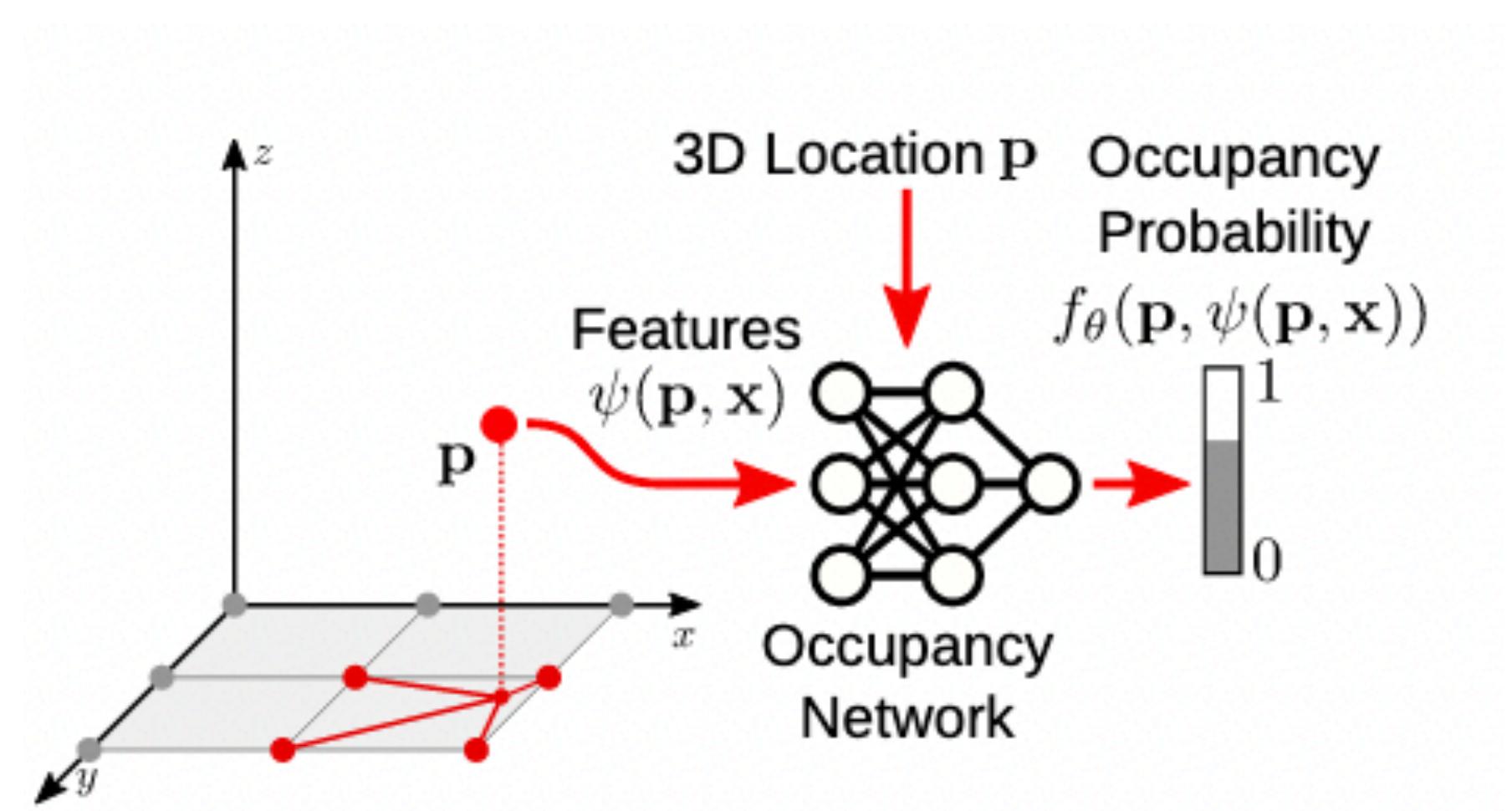
From point clouds: Conditioning on Feature Voxel grids



From point clouds: Conditioning on Feature Voxel grids

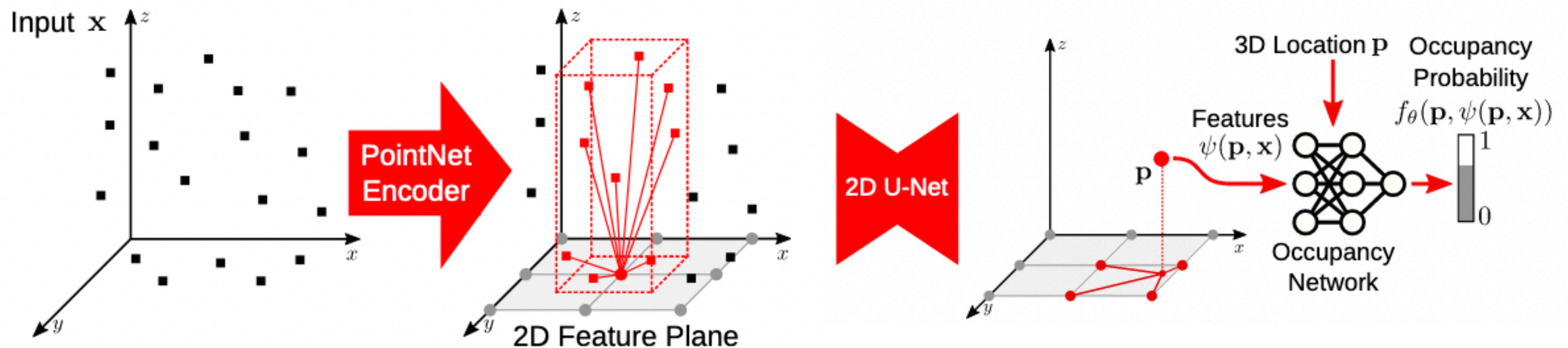


From Point clouds: Ground-plan and Tri-plane factorizations

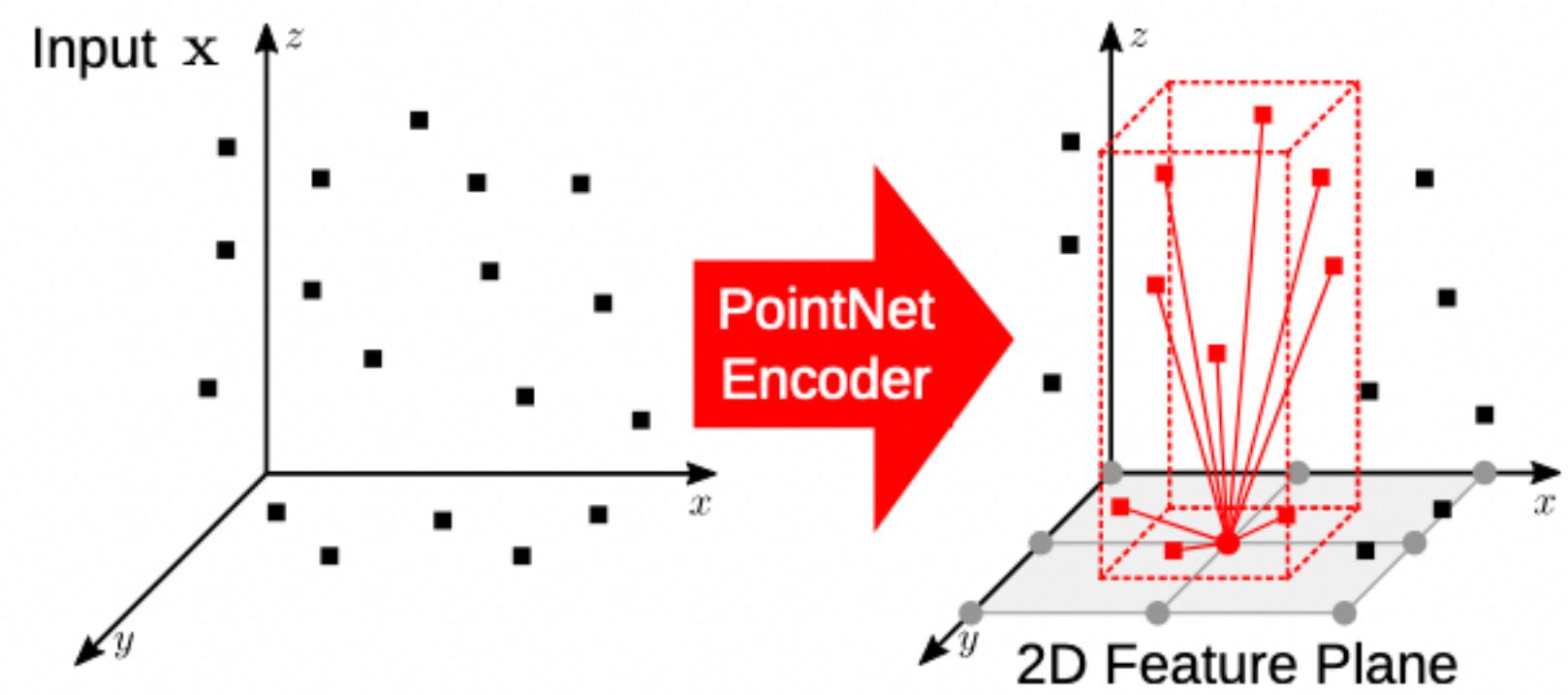


Convolutional Occupancy Networks [Peng et al. 2020]

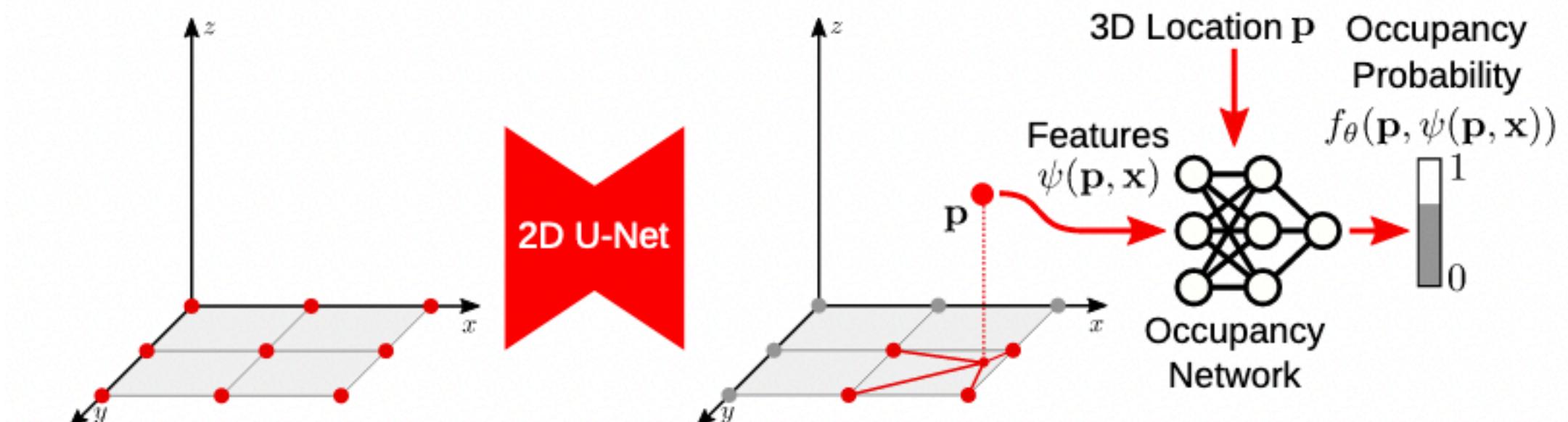
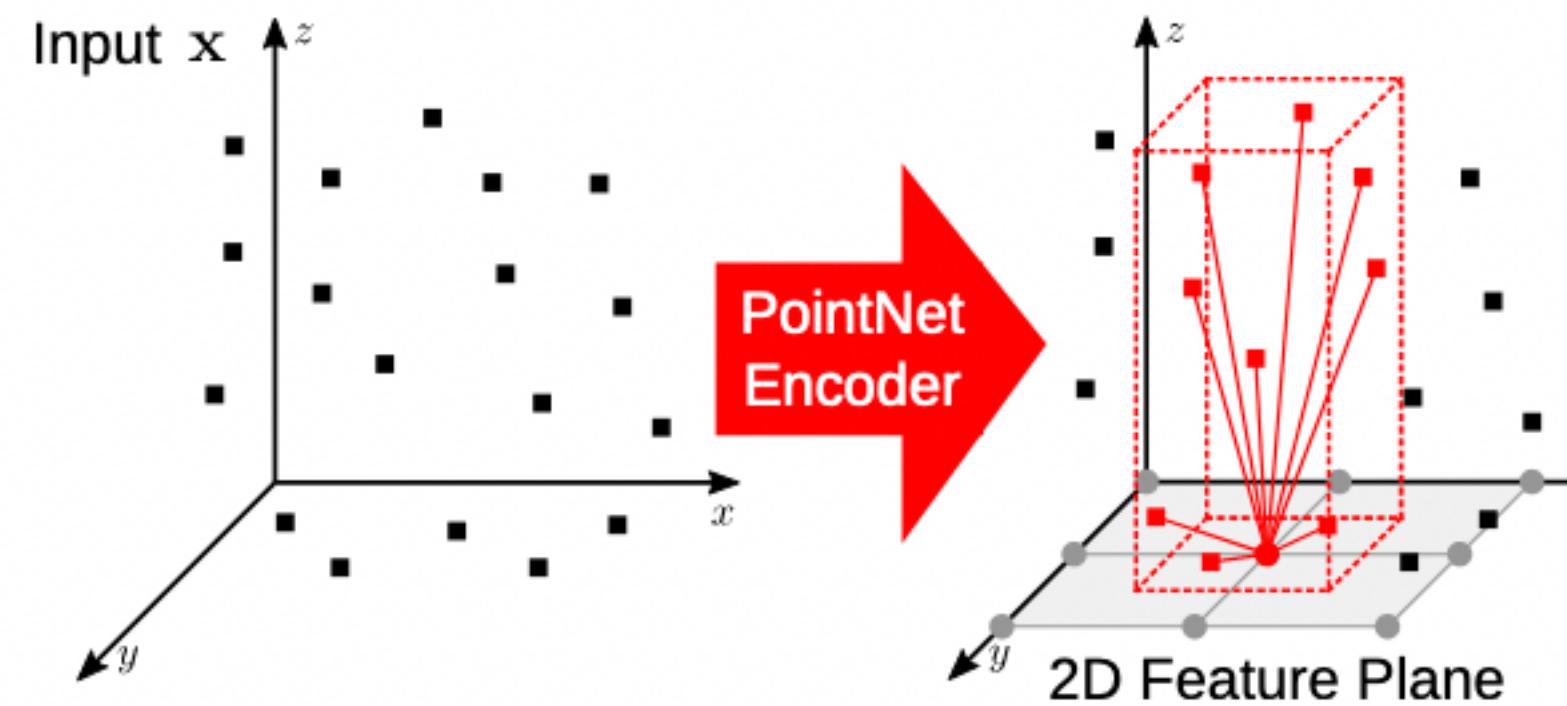
From Point clouds: Ground-plan and Tri-plane factorizations



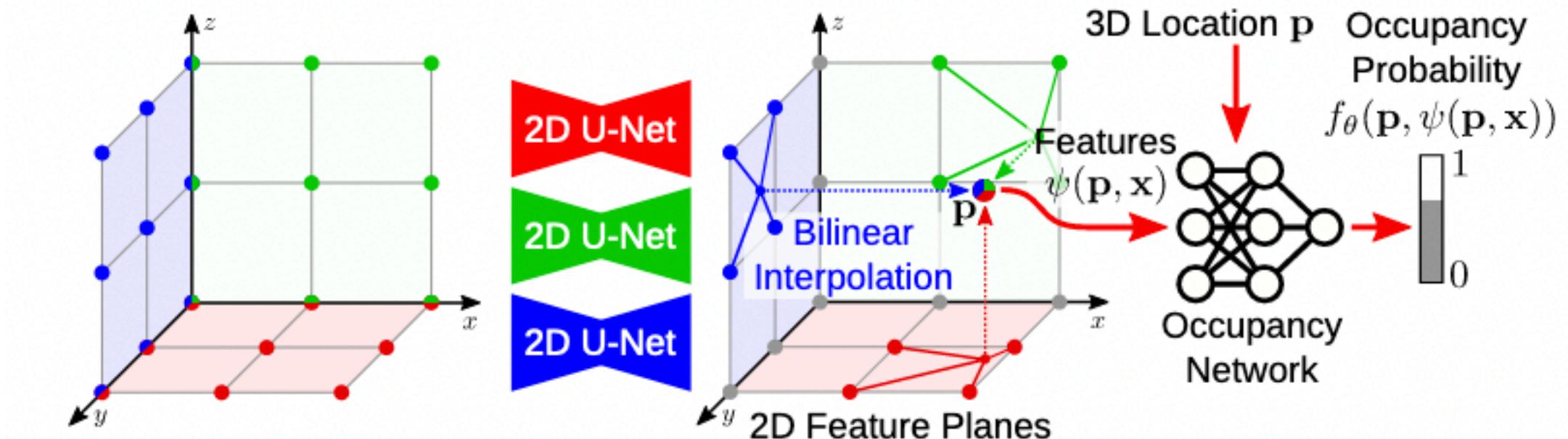
From Point clouds: Ground-plan and Tri-plane factorizations



