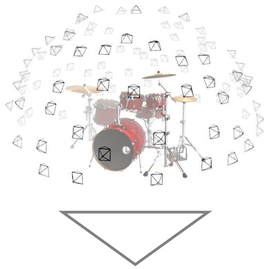
ADVANCES IN COMPUTER VISION

MODULE 1:
Geometry

Deep Learning for Single-Image 3D Reconstruction

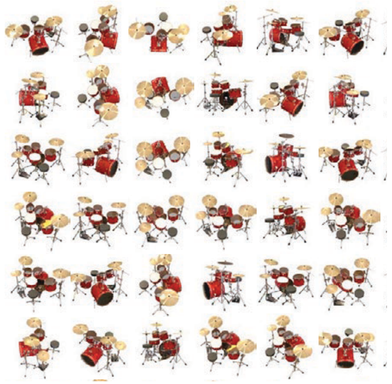
Prof. Vincent Sitzmann

Recap: Differentiable Rendering

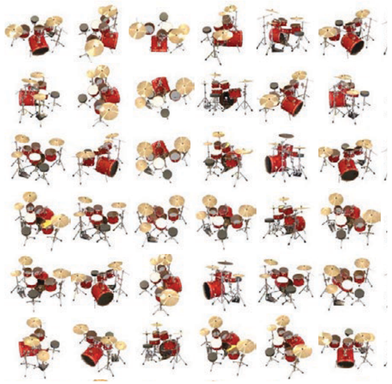


Scene Representation

Neural Renderer



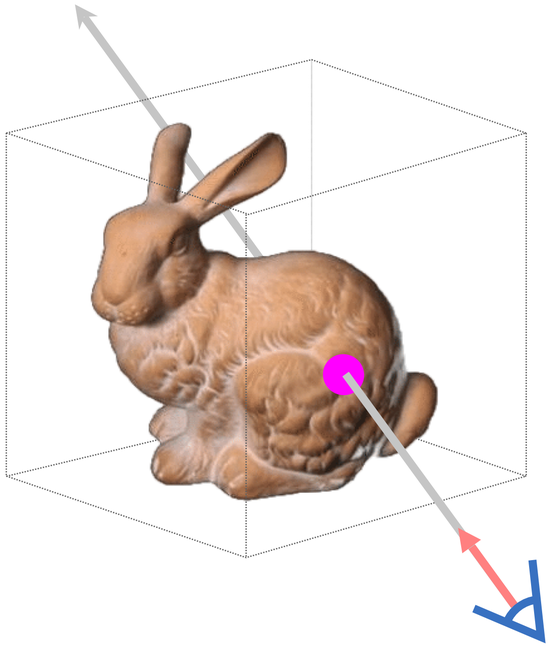
Rendered Images

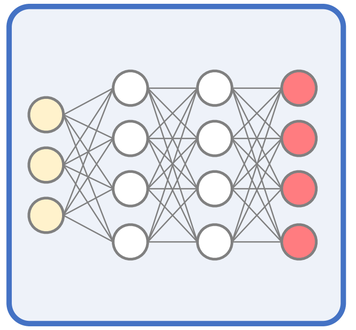


GT Images

Optimization via SGD

General structure of Neural Renderers for 3D Fields

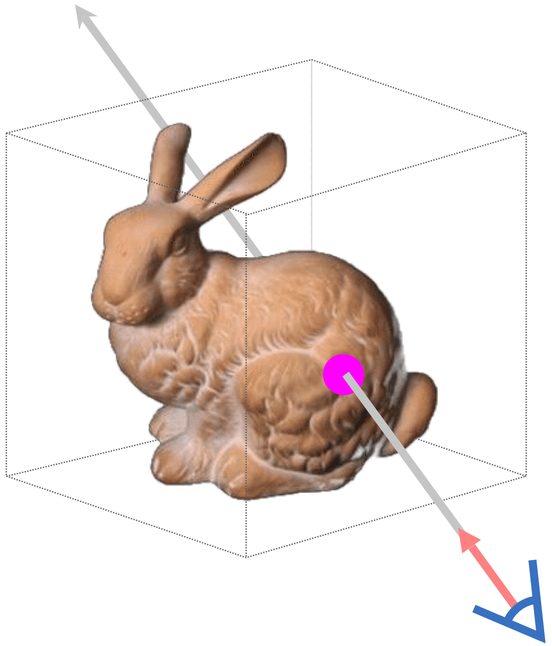


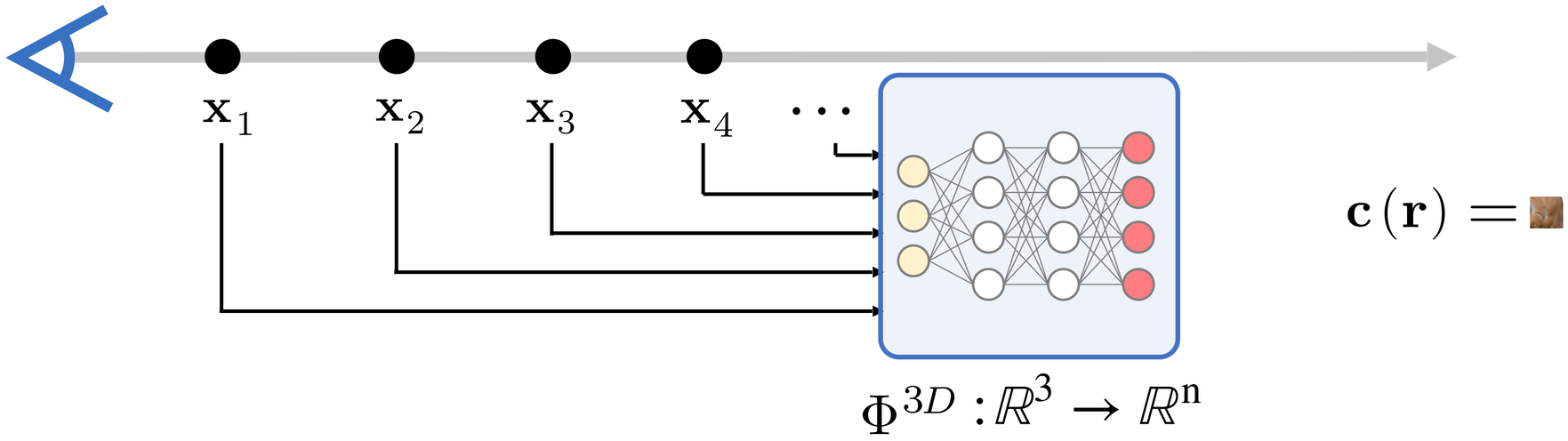


$\Phi^{3D} : \mathbb{R}^3 \rightarrow \mathbb{R}^n$

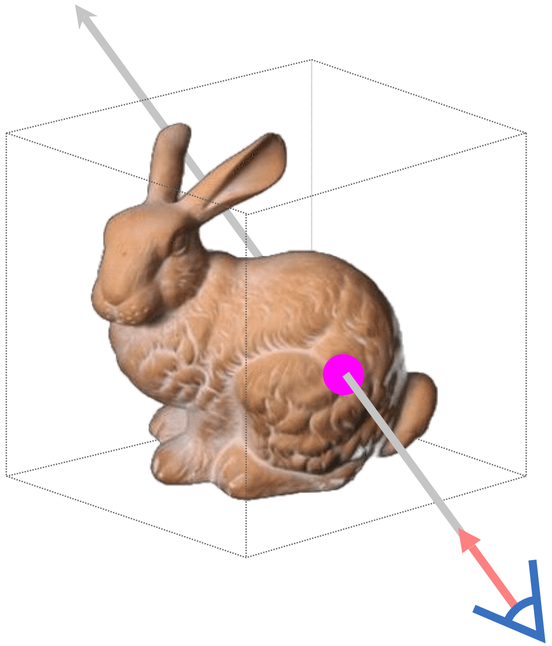
$\mathbf{c}(\mathbf{r}) =$ 

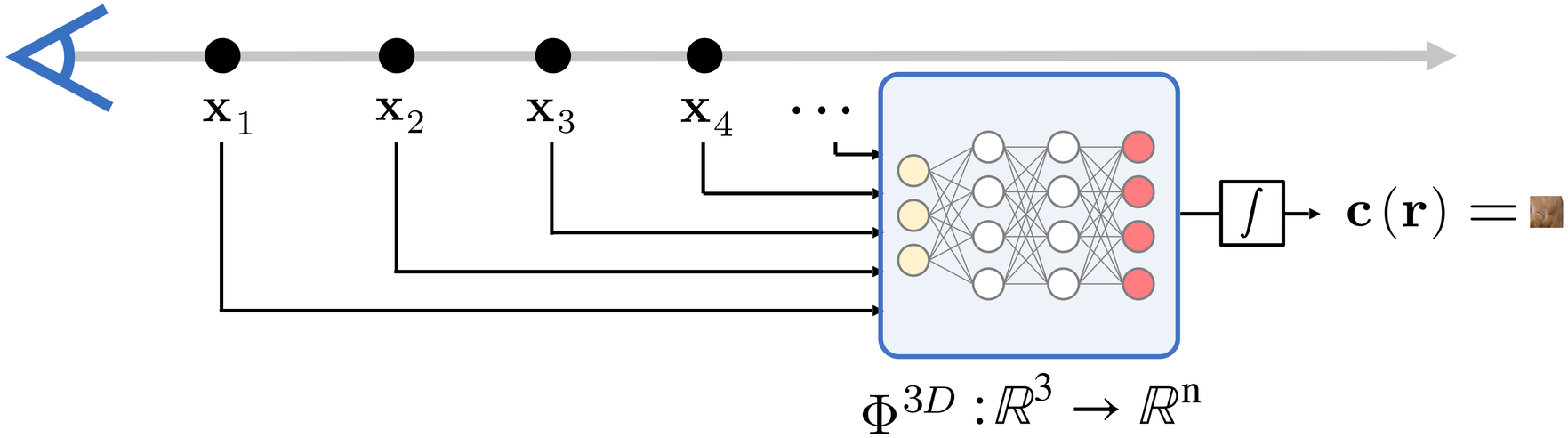
General structure of Neural Renderers for 3D Fields



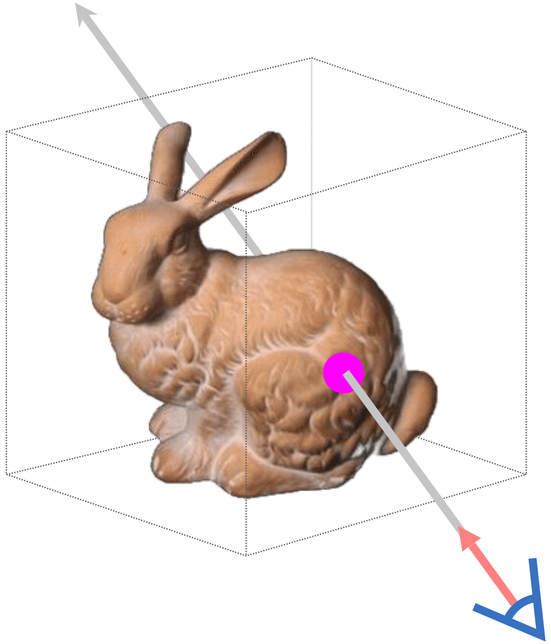


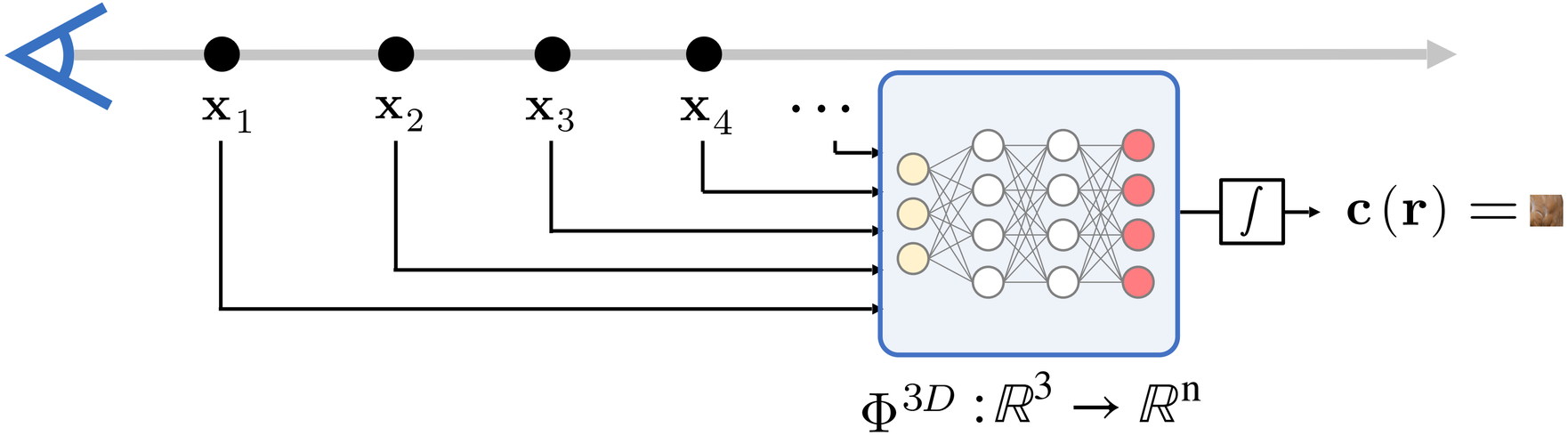
General structure of Neural Renderers for 3D Fields





General structure of Neural Renderers for 3D Fields





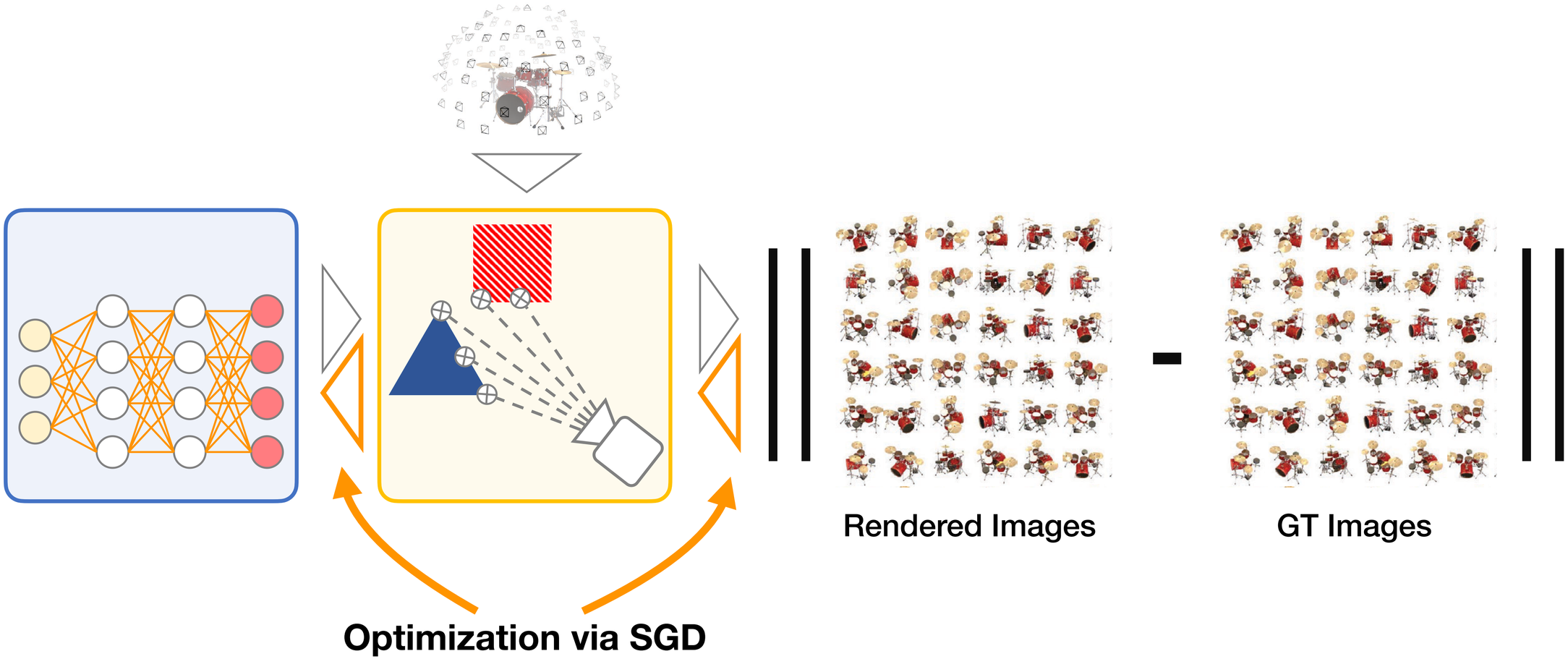
Sphere-Tracing [JC Hart, 1996]

Volumetric

Hybrid implicit-volumetric

Learned aggregation

What did we do here?

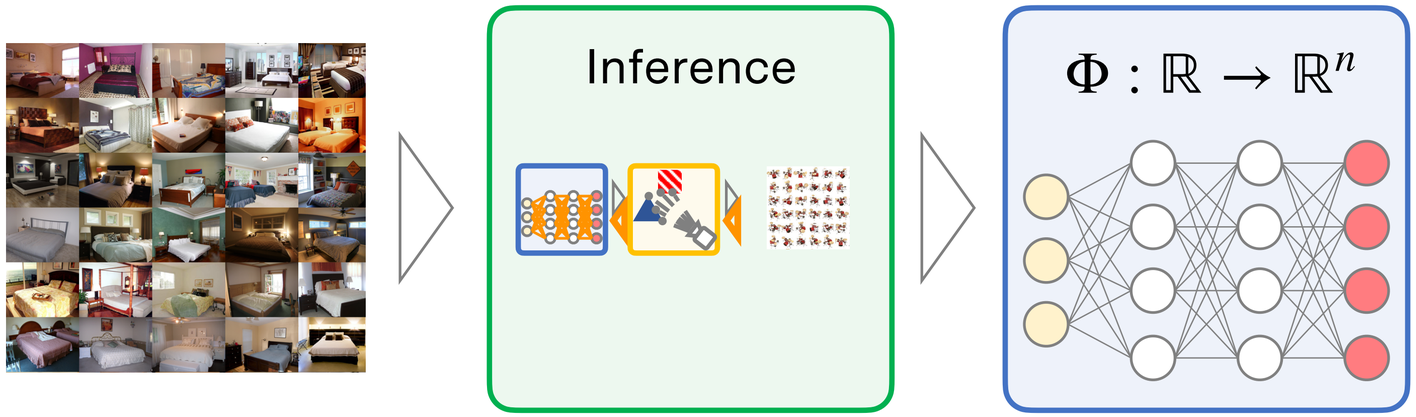


What did we do here?

Observations

What did we do here?

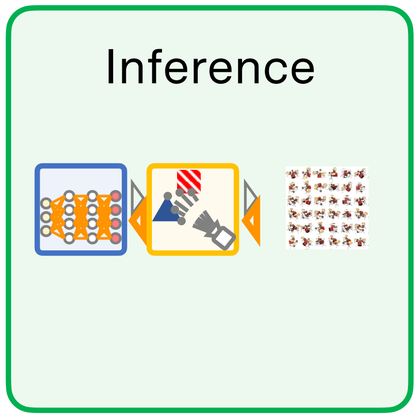
Observations

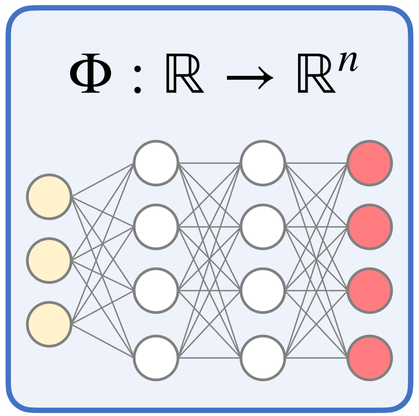


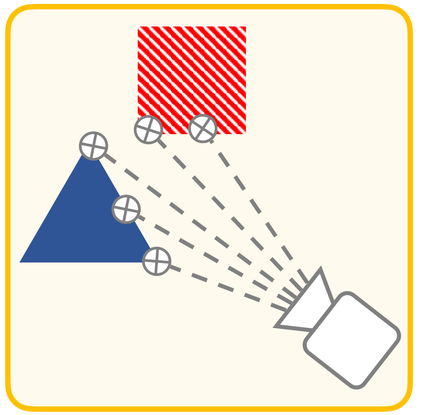
We inferred “hidden/latent” variables 3D appearance and geometry, given “observable” variables RGB images and camera poses.

What did we do here?

Observations



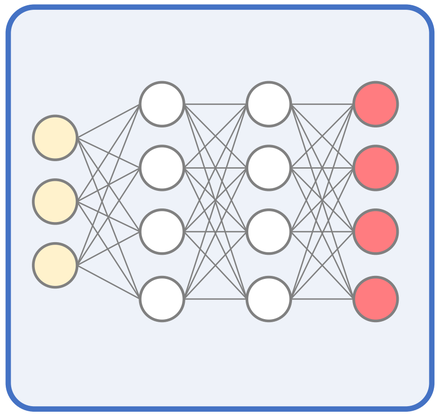




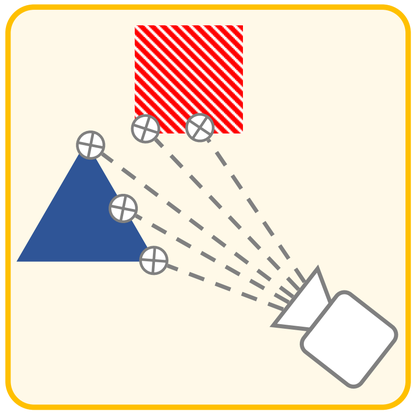
Renderings

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?

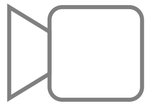


Srn_{ϕ}



Render_{θ}

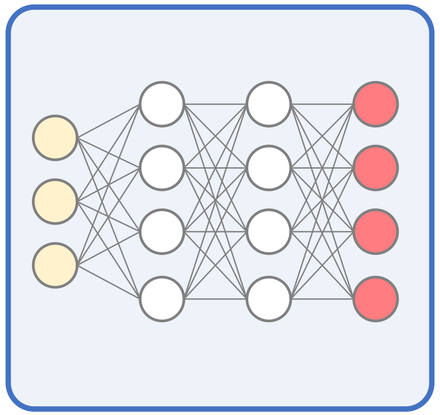
$+$



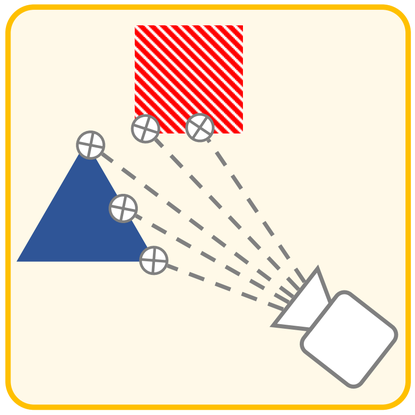
$(\mathcal{I}, \xi)$

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

What if observations don't constrain scene representation?



Srn_{ϕ}



Render_{θ}

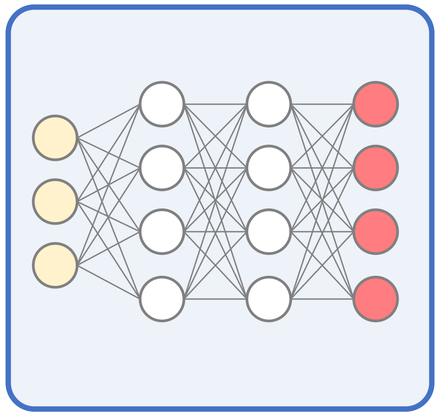
$+$

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

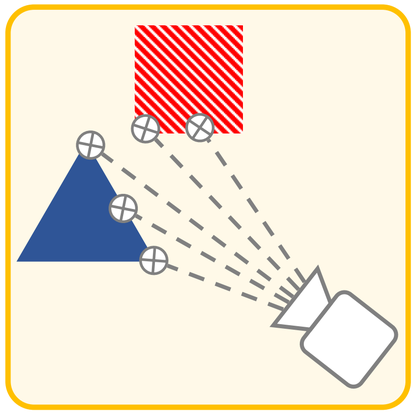
No 3D inform.

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

What if observations don't constrain scene representation?

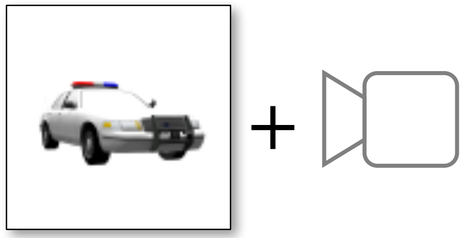


Srn_{ϕ}



Render_{θ}

Input



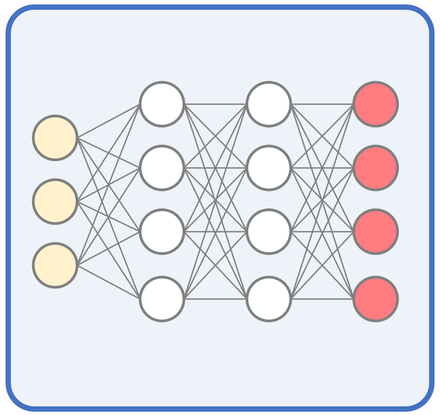
$+$

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

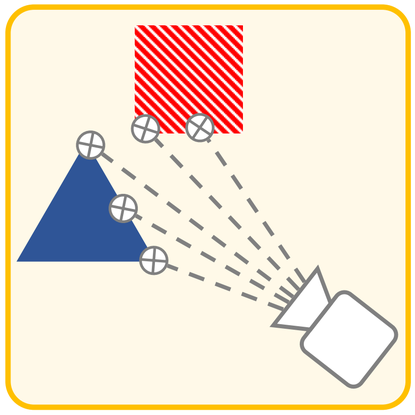
No 3D inform.

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

What if observations don't constrain scene representation?



Srn_{ϕ}



Render_{θ}

Input

$+$

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

No 3D inform.

Normal map

RGB

GT

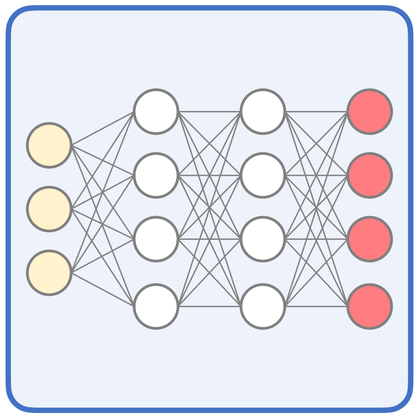
$$\underset{\phi,\theta}{\text{argmin}} \left\| \text{Render}_{\theta}(\text{Srn}_{\phi}, \xi) - \mathcal{I} \right\|$$

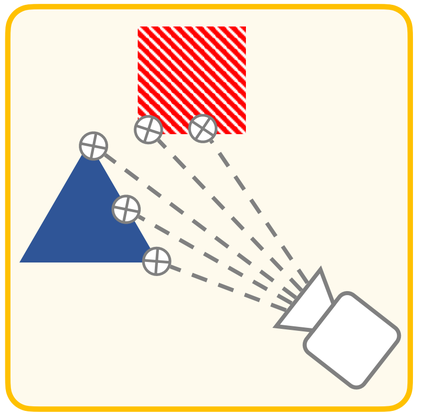
Today: *Learned* inference algorithms!

Observations

Inference

?

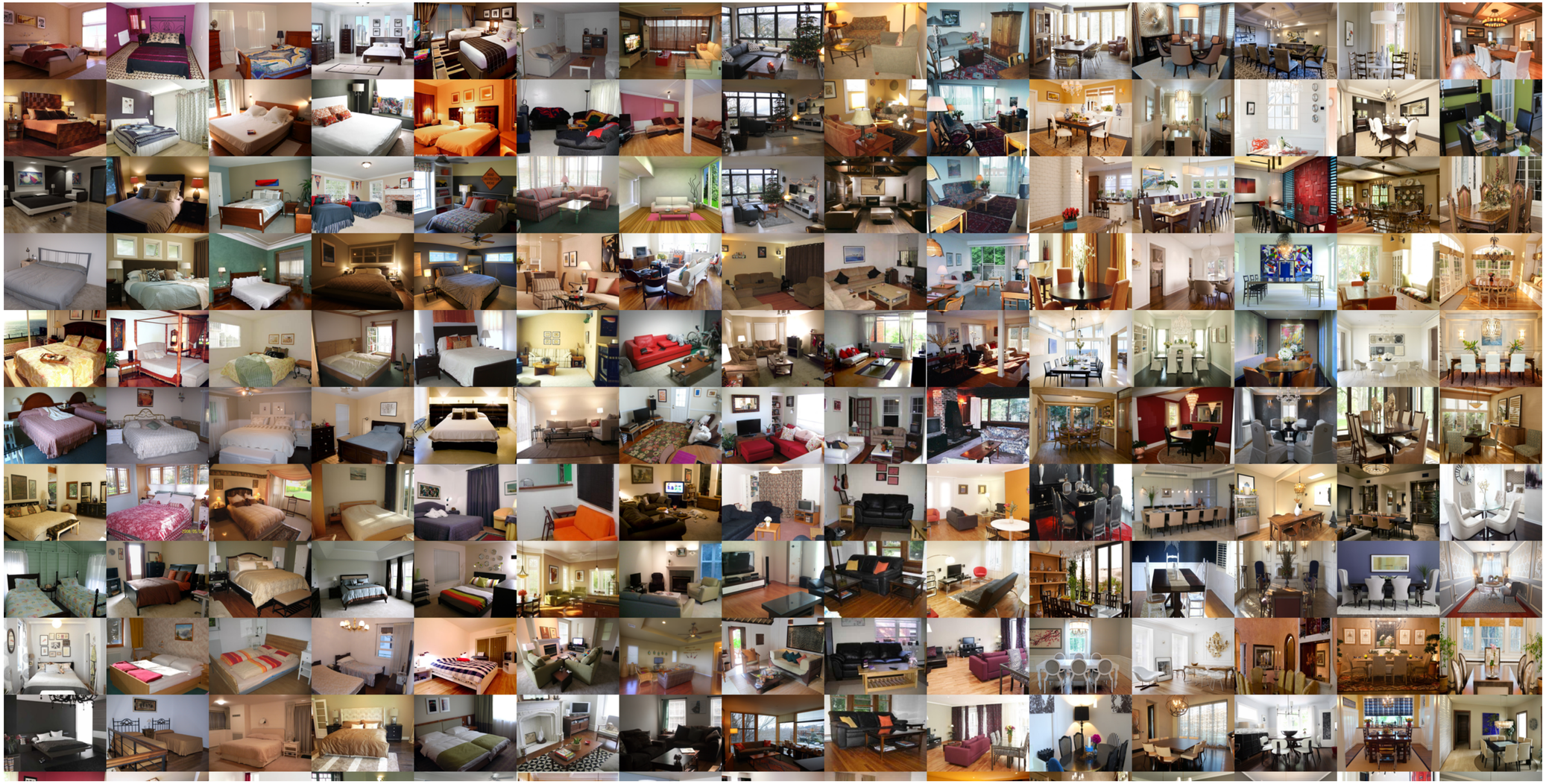




Renderings

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"