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

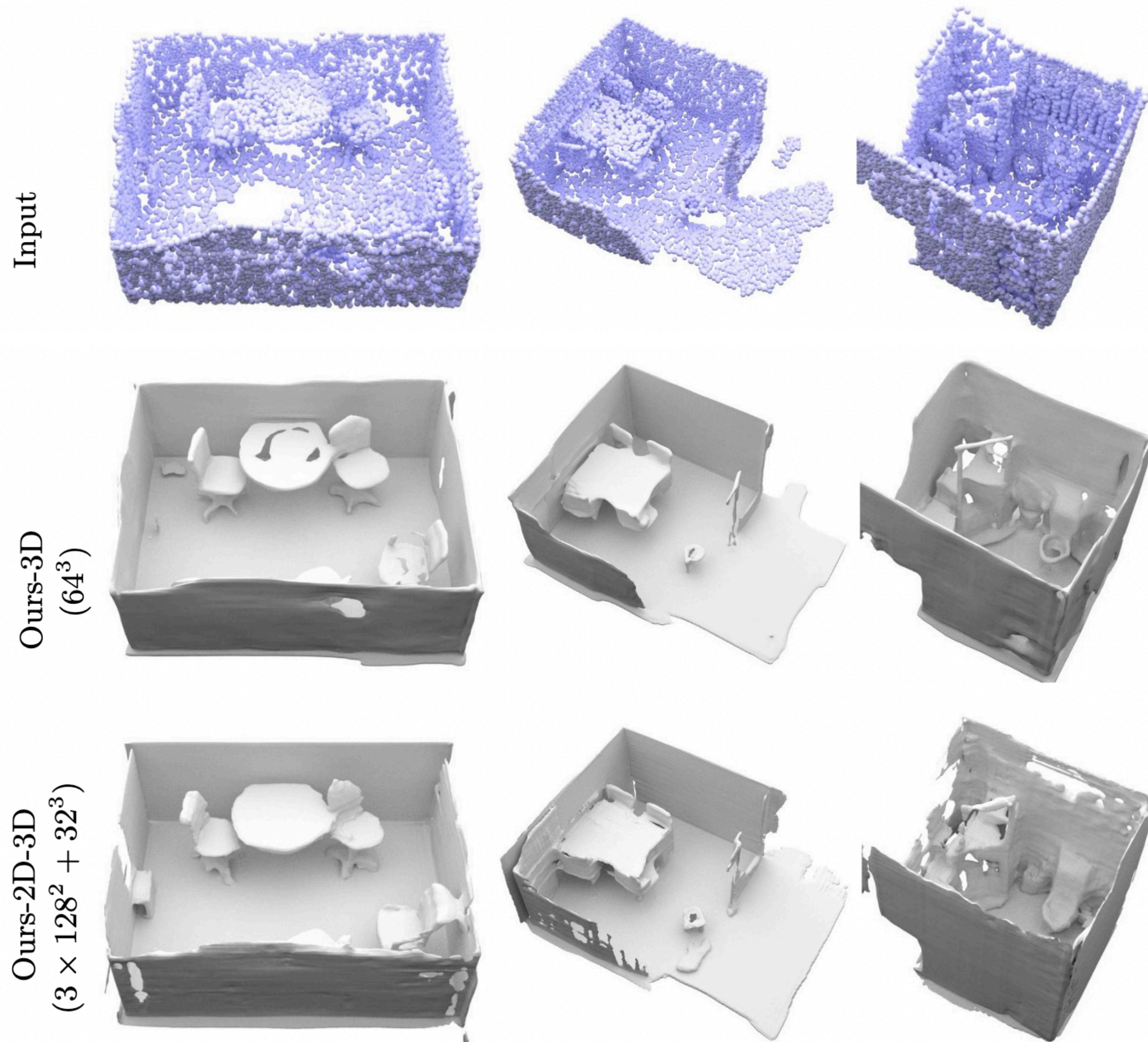


(c) Convolutional Single-Plane Decoder



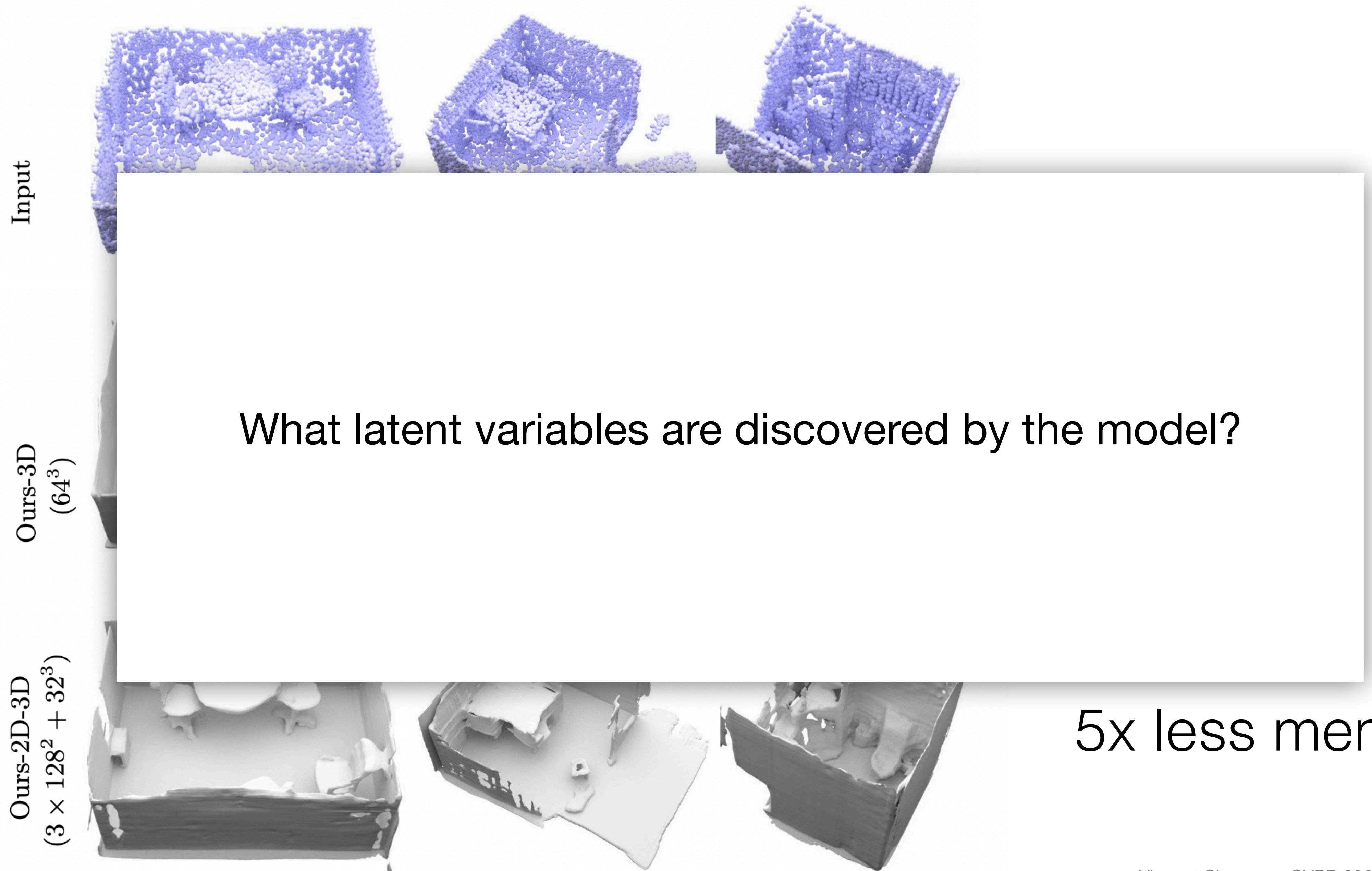
(d) Convolutional Multi-Plane Decoder

From point clouds: Conditioning on Reconstructed Voxelgrids

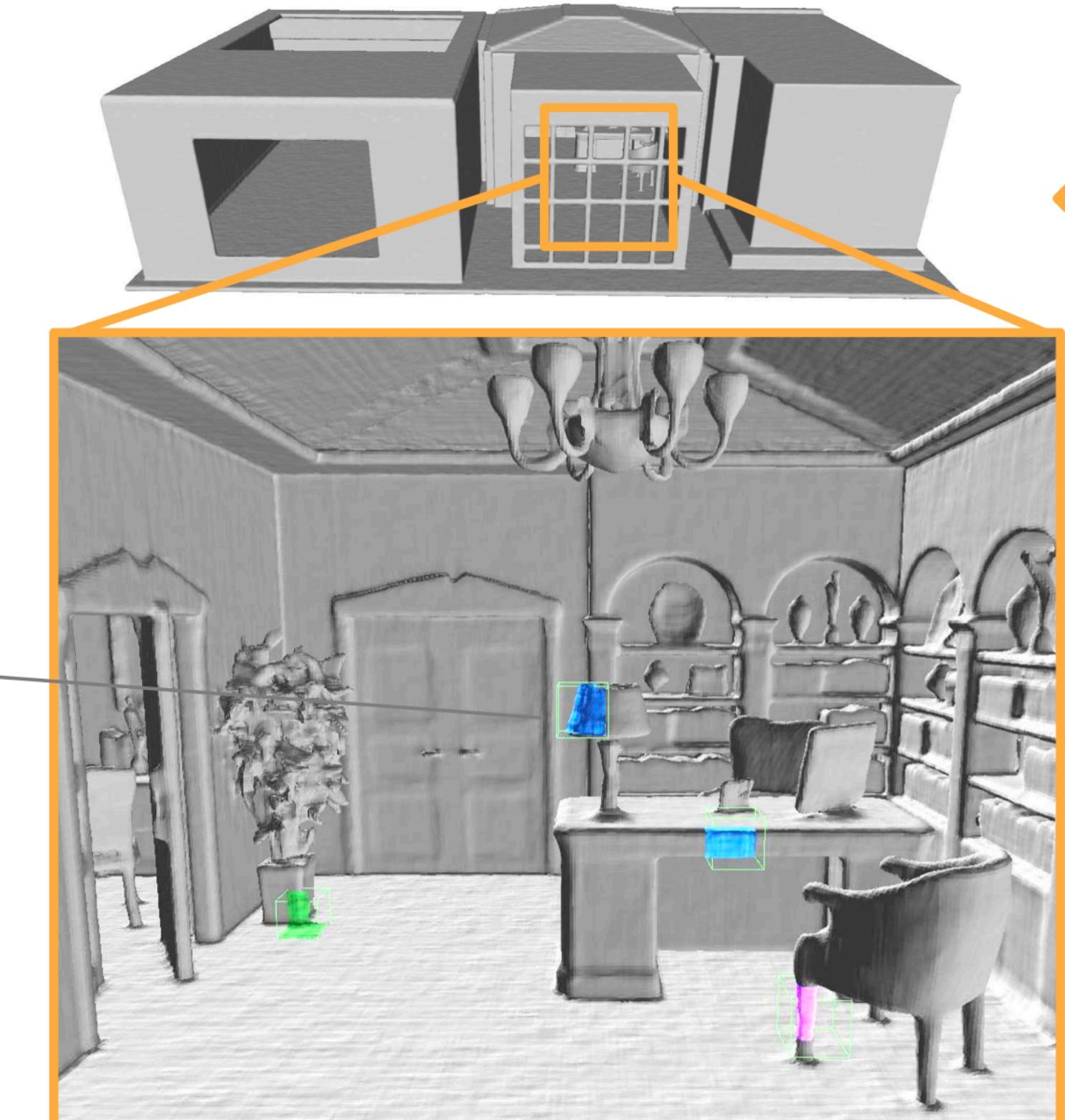
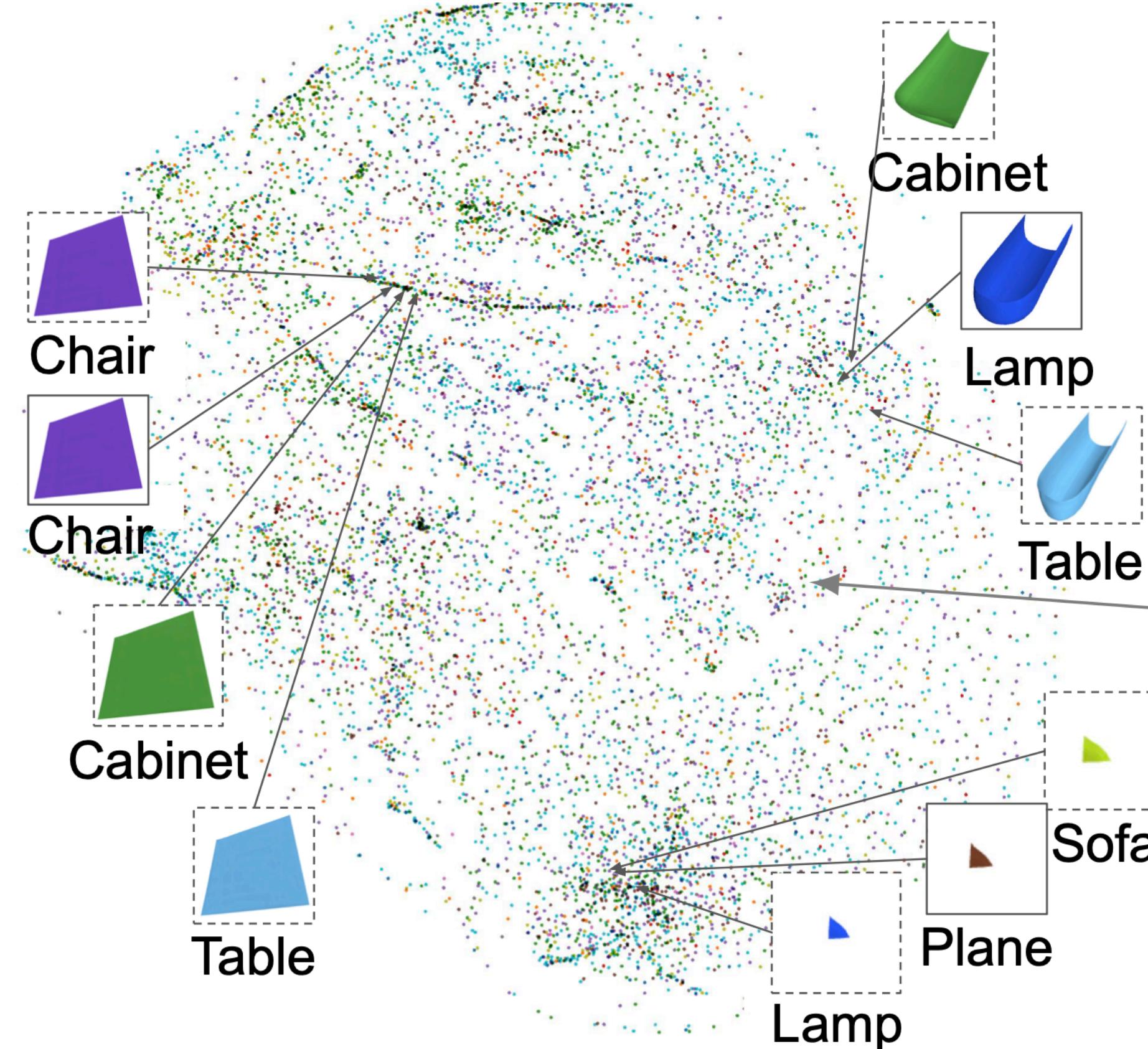
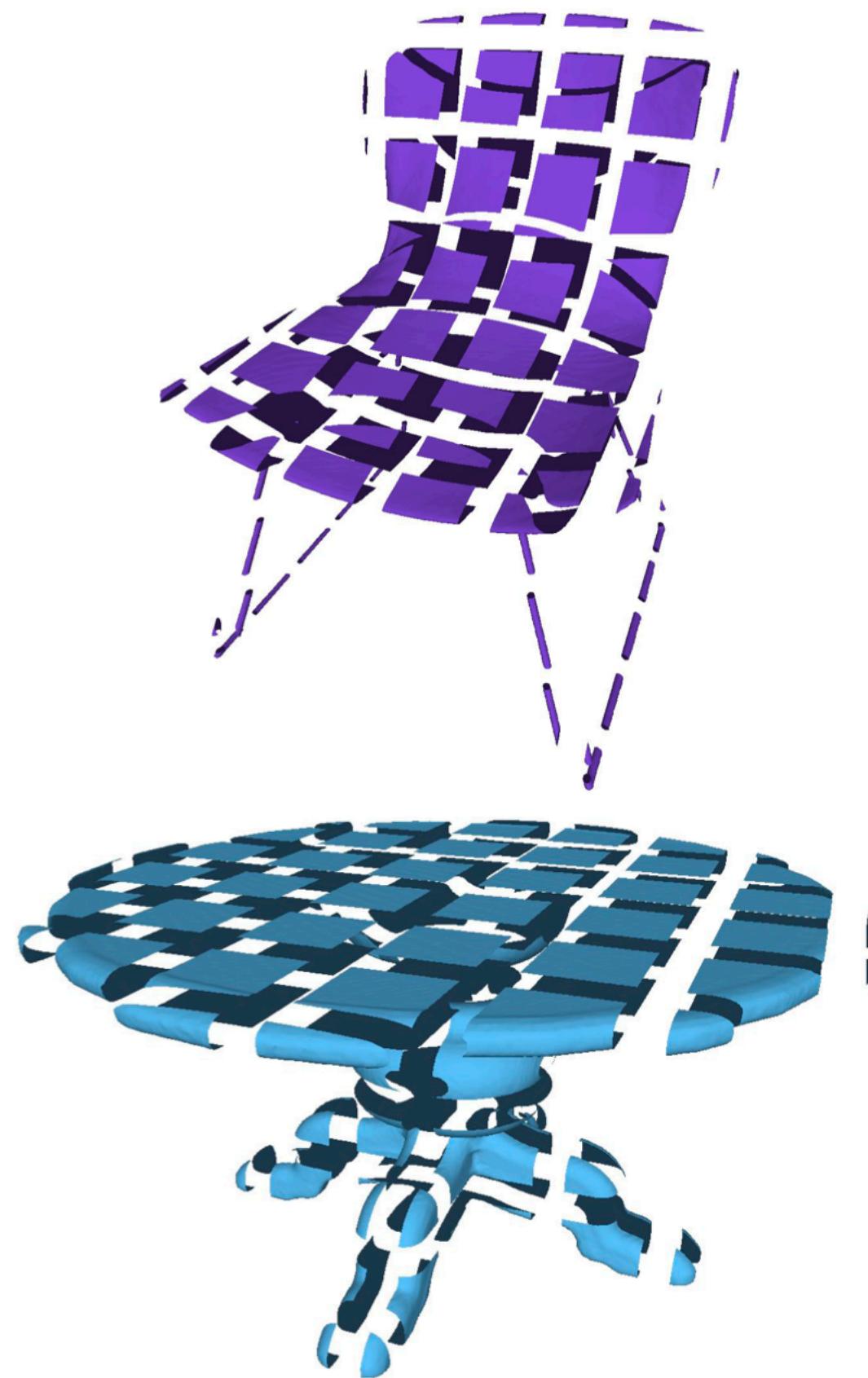


5x less memory!

From point clouds: Conditioning on Reconstructed Voxelgrids

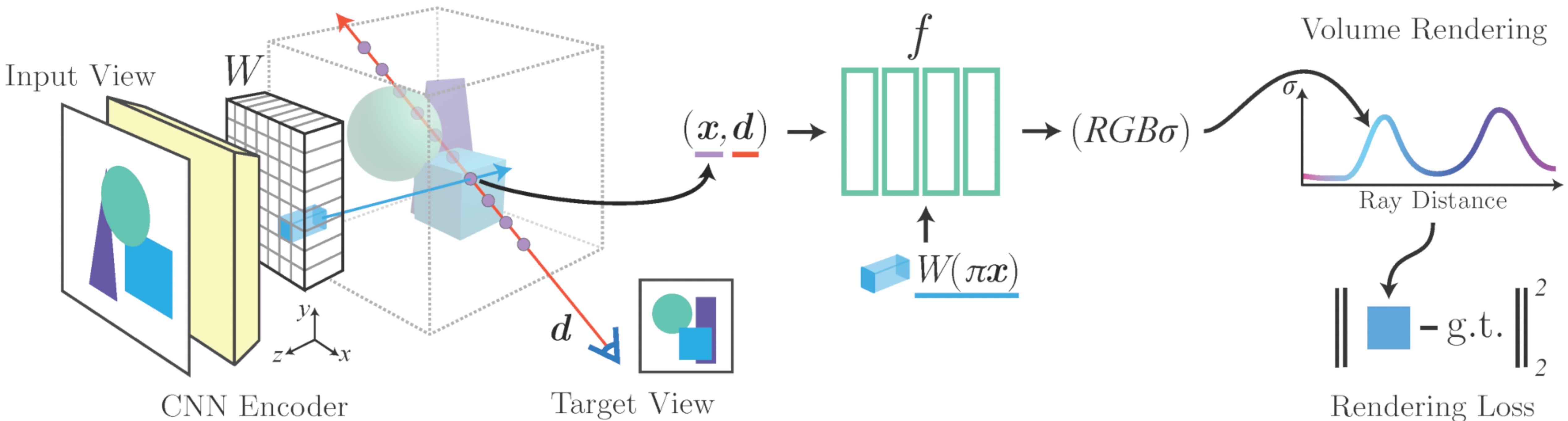


Locally Conditioned Latent Space



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

Local Conditioning: Pixel-Aligned Features.

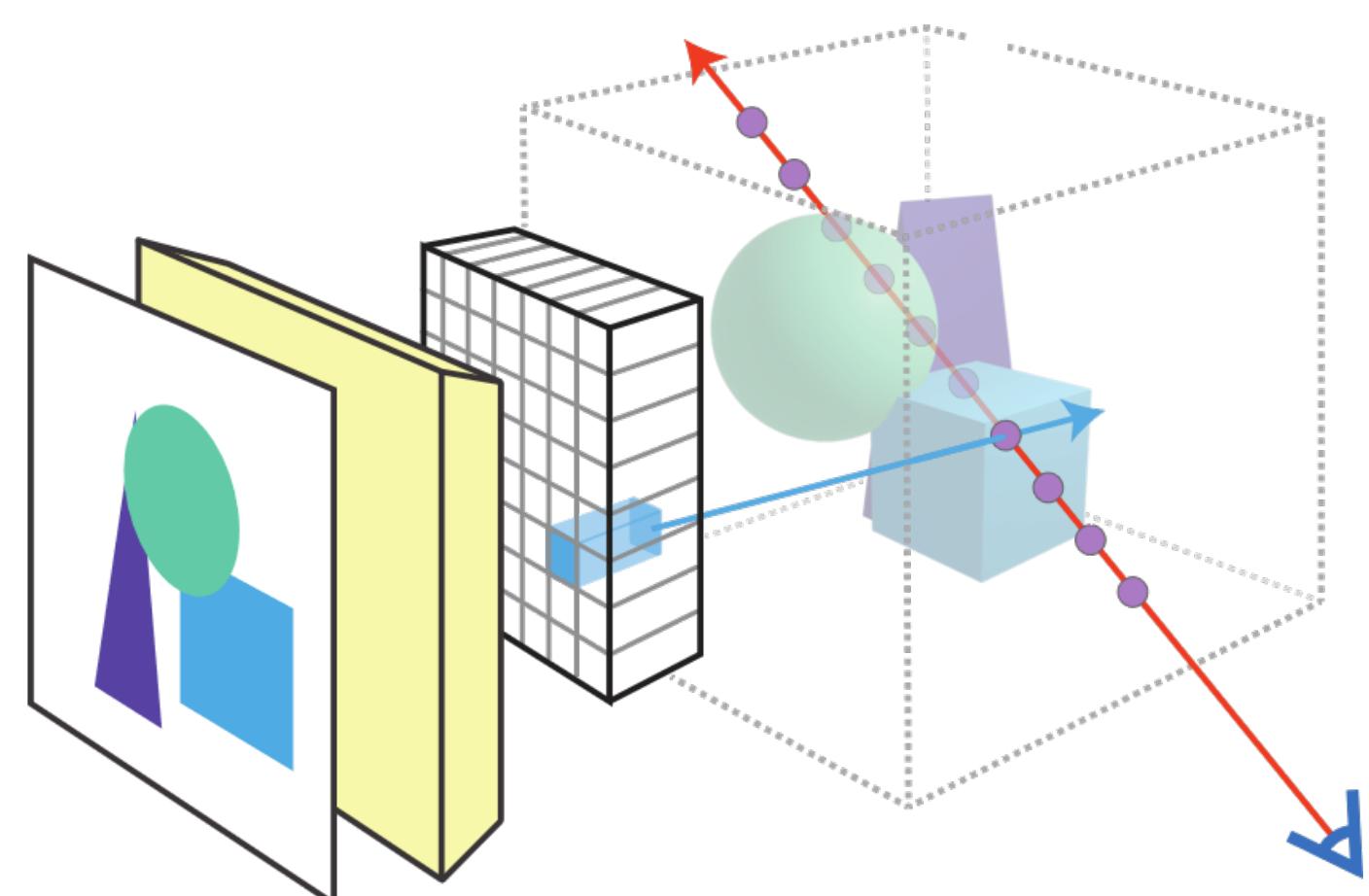
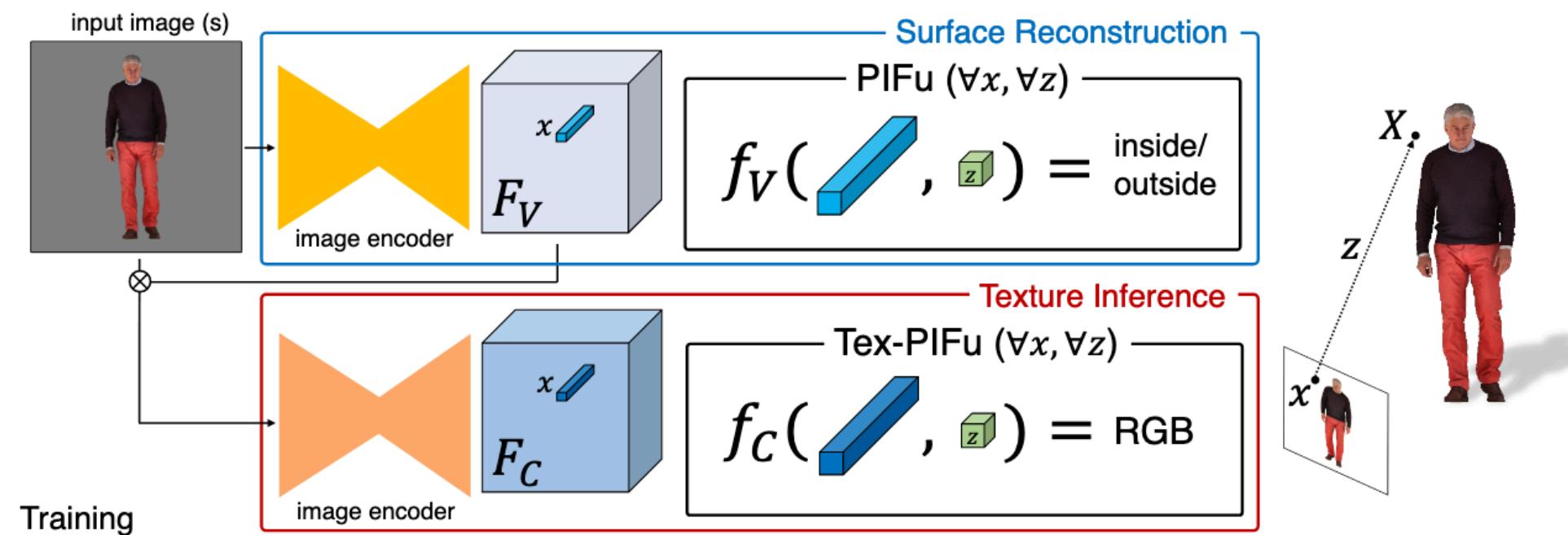


PiFU, Saito et al., ICCV 2019.

PixelNeRF, Yu et al., CVPR 2021

Grf: Learning a general radiance field..., Trevithick et al.

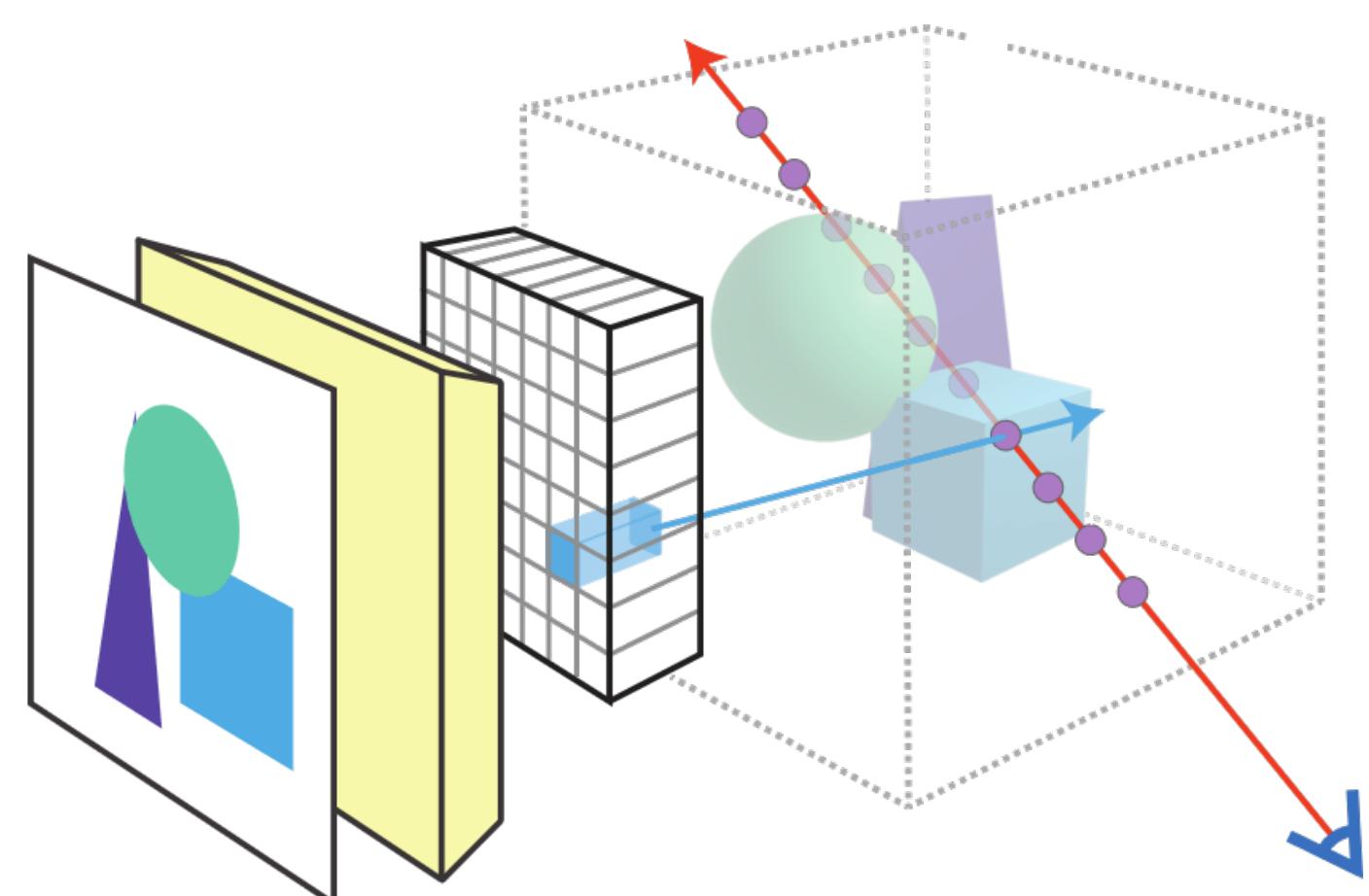
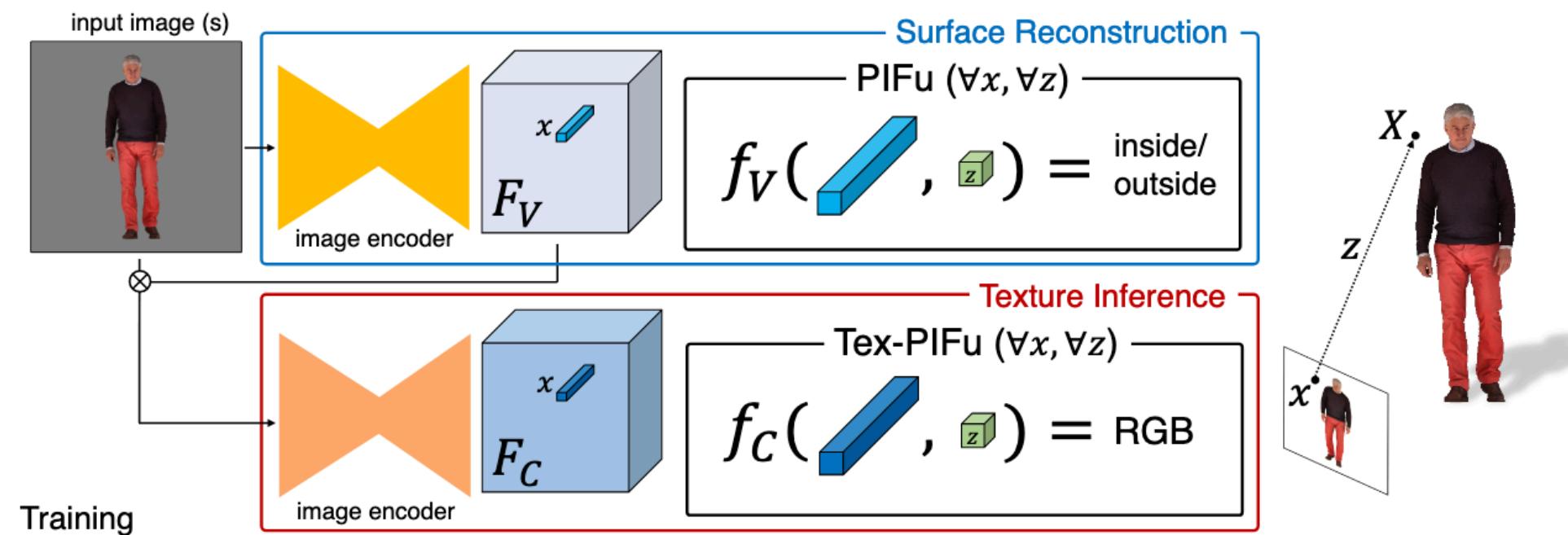
Local Conditioning: Pixel-Aligned Features.