$\mathcal{F}$

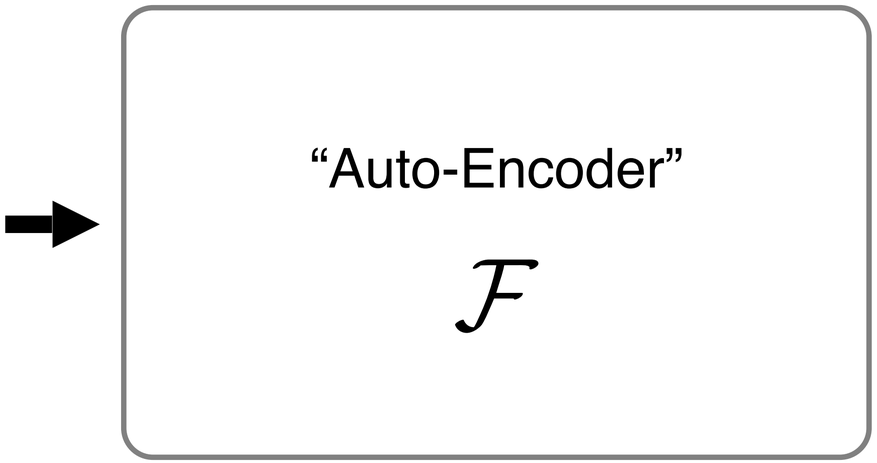
$\hat{\mathbf{x}}$

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

$\hat{\mathbf{x}}$

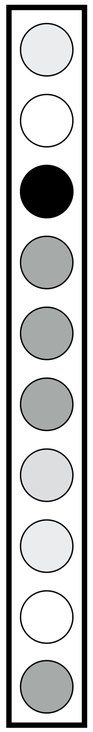
Reconstructed image

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

Auto-Encoding

$\mathbf{x}$

Image



$\hat{\mathbf{x}}$

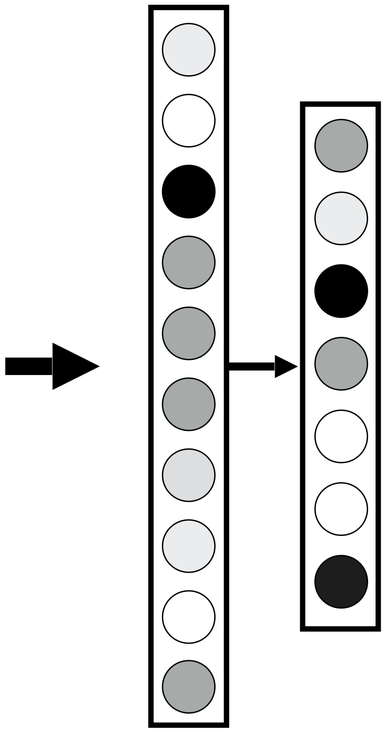
Reconstructed image

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

Auto-Encoding

$\mathbf{x}$

Image



$\hat{\mathbf{x}}$

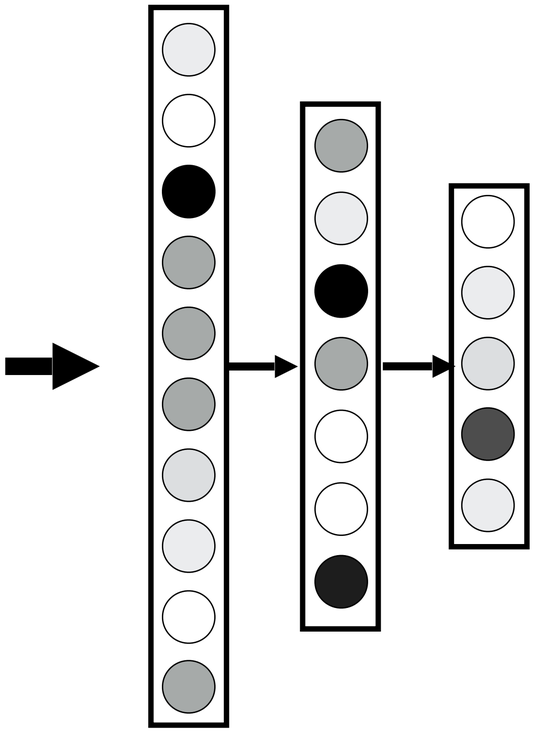
Reconstructed image

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

Auto-Encoding

$\mathbf{x}$

Image



$\hat{\mathbf{x}}$

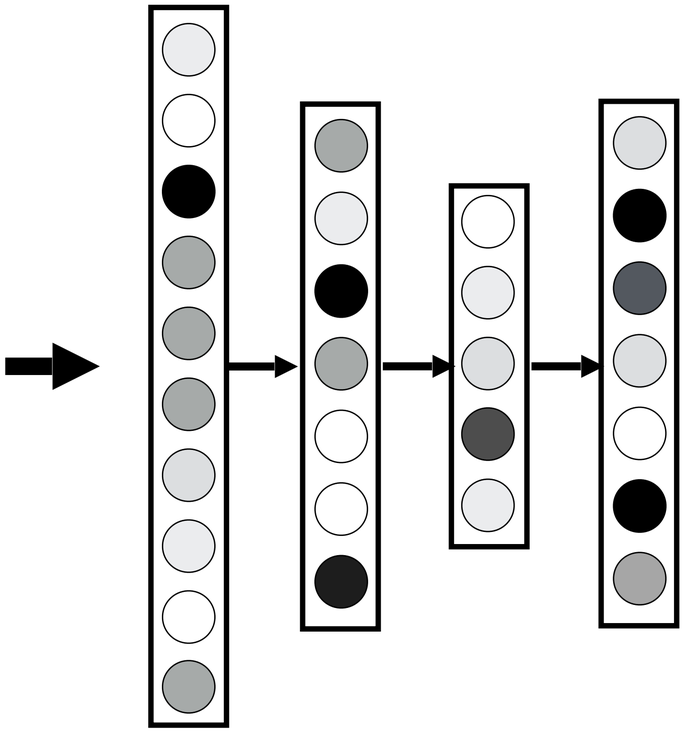
Reconstructed image

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

Auto-Encoding

$\mathbf{x}$

Image

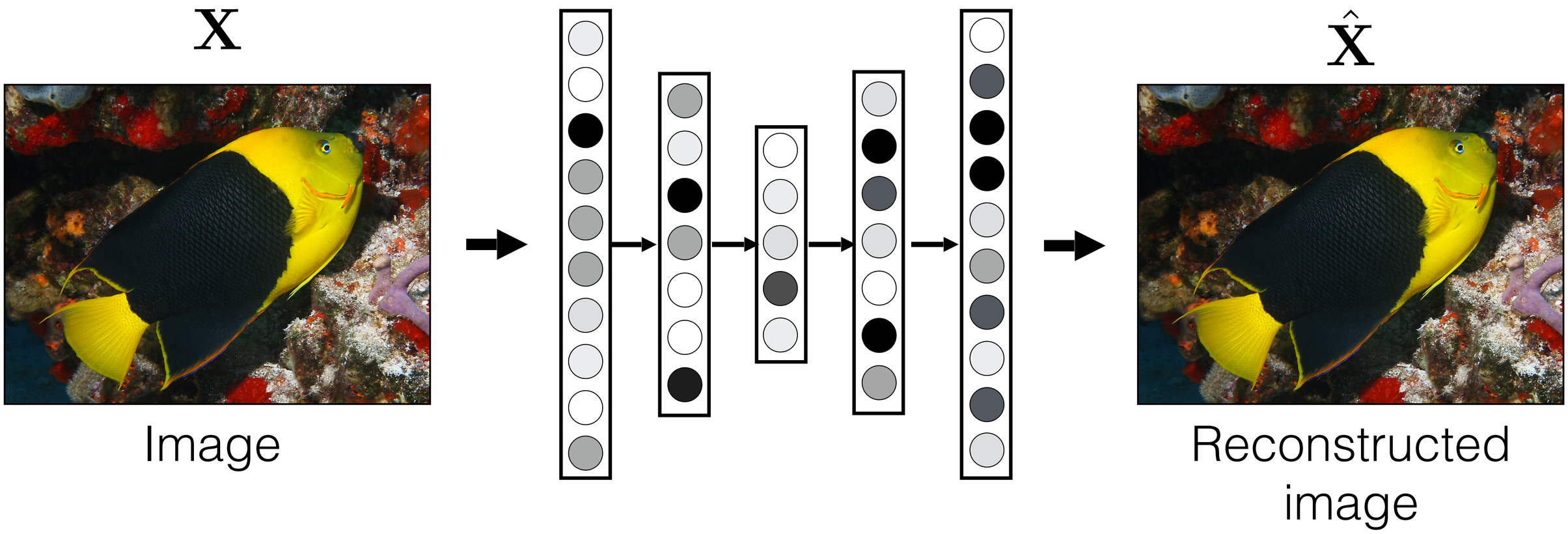


$\hat{\mathbf{x}}$

Reconstructed image

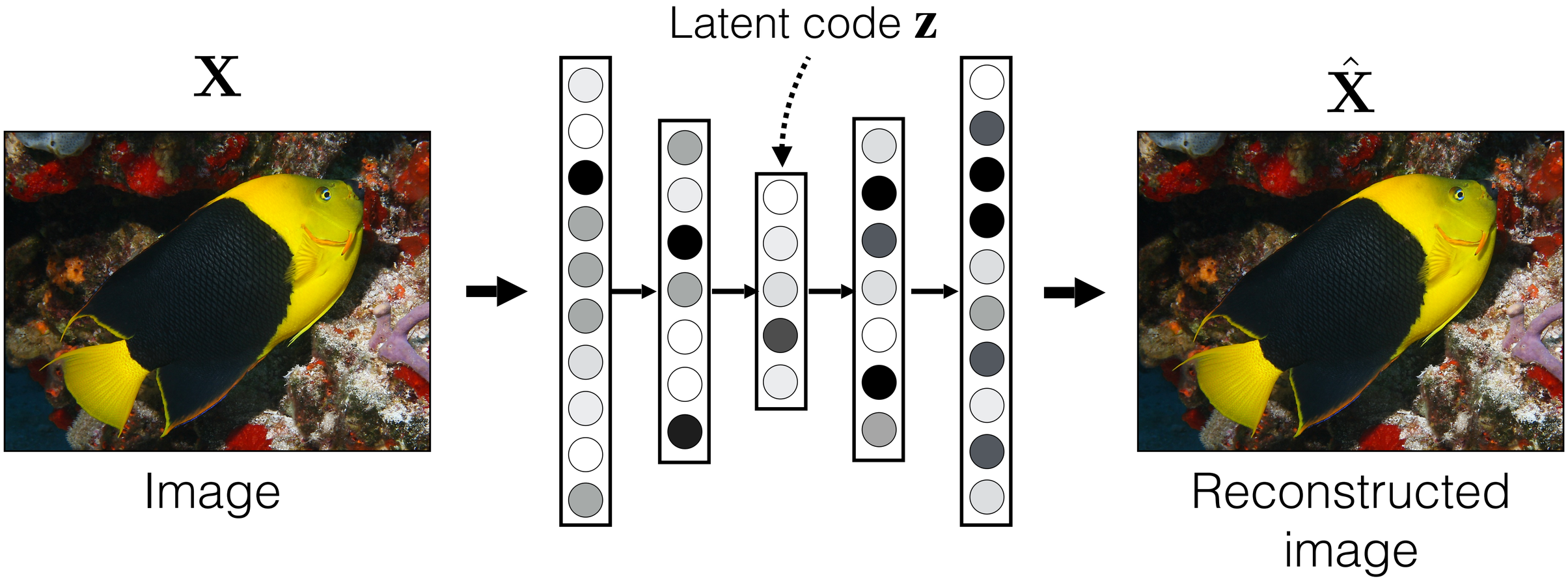
[e.g., Hinton & Salakhutdinov, Science 2006]

Auto-Encoding



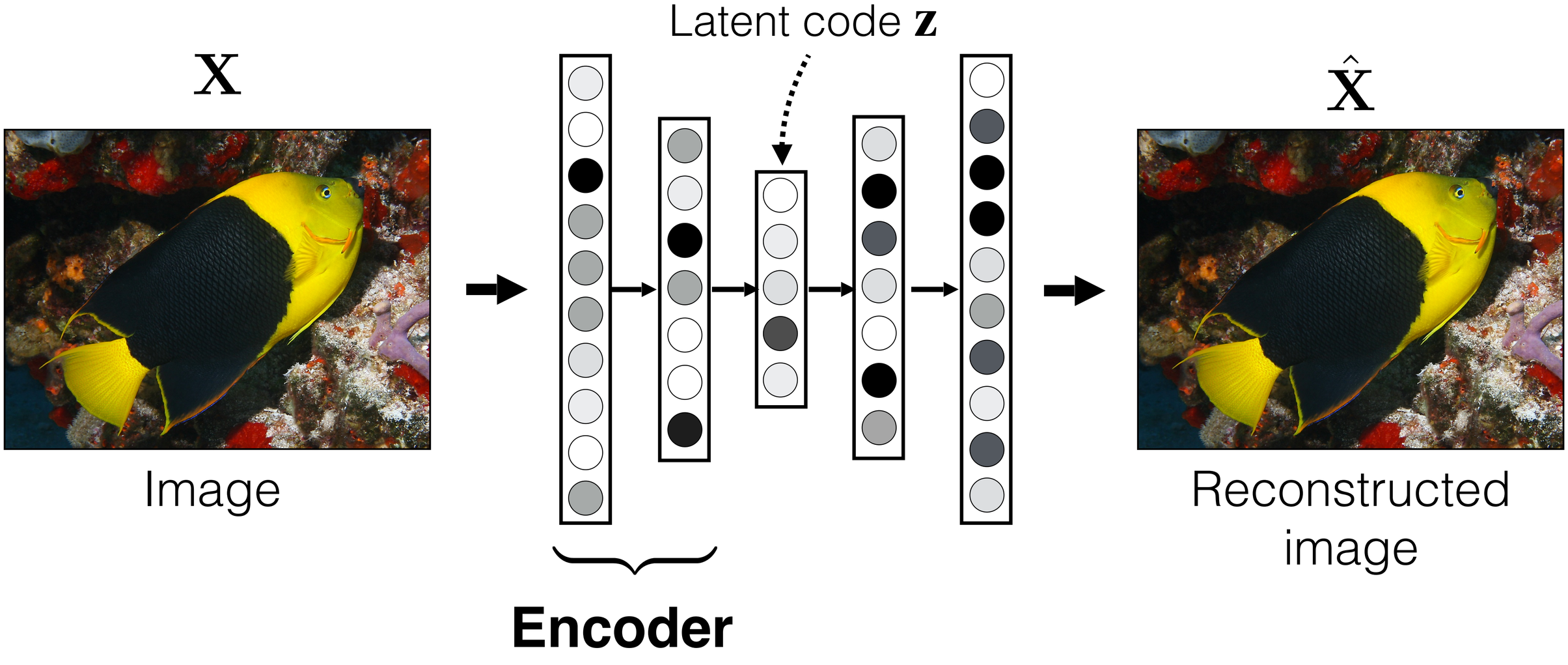
[e.g., Hinton & Salakhutdinov, Science 2006]

Auto-Encoding



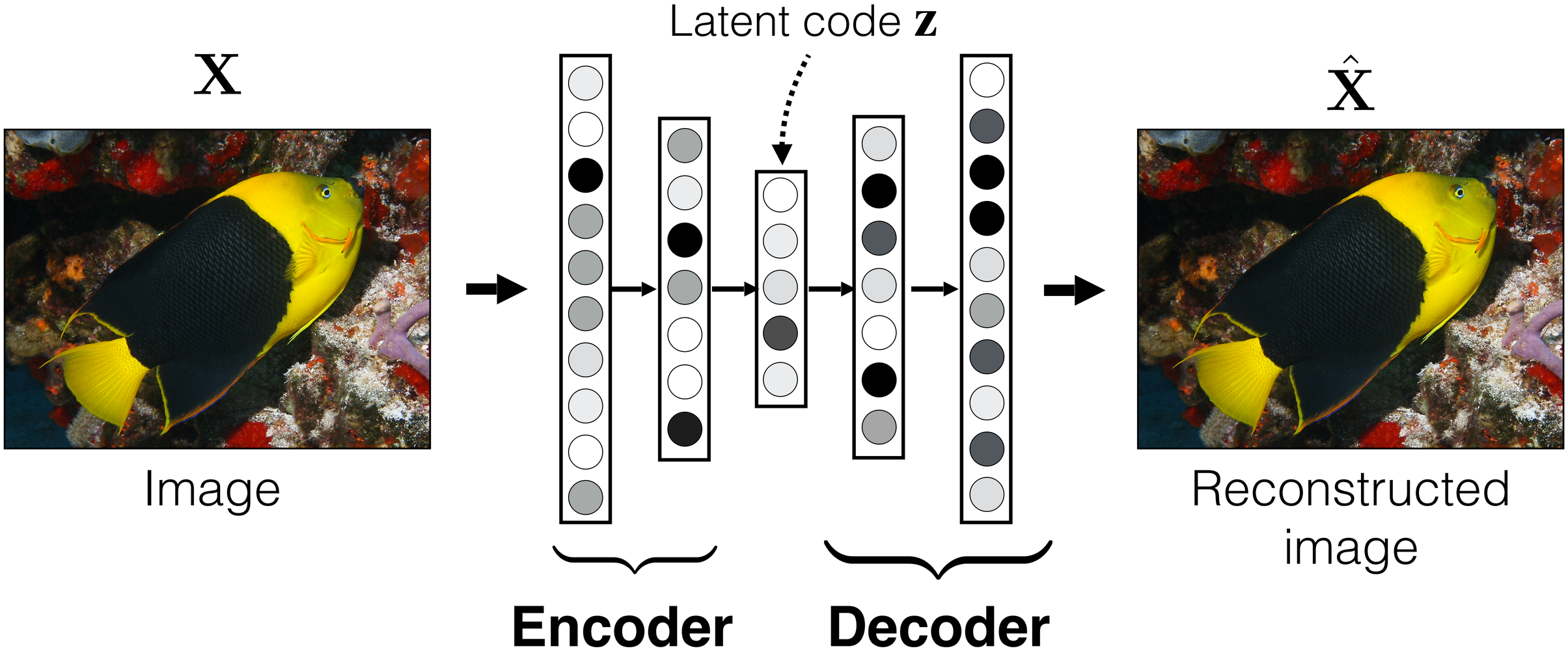
[e.g., Hinton & Salakhutdinov, Science 2006]

Auto-Encoding

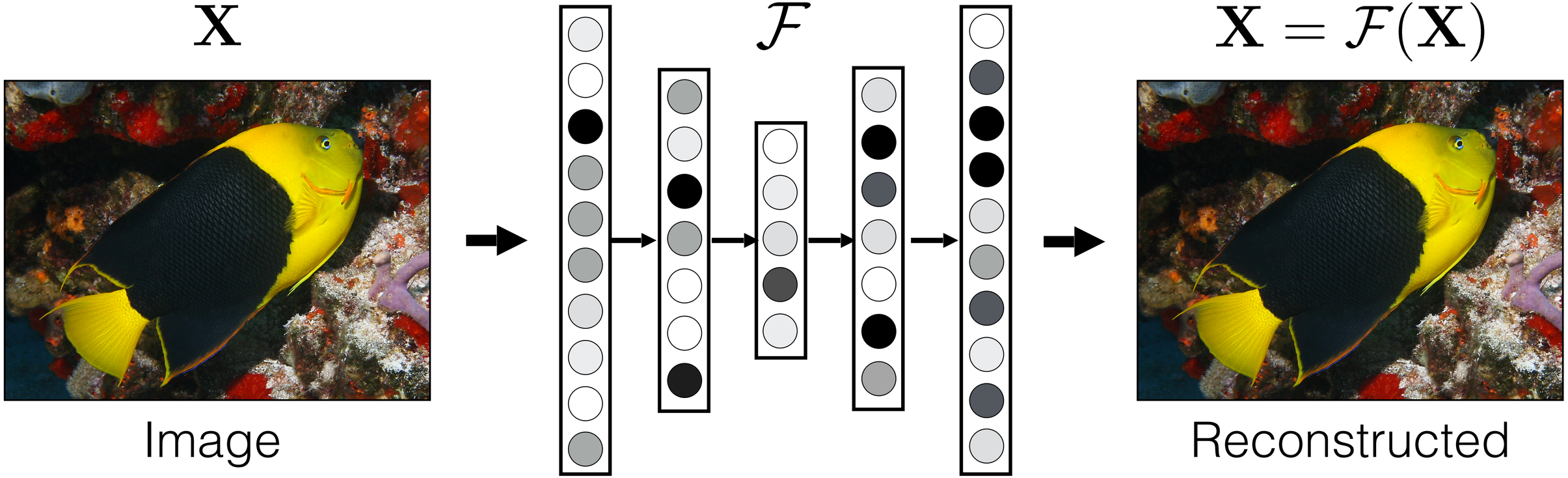


[e.g., Hinton & Salakhutdinov, Science 2006]

Auto-Encoding



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

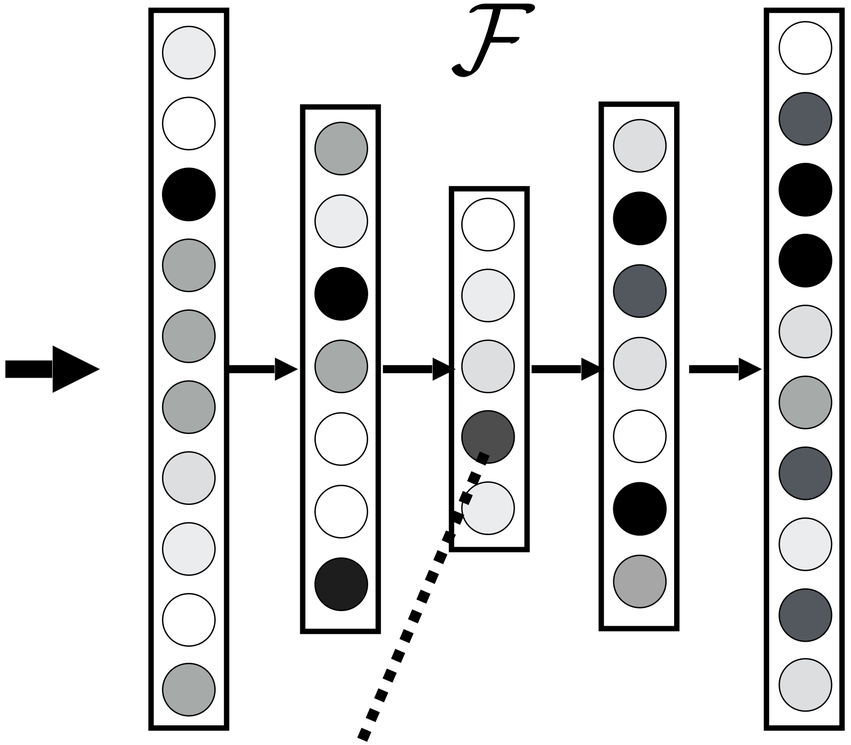


Learning Signal: **Forced Compression**

$\mathbf{X}$

Image

$\mathcal{F}$



“Fish”

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

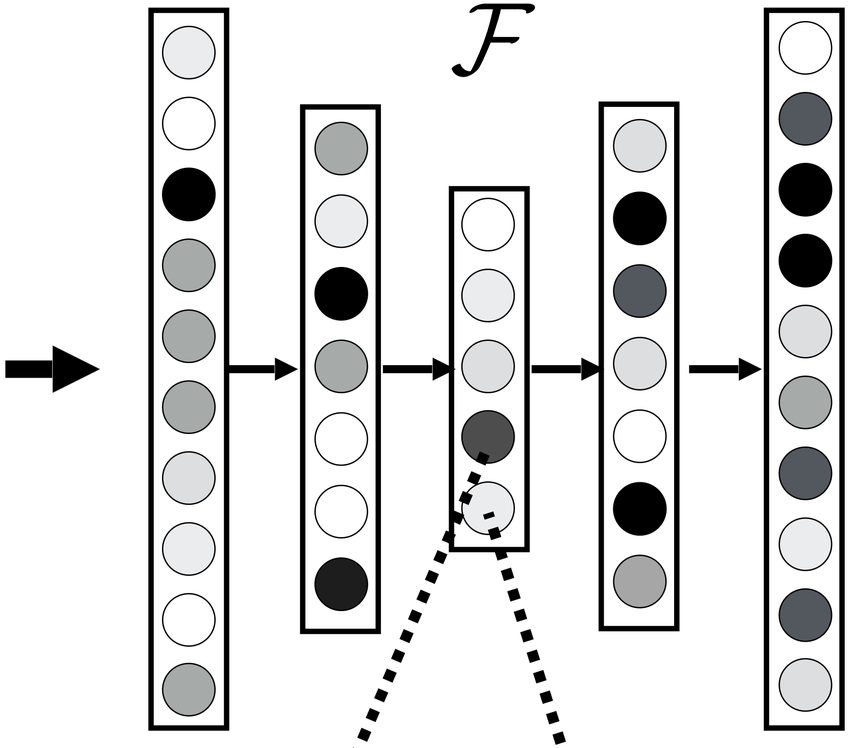
Reconstructed image

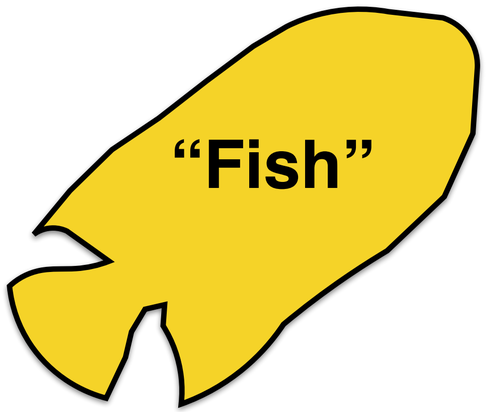
Learning Signal: **Forced Compression**

$\mathbf{X}$

Image

$\mathcal{F}$



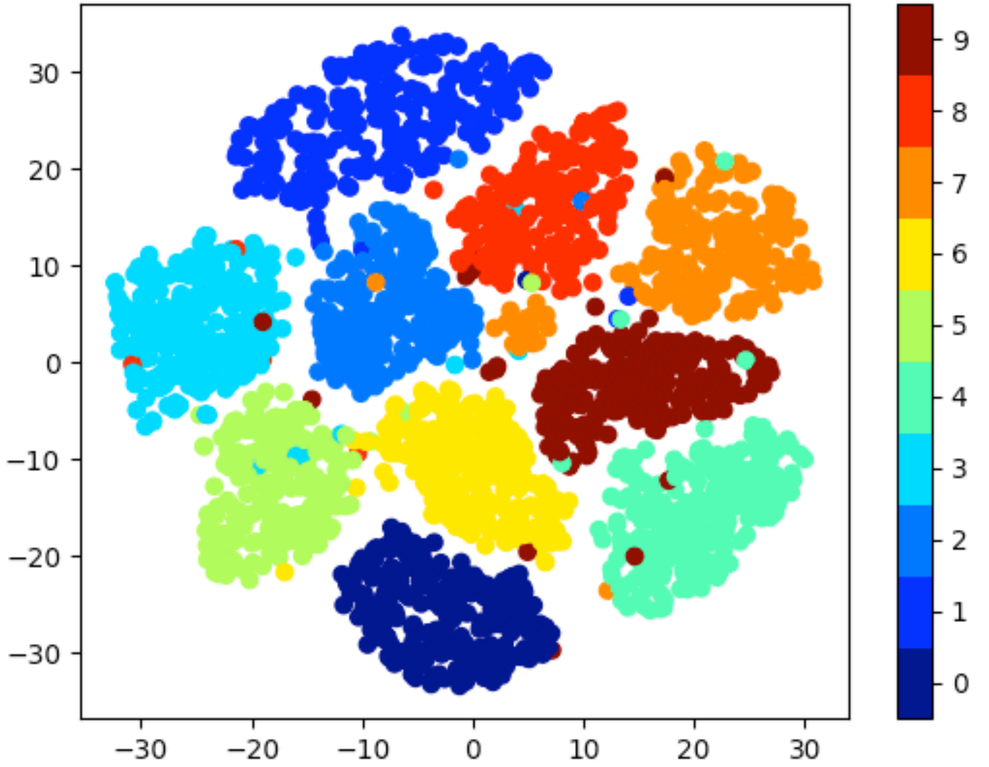


$\hat{\mathbf{X}} = \mathcal{F}(\mathbf{X})$

Reconstructed image

Learning Signal: **Forced Compression**

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



Self-supervised Scene Representation Learning

Latent 3D Scenes

{  ,  , . . . }

Self-supervised Scene Representation Learning

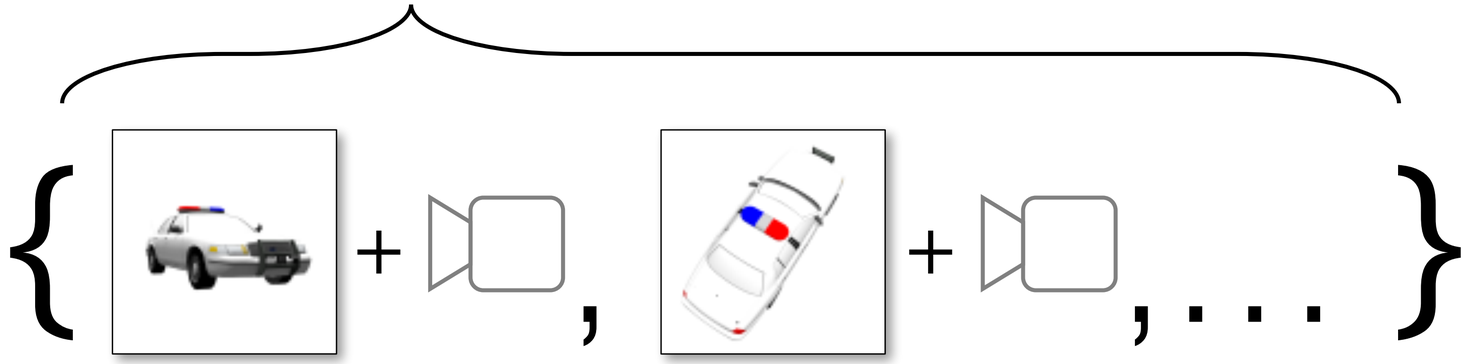
Latent 3D Scenes

{  ,  , . . . }

Self-supervised Scene Representation Learning

Latent 3D Scenes

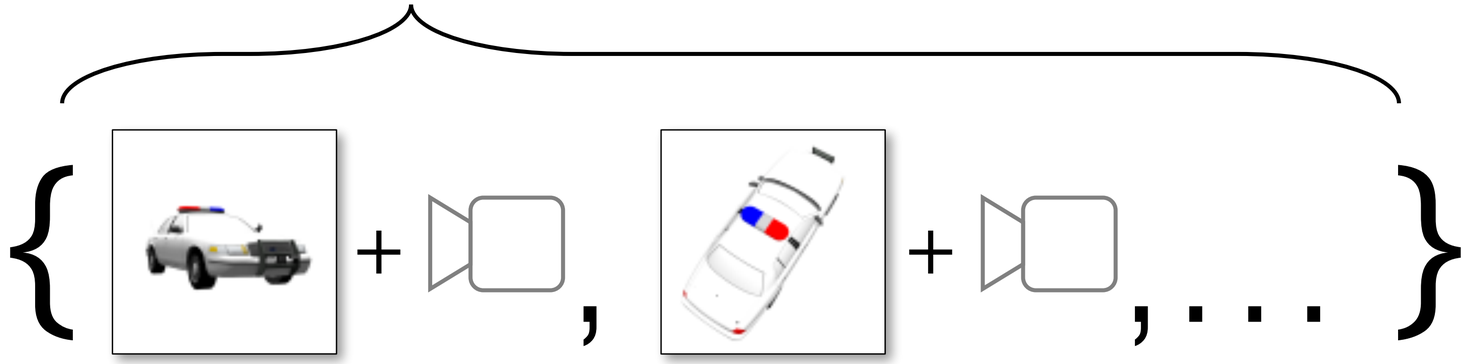
Observations
Image + Pose & Intrinsics



Self-supervised Scene Representation Learning

Latent 3D Scenes

Observations
Image + Pose & Intrinsics



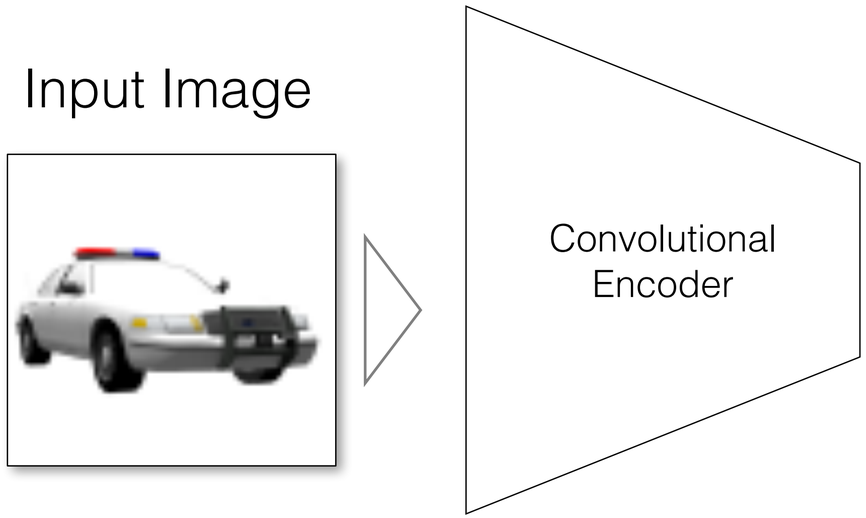
What can we learn about latent 3D scenes from observations?

Vision: Learn rich representations just by watching video!