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



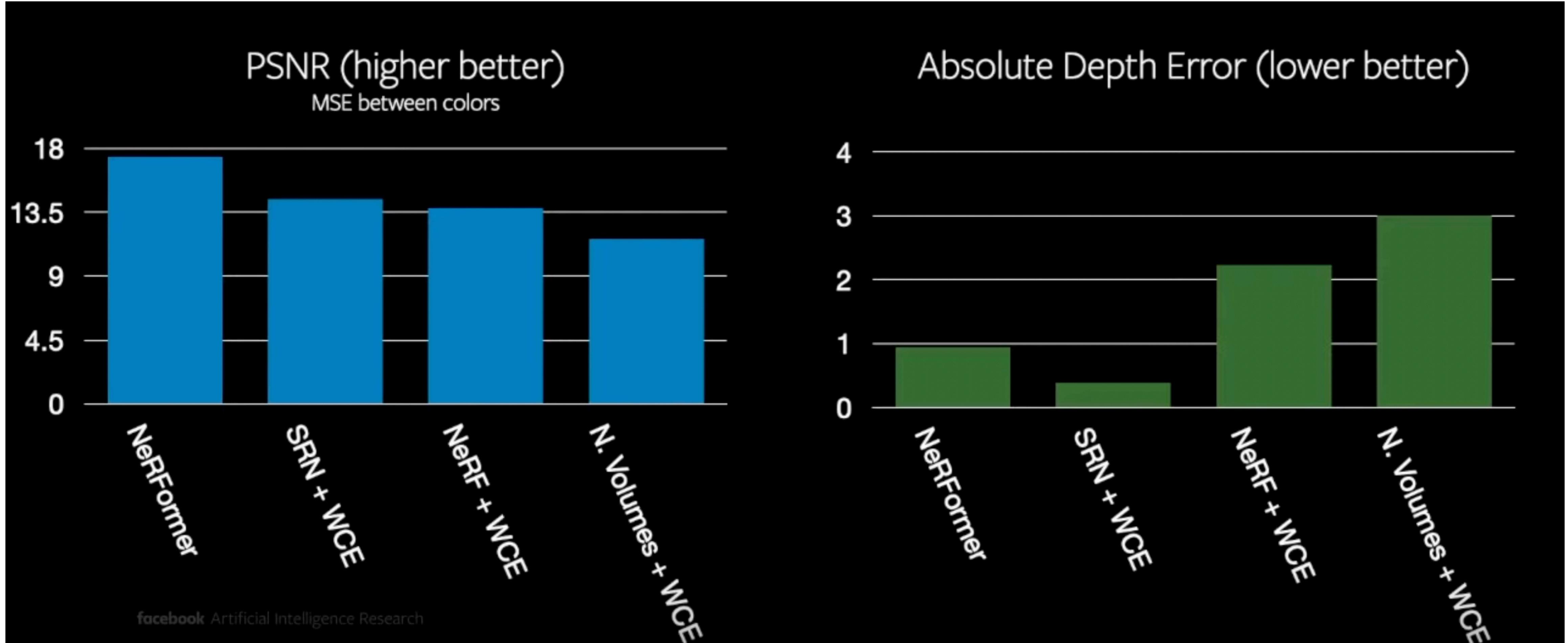
Local Conditioning: Pixel-Aligned Features.



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

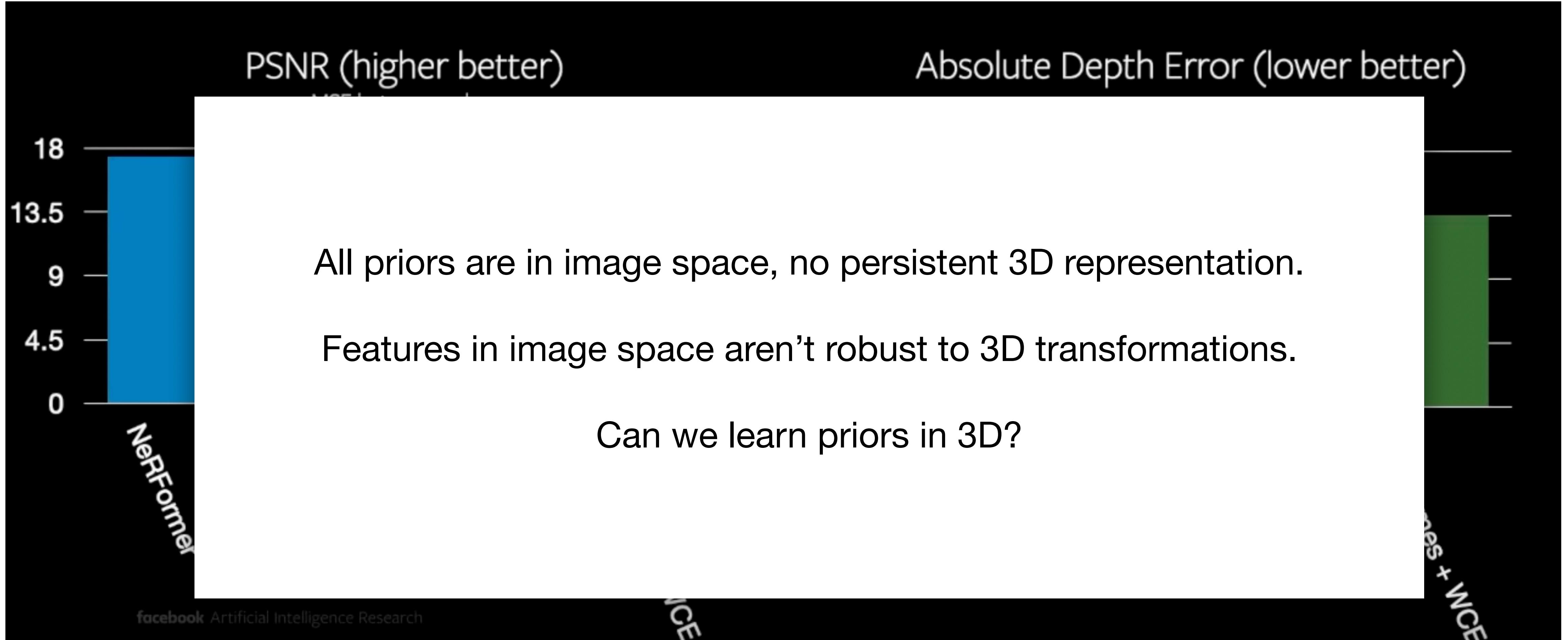


Can combine Scene Representation Networks with pixel-aligned features as well!



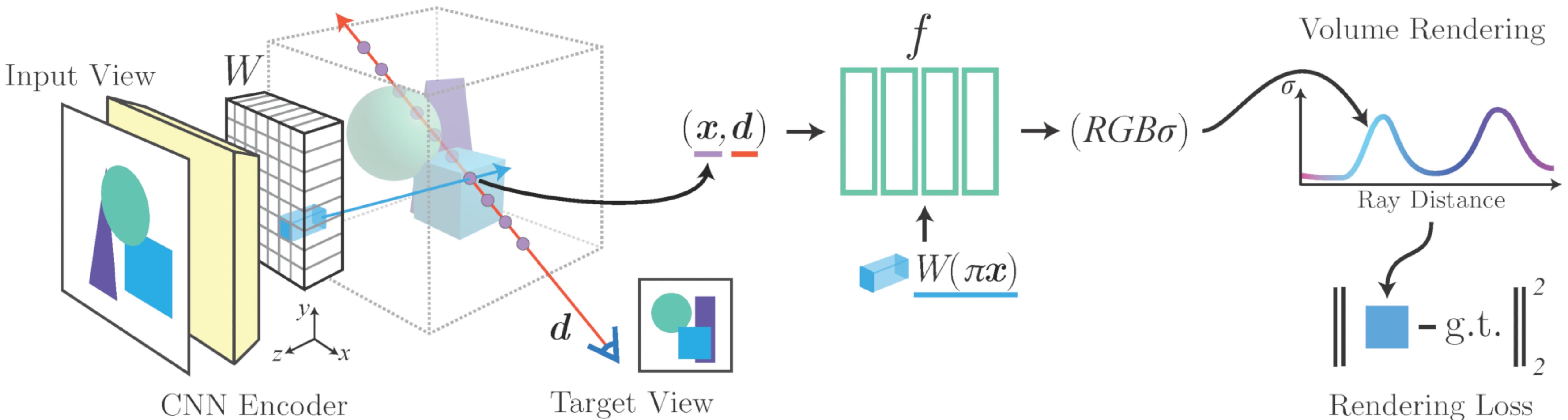
Common Objects in 3D: Large-Scale Learning and Evaluation of Real-life 3D Category Reconstruction

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.

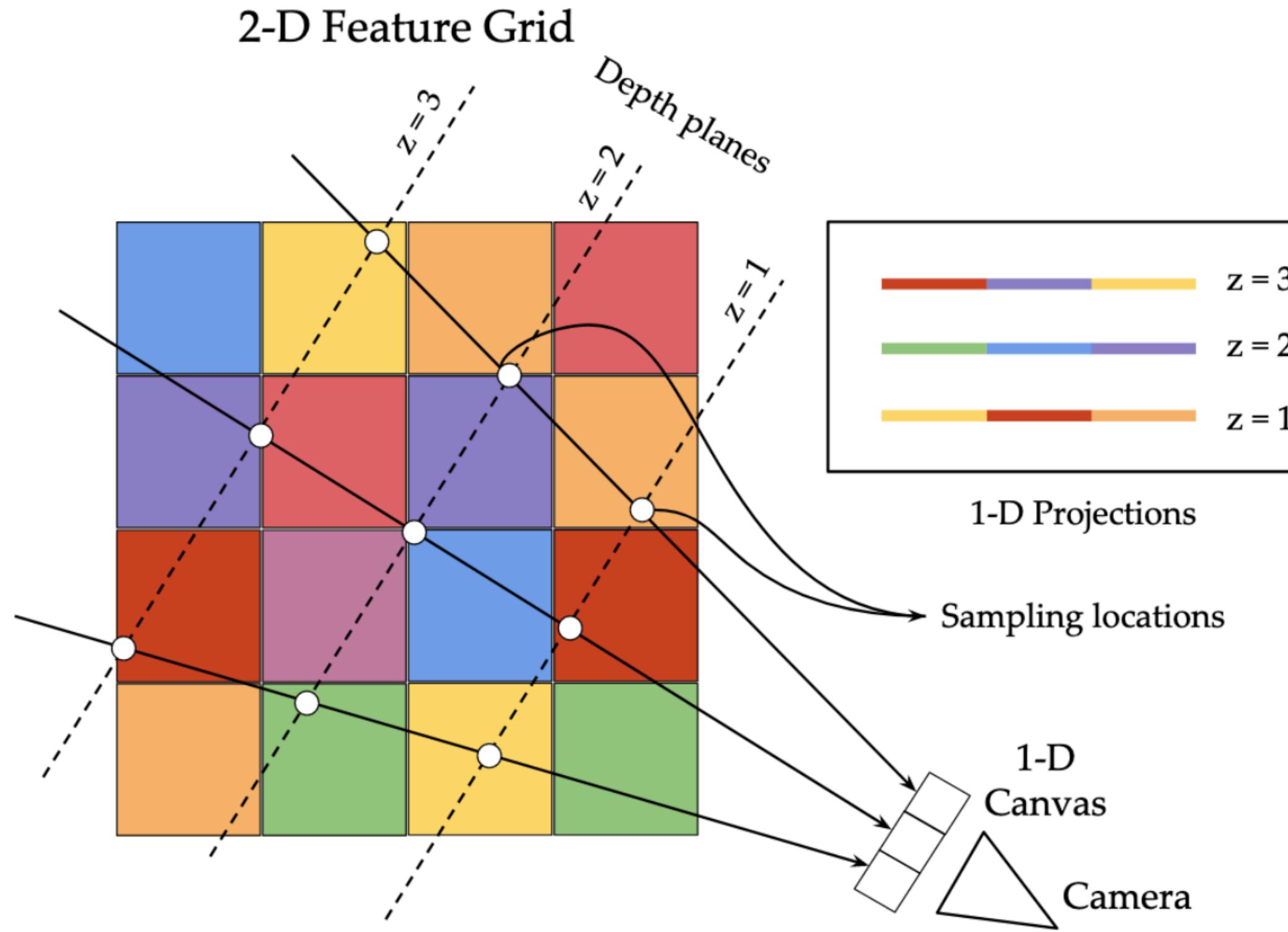


PiFu, Saito et al., ICCV 2019.