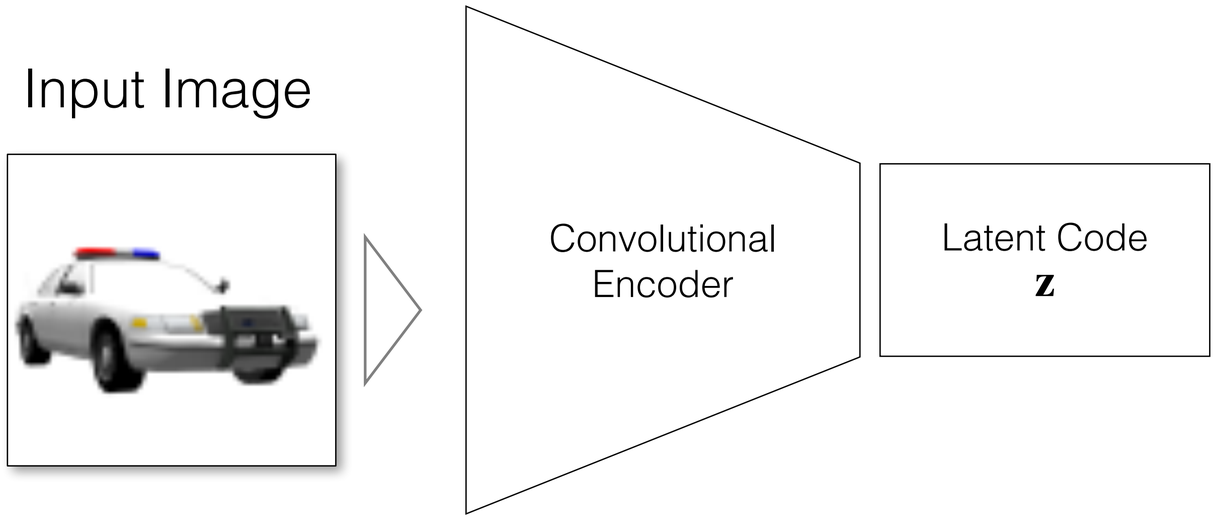
2D baseline: Auto-Encoder-Like model

Input Image

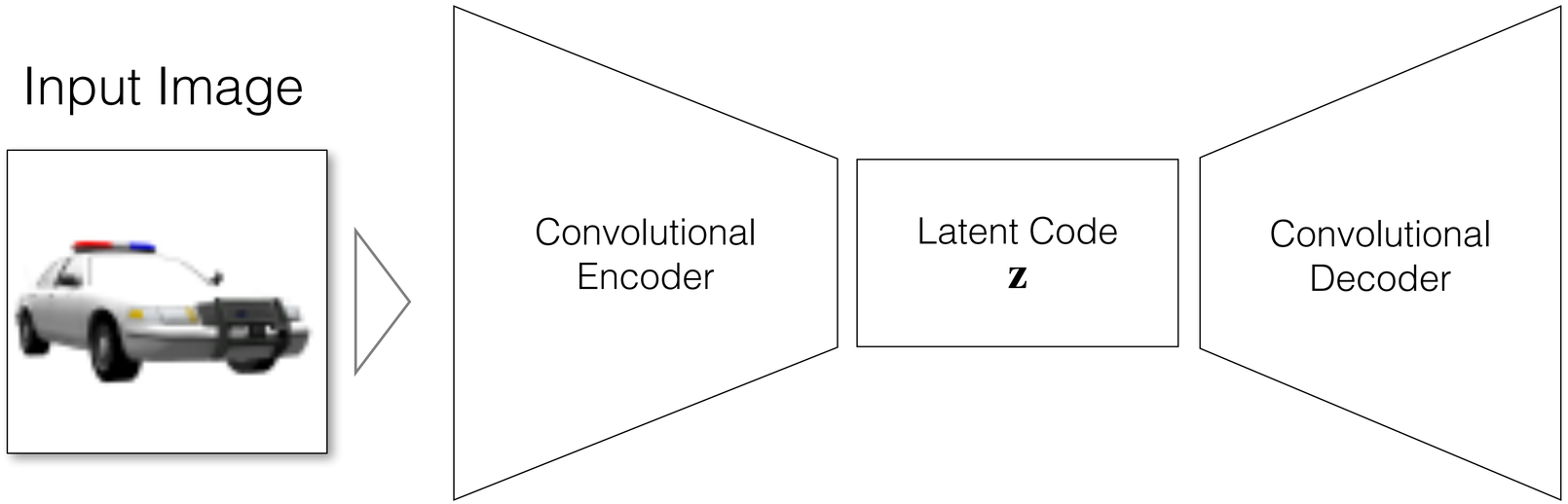
2D baseline: Auto-Encoder-Like model



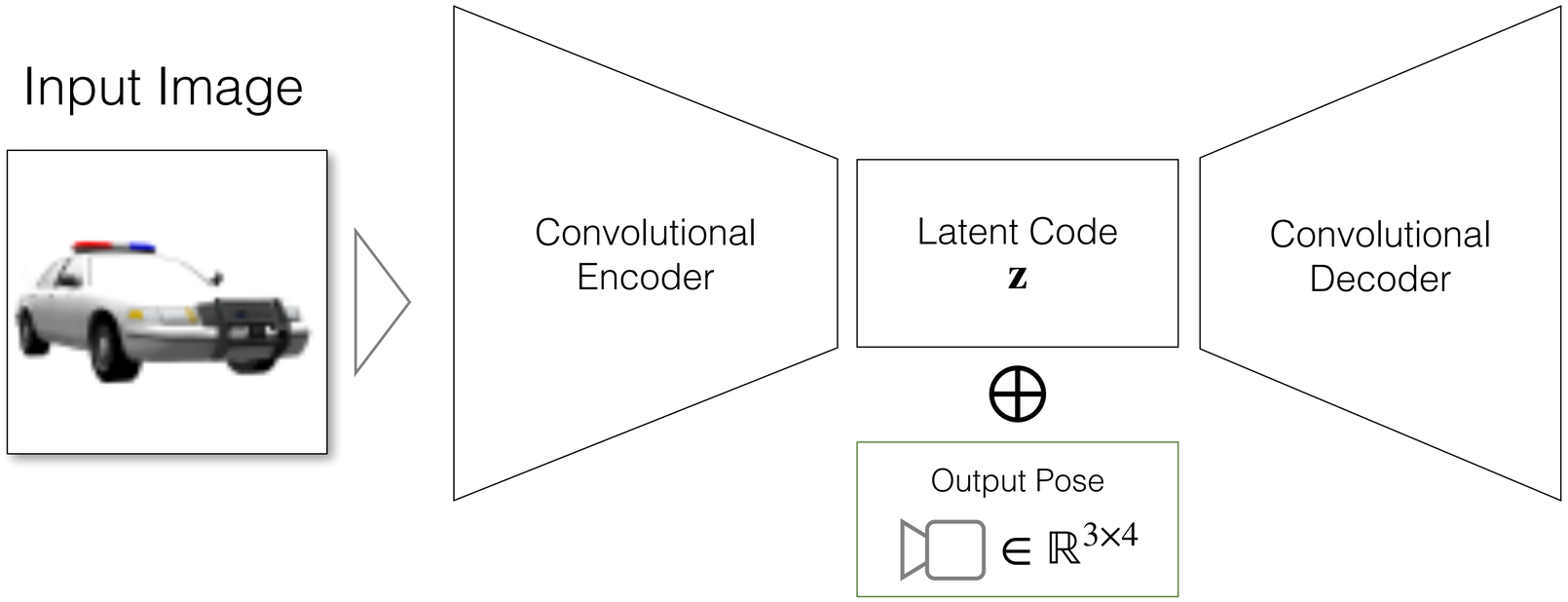
2D baseline: Auto-Encoder-Like model



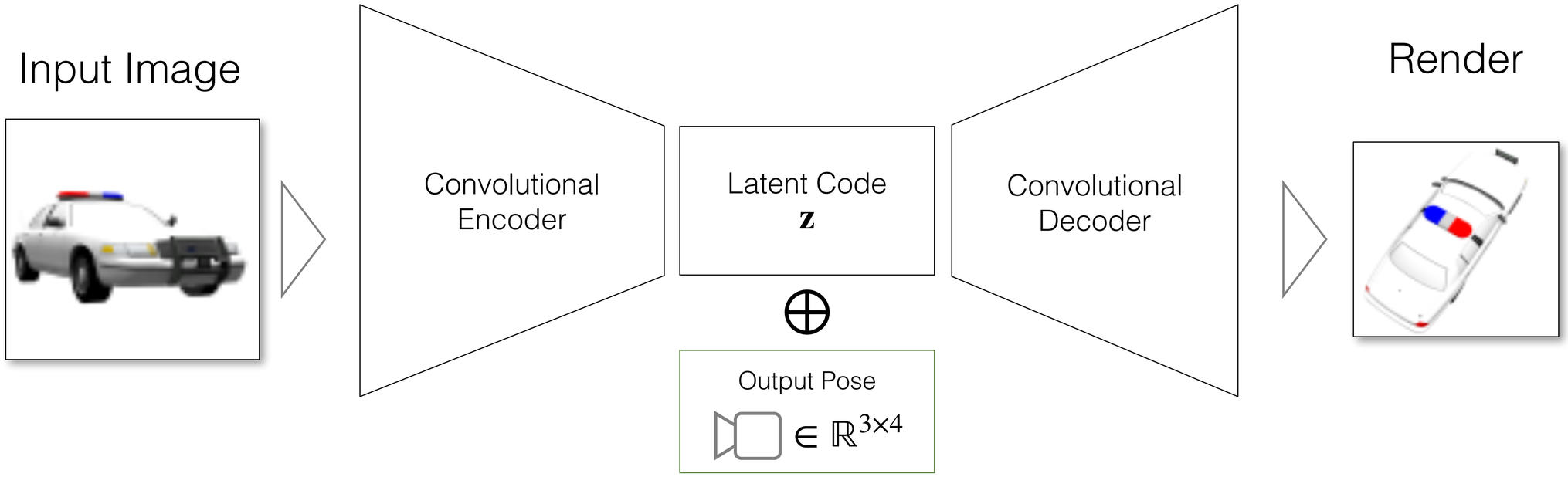
2D baseline: Auto-Encoder-Like model



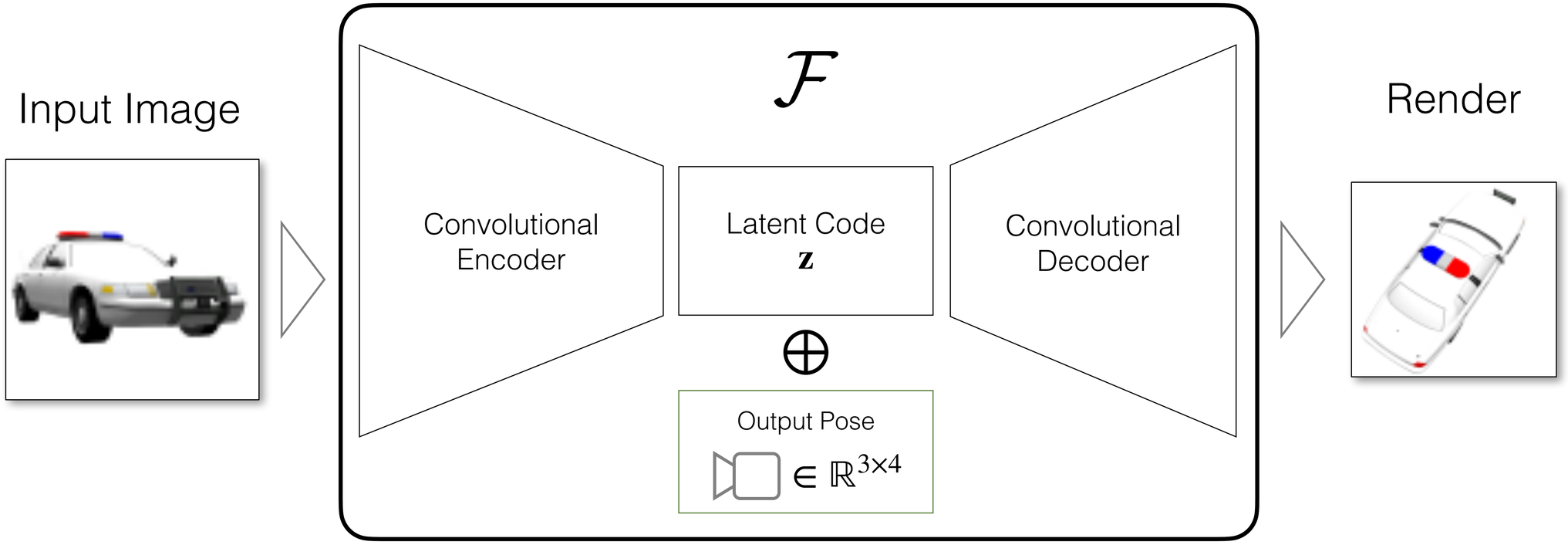
2D baseline: Auto-Encoder-Like model



2D baseline: Auto-Encoder-Like model

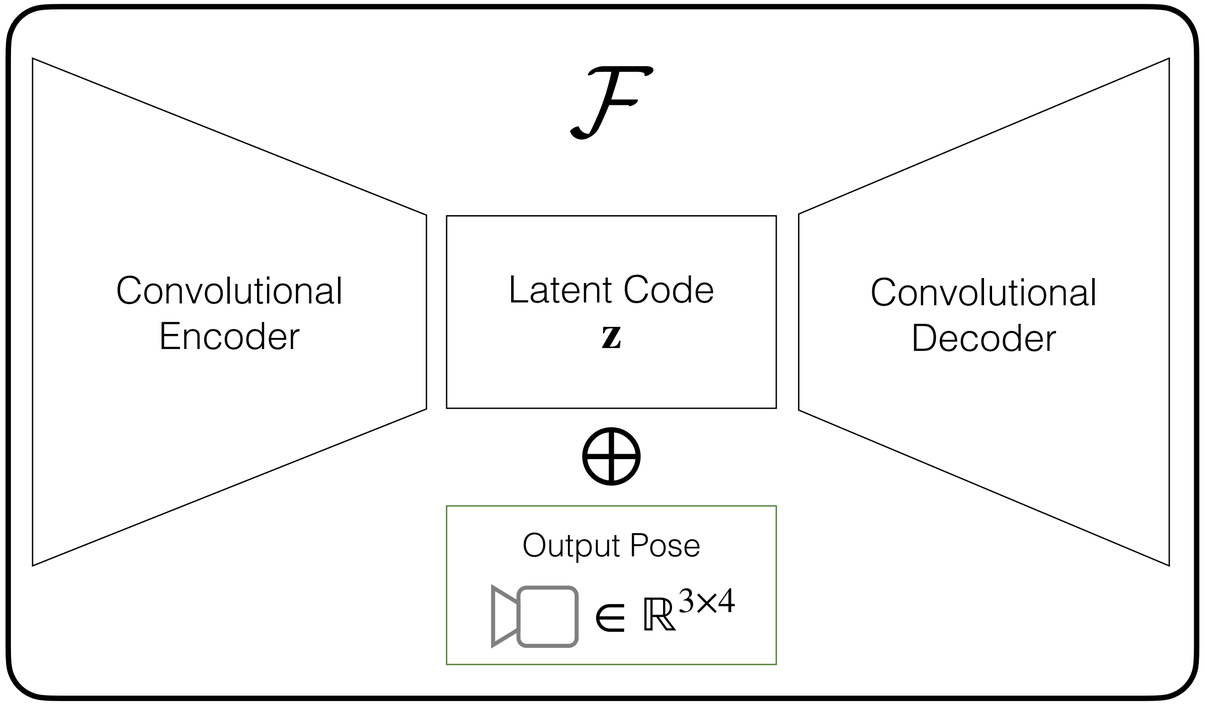


2D baseline: Auto-Encoder-Like model

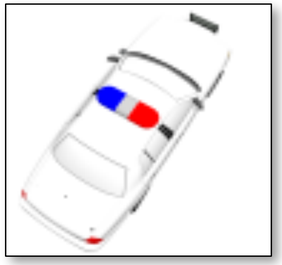


2D baseline: Auto-Encoder-Like model

Input Image

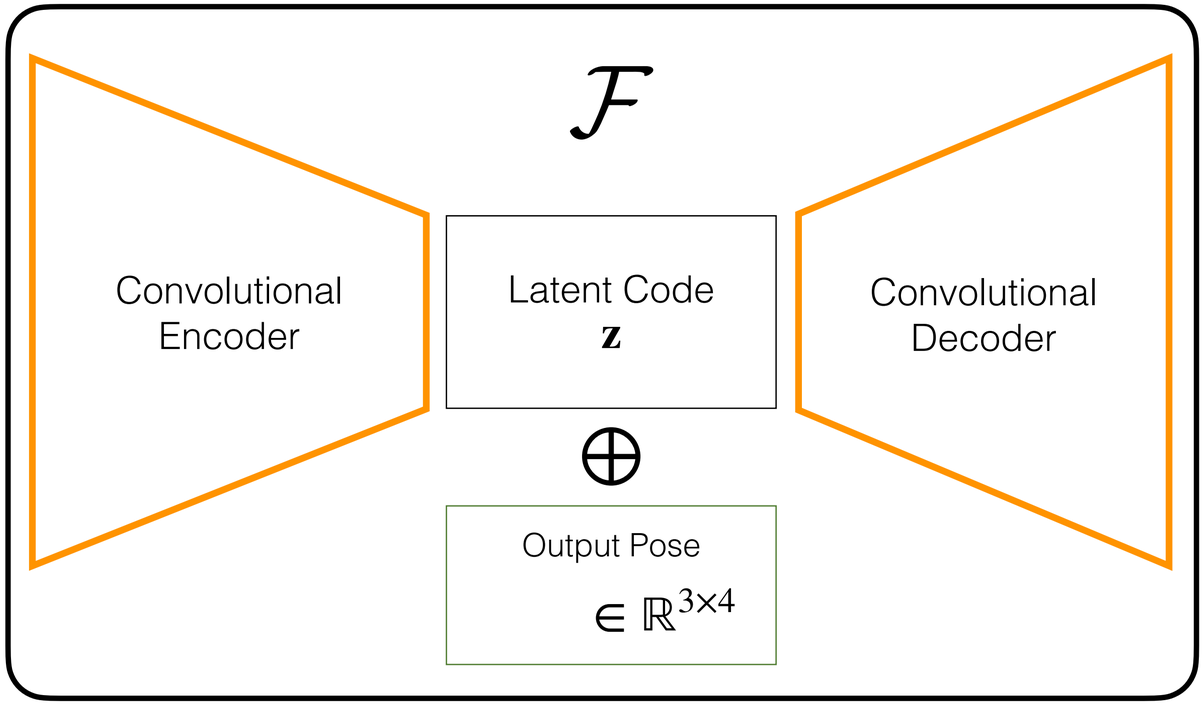


Render



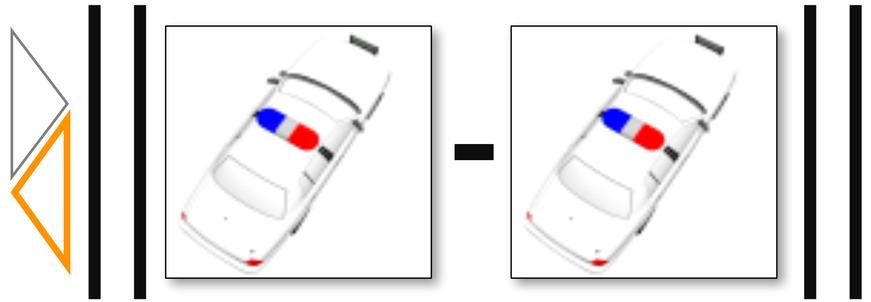
2D baseline: Auto-Encoder-Like model

Input Image

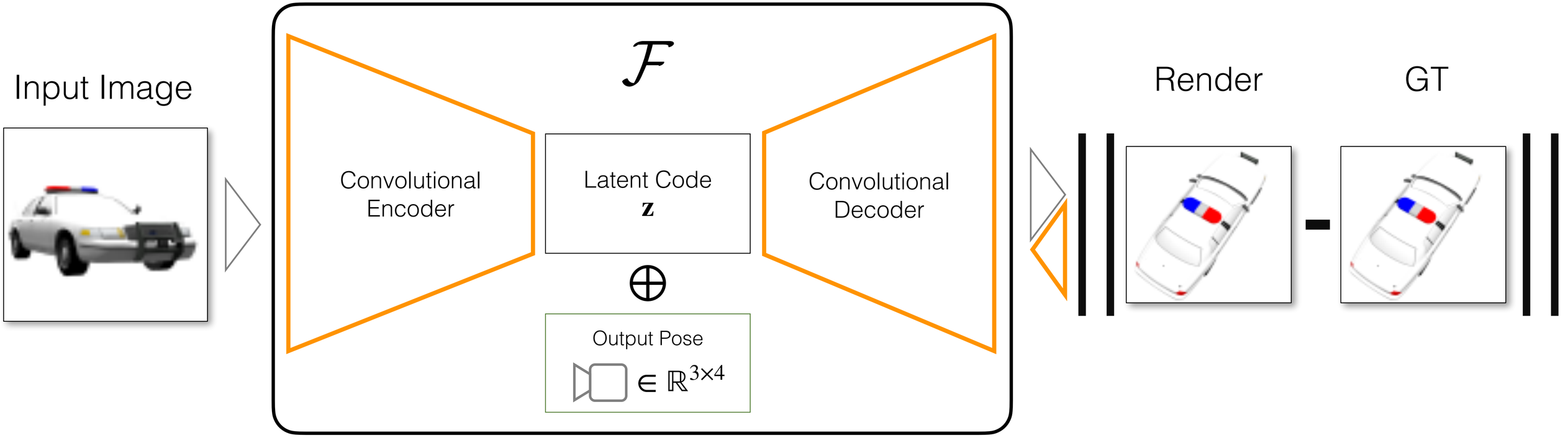


Render

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.

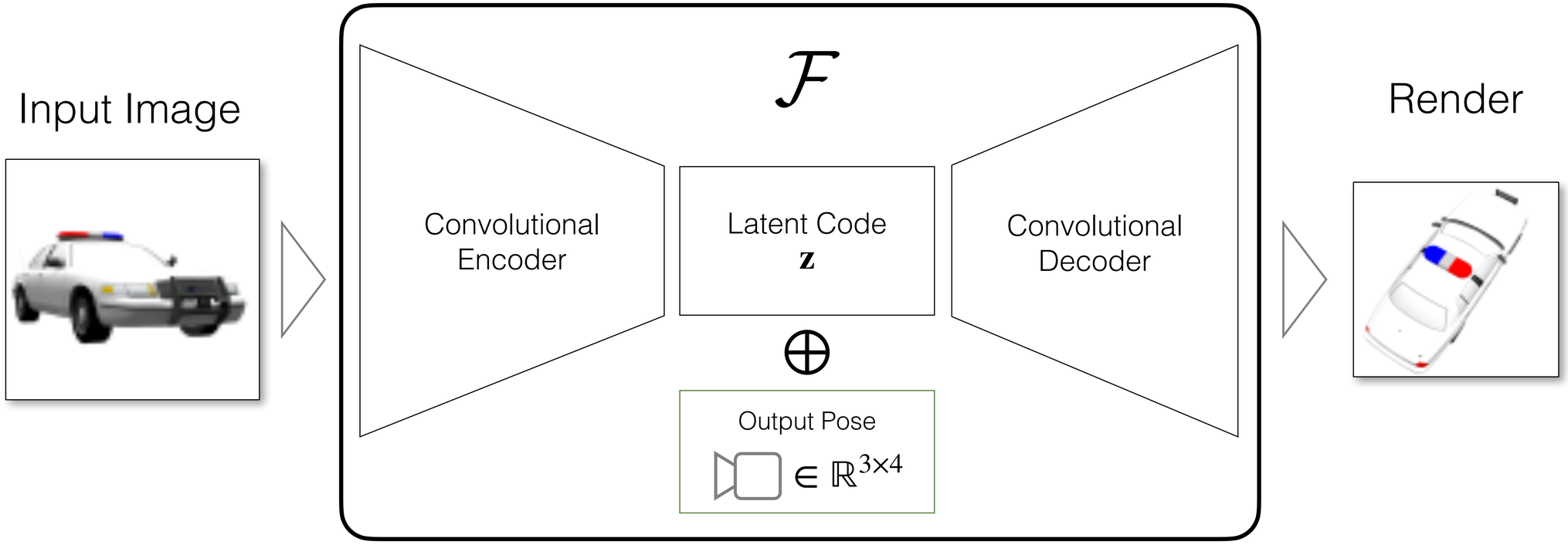
Trained on ~2500 shapenet cars with 50 observations each.

Doesn’t capture 3D properties of scenes.

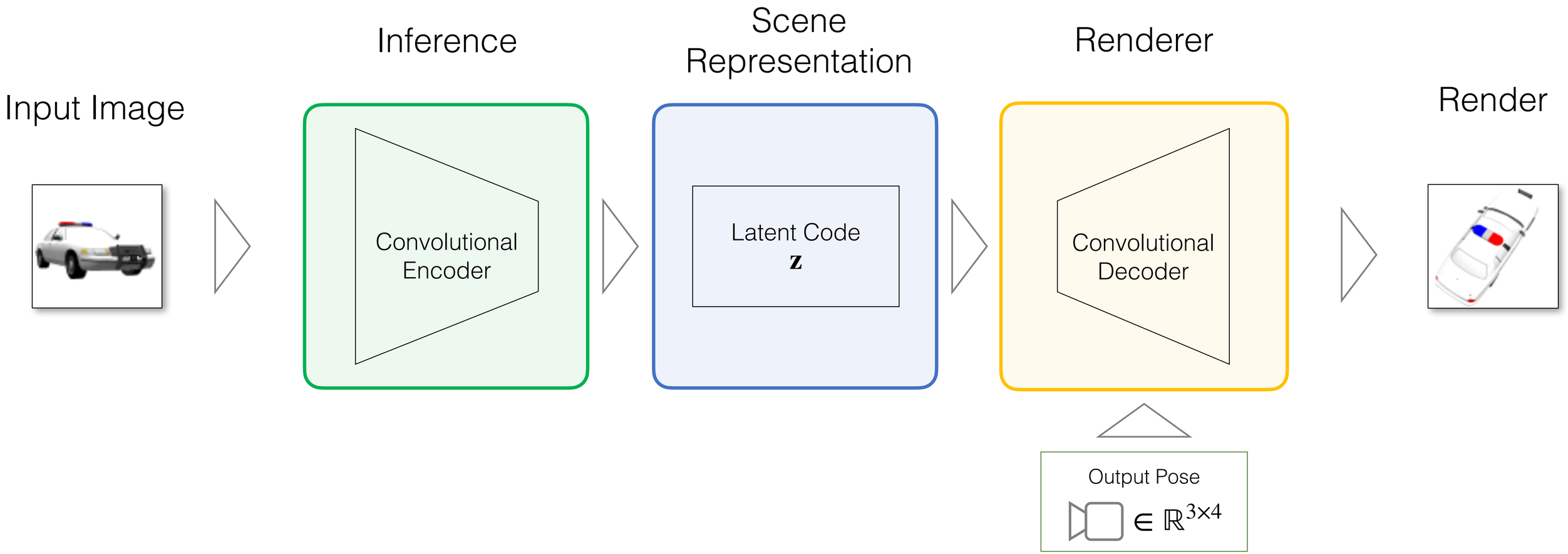
Trained on ~2500 shapenet cars with 50 observations each.

First hypothesis: Doesn’t have 3D structure, therefore, just finds some function that explains the training set that ends up being different from the “true” function.

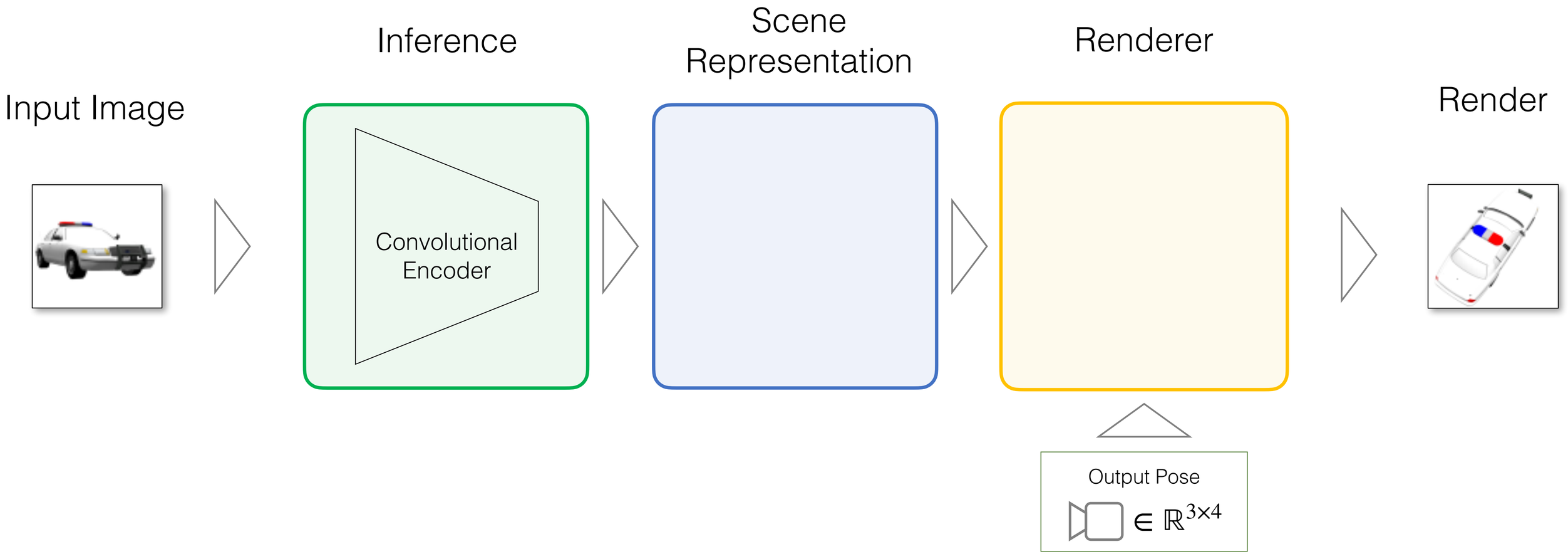
2D baseline: Auto-Encoder-Like model



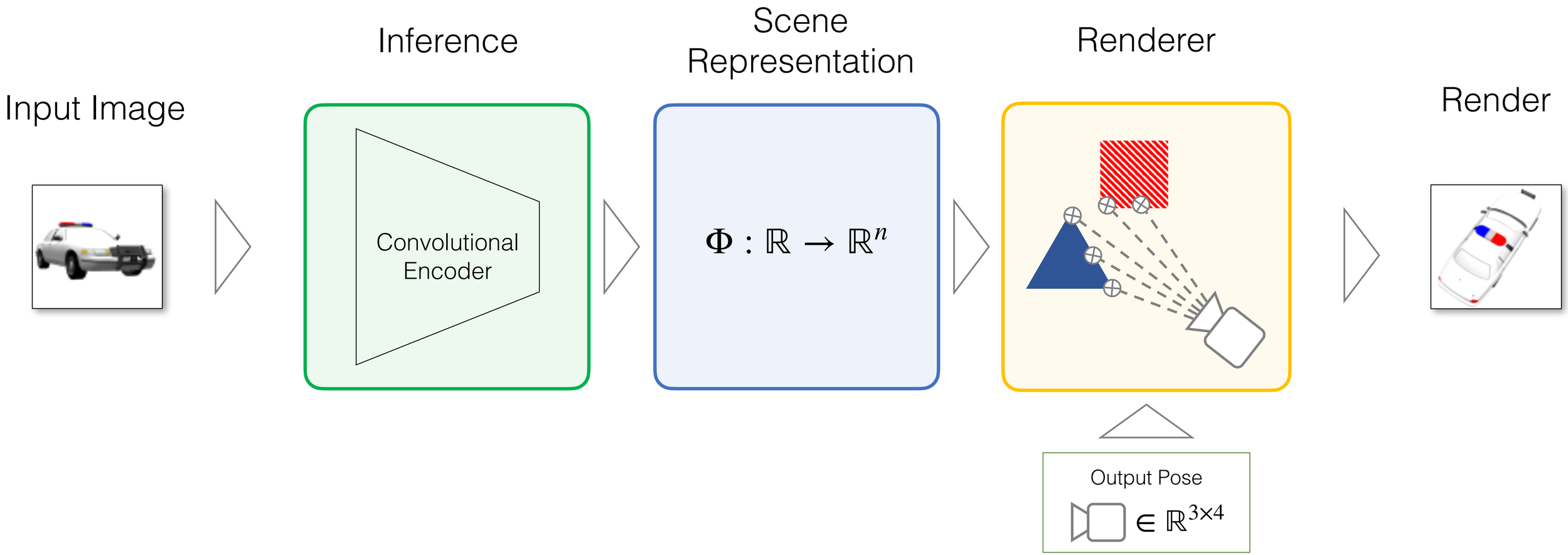
2D baseline: Auto-Encoder-Like model



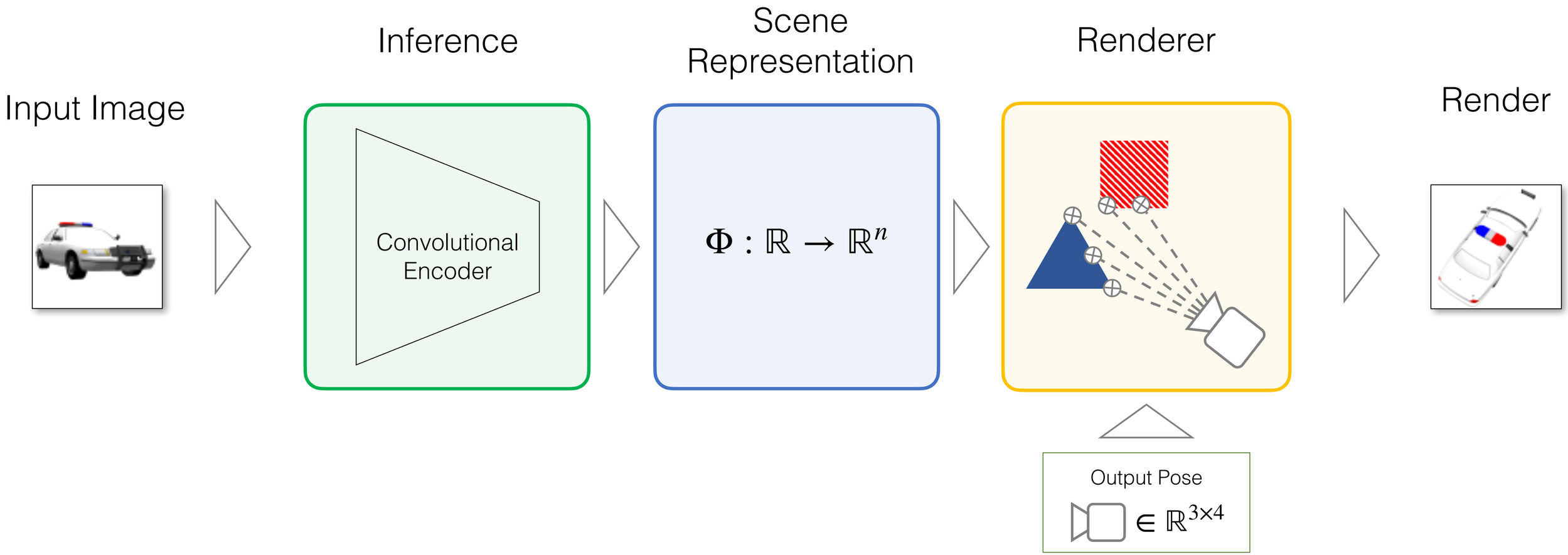
2D baseline: Auto-Encoder-Like model



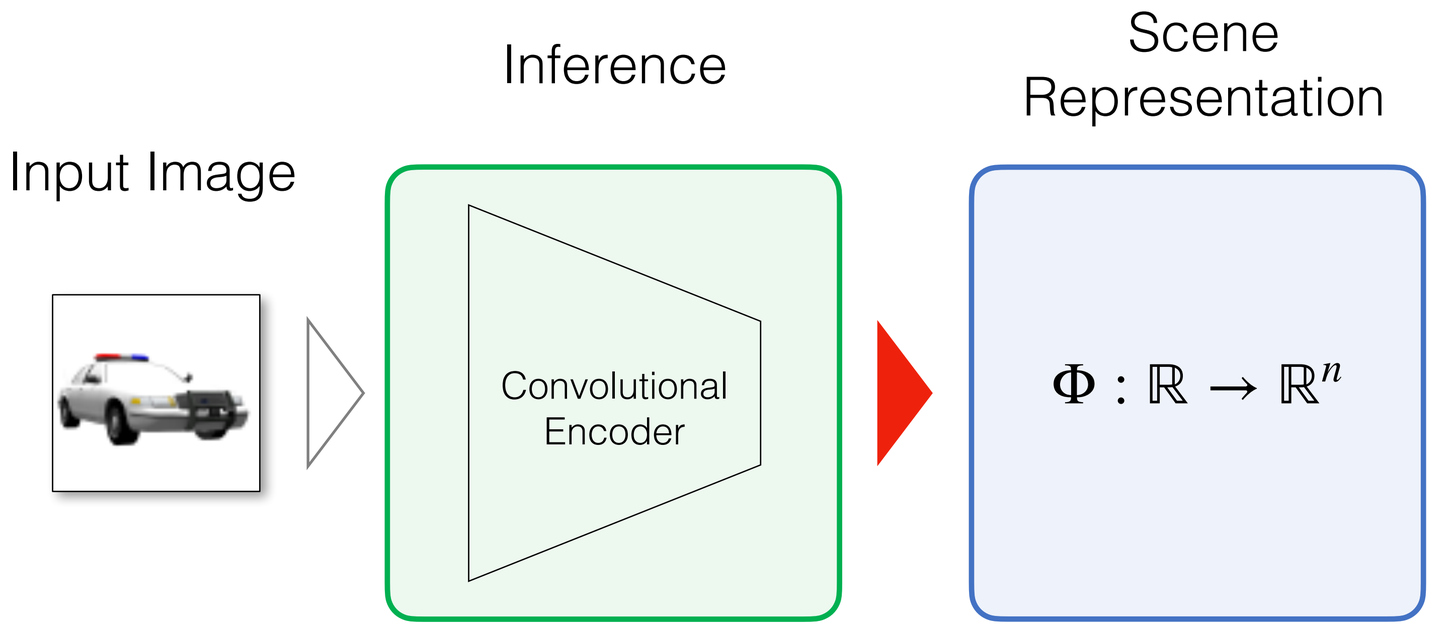
3D-Structured Decoder



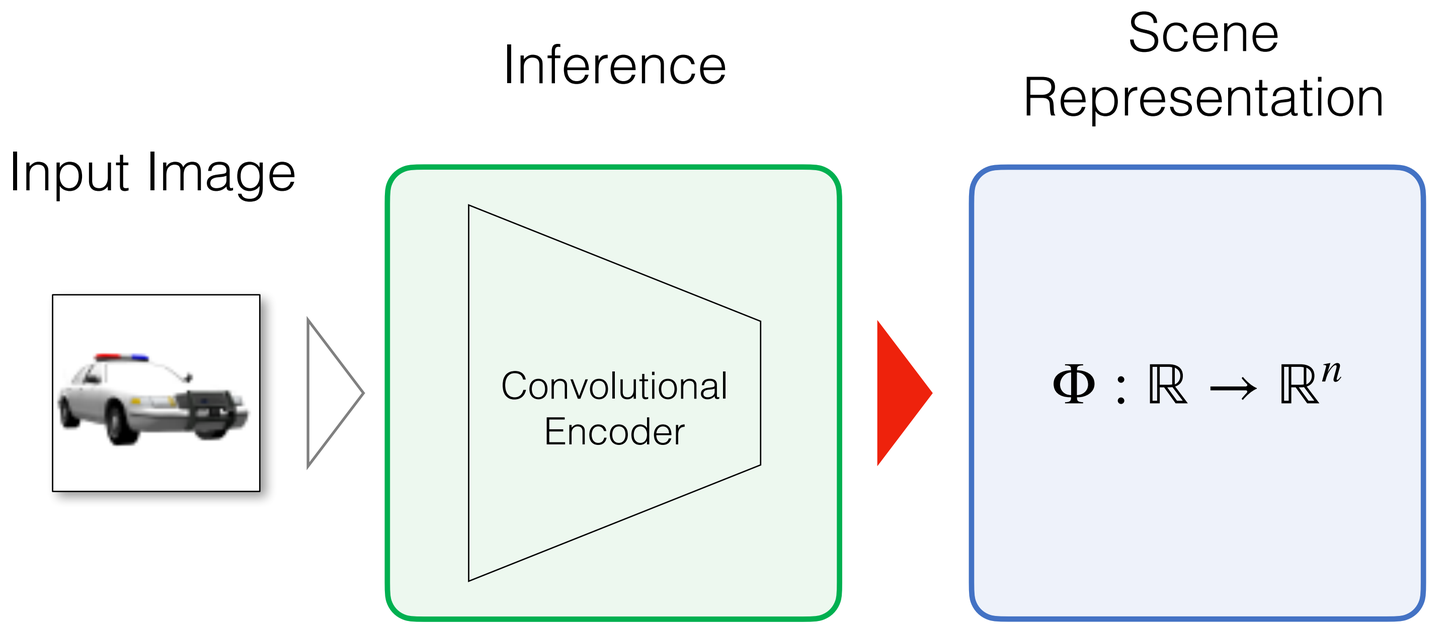
3D-Structured Decoder



3D-Structured Decoder

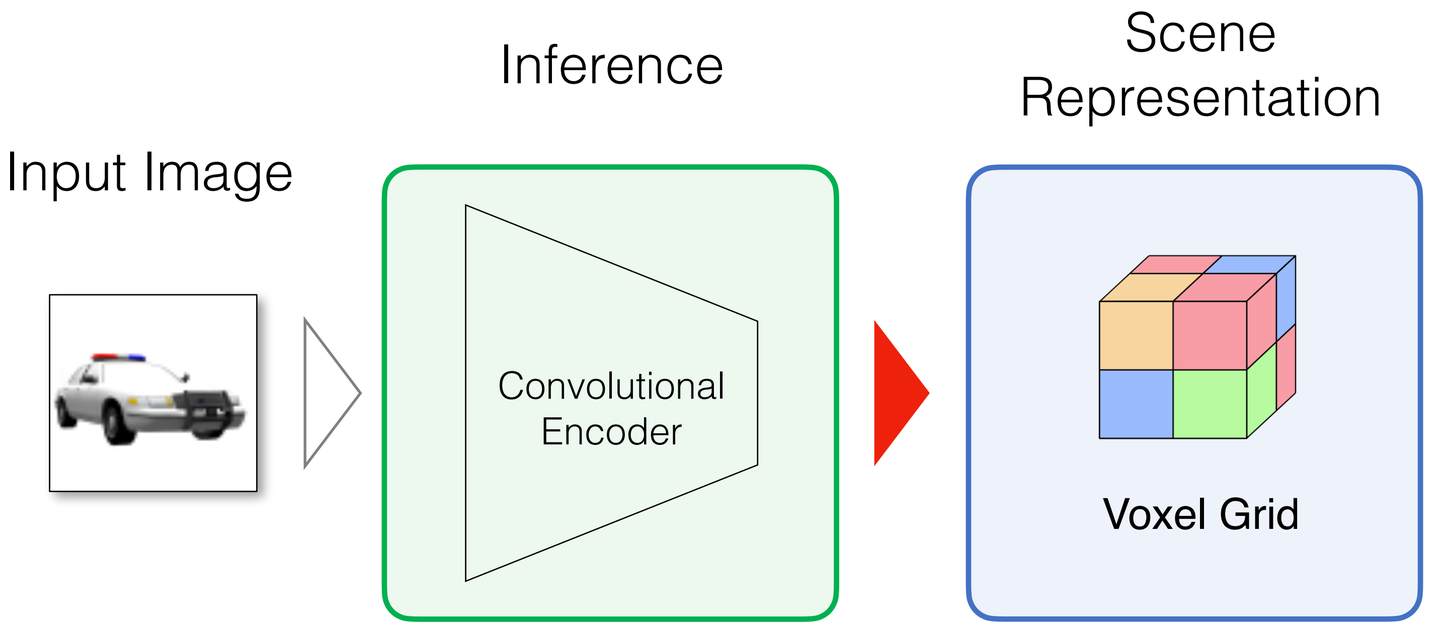


3D-Structured Decoder

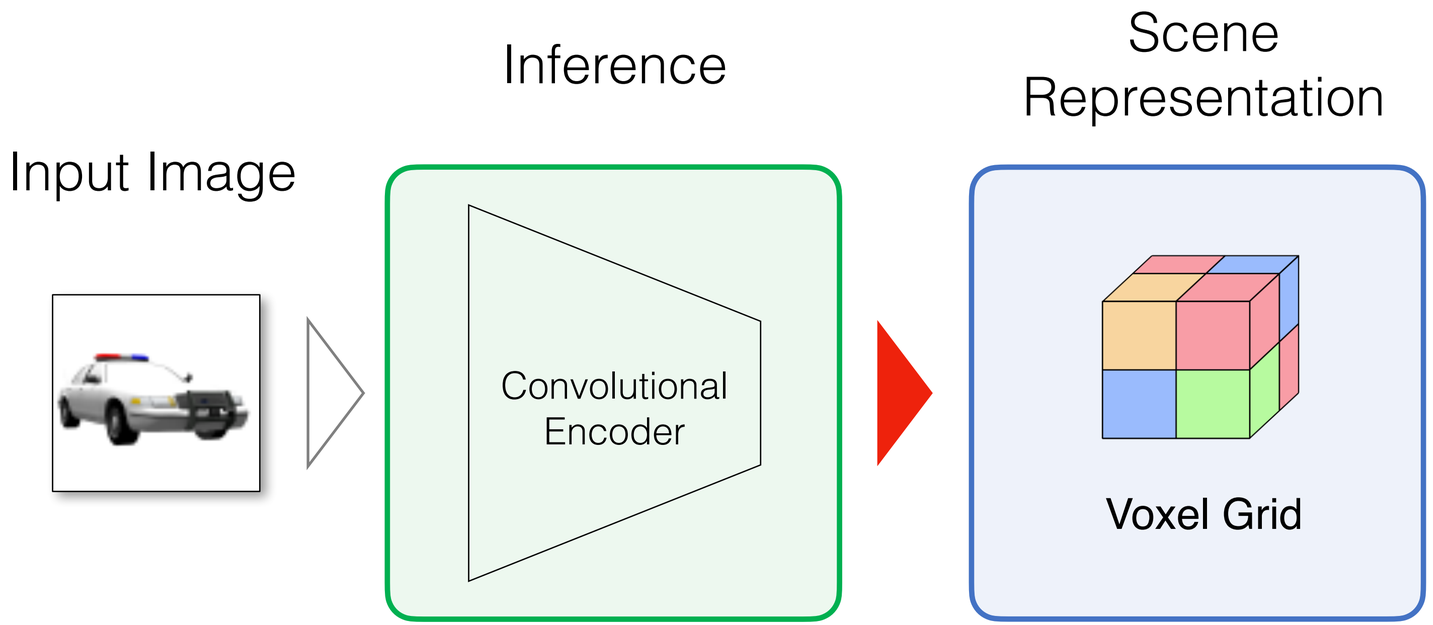


How can we have the convolution encoder “output” the Scene Representation?

3D-Structured Decoder

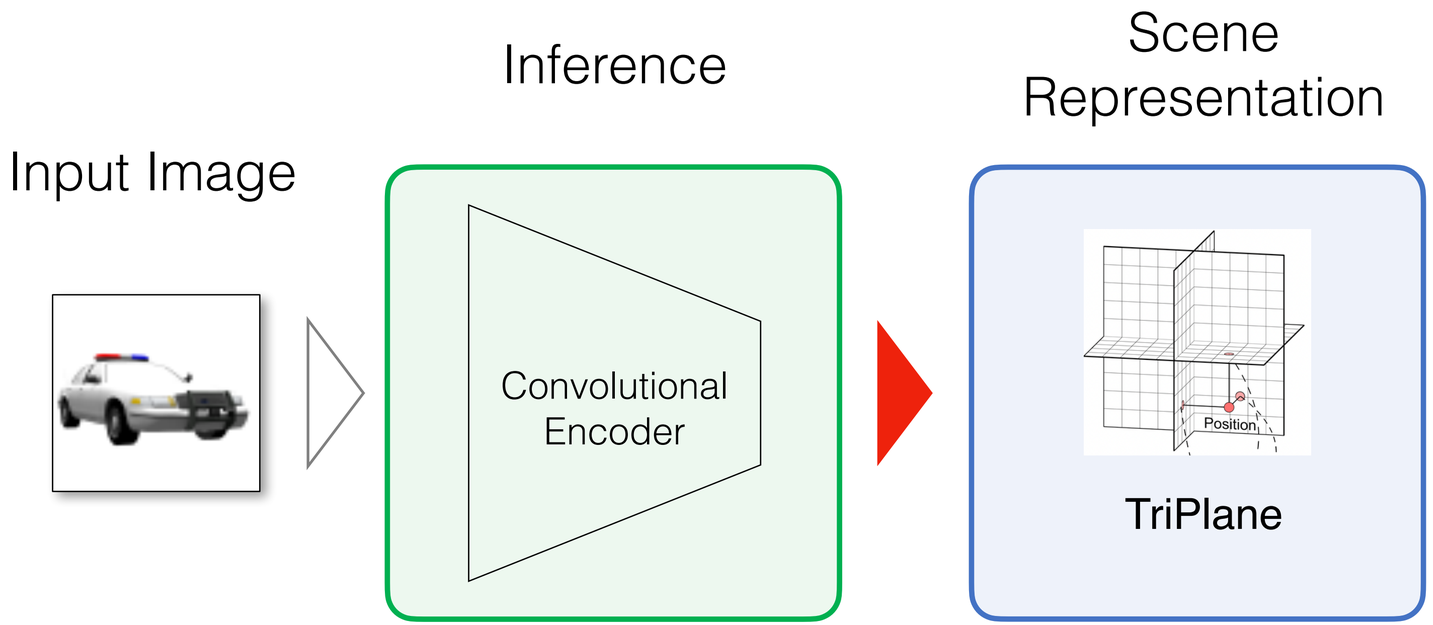


3D-Structured Decoder

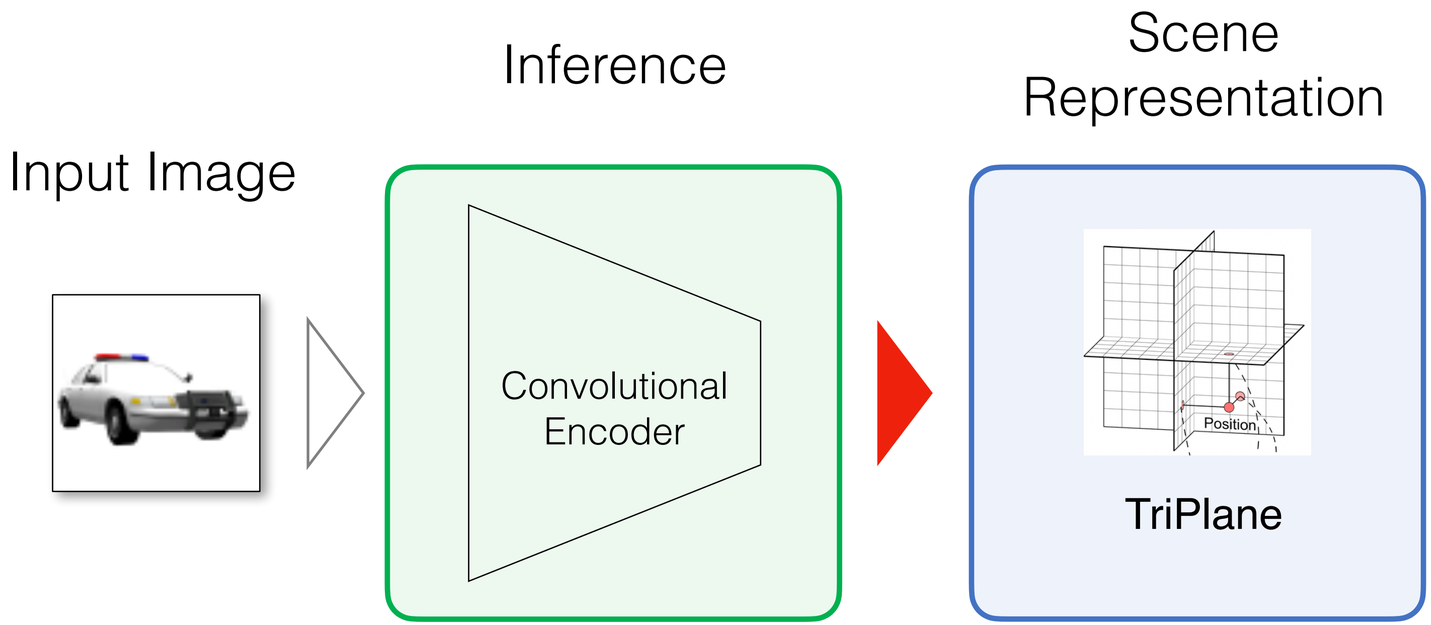


Just predict *all the parameters of that scene representation*.

3D-Structured Decoder

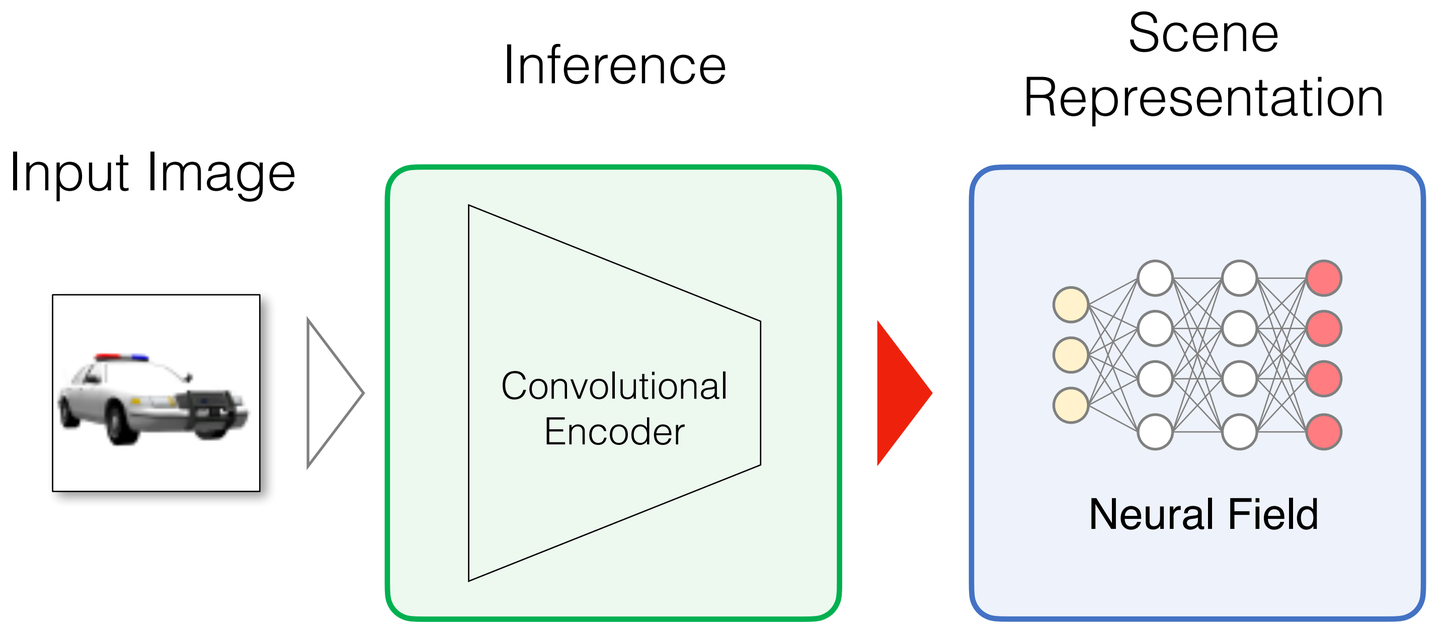


3D-Structured Decoder

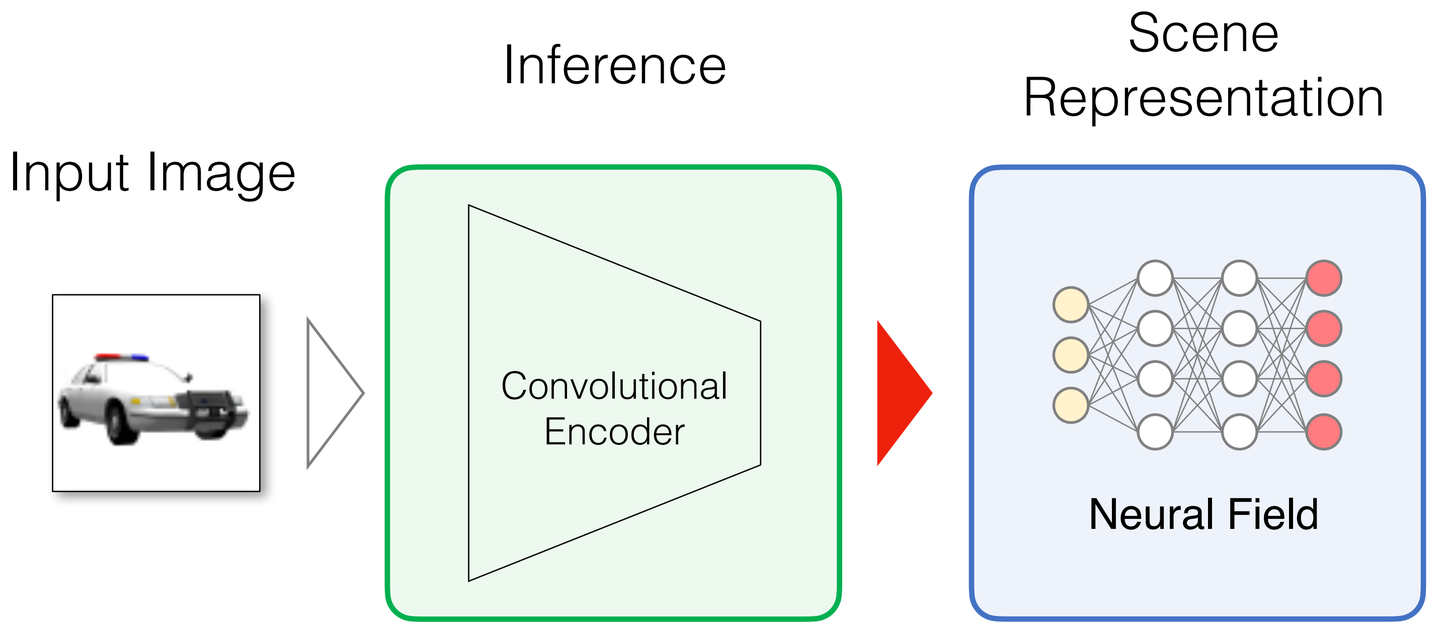


Just predict *all the parameters of that scene representation*.

3D-Structured Decoder

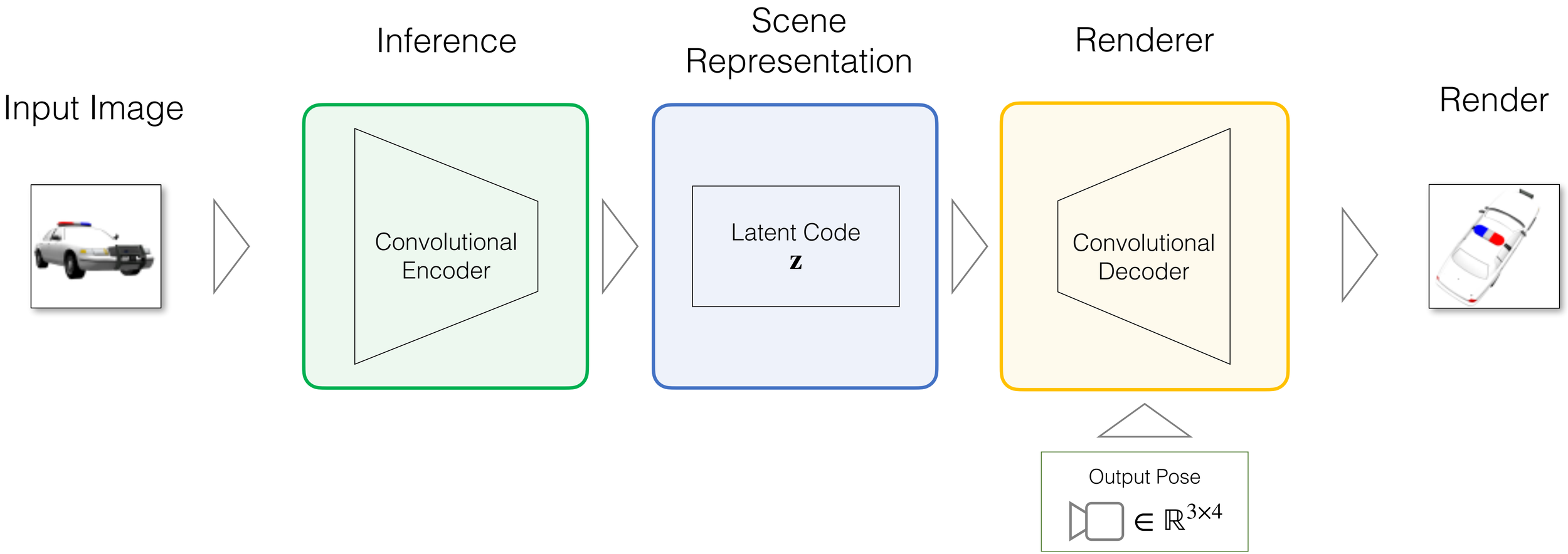


3D-Structured Decoder

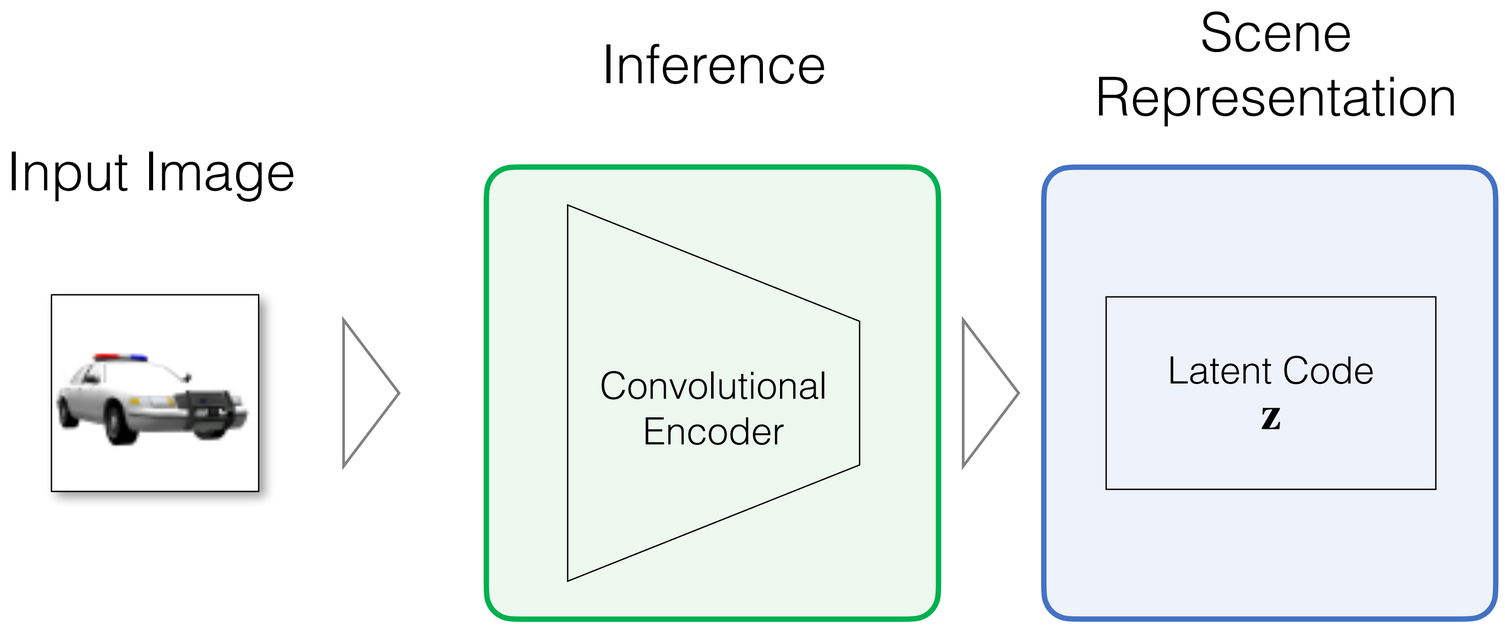


Just predict *all the parameters of that scene representation*.

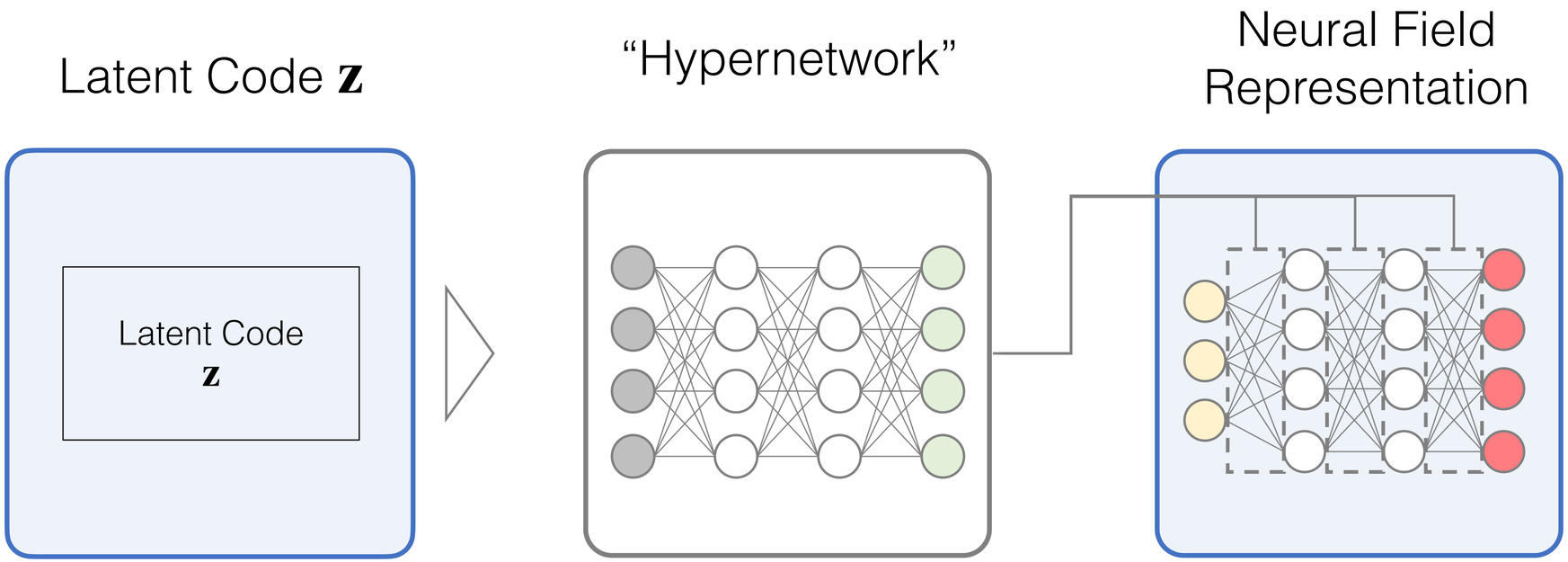
2D baseline: Auto-Encoder-Like model



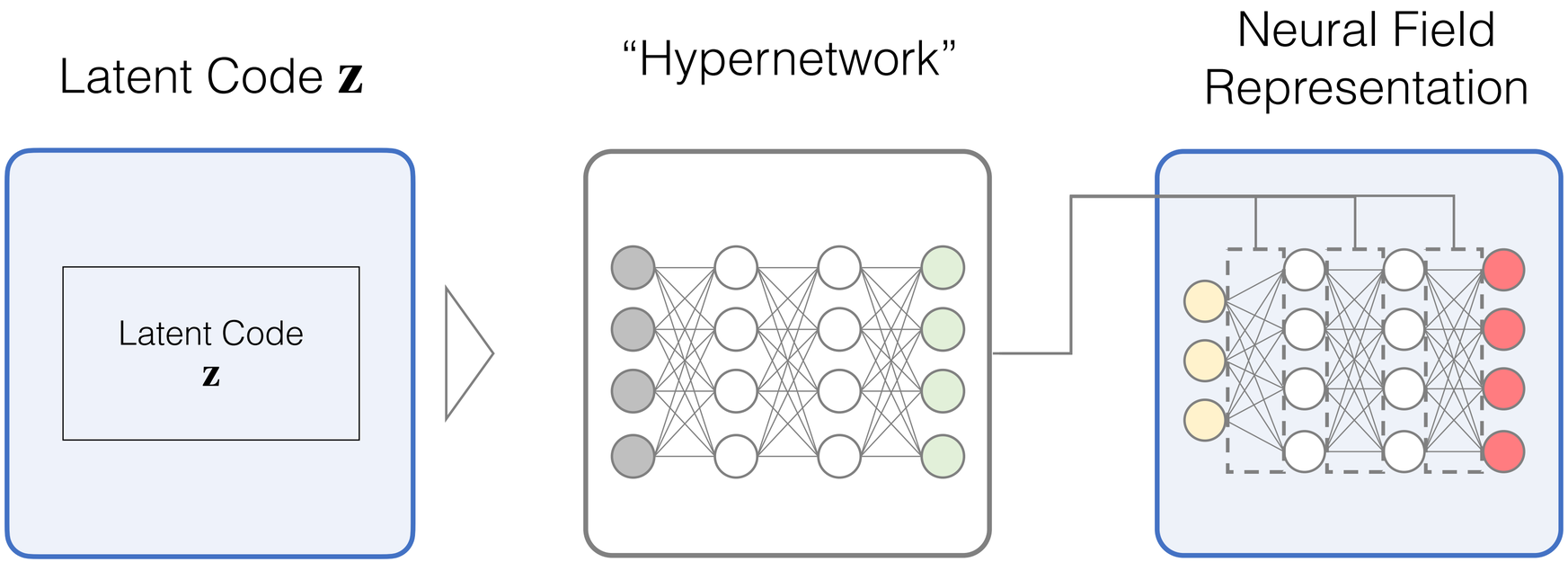
2D baseline: Auto-Encoder-Like model



Conditioning Neural Fields by predicting parameters



Conditioning Neural Fields by predicting parameters



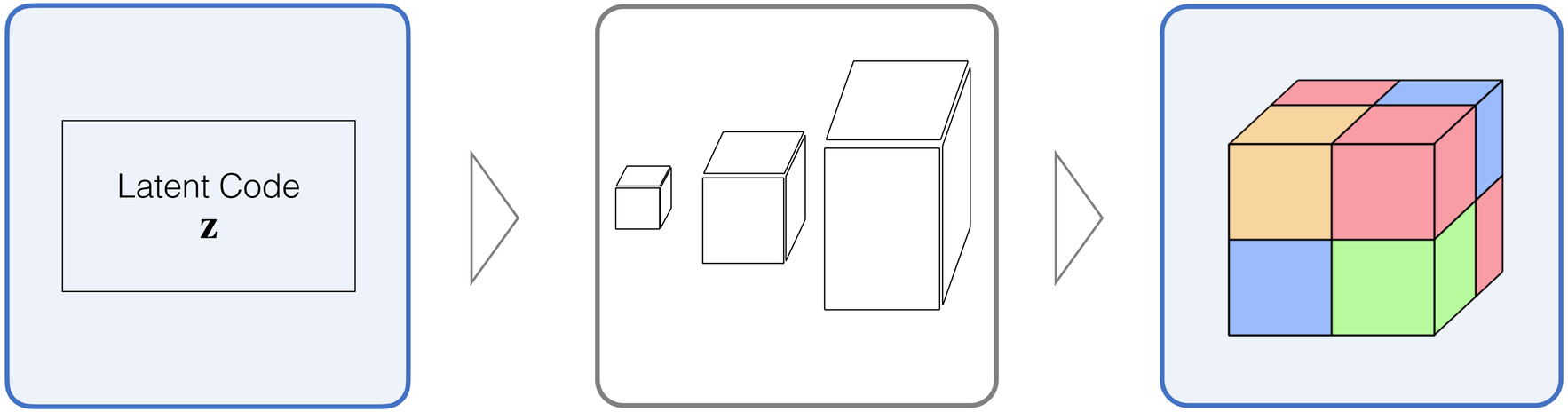
To condition neural field, can literally have MLP that takes as input $\mathbf{z}$ and outputs *all* of the parameters (weights, biases) of the neural field.