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

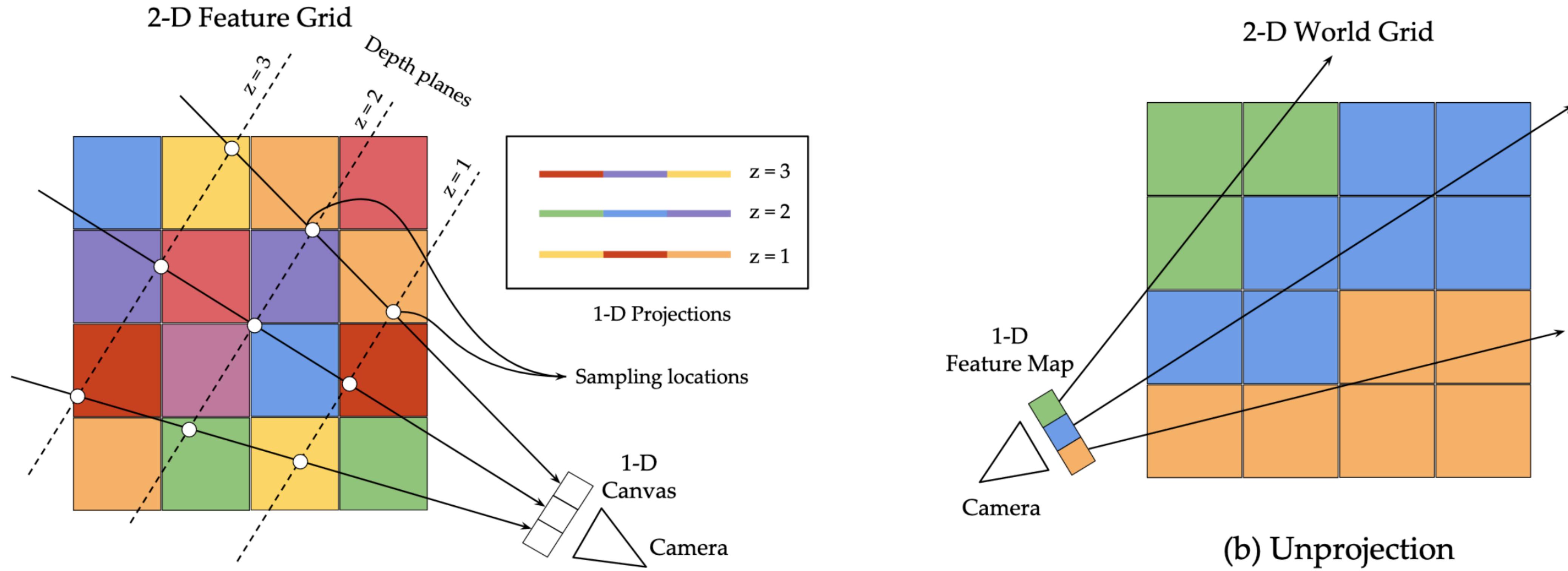


Learning a Multi-View Stereo Machine, Kar et al. 2017

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

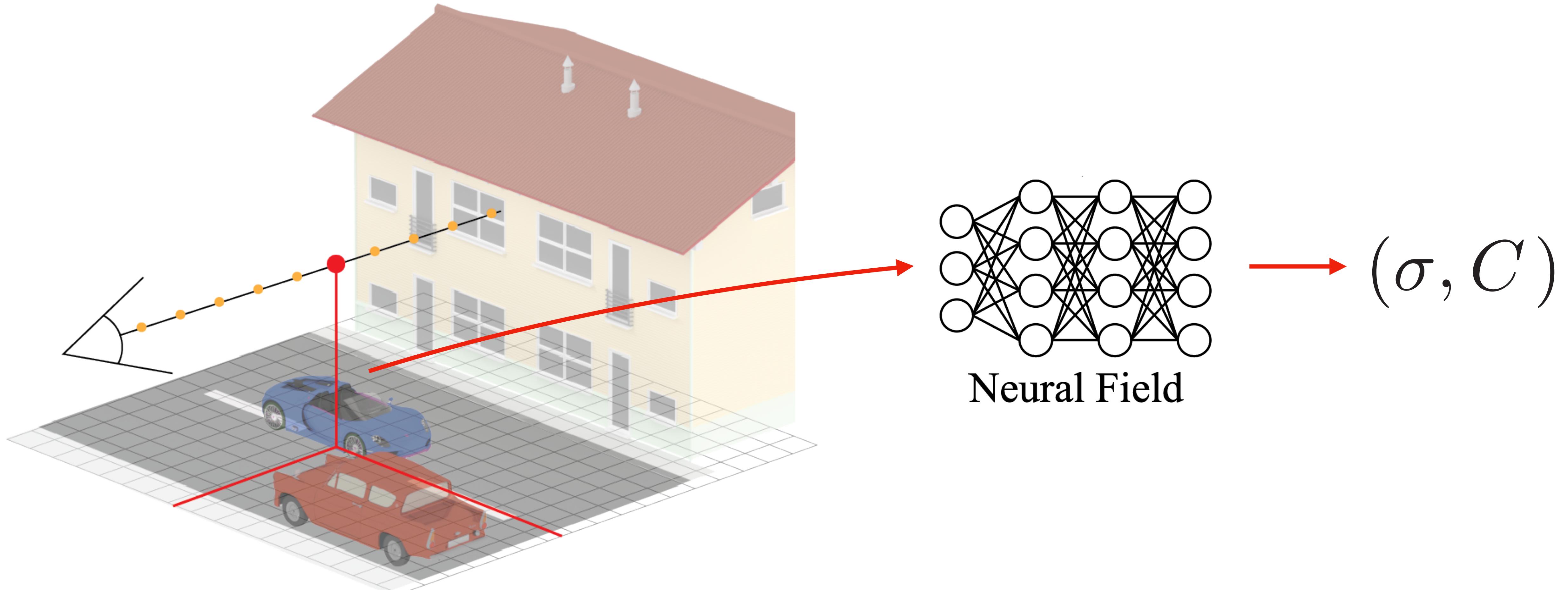


Learning a Multi-View Stereo Machine, Kar et al. 2017

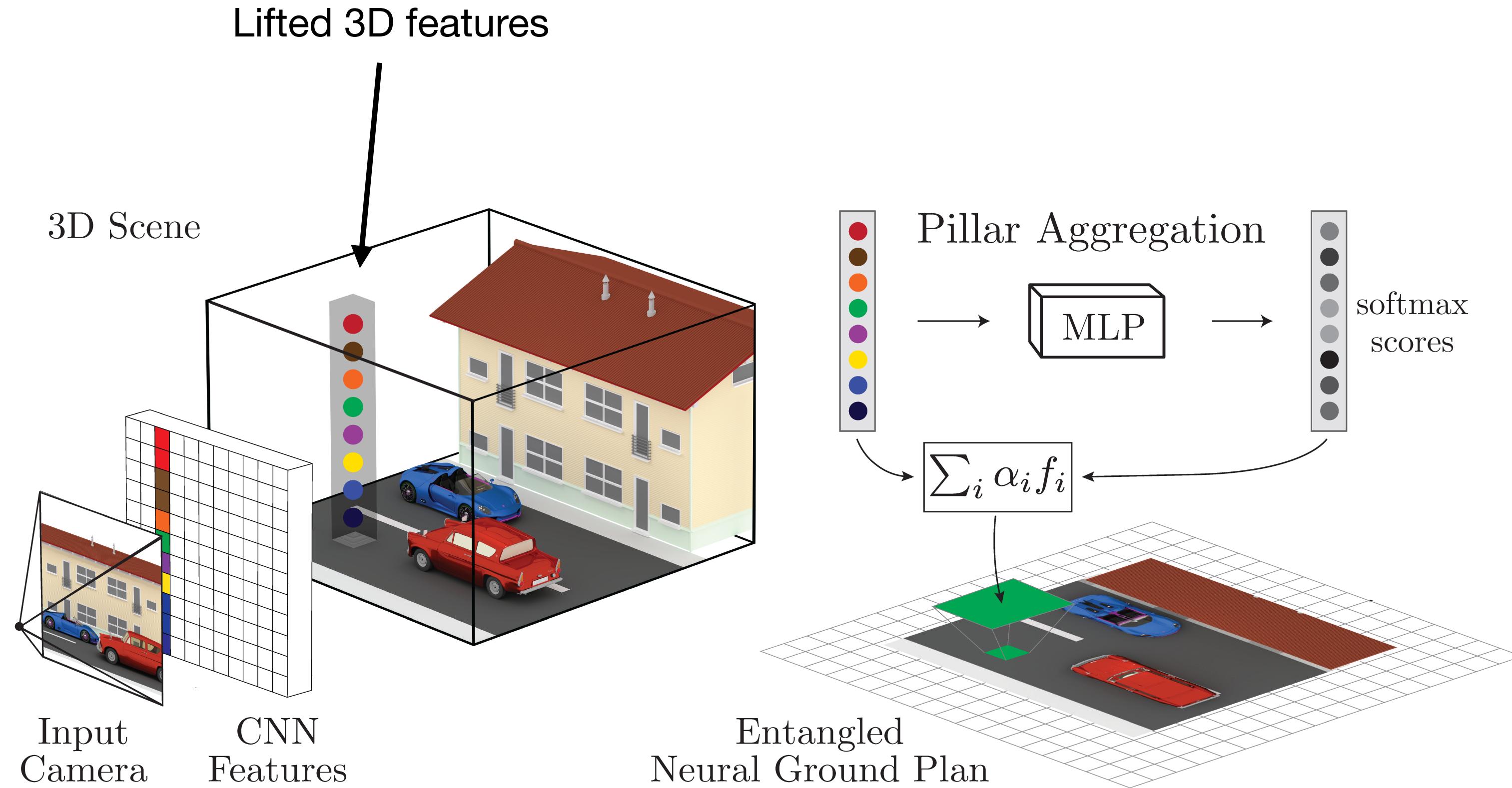
DeepVoxels, Sitzmann et al. 2019

Learning Spatial Common Sense with Geometry-Aware Recurrent Networks, Tung et al. 2019

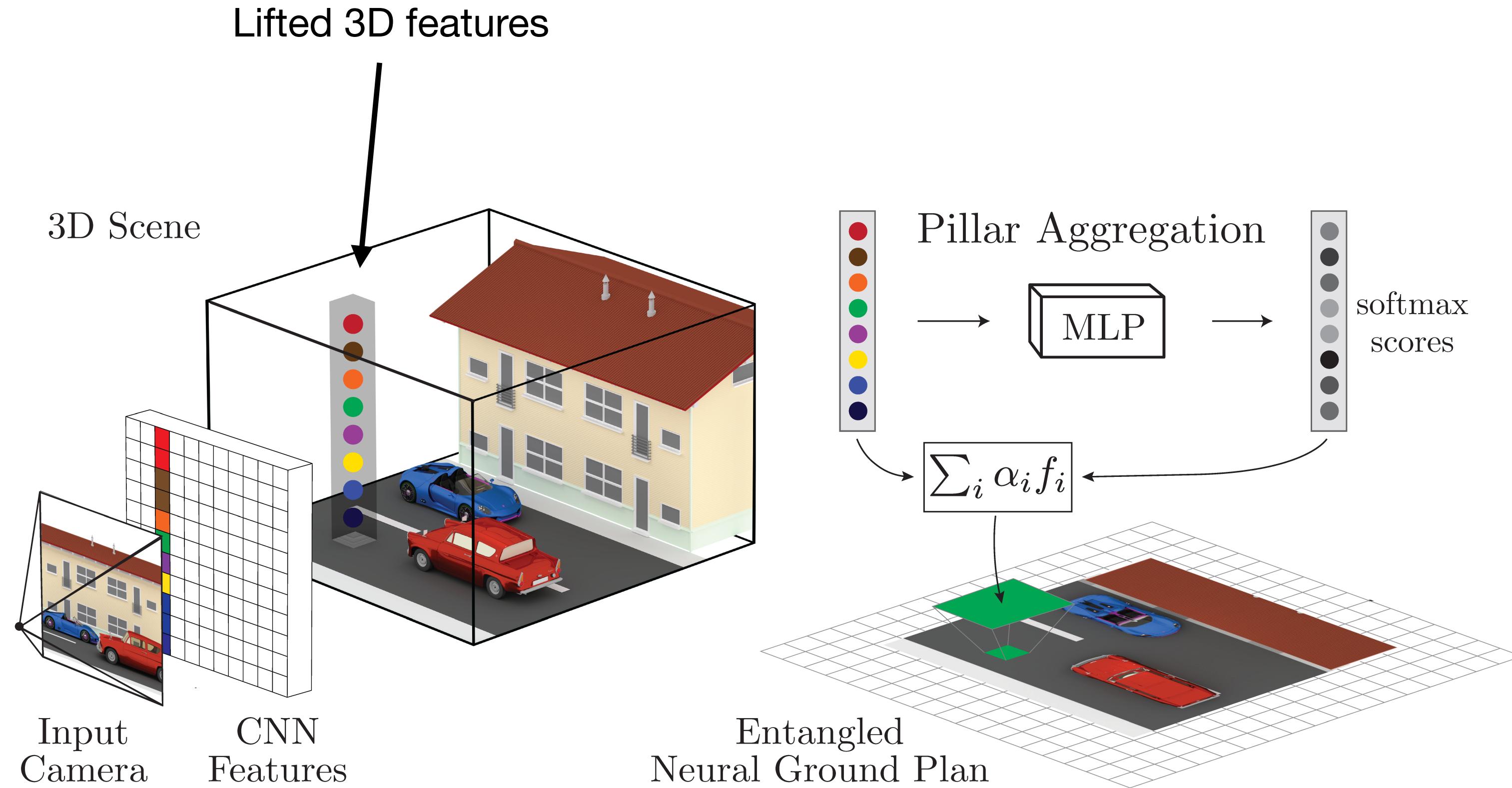
Conditional Ground Plans for Single-Image 3D Reconstruction



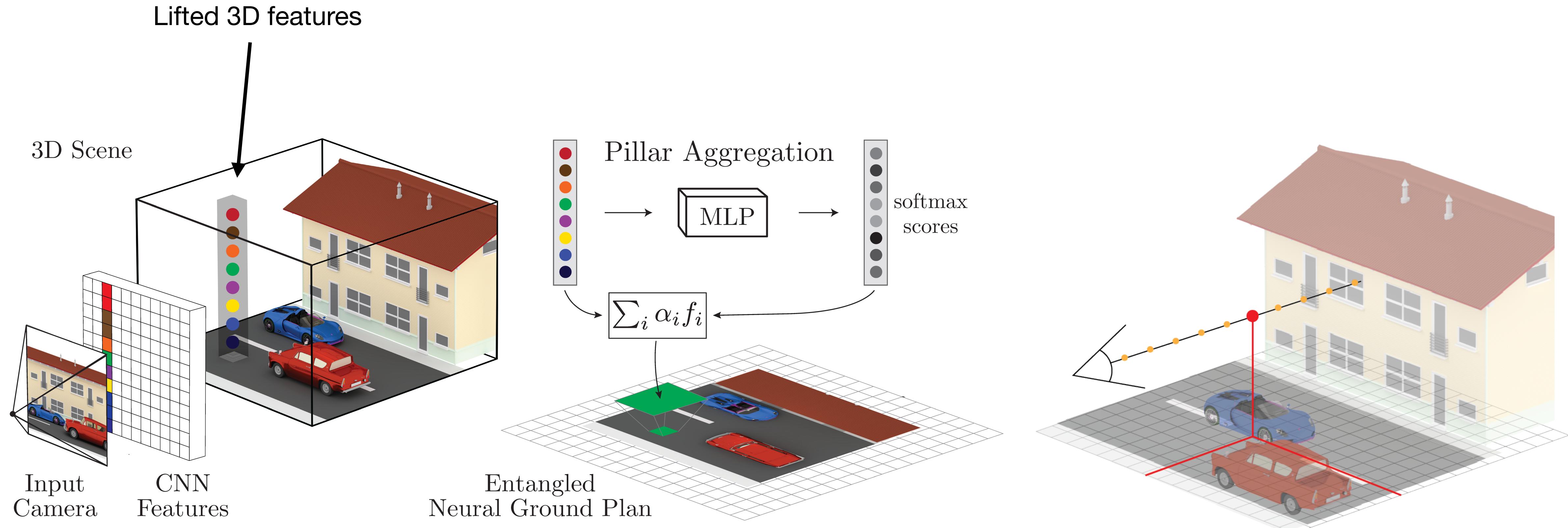
Conditional Ground Plans for Single-Image 3D Reconstruction



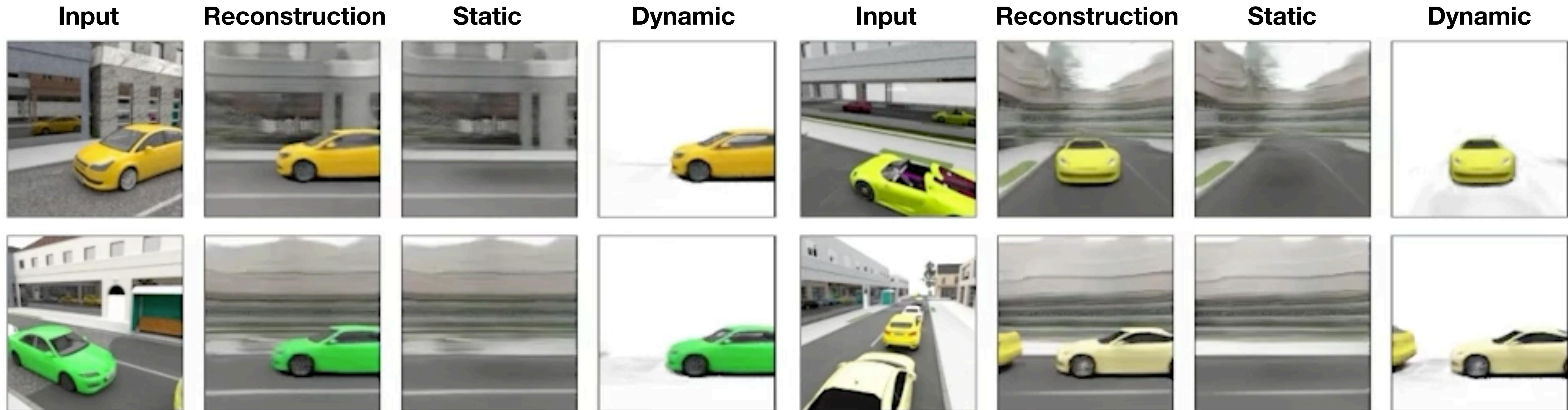
Conditional Ground Plans for Single-Image 3D Reconstruction



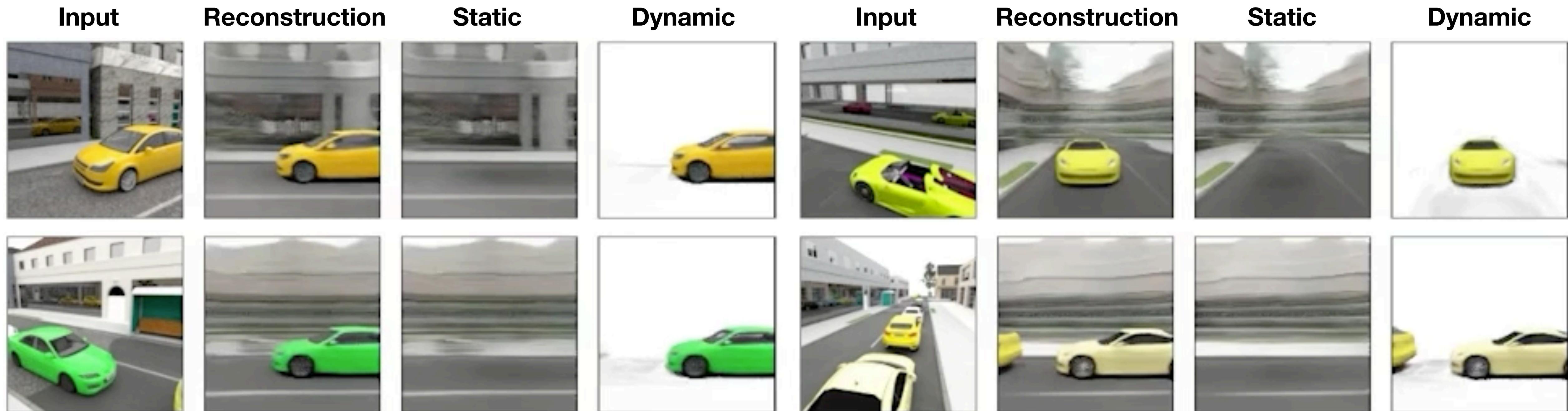
Conditional Ground Plans for Single-Image 3D Reconstruction



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



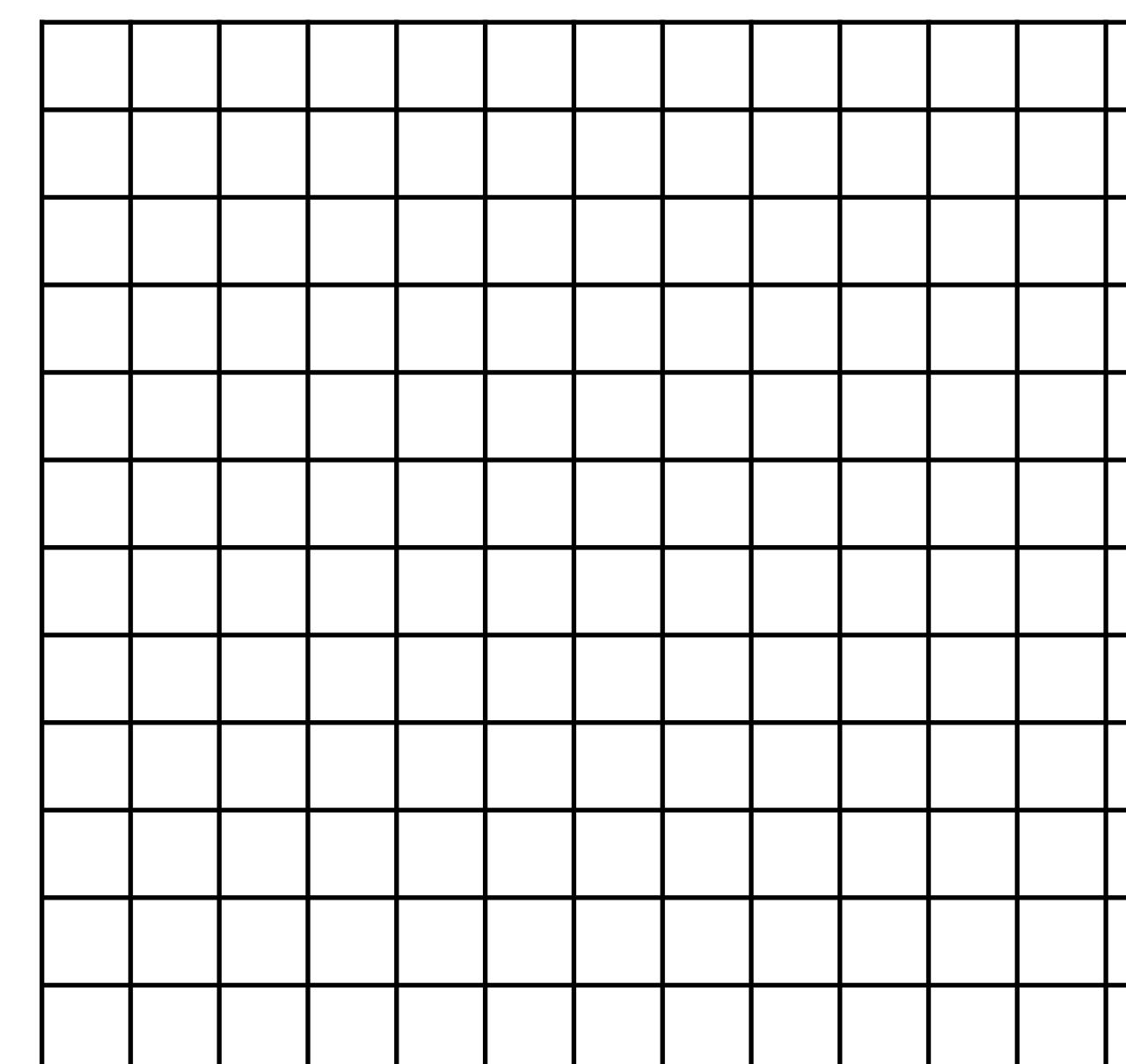
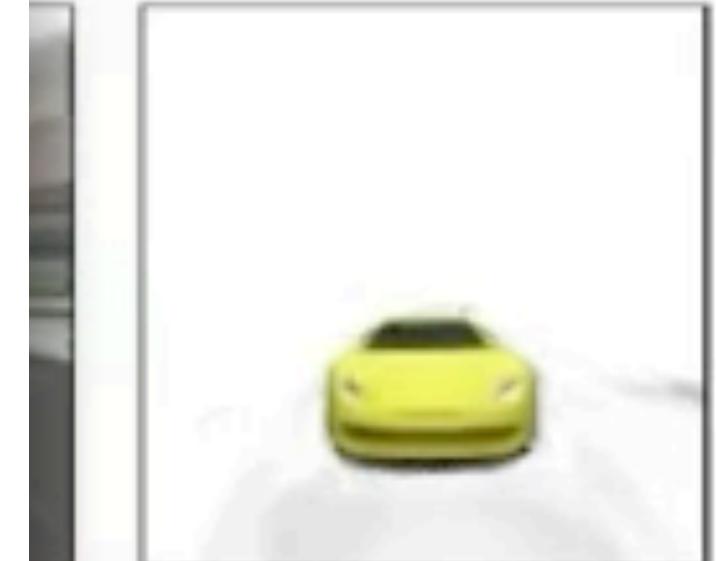
Static-dynamic disentanglement from a **single** image!