Conditioning Voxel Grids by predicting parameters

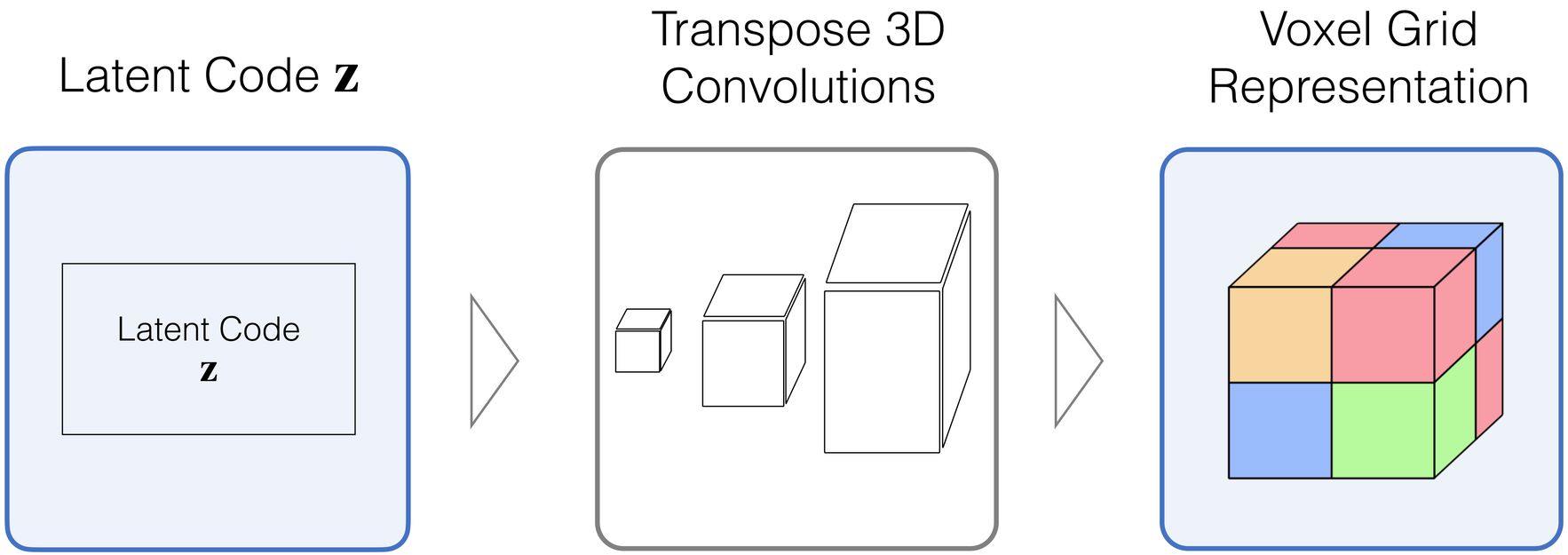
Latent Code $\mathbf{z}$

Transpose 3D Convolutions

Voxel Grid Representation

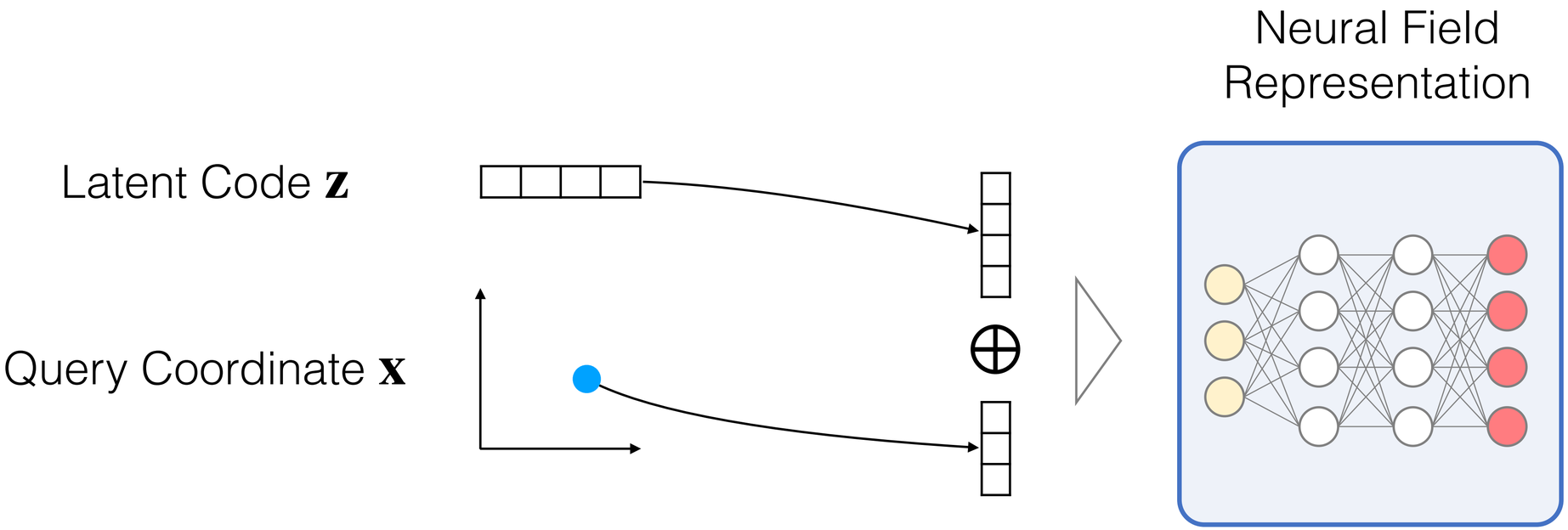


Conditioning Voxel Grids by predicting parameters

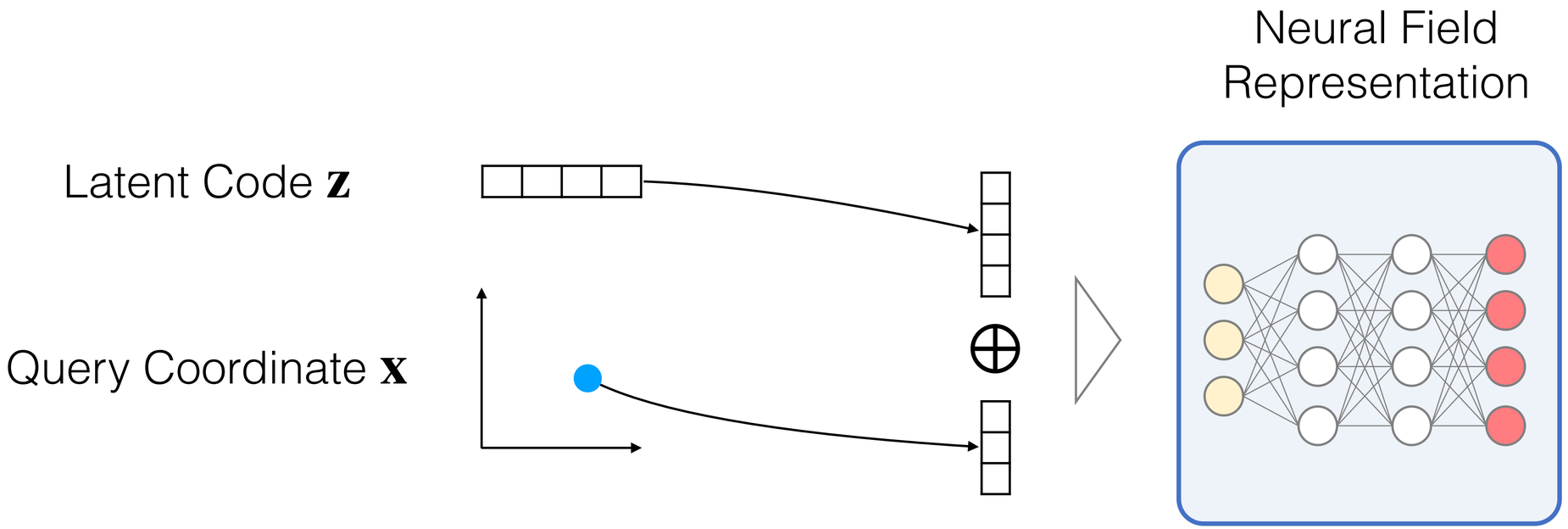


To condition voxel grid, can apply 3D de-convolutional neural network (aka transpose convolutions) to upsample $1 \times 1 \times 1 \times ch$ latent code to $N \times N \times N \times ch$.

Conditioning Neural Fields: Conditioning via Concatenation



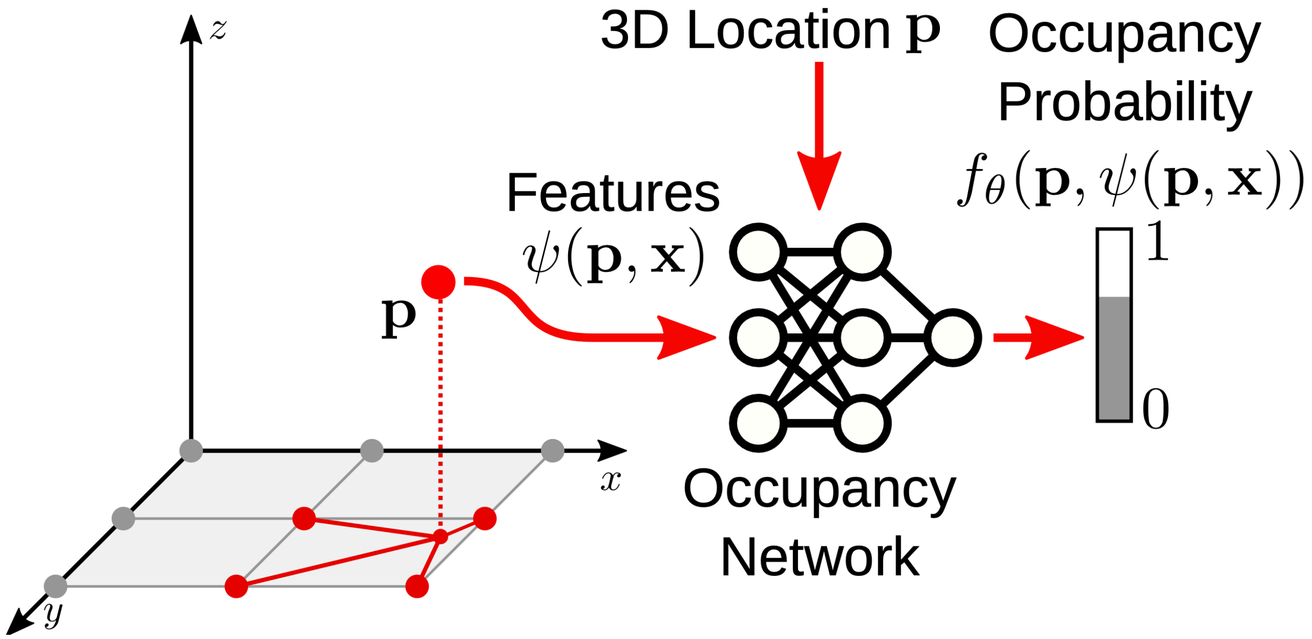
Conditioning Neural Fields: Conditioning via Concatenation



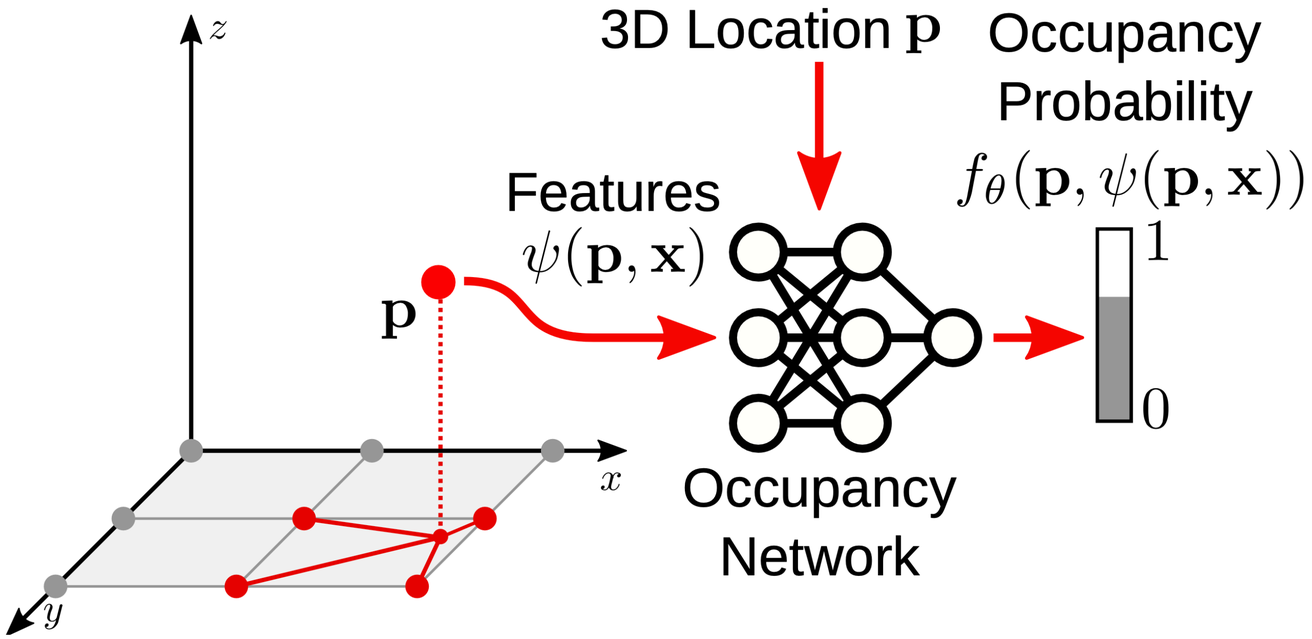
For neural fields, can concatenate latent code and query coordinate as input.

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

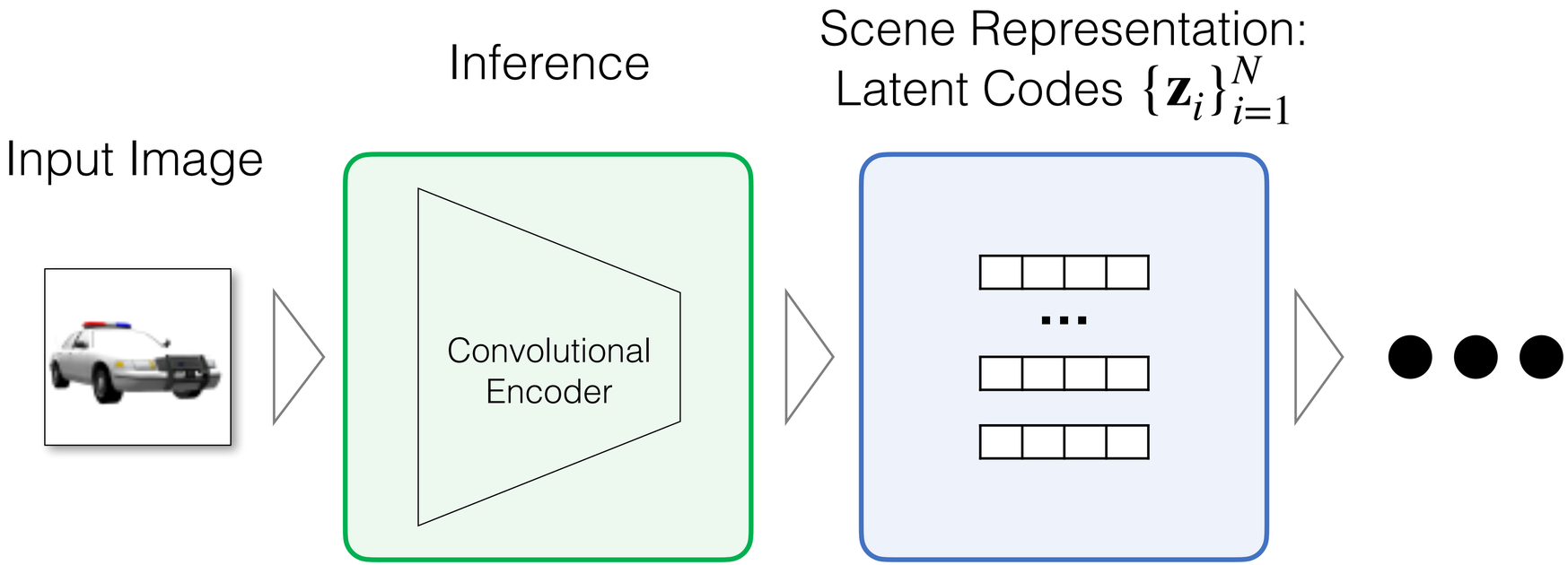


Conditioning Neural Fields: Conditioning via Concatenation

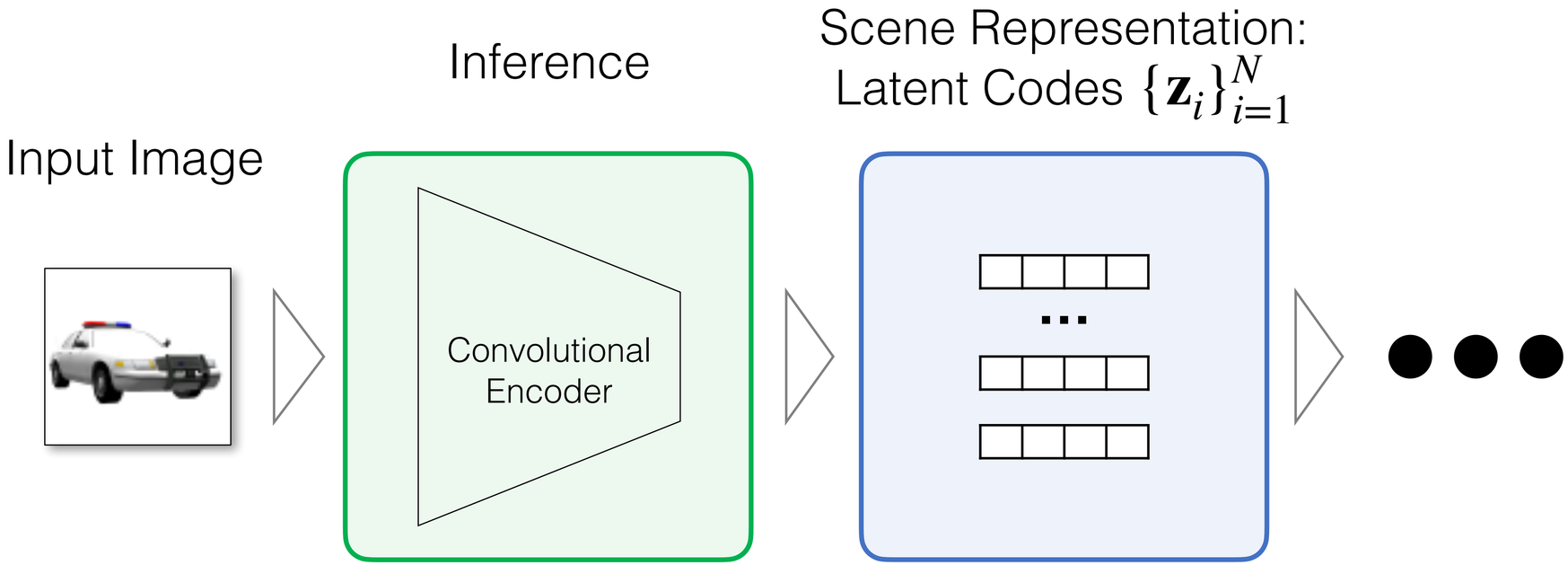


You already know this: That's what's usually done in hybrid discrete-continuous Representations!

Note on Terminology



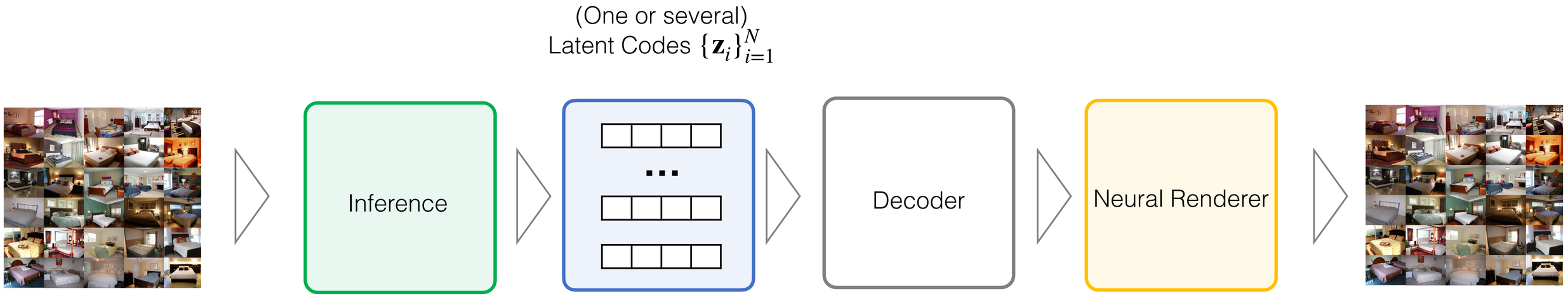
Note on Terminology



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.**

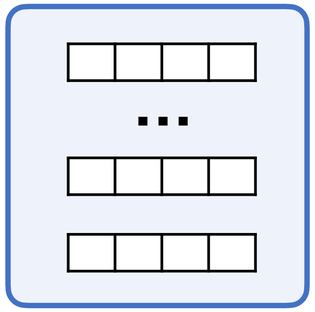
Summary: General Framework



Summary: General Framework

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

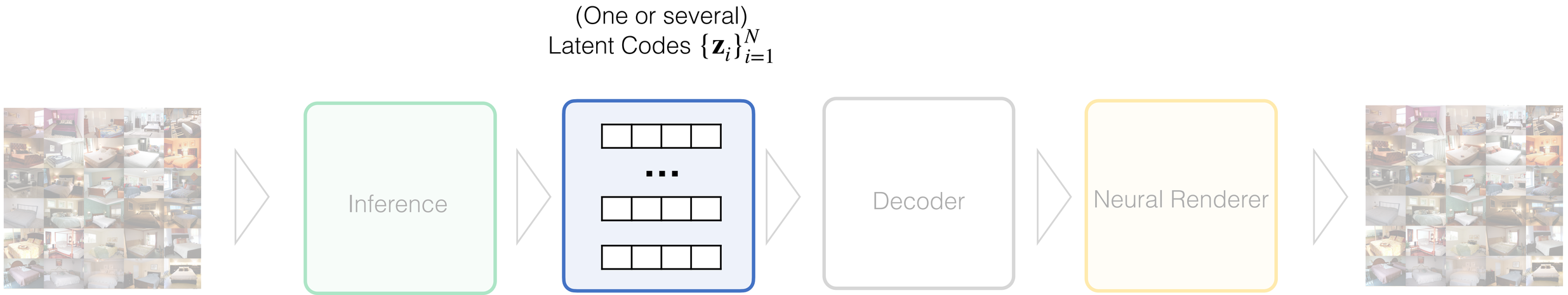
Inference



Decoder

Neural Renderer

Summary: General Framework

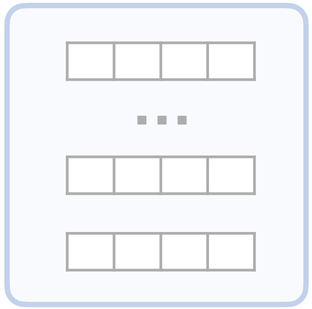


What are the **Latent Codes and how many should there be?**

Summary: General Framework

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

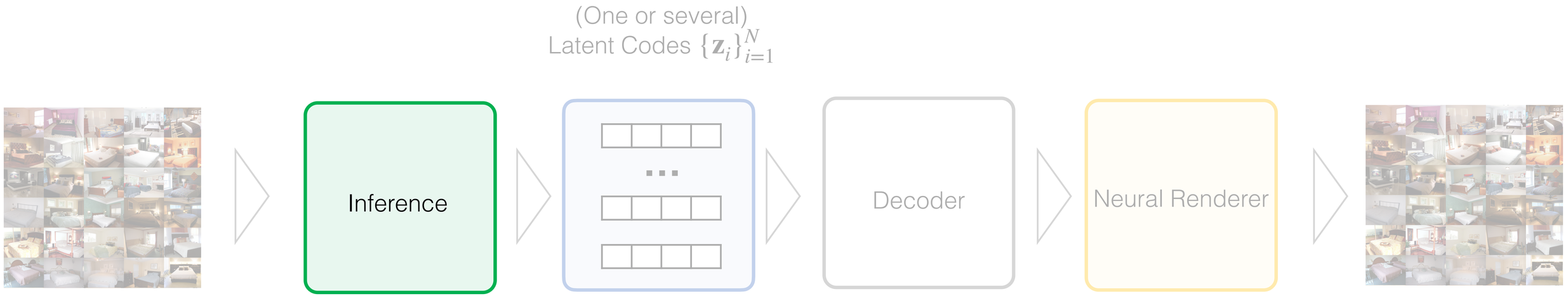
Inference



Decoder

Neural Renderer

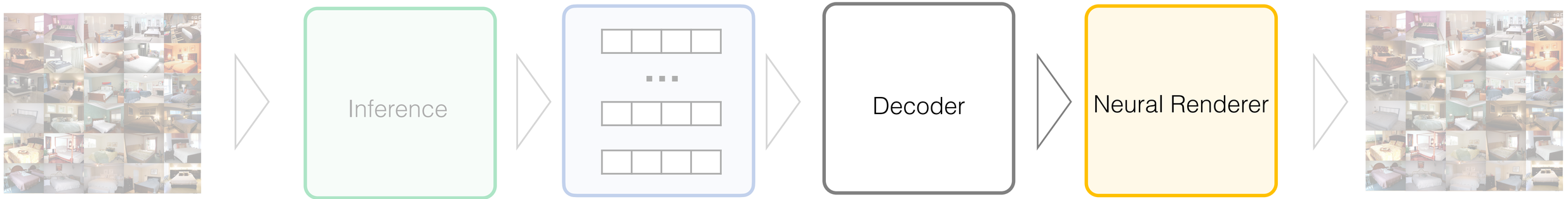
Summary: General Framework



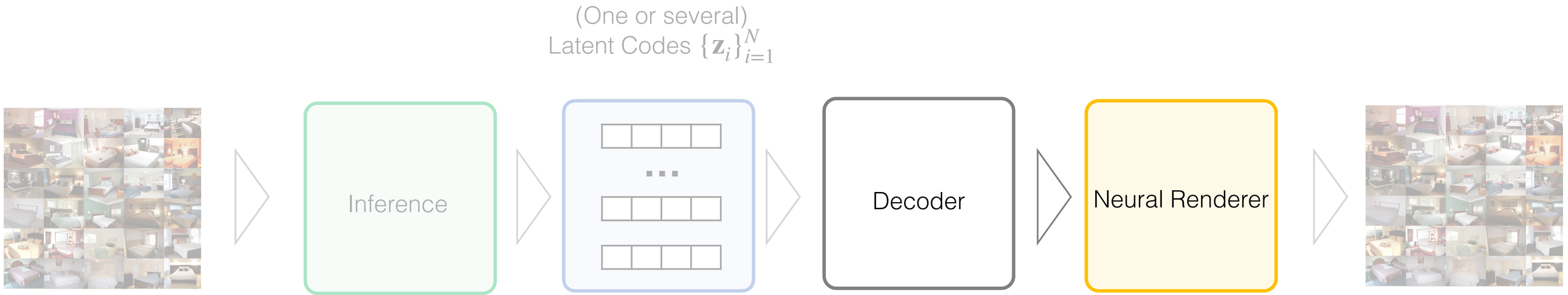
How to get the latent codes from images?

How to guarantee generalization?

Summary: General Framework



Summary: General Framework

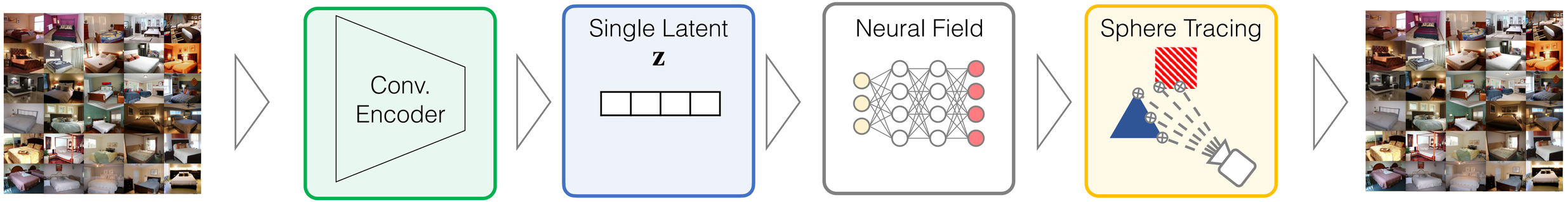


What forms & shapes can the decoder take?

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

Do we need 3D volumetric renderers?

Specific Model



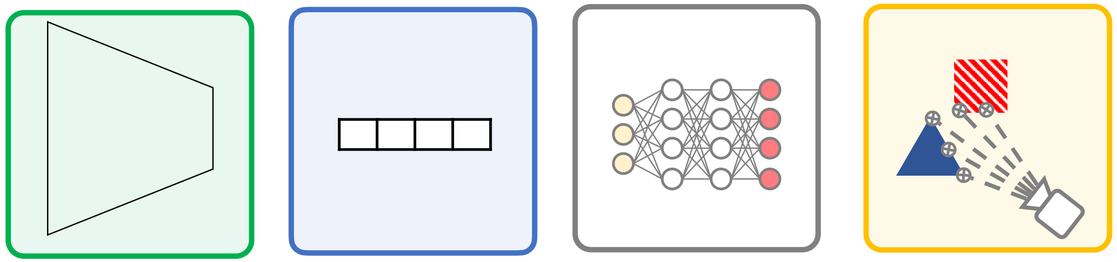
Learn inference model from many images of many scenes.

Learn inference model from many images of many scenes.

Model

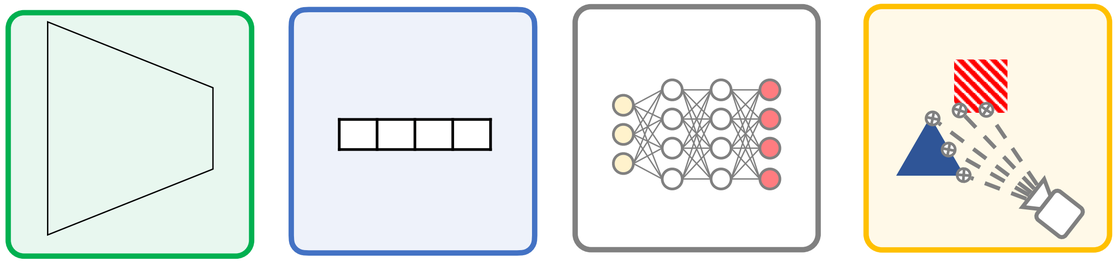
Learn inference model from many images of many scenes.

Model



Learn inference model from many images of many scenes.

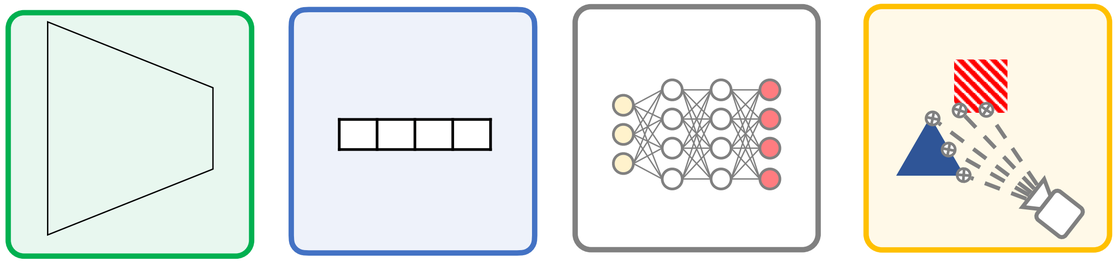
Model



Dataset

Learn inference model from many images of many scenes.

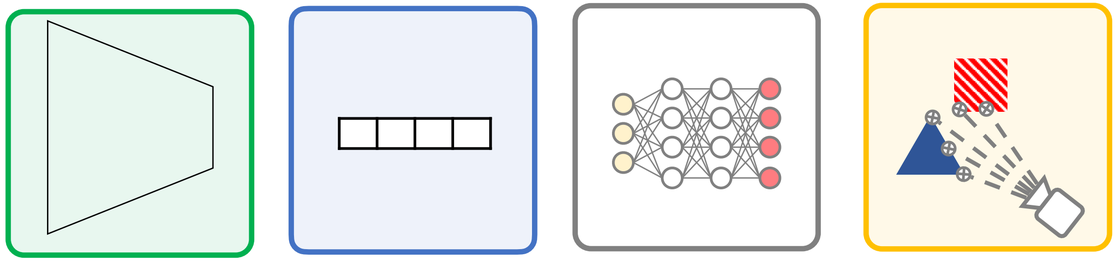
Model



Dataset

Learn inference model from many images of many scenes.

Model

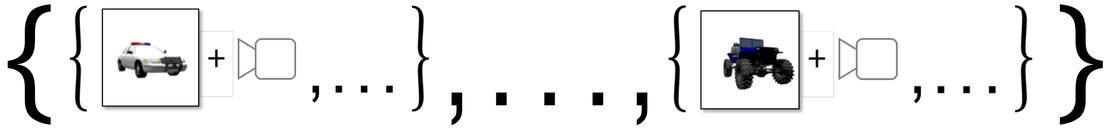


Input view

Normal map

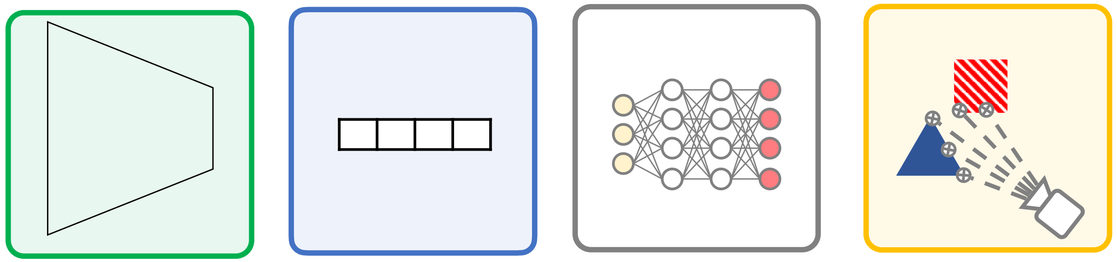
RGB

Dataset



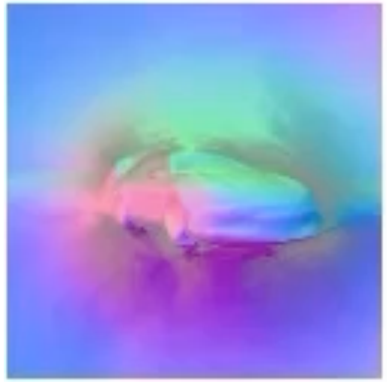
Learn inference model from many images of many scenes.

Model



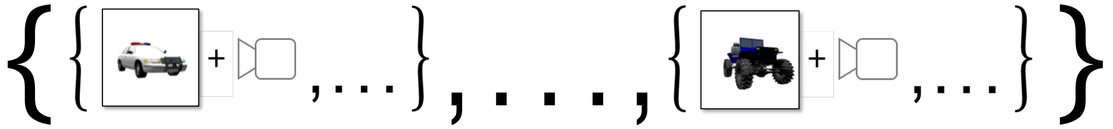
Input view

Normal map



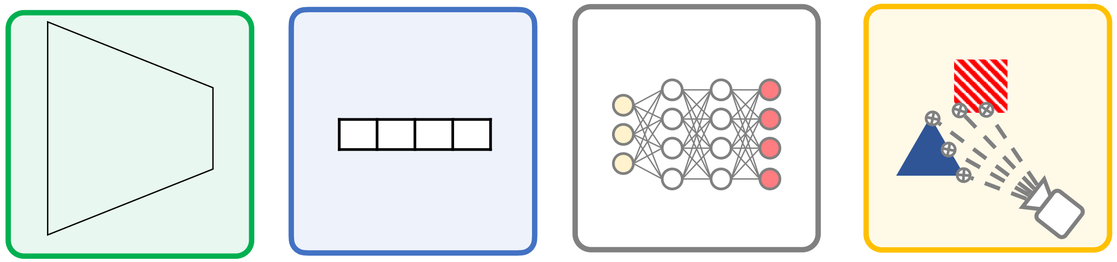
RGB

Dataset



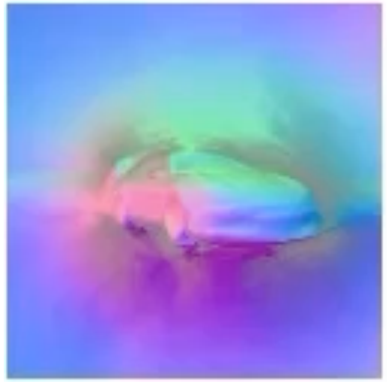
Learn inference model from many images of many scenes.

Model



Input view

Normal map



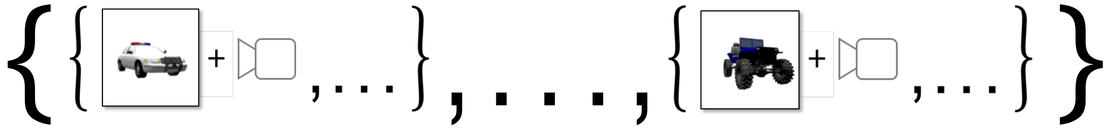
RGB

Out-of-distribution
Input view

Normal map

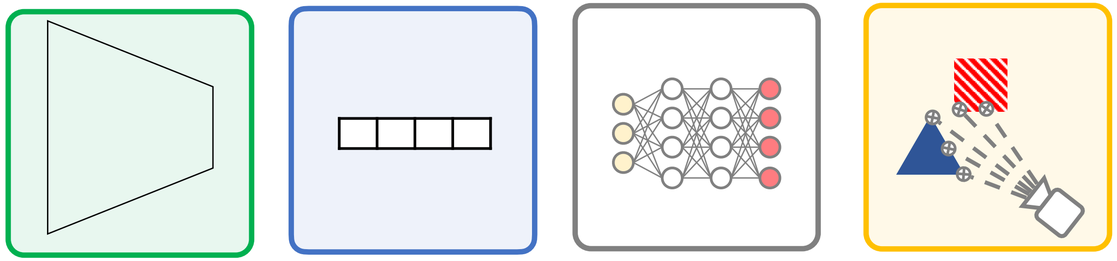
RGB

Dataset



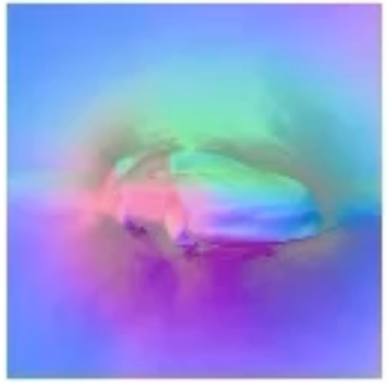
Learn inference model from many images of many scenes.

Model



Input view

Normal map



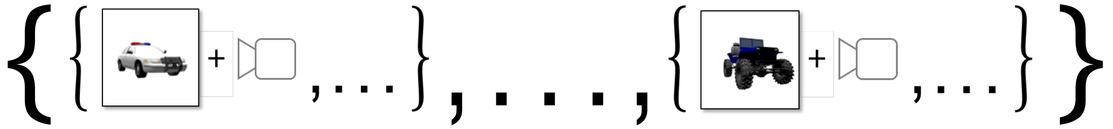
RGB

Out-of-distribution
Input view

Normal map

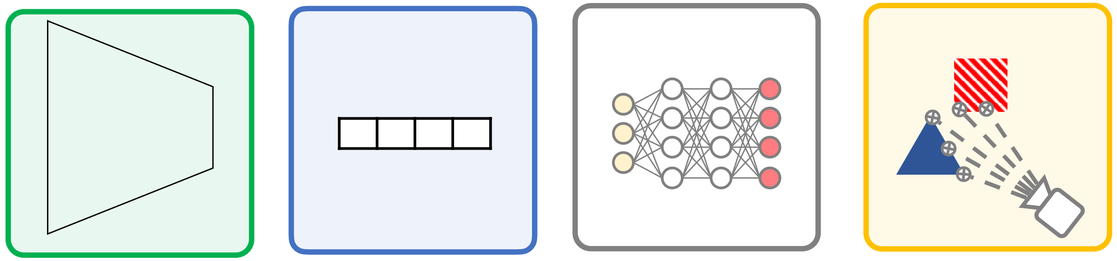
RGB

Dataset

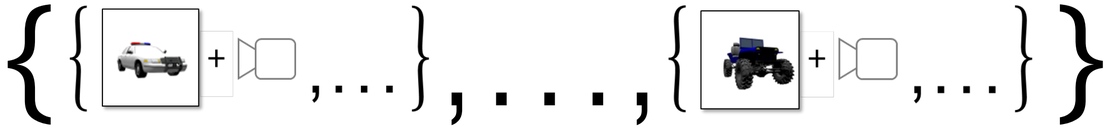


Learn inference model from many images of many scenes.

Model



Dataset



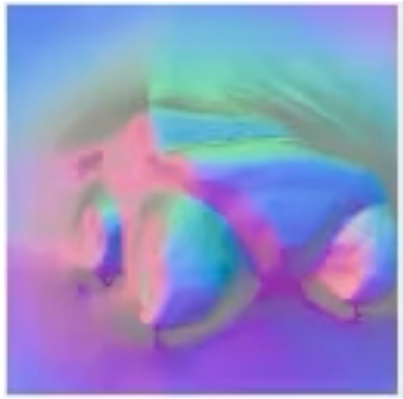
Input view

Normal map

RGB

Out-of-distribution
Input view

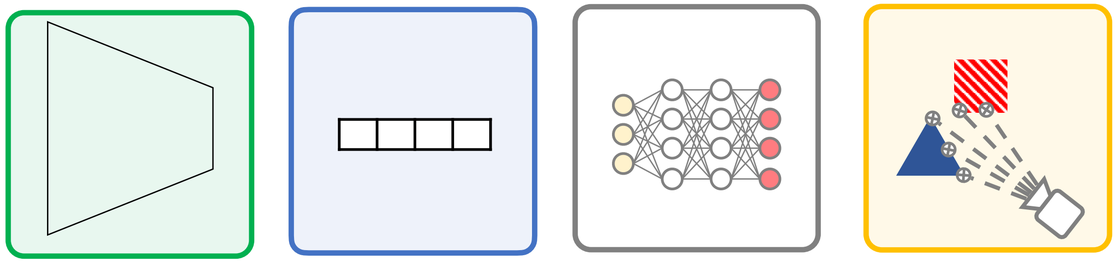
Normal map



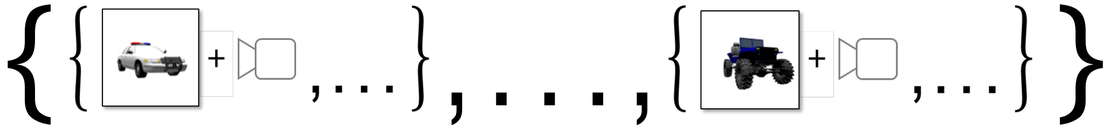
RGB

Learn inference model from many images of many scenes.

Model



Dataset



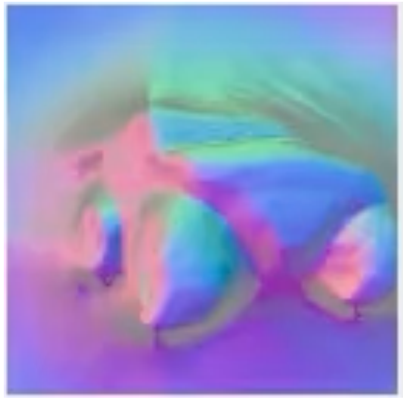
Input view

Normal map

RGB

Out-of-distribution
Input view

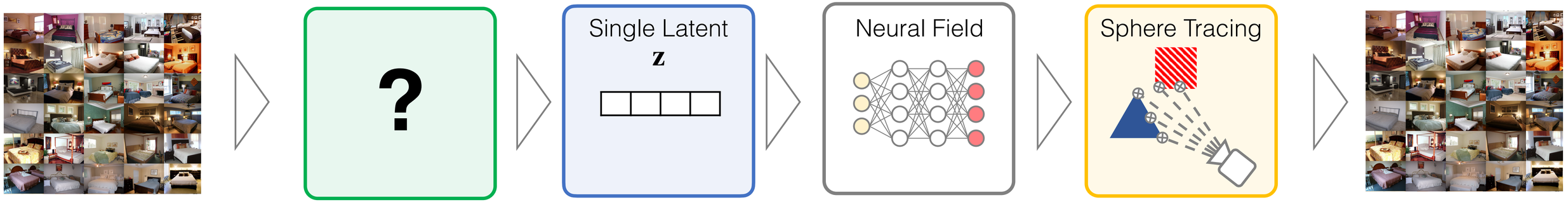
Normal map



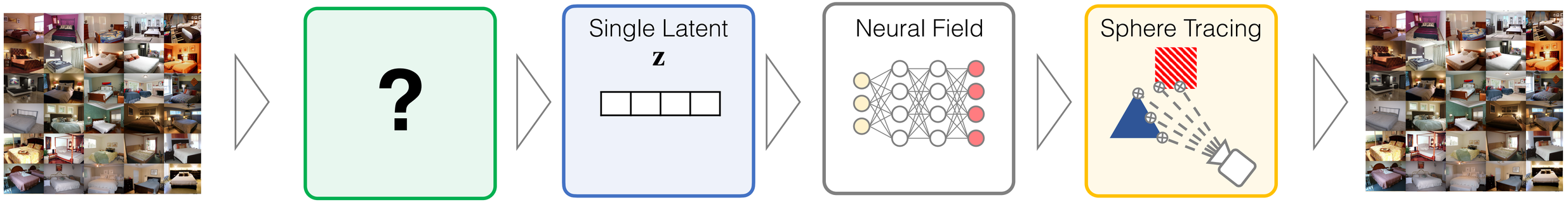
RGB

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

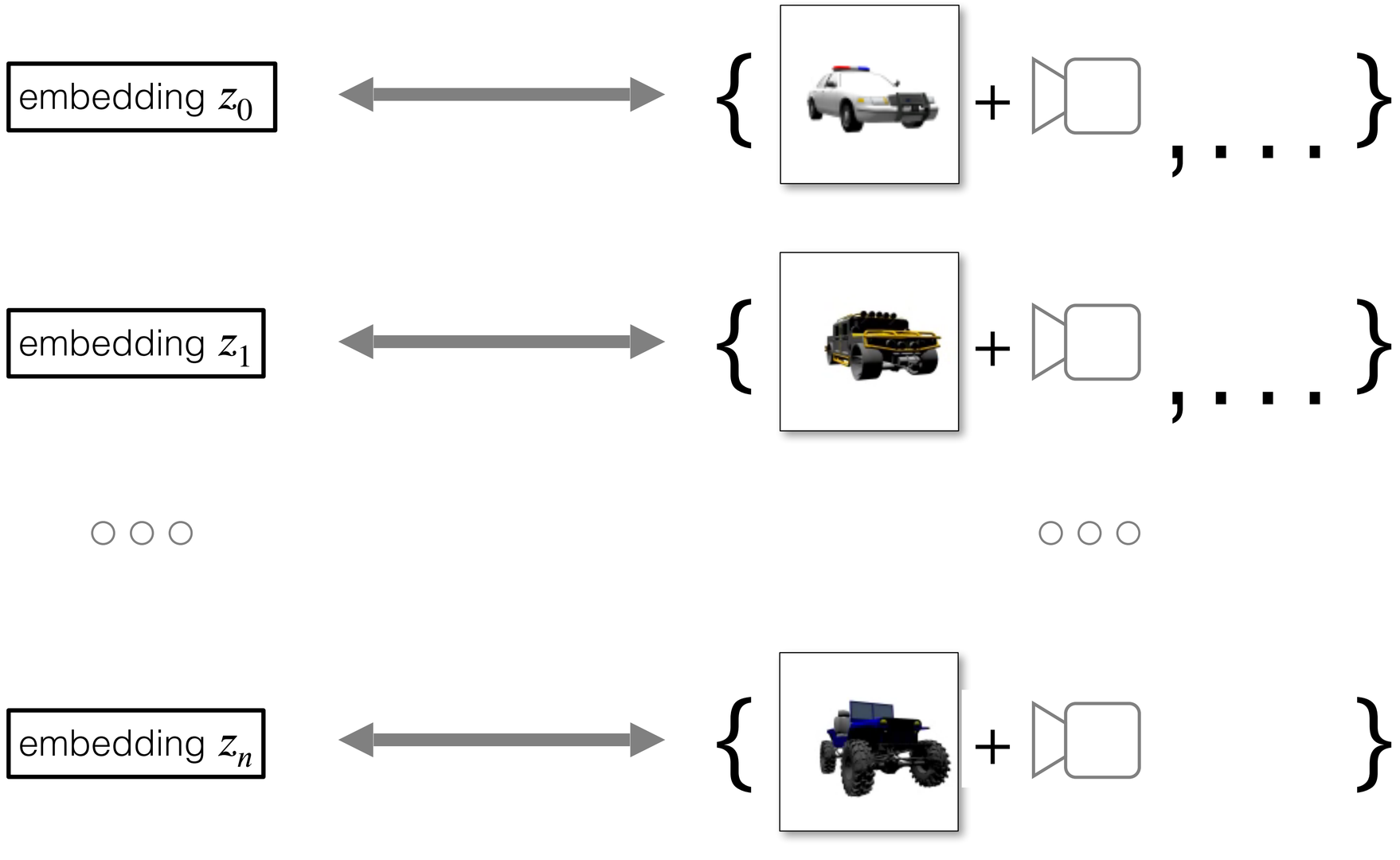
embedding z_1  +

○ ○ ○ ○ ○ ○

embedding z_n  +

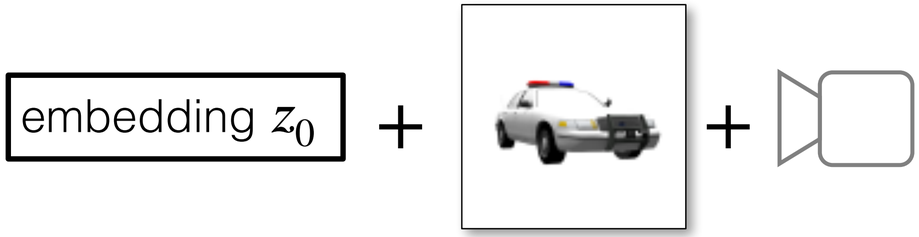
$$z_{j=0,\ldots,n} \sim \mathcal{N}(0, \sigma^2)$$

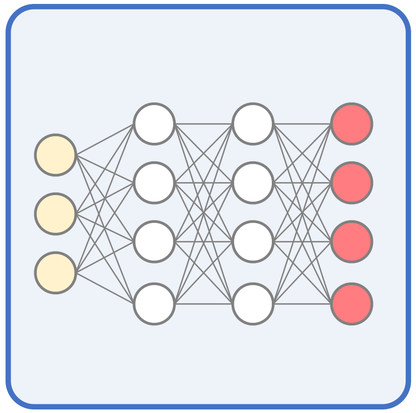
Training: Initialize one embedding per scene



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

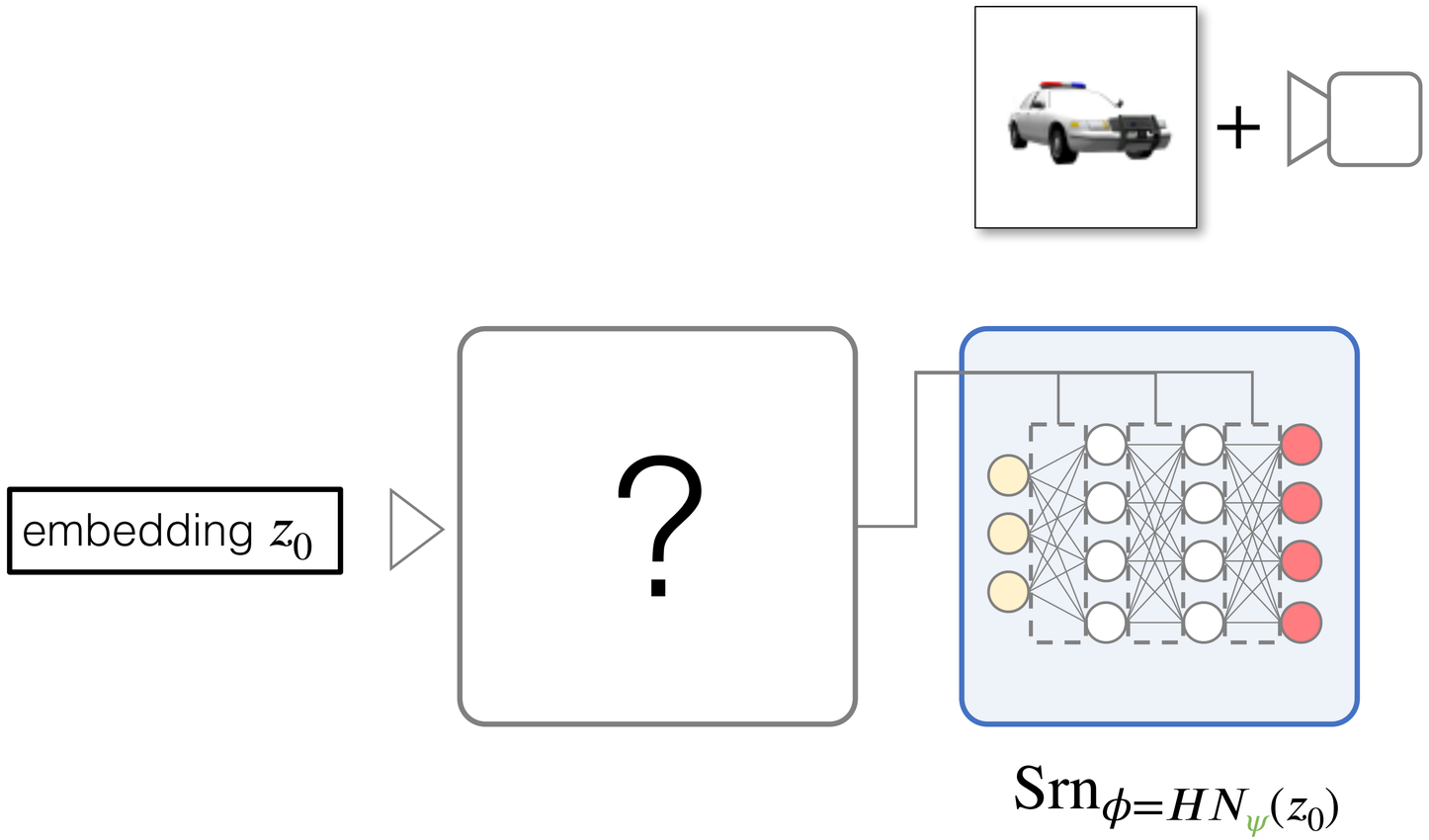
Decode embedding into scene representation



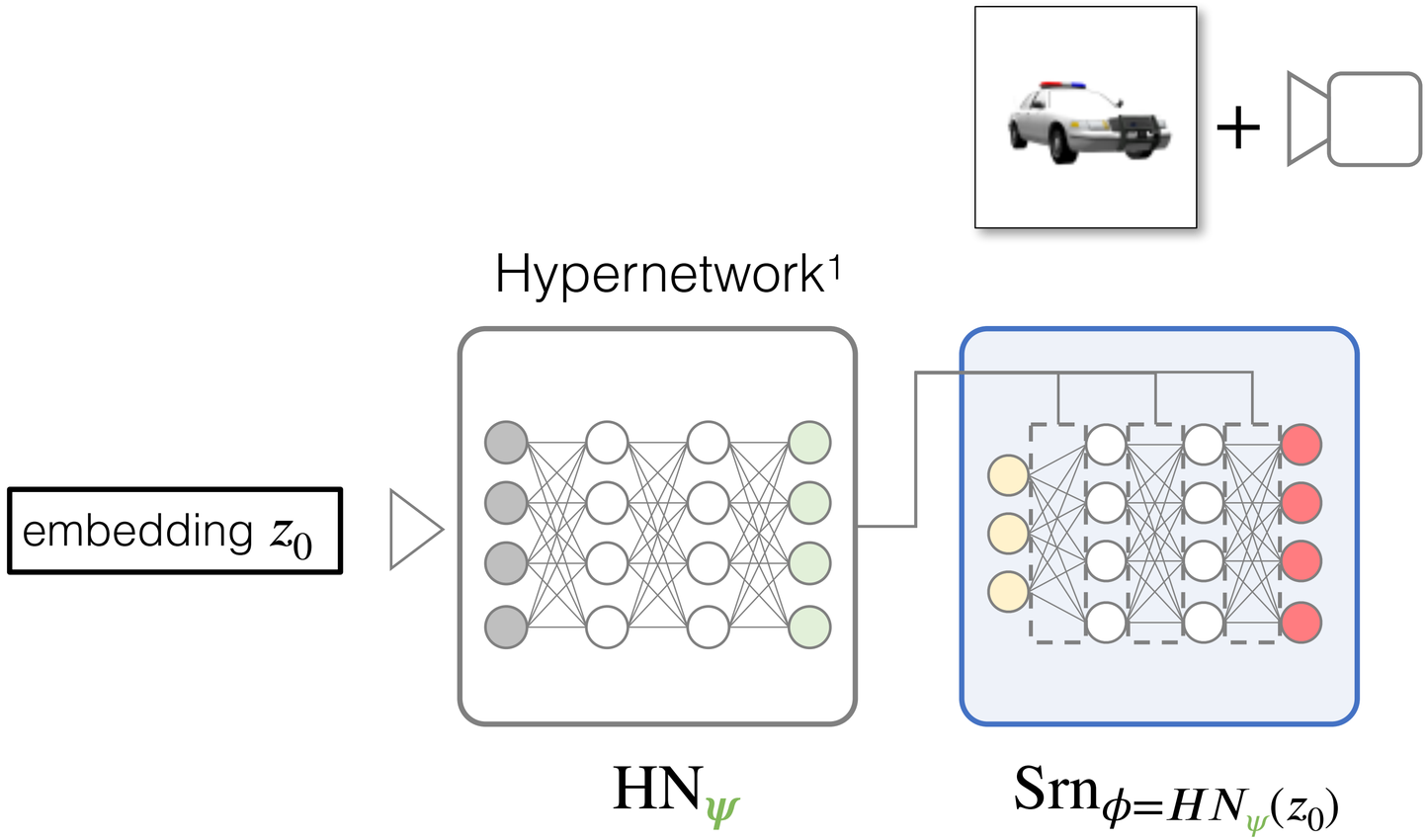


$$\text{Srn}_{\phi=HN_{\psi}(z_0)}$$

Decode embedding into scene representation

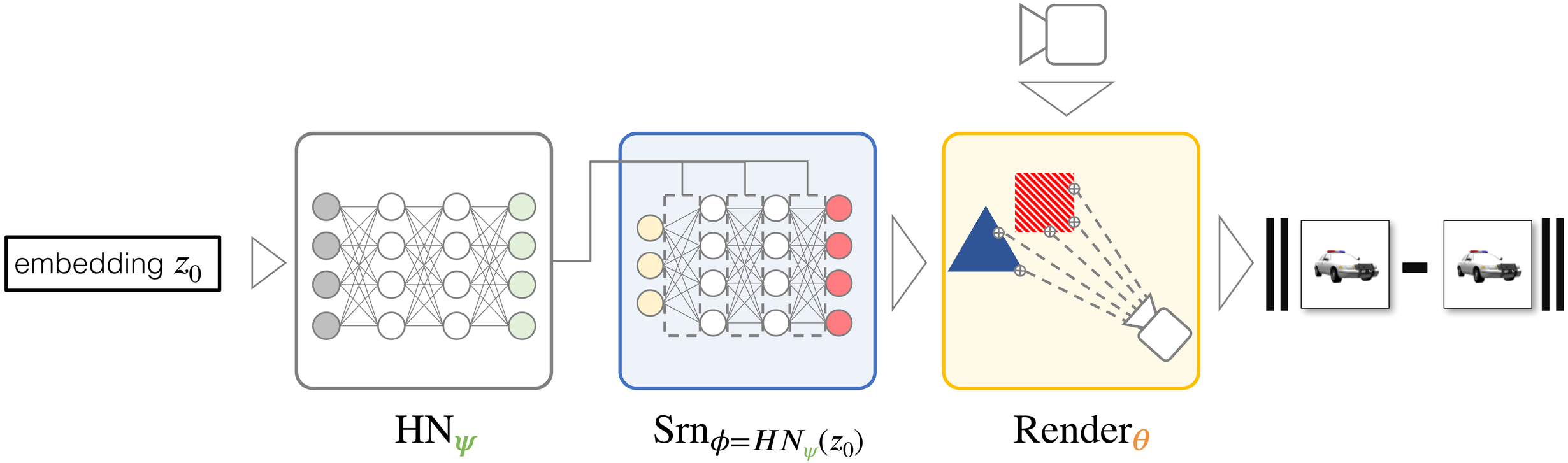


Decode embedding into scene representation

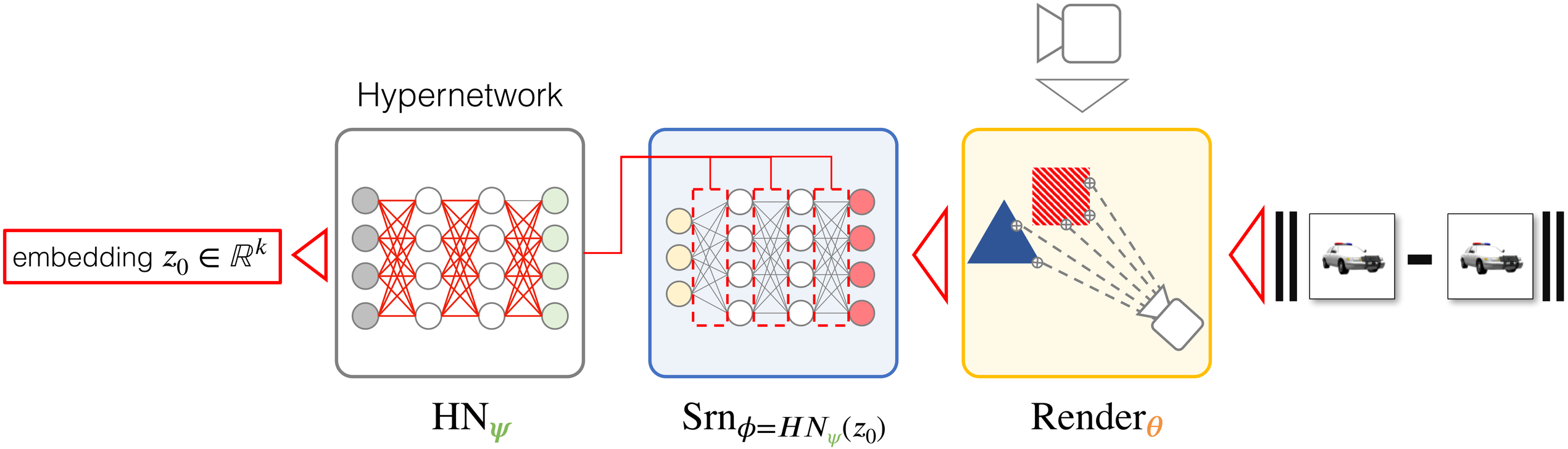


[1][Schmidhuber et al. 1992, Schmidhuber et al. 1993, Stanley et al. 2009, Ha et al., 2016]

Render training view

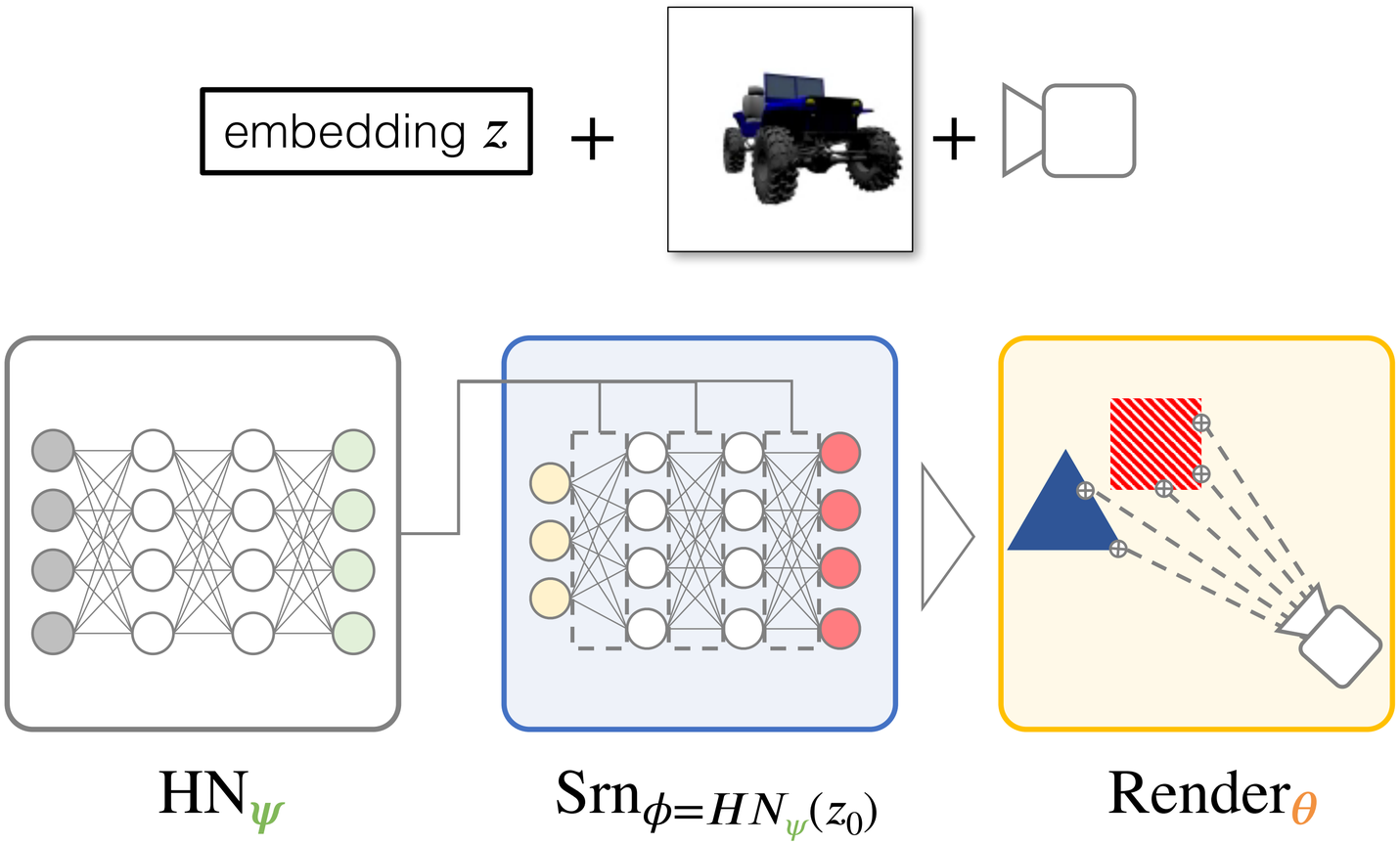


Co-optimize embeddings and model weights!