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

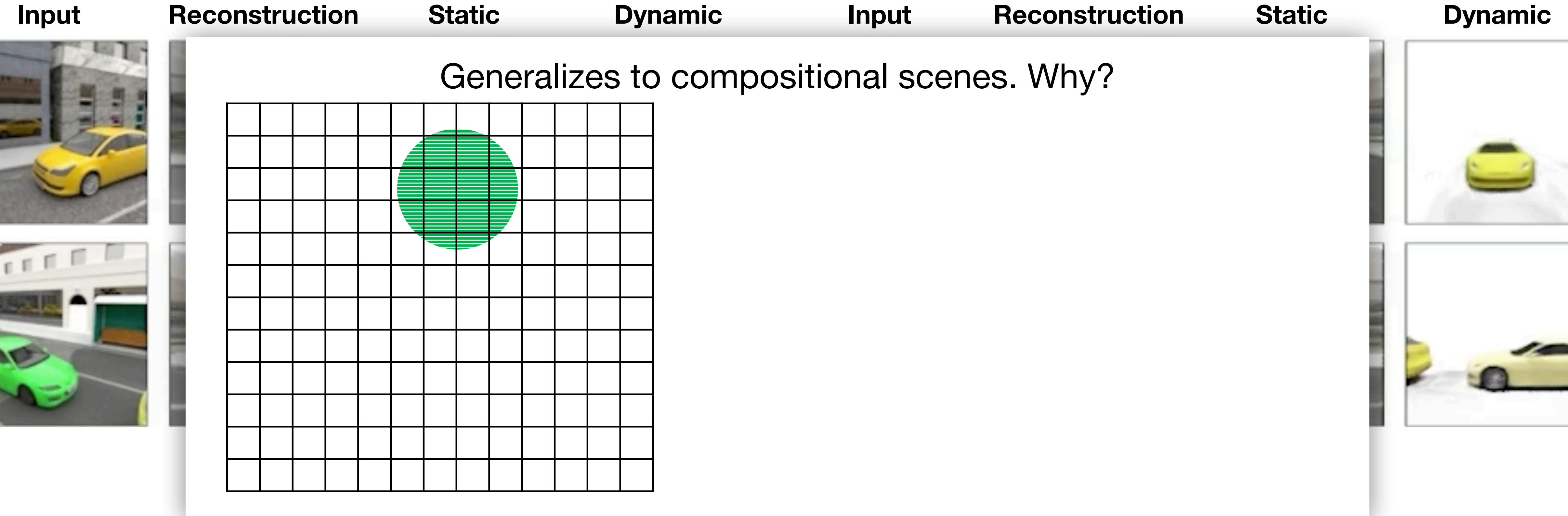
Input	Reconstruction	Static	Dynamic	Input	Reconstruction	Static	Dynamic
							
Generalizes to compositional scenes. Why?							

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

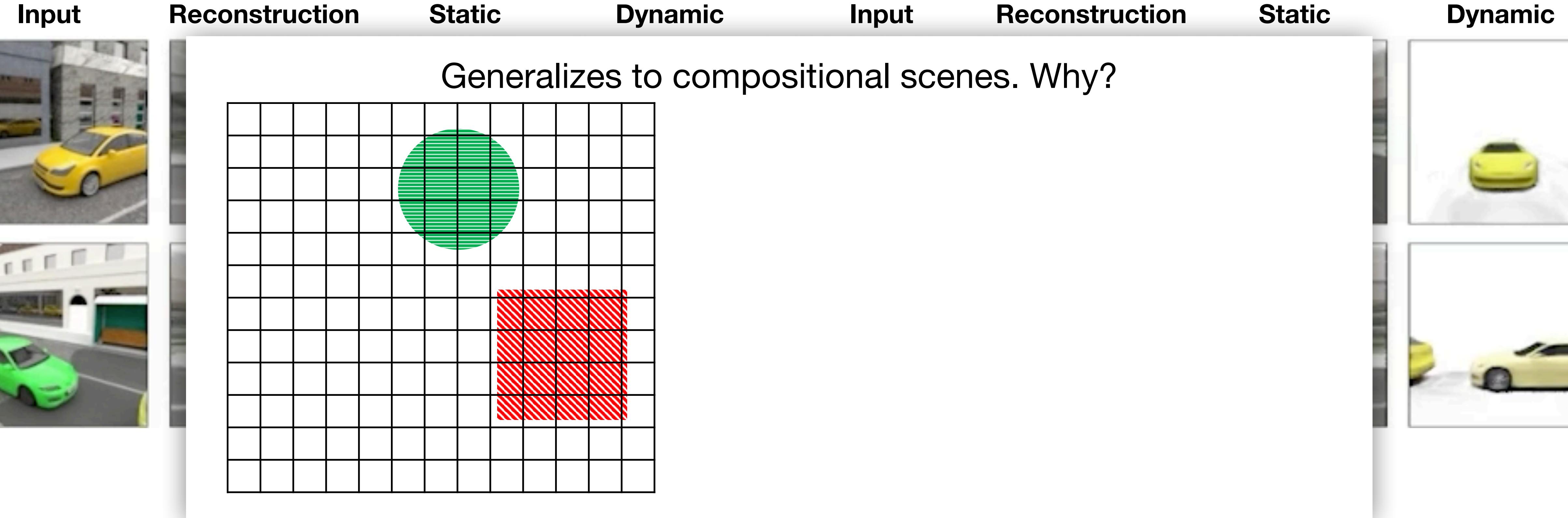
Input	Reconstruction	Static	Dynamic	Input	Reconstruction	Static	Dynamic
							
							

Generalizes to compositional scenes. Why?

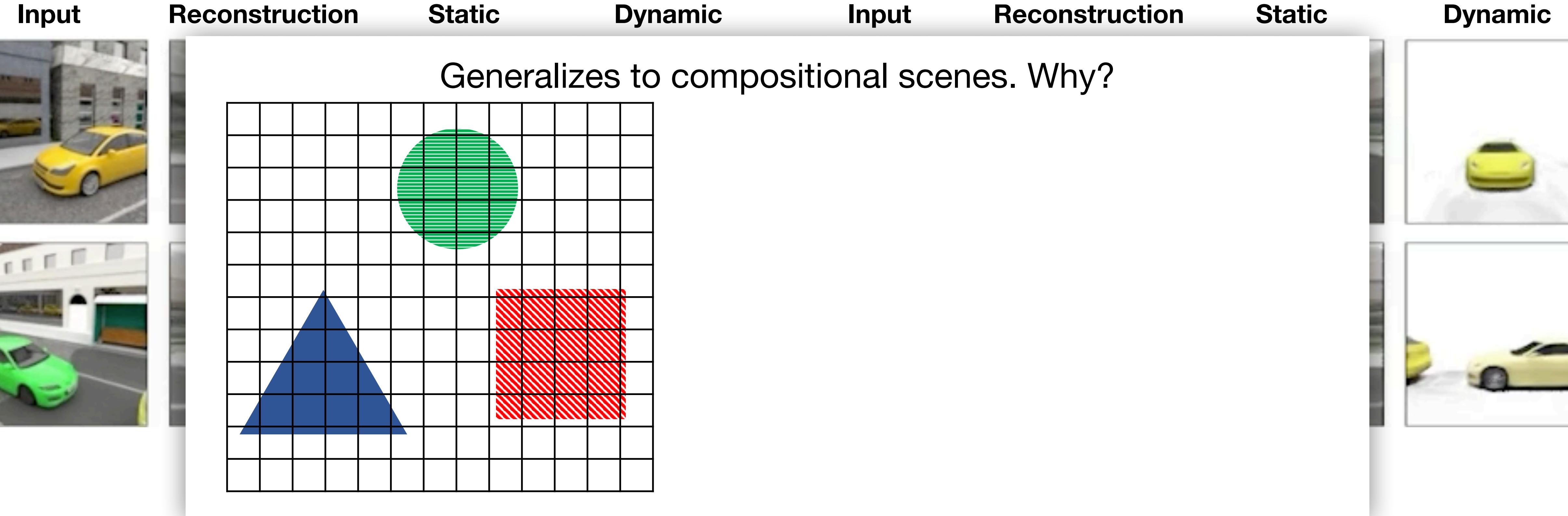
Static-dynamic disentanglement from a **single** image!



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



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



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

Input

Reconstruction

Static

Dynamic

Input

Reconstruction

Static

Dynamic

Generalizes to compositional scenes. Why?

Locality + shift-equivariance!

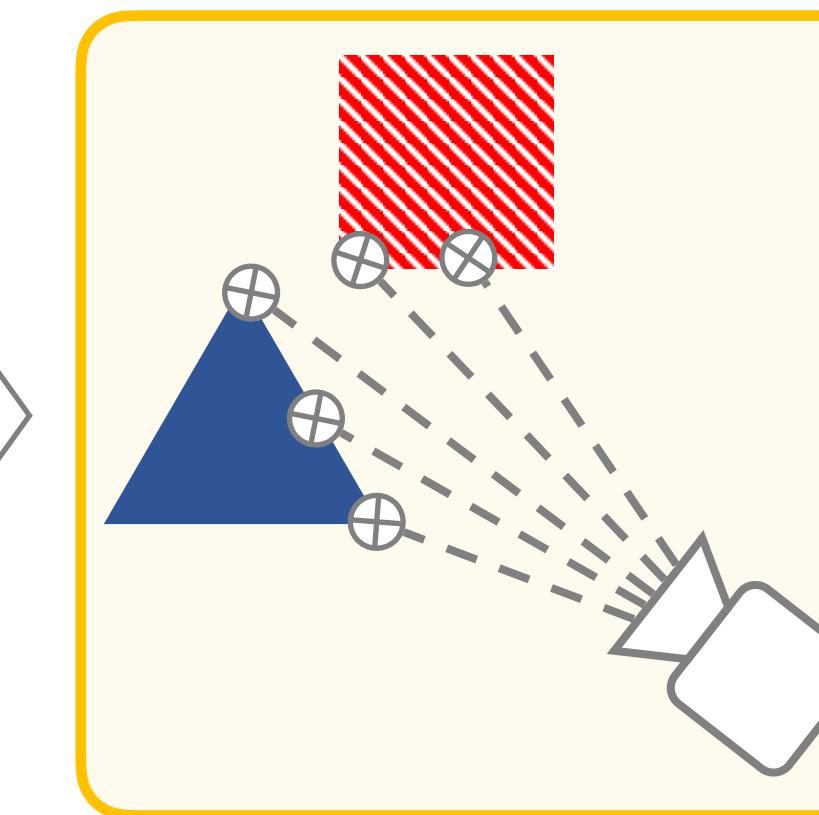
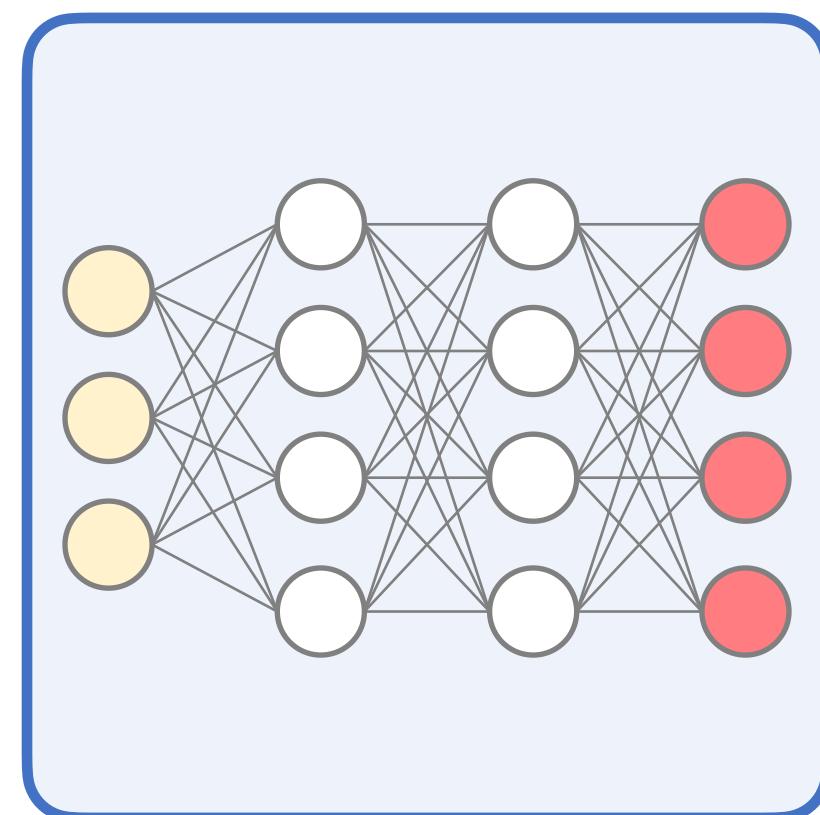
Today: *Learned* inference algorithms!

Observations



Inference

?



Renderings



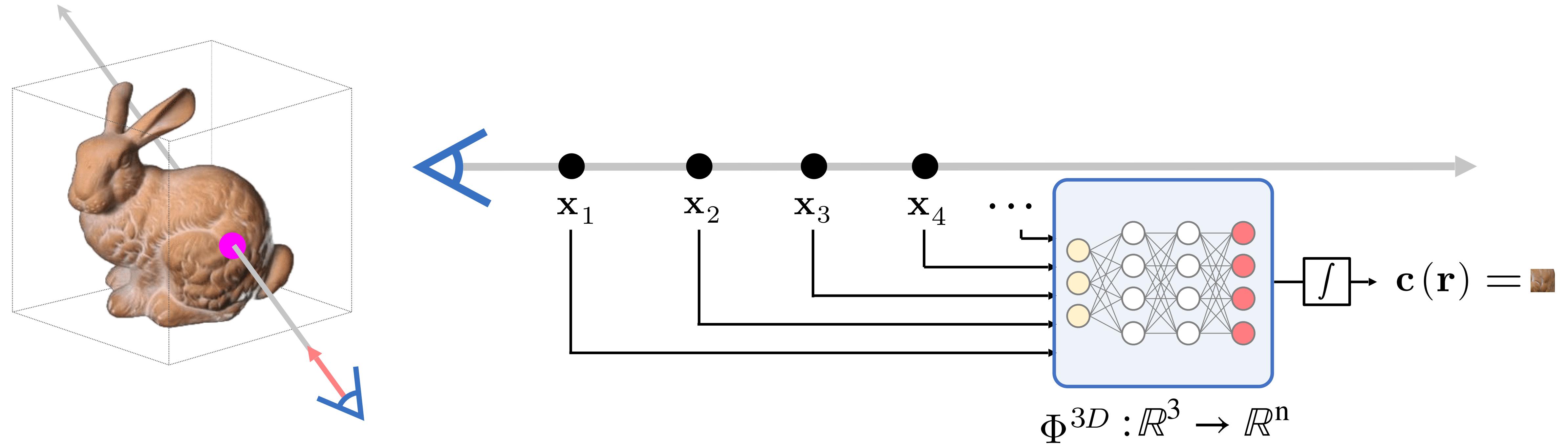
Why?