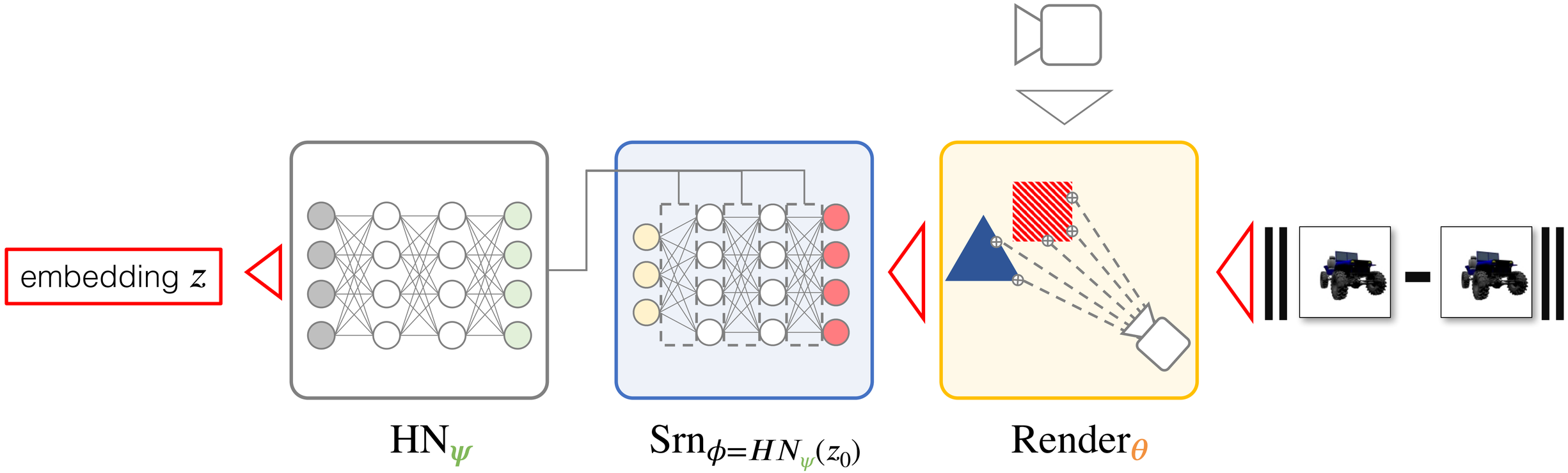


$$\arg \min_{\{z_j\}_{j=1}^{M},\psi,\theta} \sum_j \sum_i \left\| \text{Render}_{\theta}(\text{Srn}_{\phi=HN_{\psi}(z_j)}, \xi_i) - \mathcal{I}_i^j \right\|$$

Test time: initialize new embedding



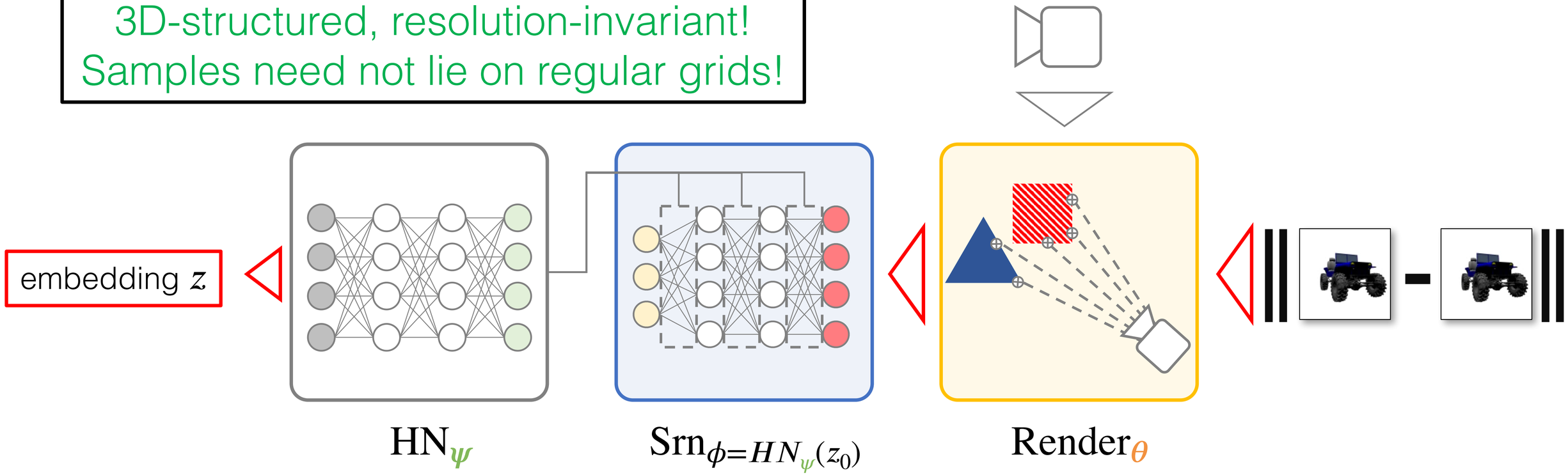
Inference: Freeze model weights, backprop into embedding only!



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

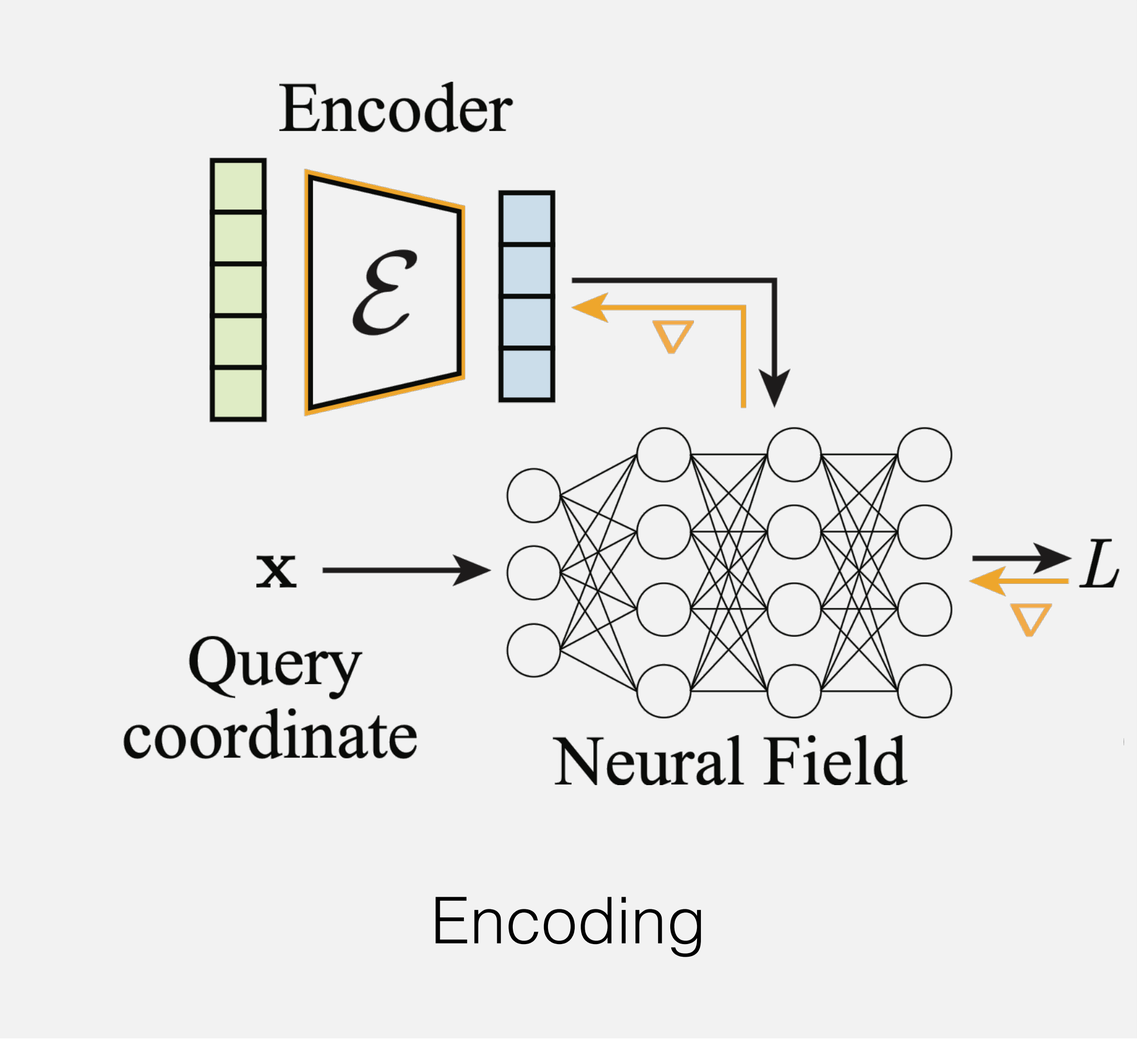
Inference: Freeze model weights, backprop into embedding only!

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



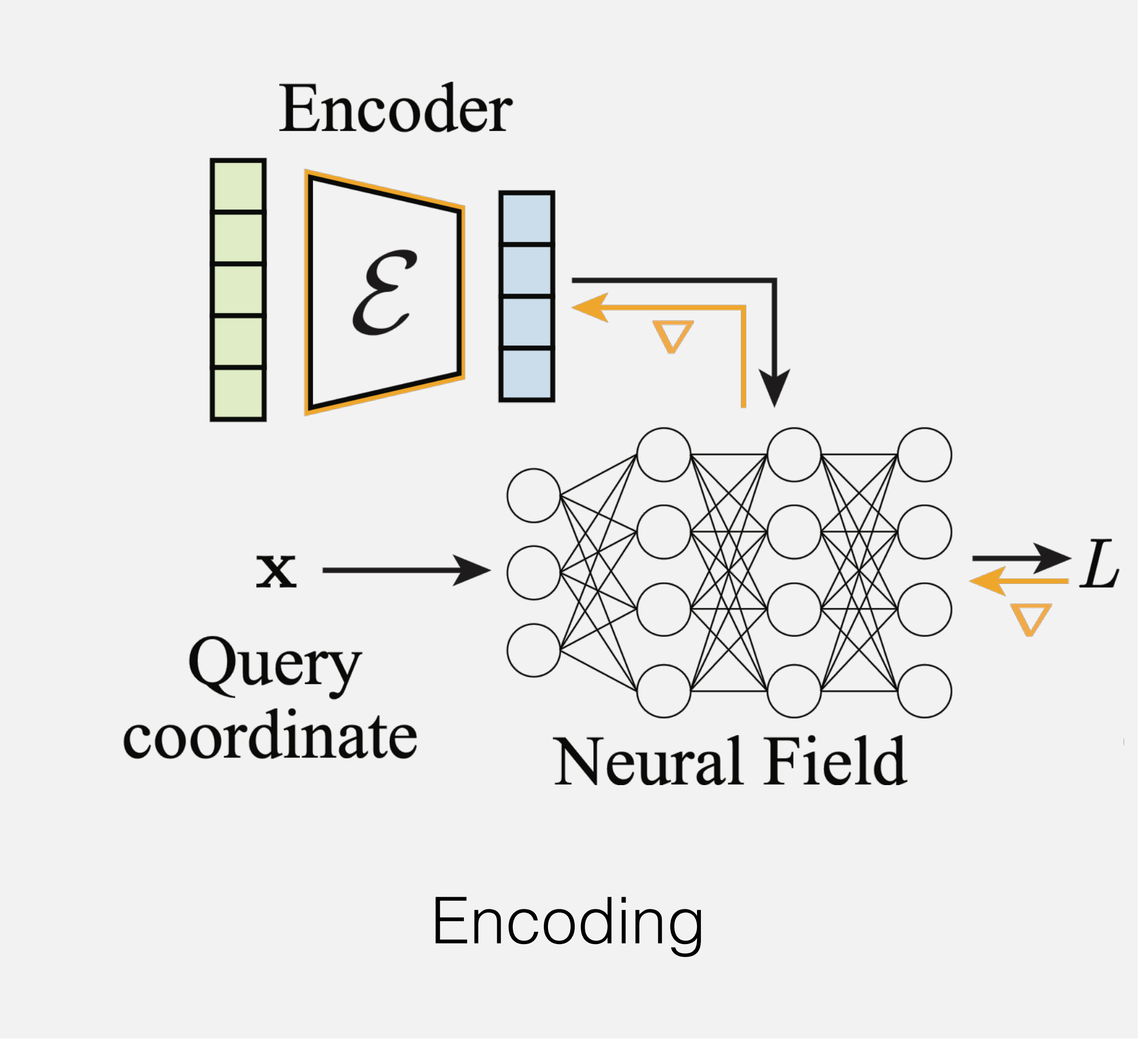
$$\hat{z} = \underset{z}{\text{argmin}} \left\| \text{Render}_{\theta}(\text{Srn}_{\phi=HN_{\psi}(z)}, \xi) - \mathcal{I} \right\|$$

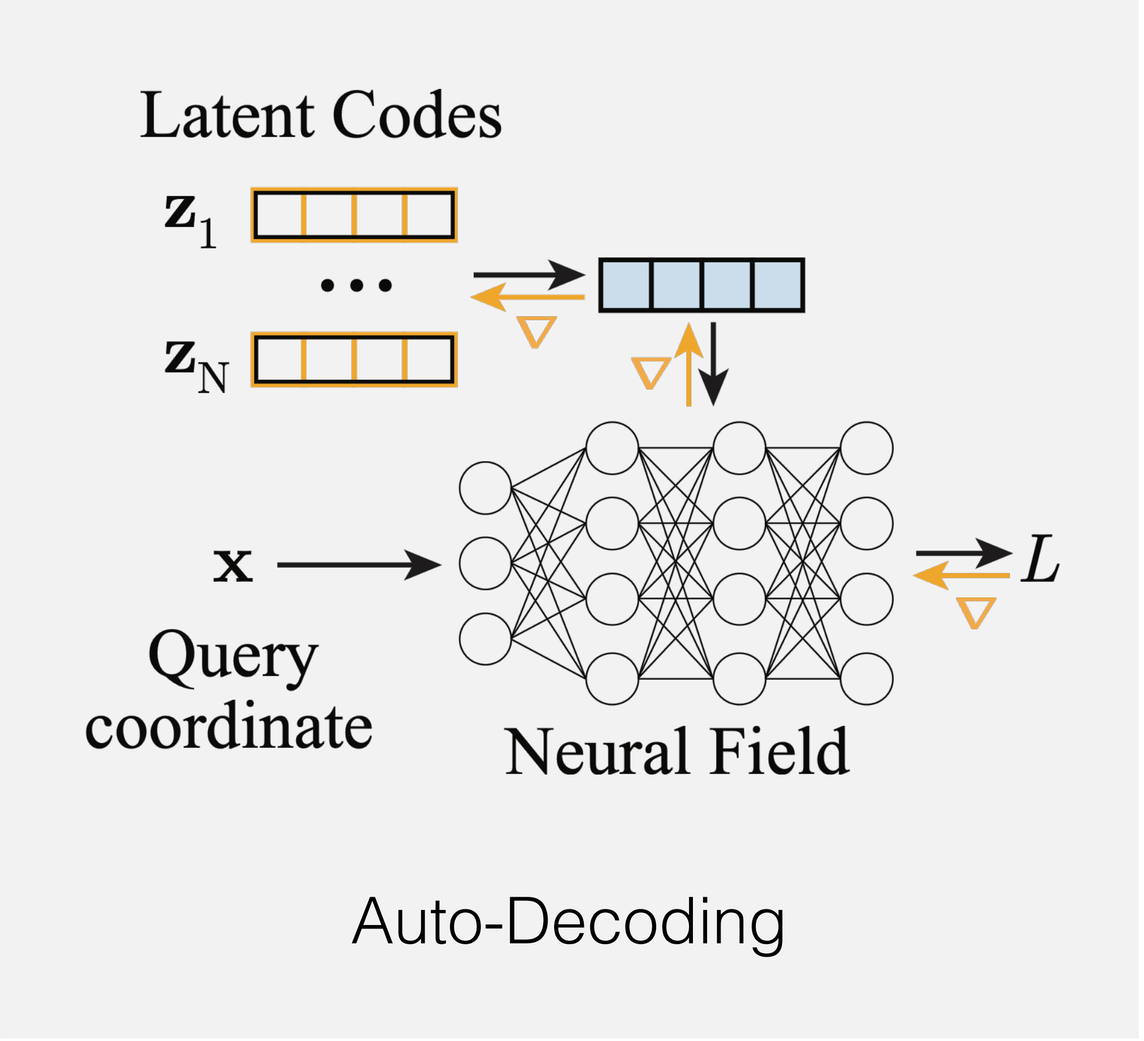
Encoding & Auto-Decoding



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

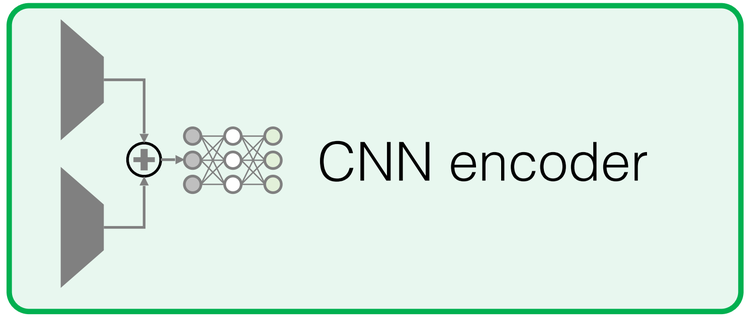
Encoding & Auto-Decoding





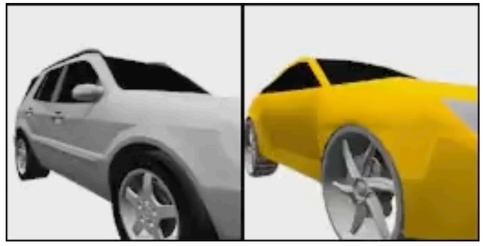
Neural Fields in Visual Computing and Beyond, Xie et al., EG STAR 2022

Out-of-distribution generalization

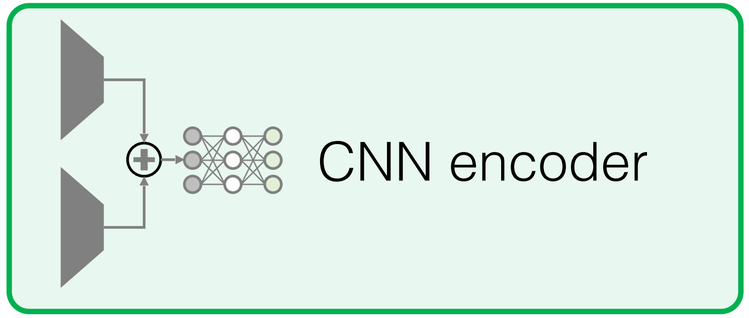


Input

Reconstructions

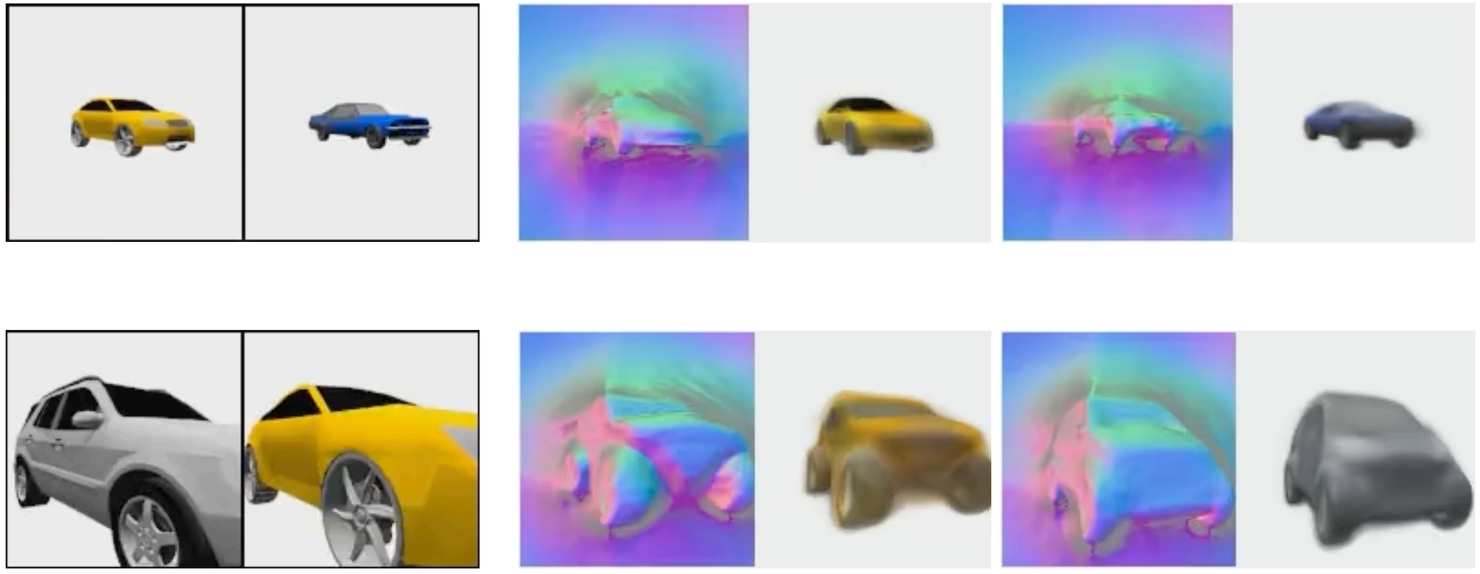


Out-of-distribution generalization

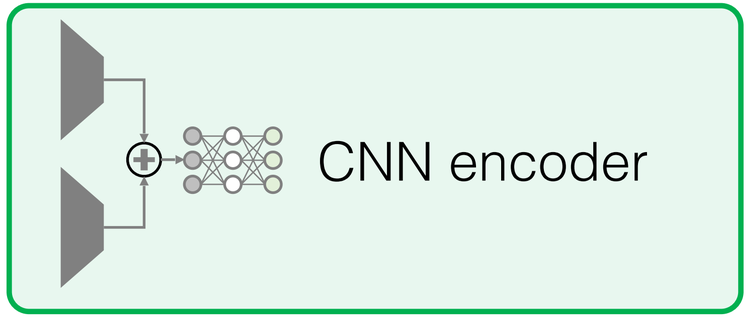


Input

Reconstructions



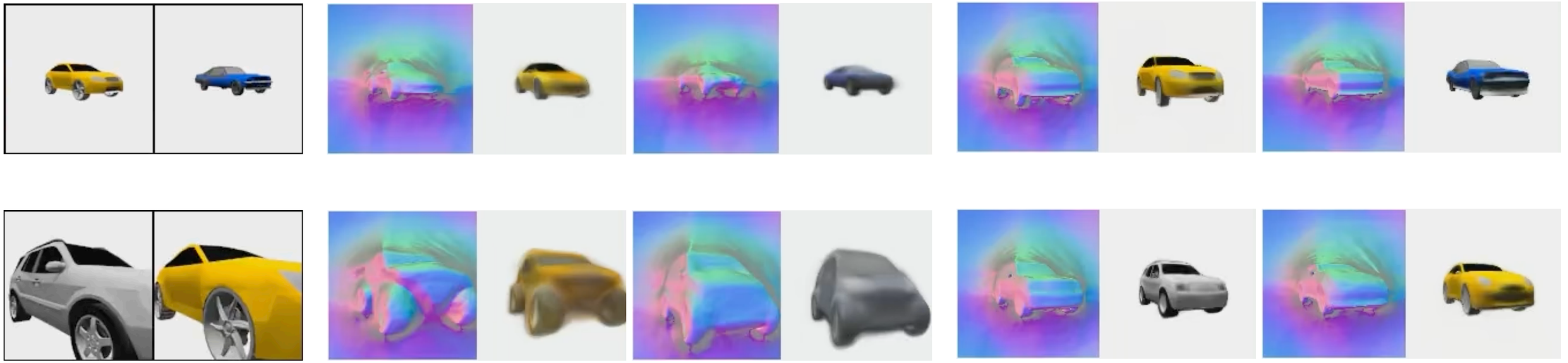
Out-of-distribution generalization



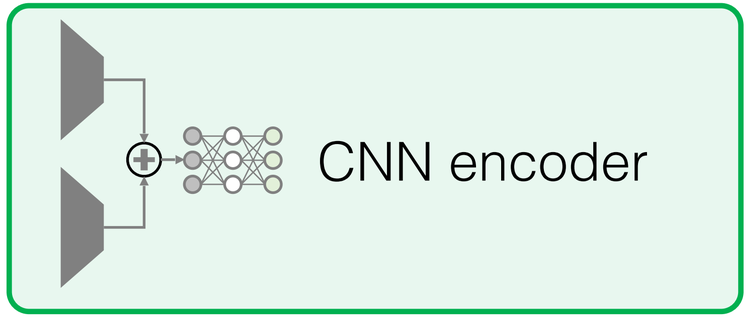
$$argmin_z ||Render(SRN, \xi) - \mathcal{I}||$$

Input

Reconstructions



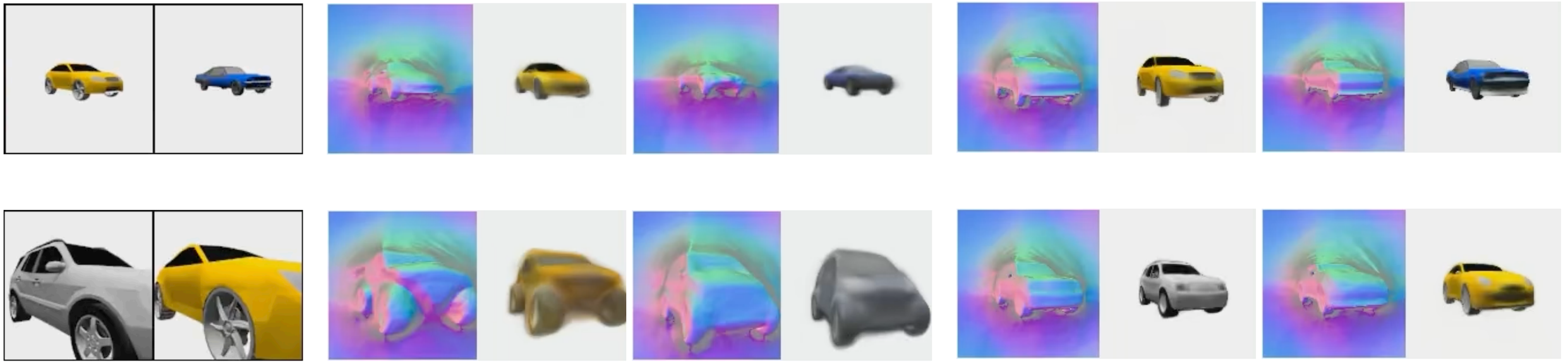
Out-of-distribution generalization



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

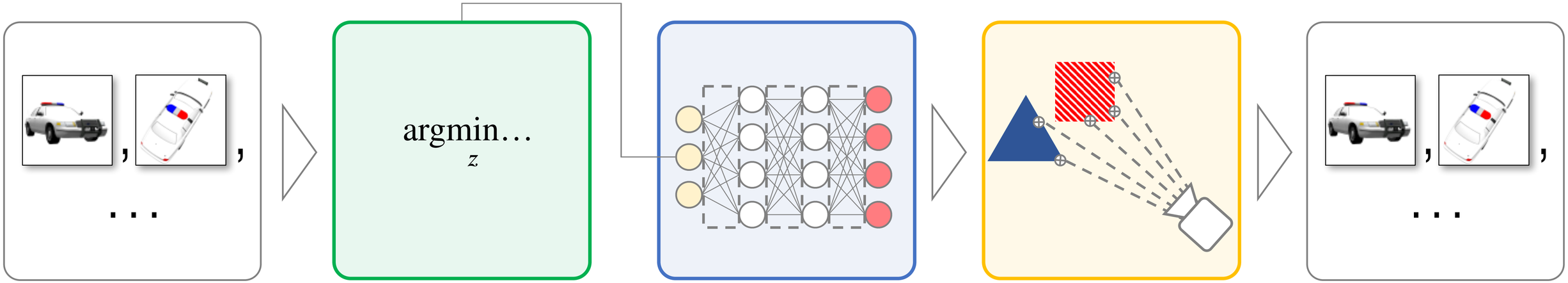
Input

Reconstructions

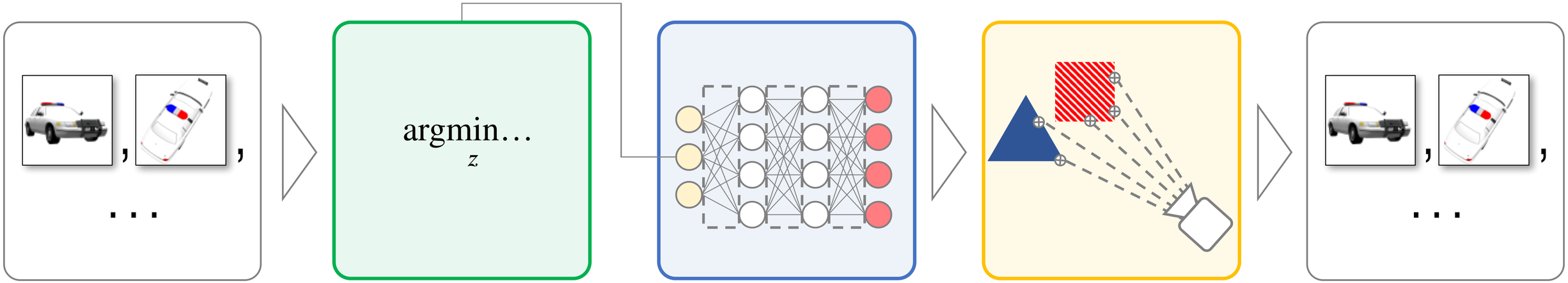


3D structure enables <u>generalization to out-of-distribution camera poses</u>!