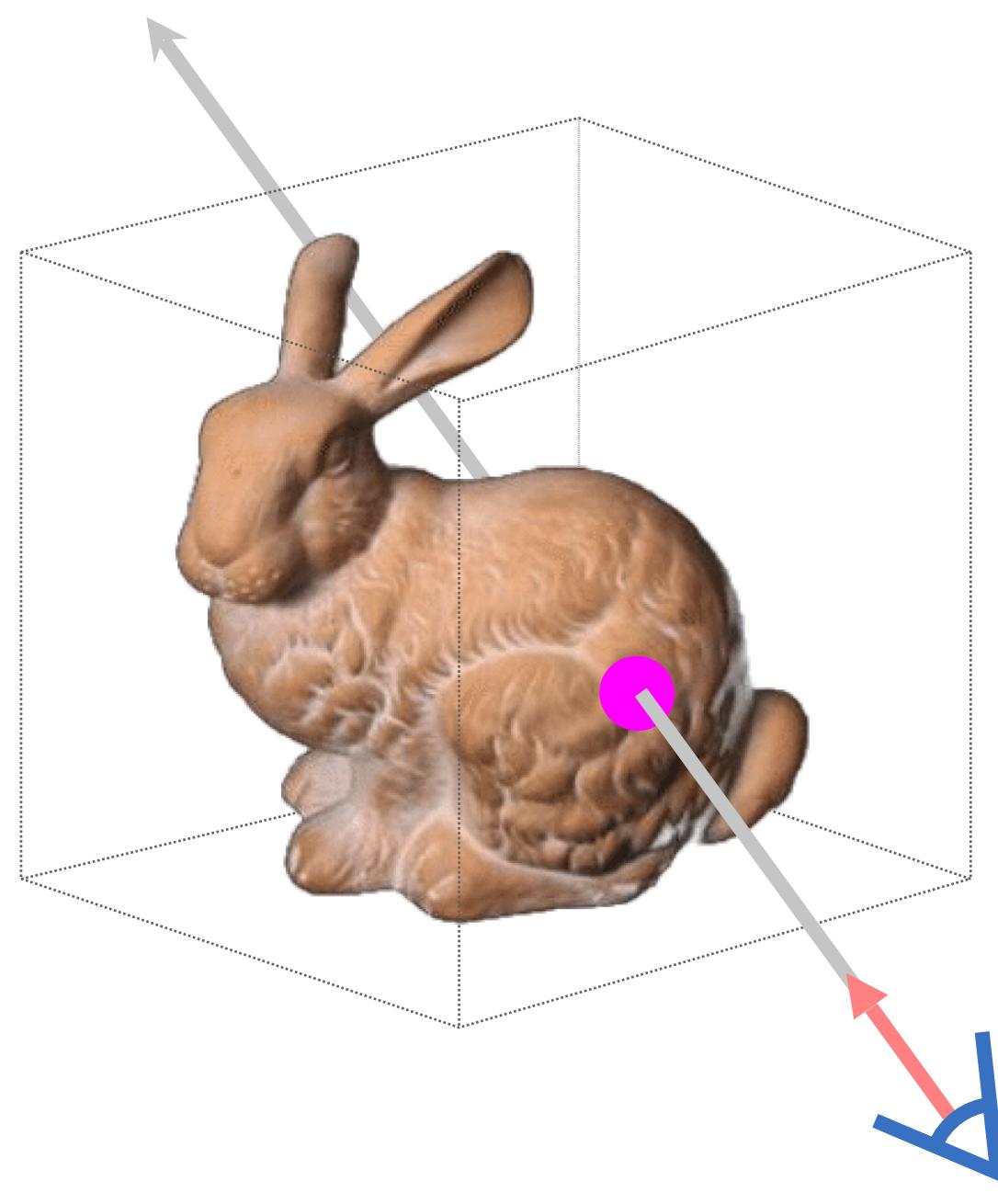
We humans can reconstruct 3D from incomplete observations, by using knowledge that we have learned about the world. Deep learning is the best way we know to date to learn such priors from data.

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





A

x_1

x_2

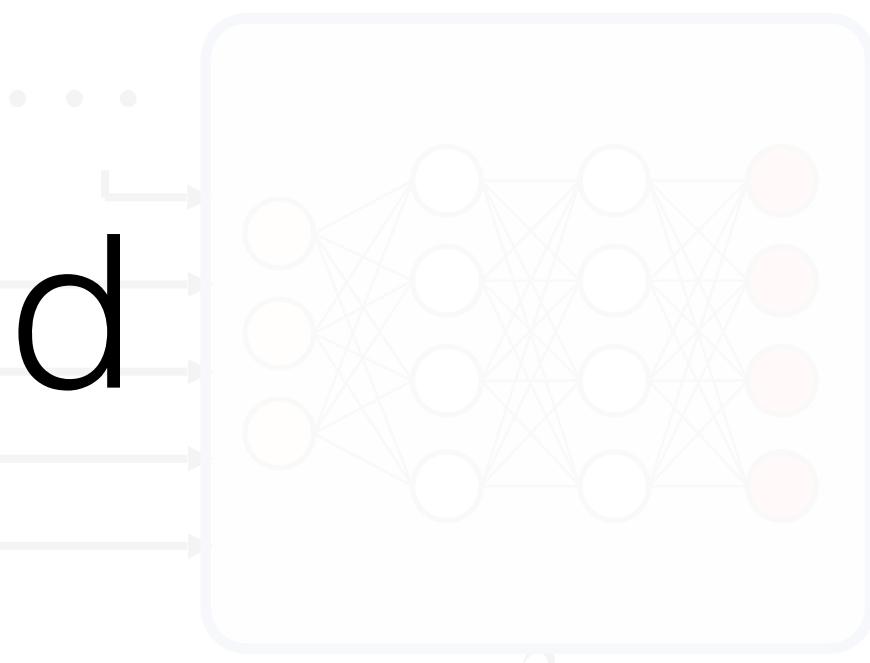
x_3

x_4

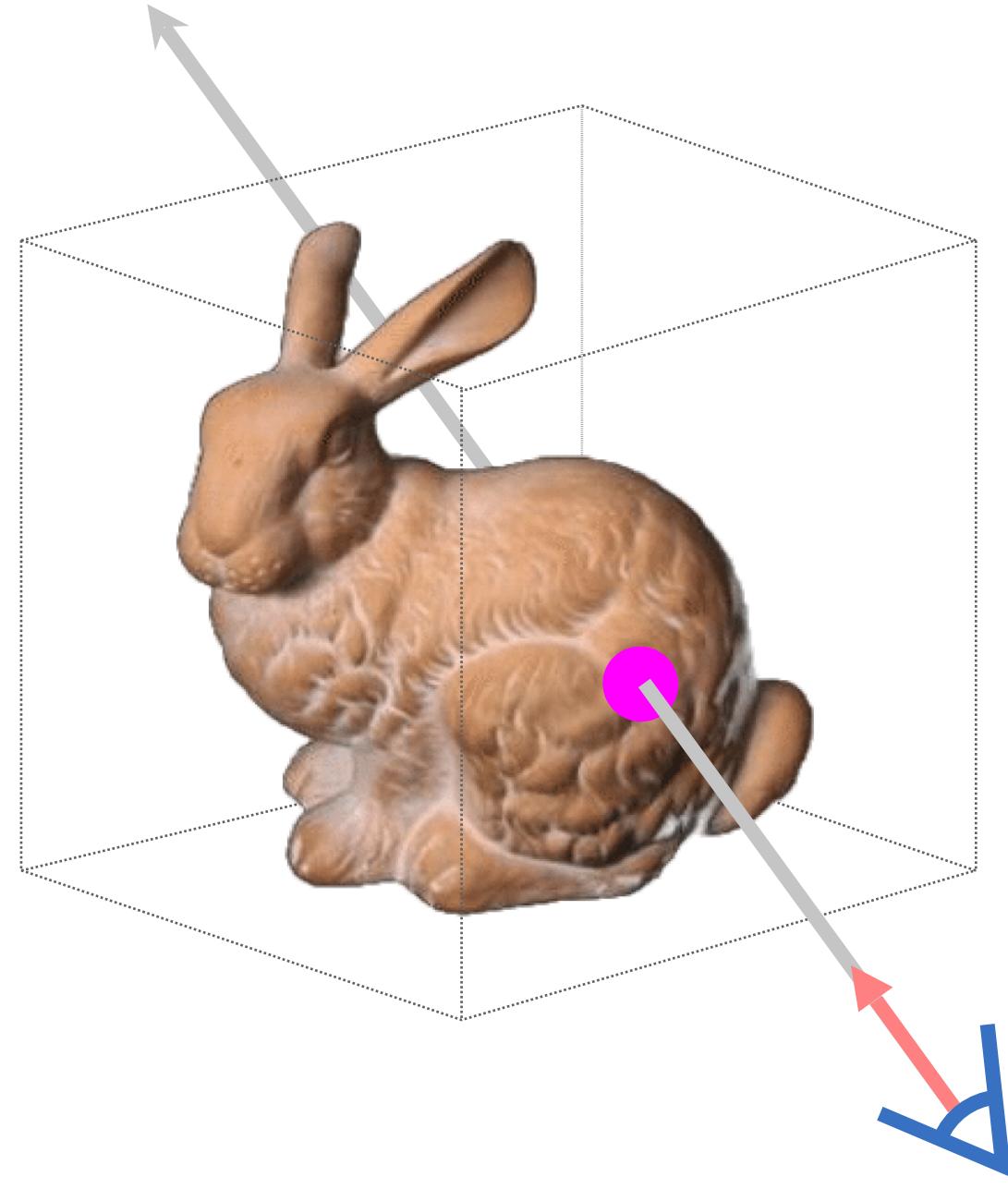
...

Light Field

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



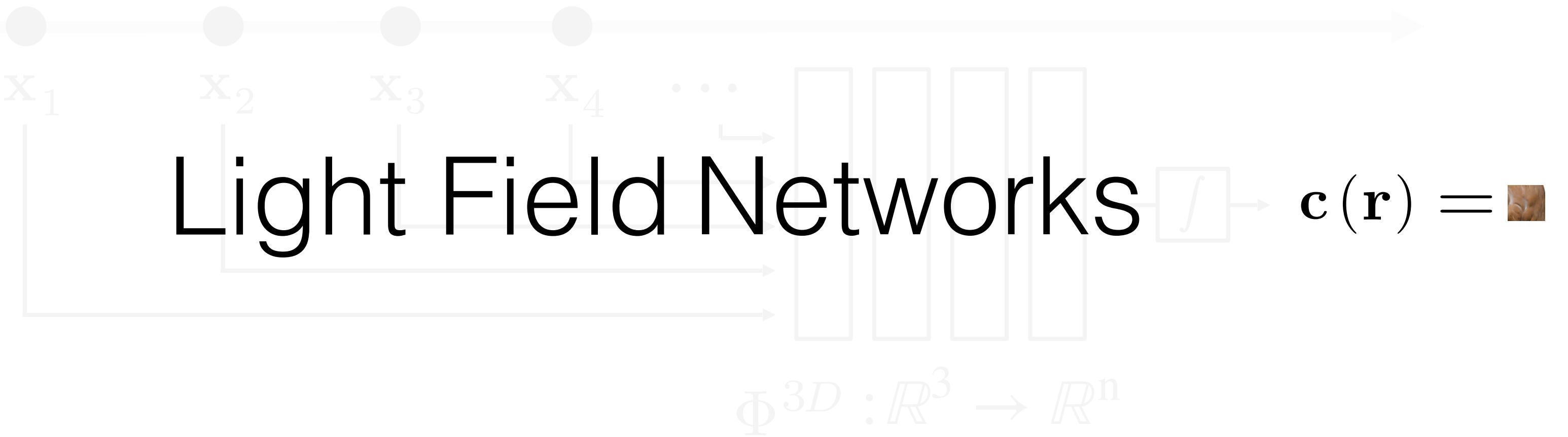
$$c(\mathbf{r}) =$$



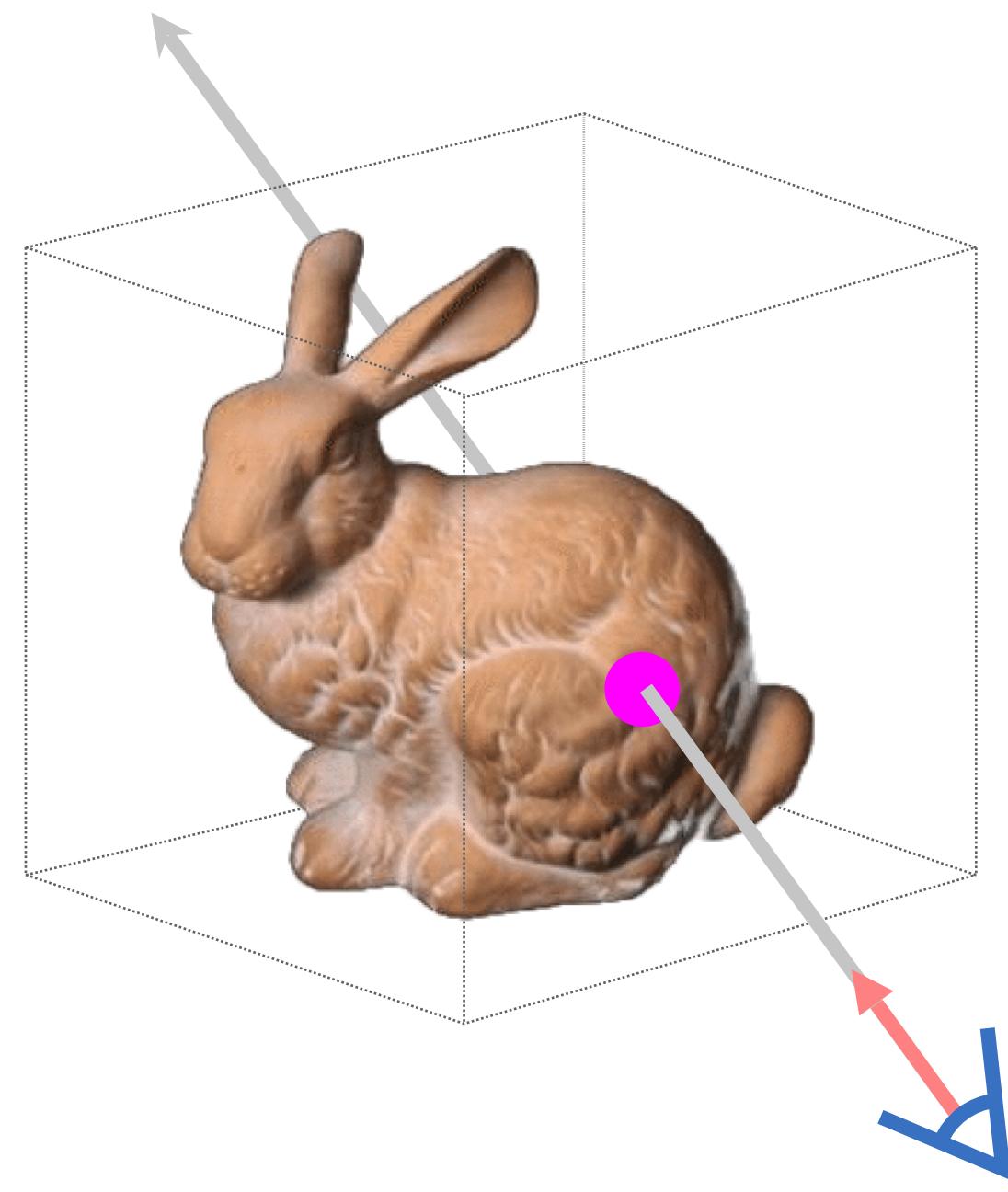
A

x_1

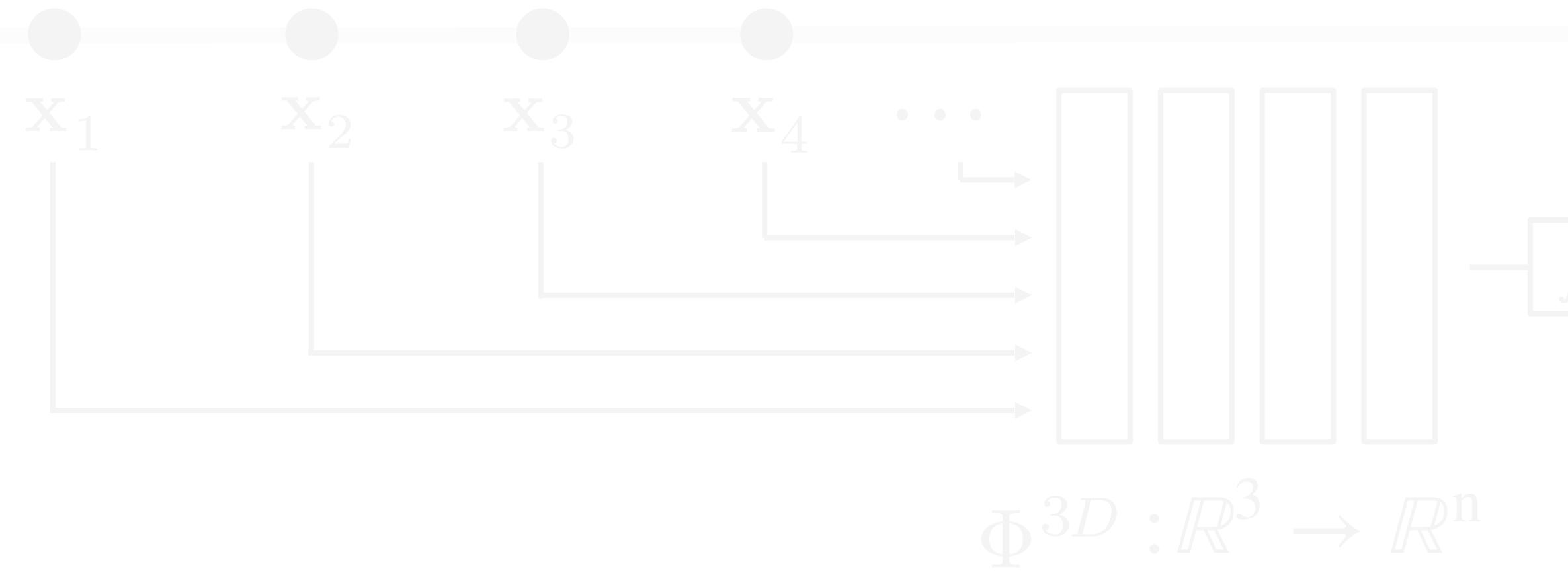
Light Field Networks



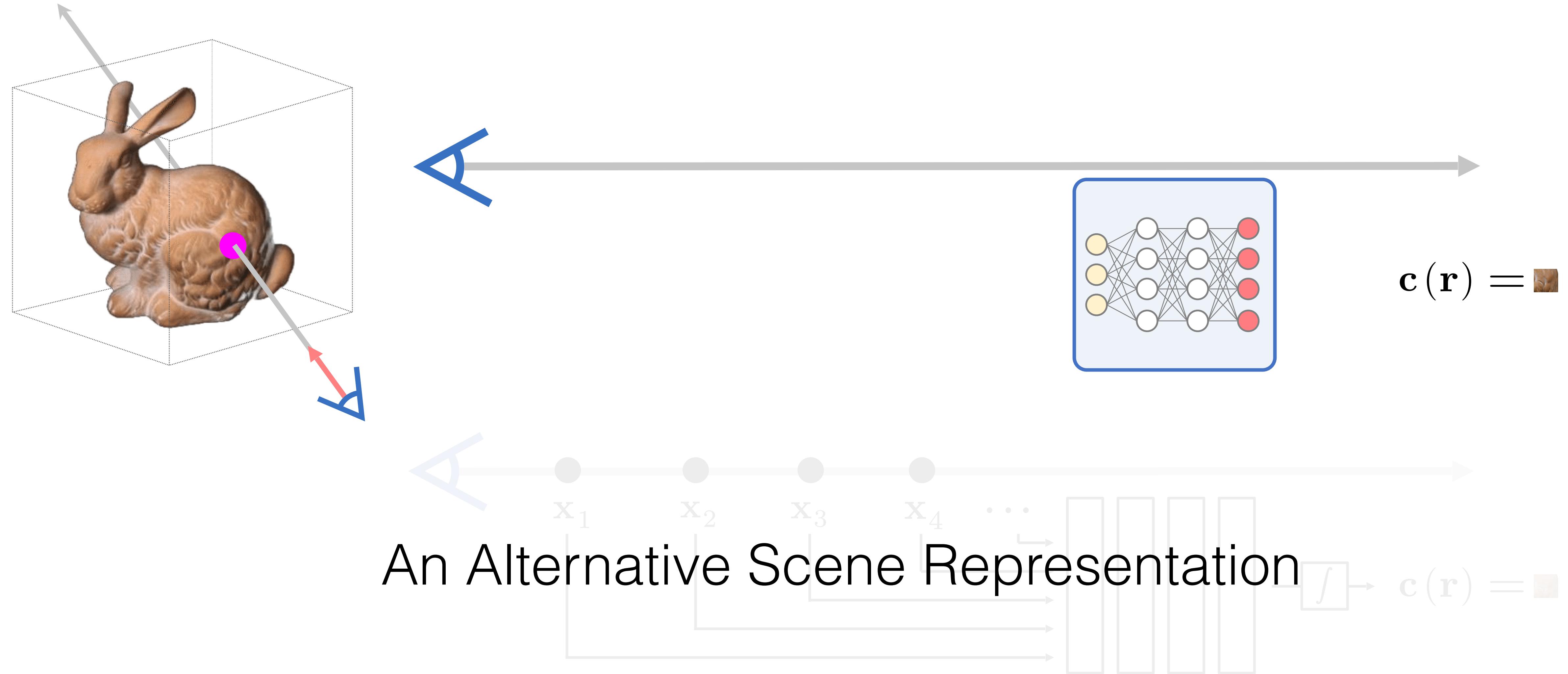
Light Field Networks



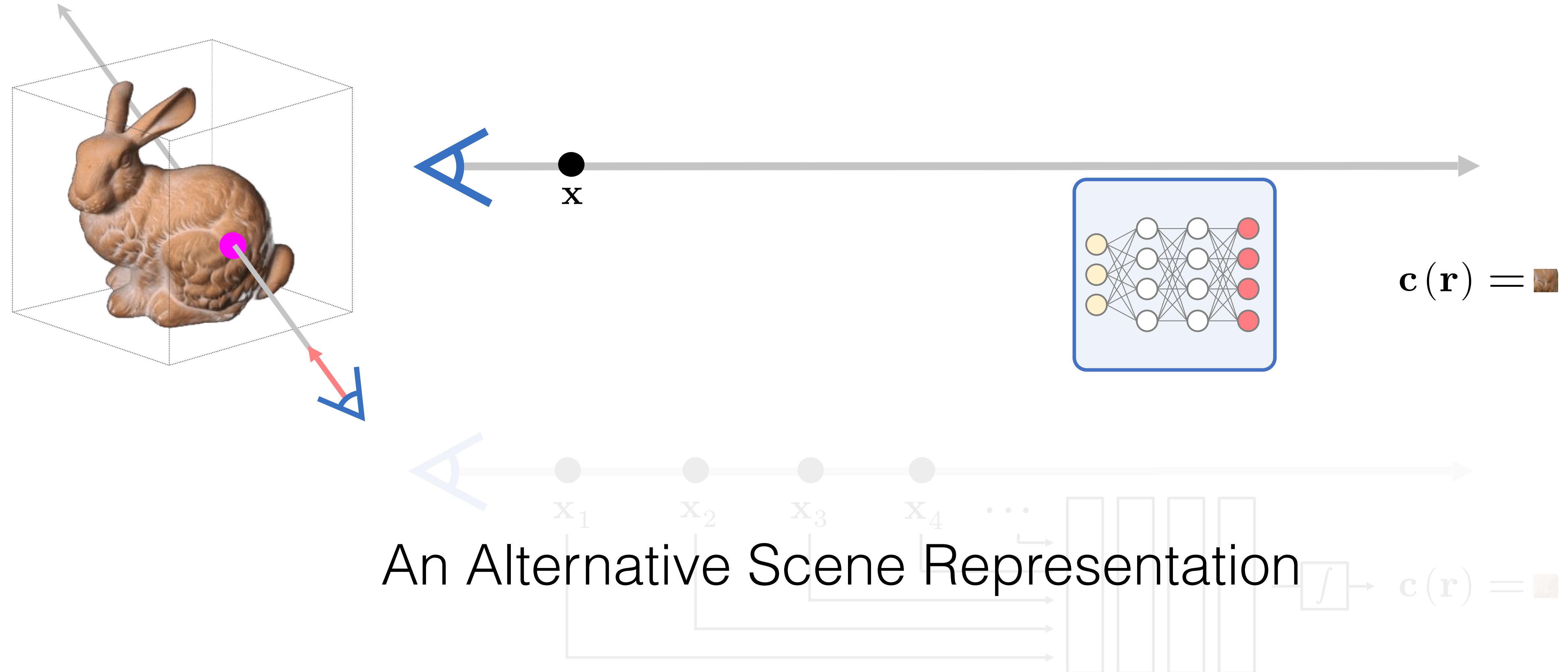
A



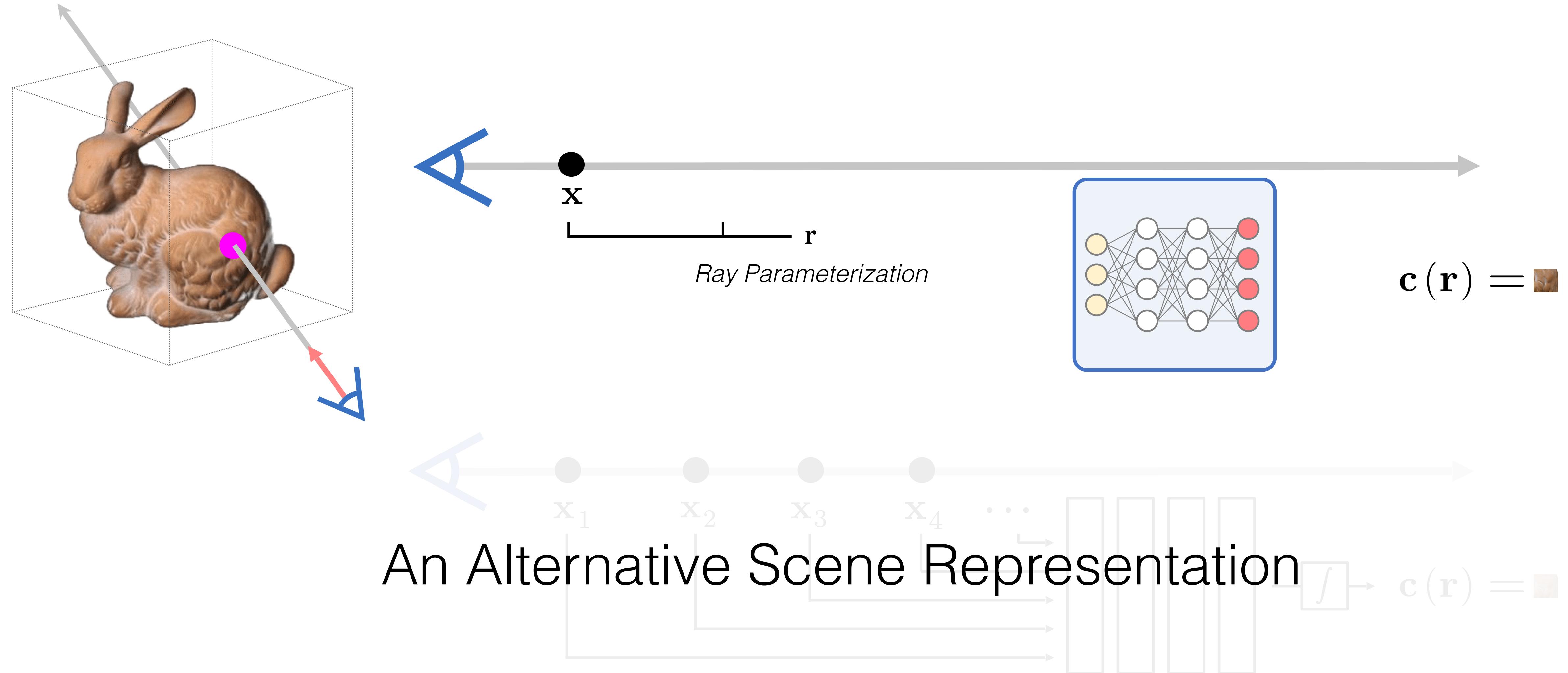
Light Field Networks



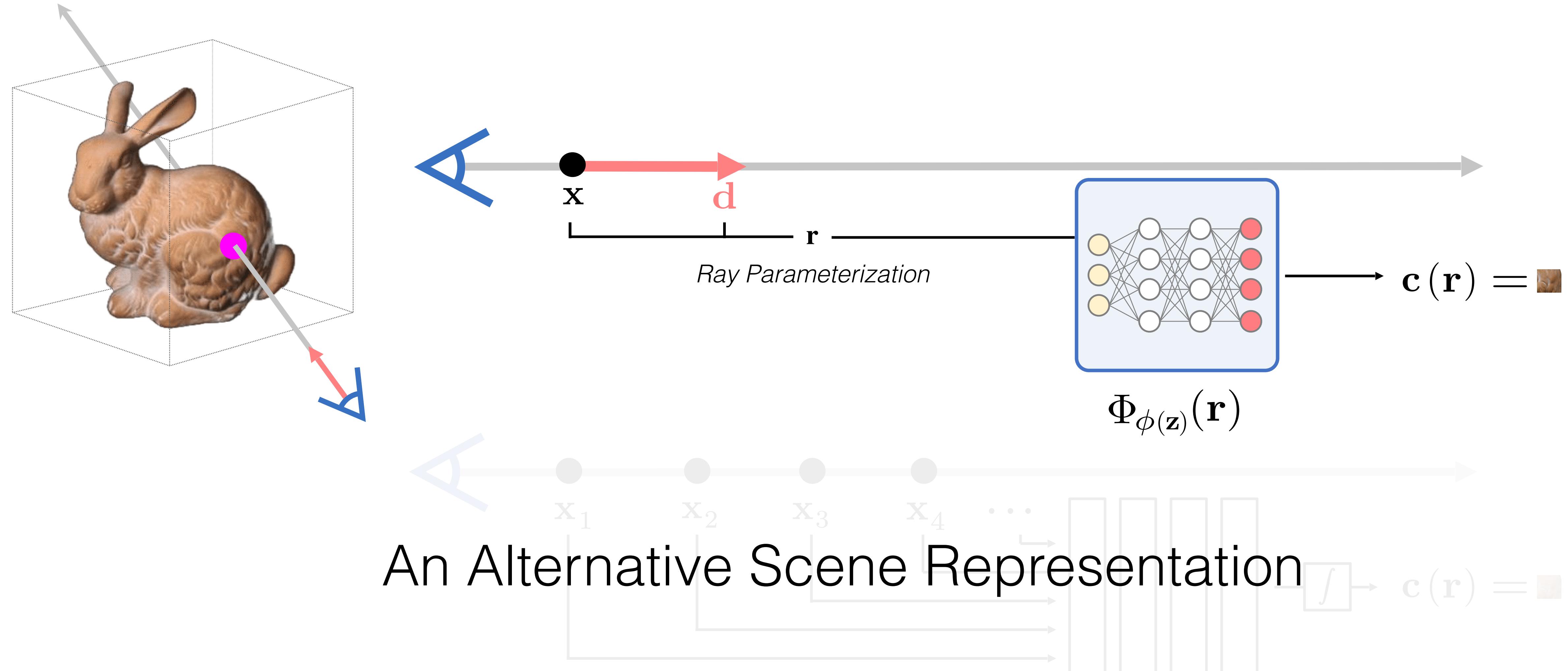
Light Field Networks



Light Field Networks



Light Field Networks



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

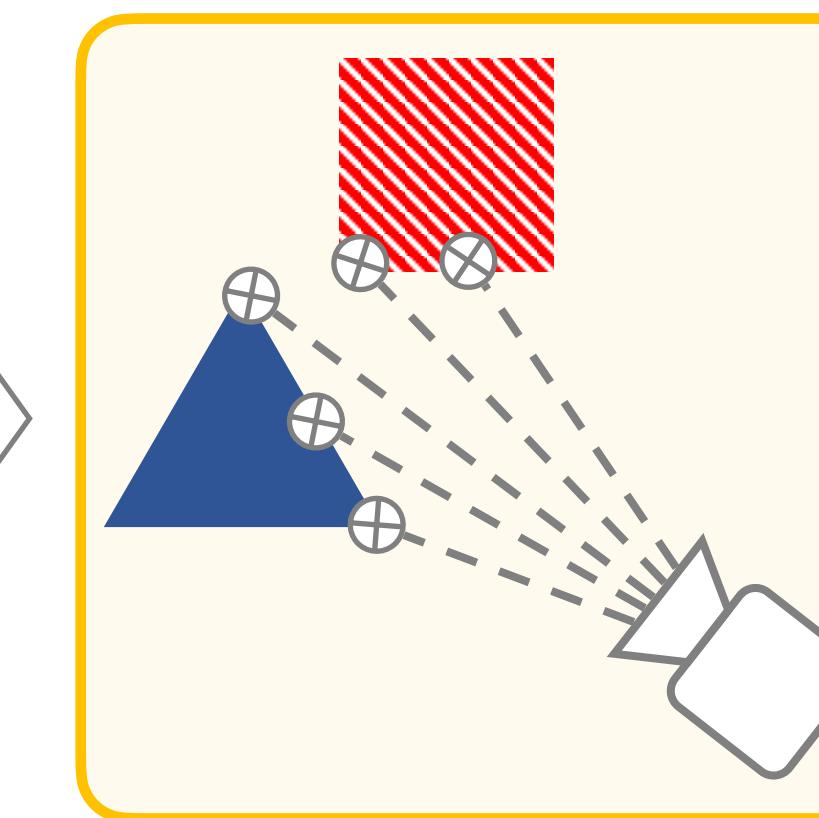
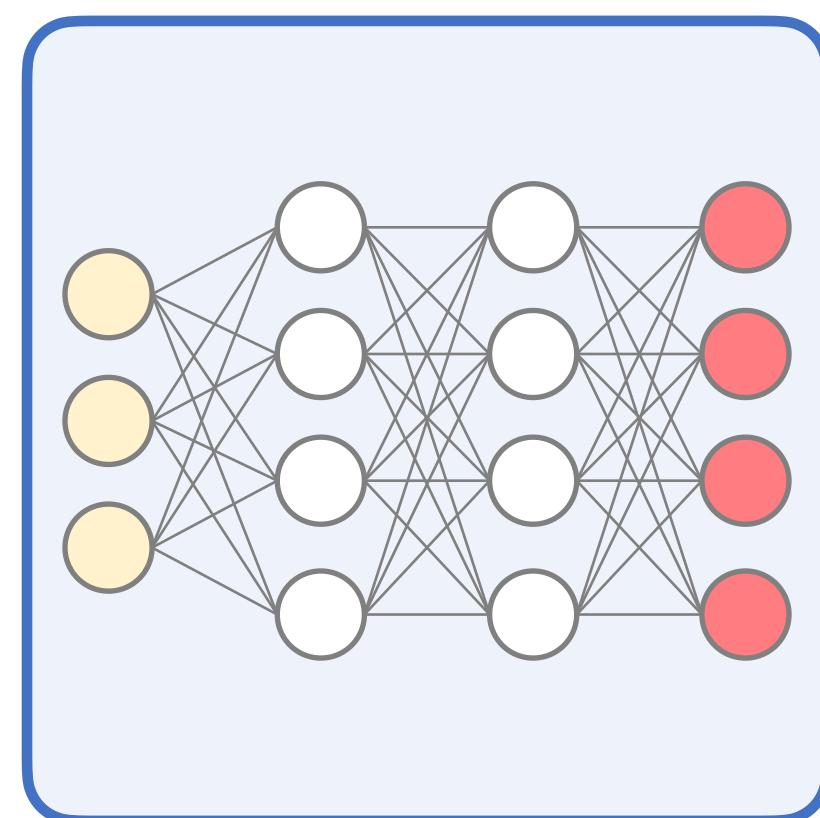
Today: *Learned* inference algorithms!

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