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

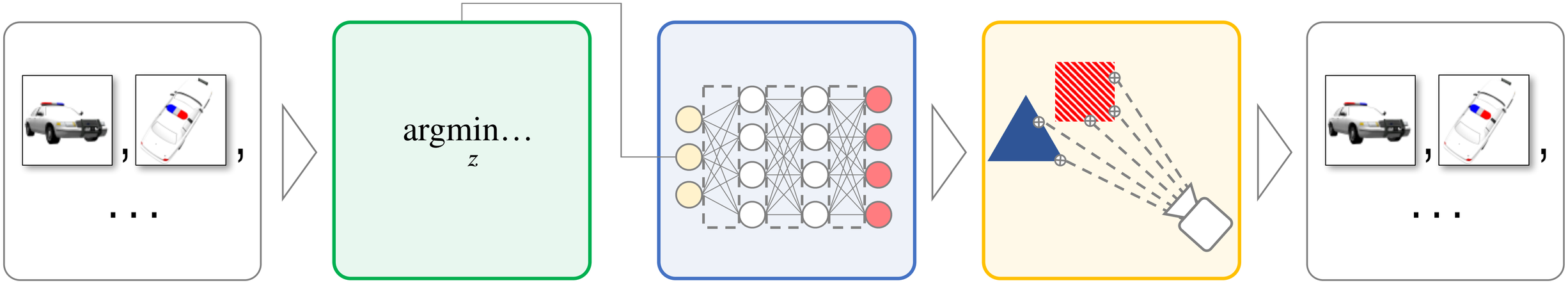


Scene Representation Networks (Sitzmann et al., Neurips 2019)



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)

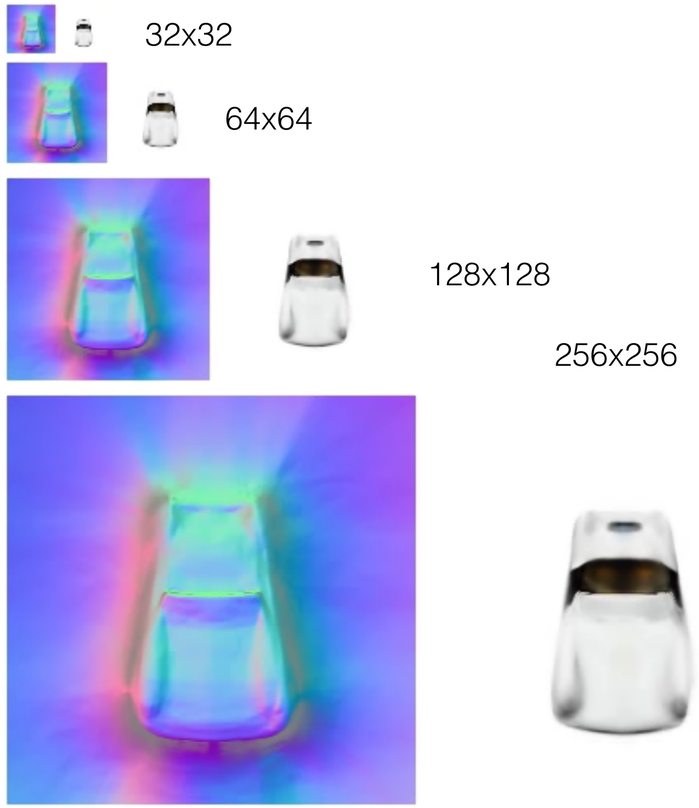
Tatarchenko et al. 2015

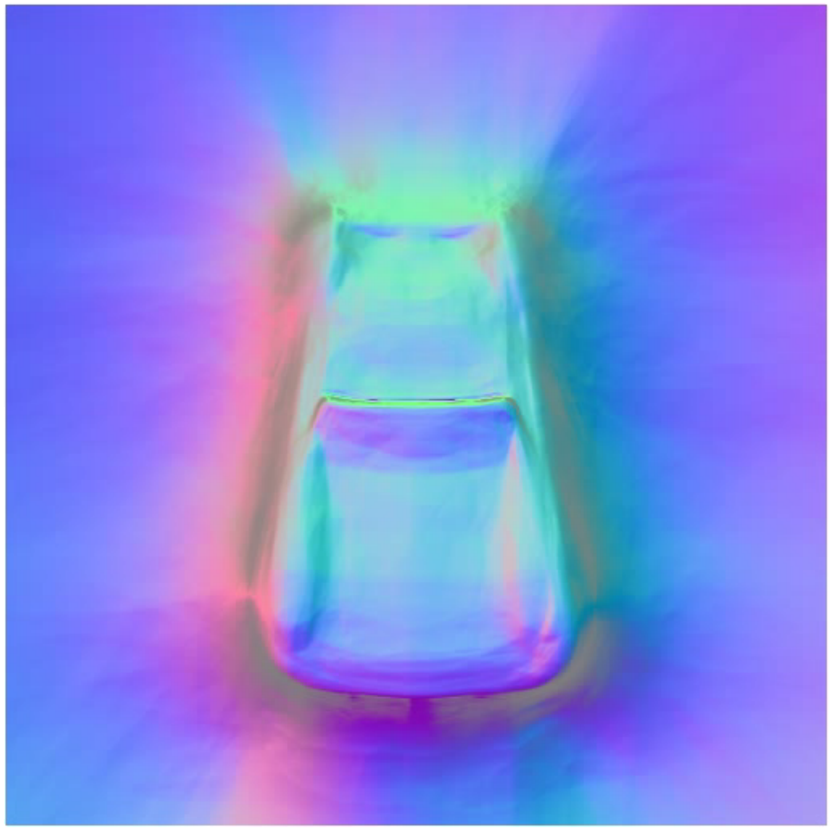
Worrall et al. 2017

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



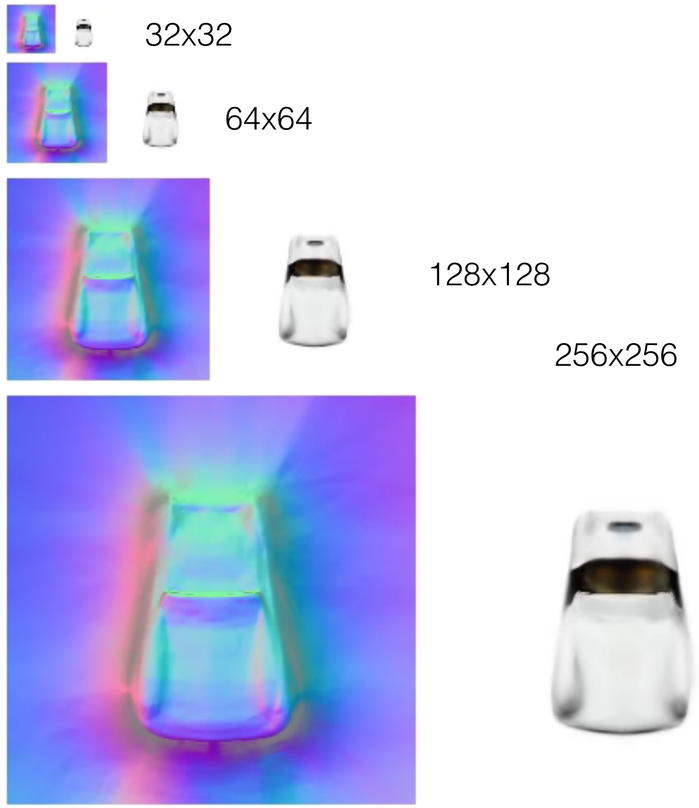


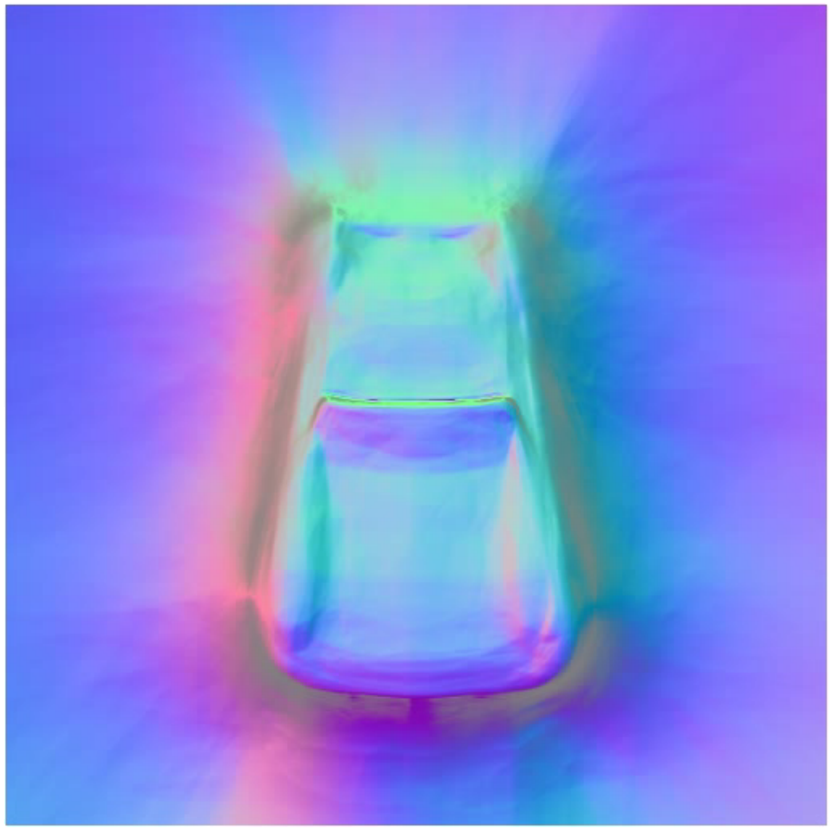
Surface Normals

RGB

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

Sampling at arbitrary resolutions





Surface Normals

RGB

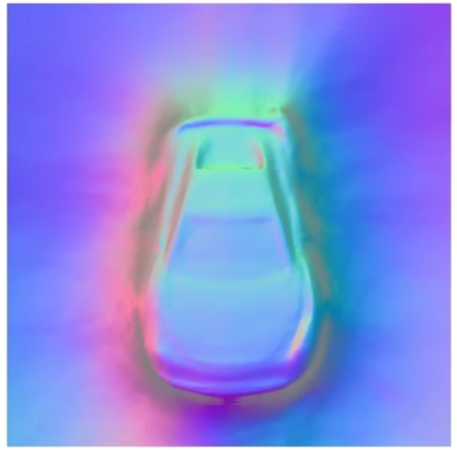
Scene Representation Networks: Continuous 3D-structure Aware Neural Scene Representations, Sitzmann et al., NeurIPS

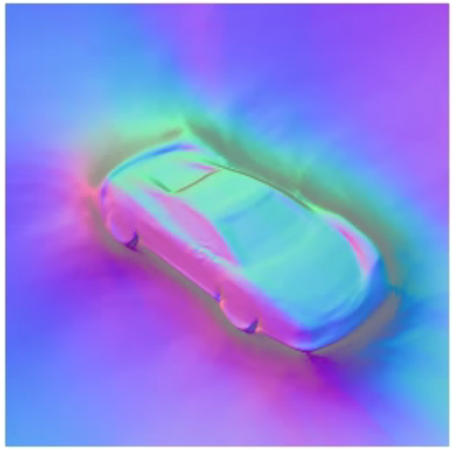
Generalization to unseen camera poses

Camera close-up

Camera Roll

SRNs





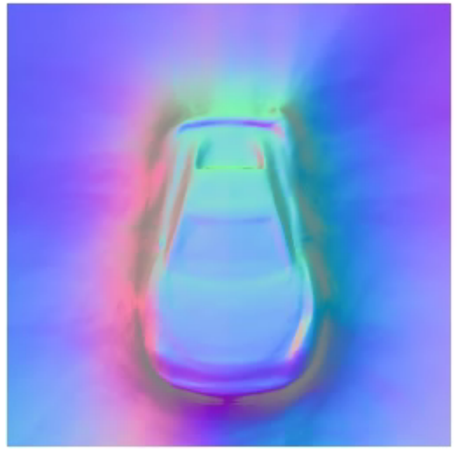
Scene Representation Networks: Continuous 3D-structure Aware Neural Scene Representations, Sitzmann et al., NeurIPS

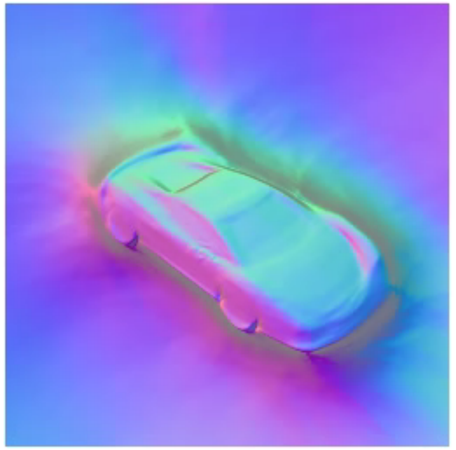
Generalization to unseen camera poses

Camera close-up

Camera Roll

SRNs





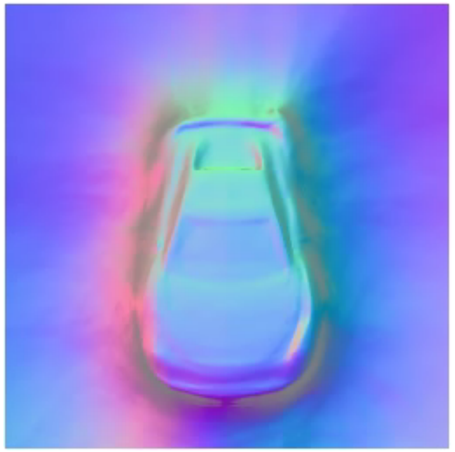
Scene Representation Networks: Continuous 3D-structure Aware Neural Scene Representations, Sitzmann et al., NeurIPS

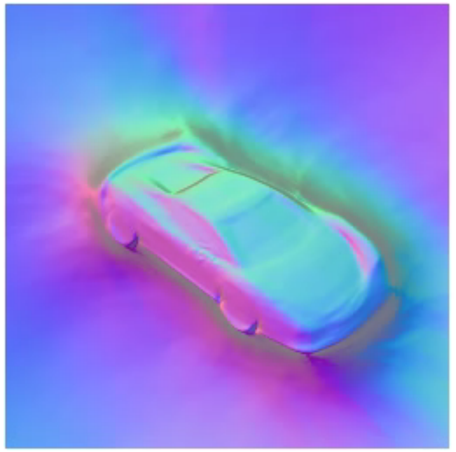
Generalization to unseen camera poses

Camera close-up

Camera Roll

SRNs





Tatarchenko et al.

Doesn’t reconstruct geometry

Doesn’t reconstruct geometry

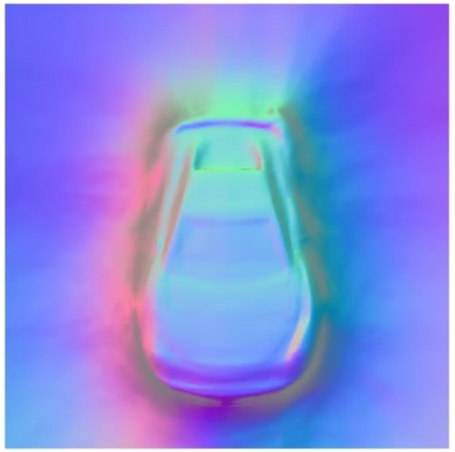
Scene Representation Networks: Continuous 3D-structure Aware Neural Scene Representations, Sitzmann et al., NeurIPS

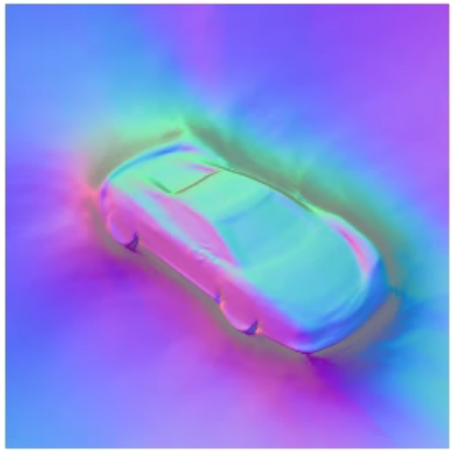
Generalization to unseen camera poses

Camera close-up

Camera Roll

SRNs





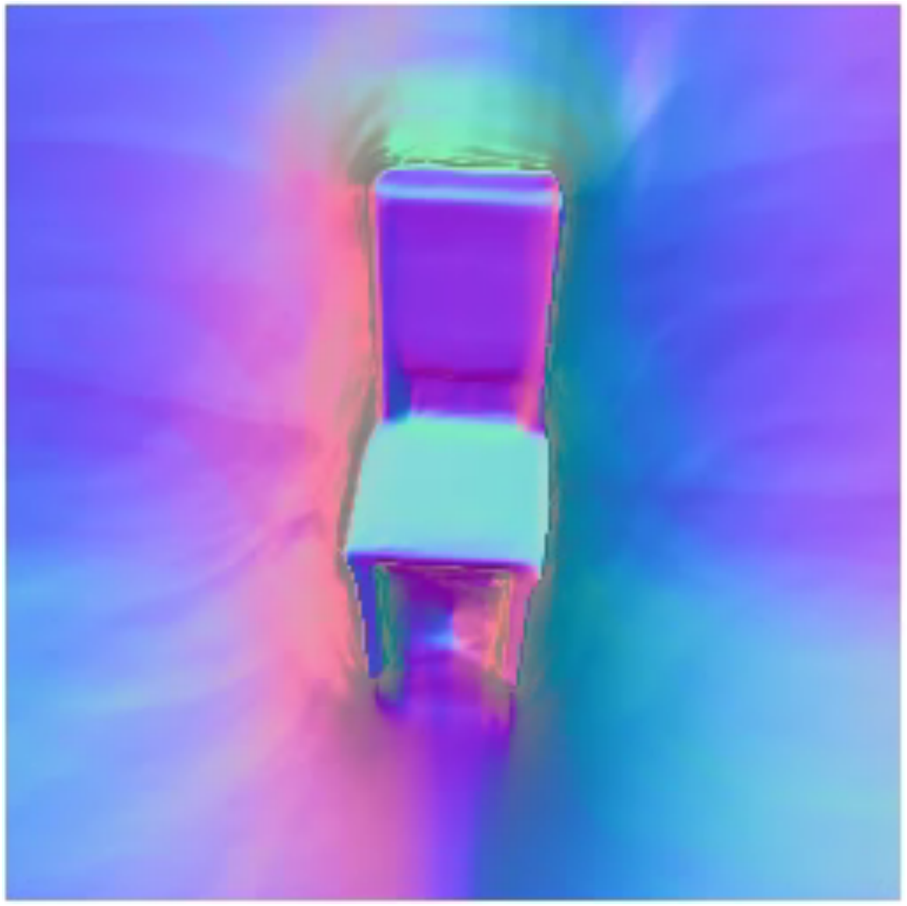
Tatarchenko et al.

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

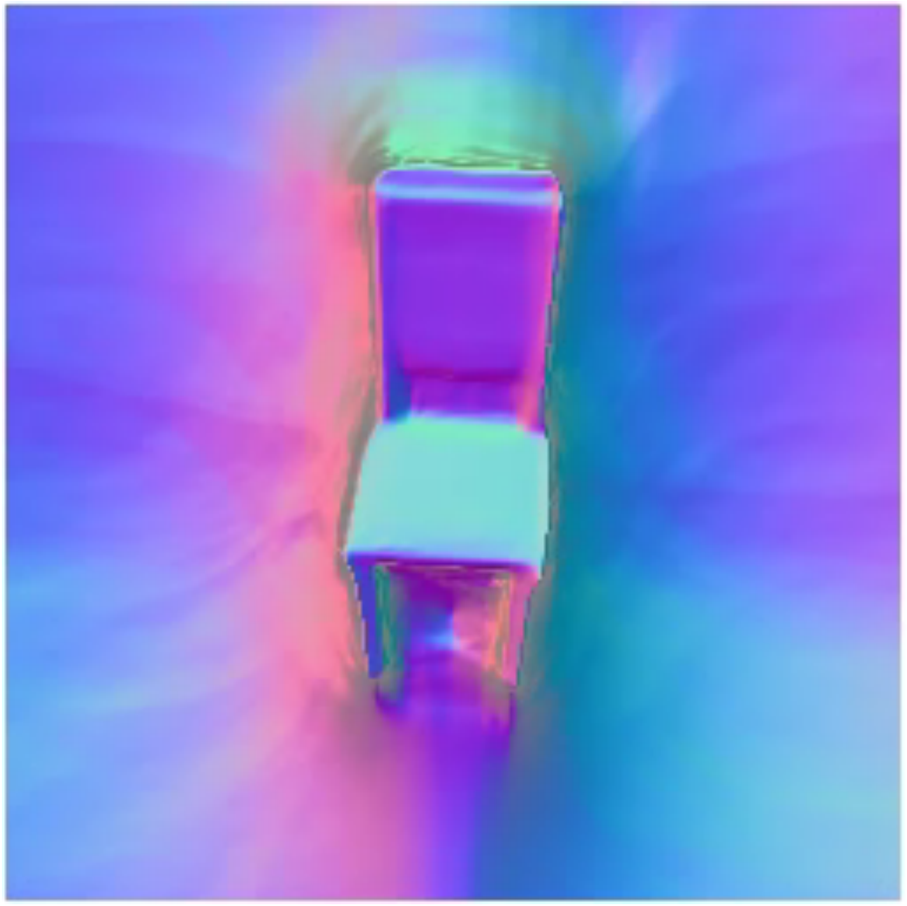


Surface Normals

RGB

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

Latent code interpolation

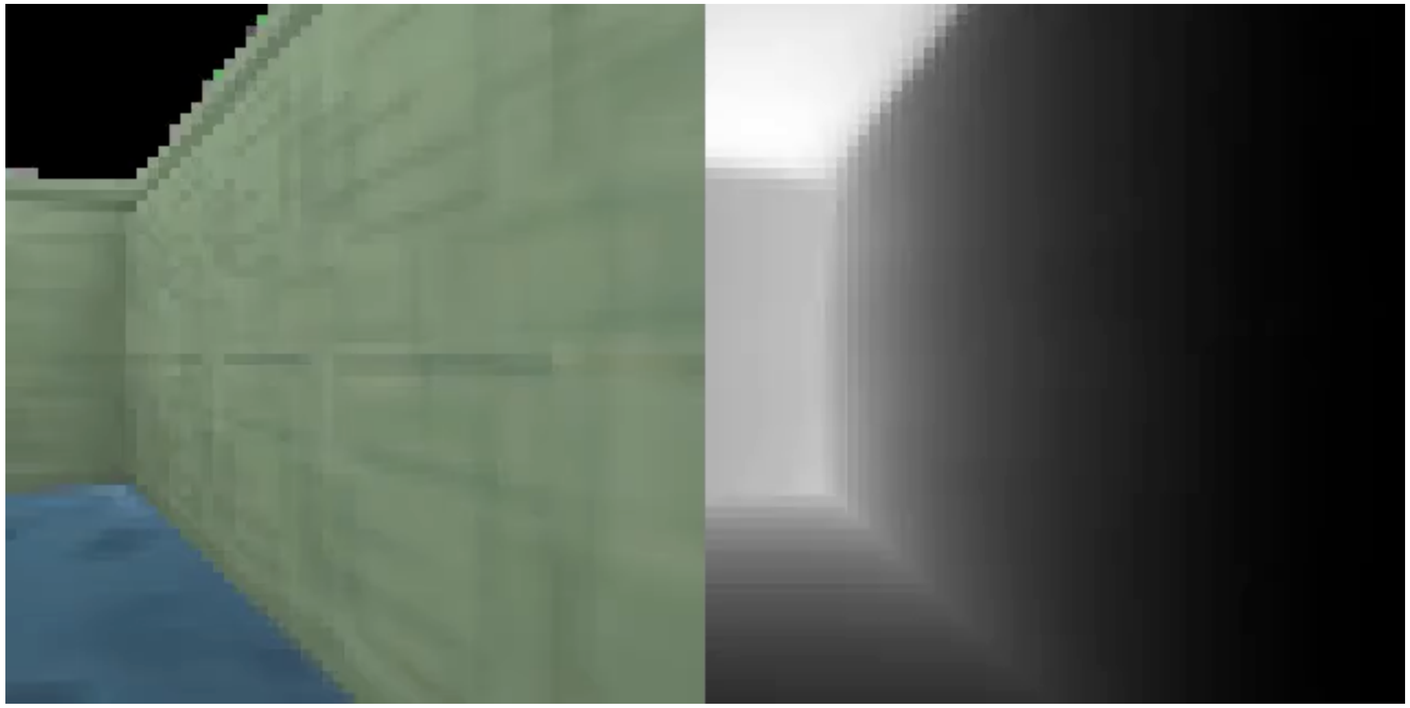


Surface Normals

RGB

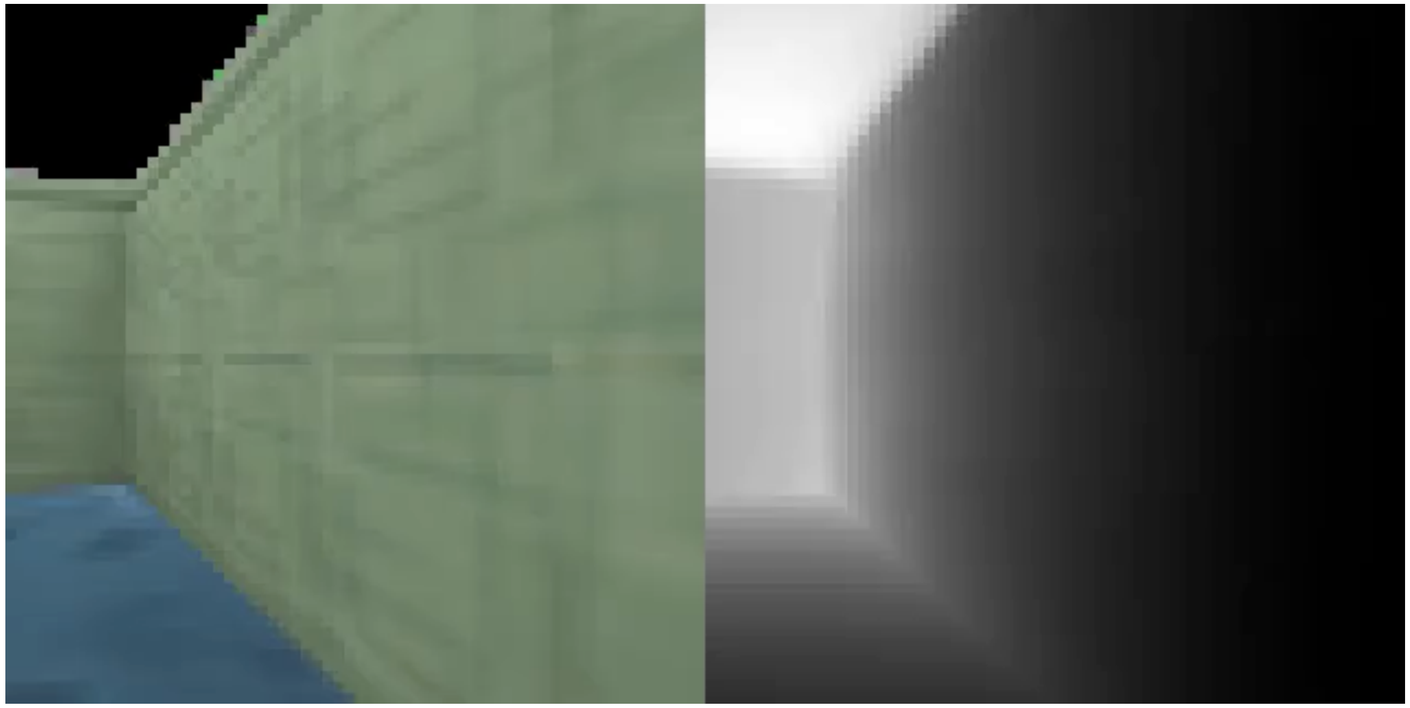
Scene Representation Networks: Continuous 3D-structure Aware Neural Scene Representations, Sitzmann et al., NeurIPS

Can generalize across simple room-scale scenes.



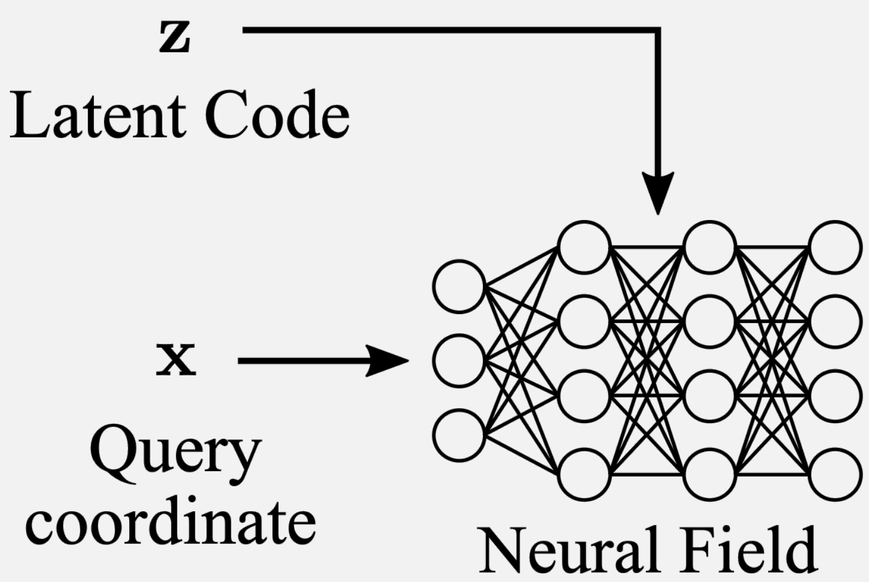
Scene Representation Networks: Continuous 3D-structure Aware Neural Scene Representations, Sitzmann et al., NeurIPS

Can generalize across simple room-scale scenes.

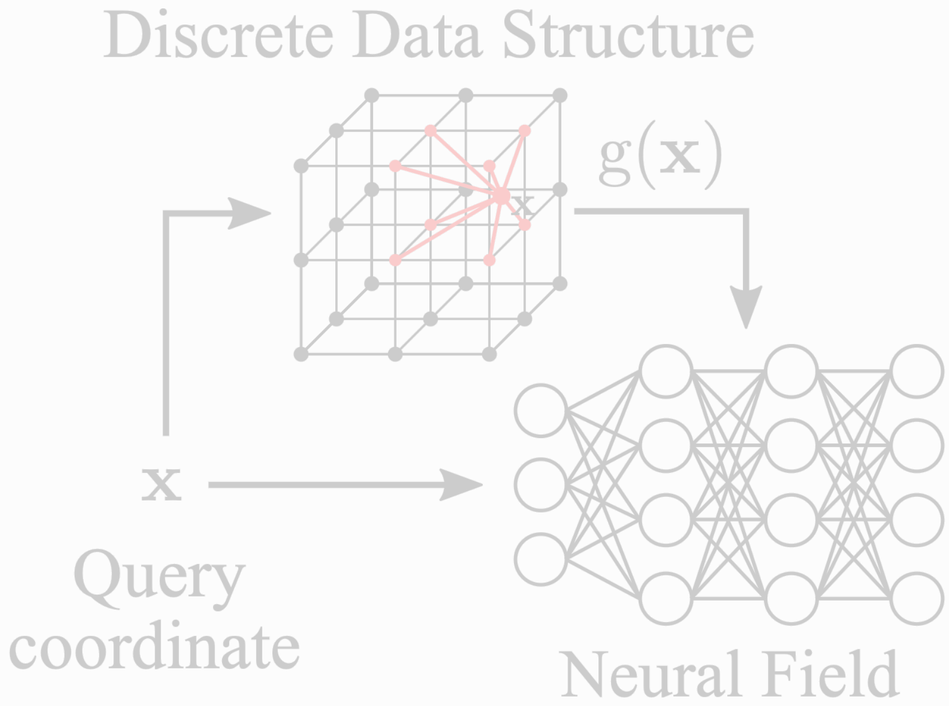


Scene Representation Networks: Continuous 3D-structure Aware Neural Scene Representations, Sitzmann et al., NeurIPS

Global Conditioning: Single Latent Code for whole 3D Scene



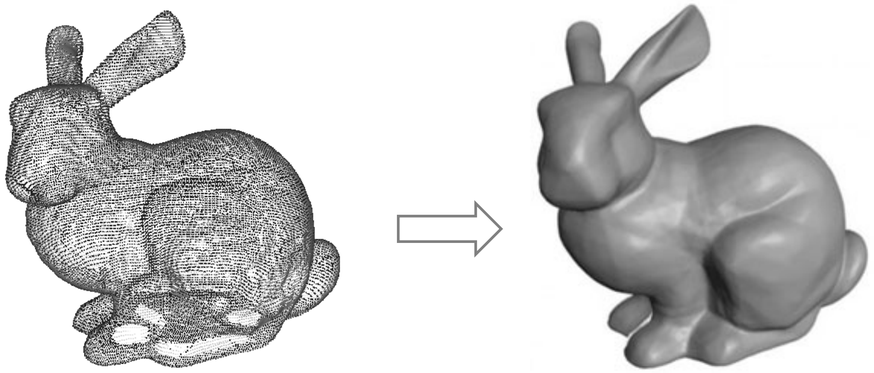
Global Conditioning



Local Conditioning

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

Global Latent Codes: Enables reconstruction from partial observations!

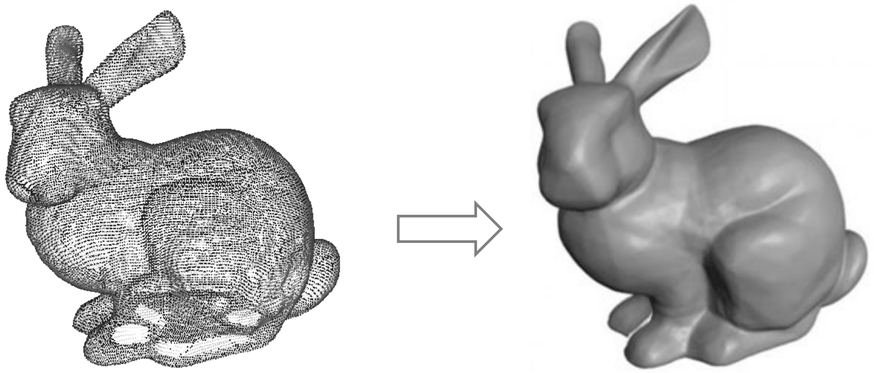


DeepSDF, Occupancy Networks, IM-Net

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!

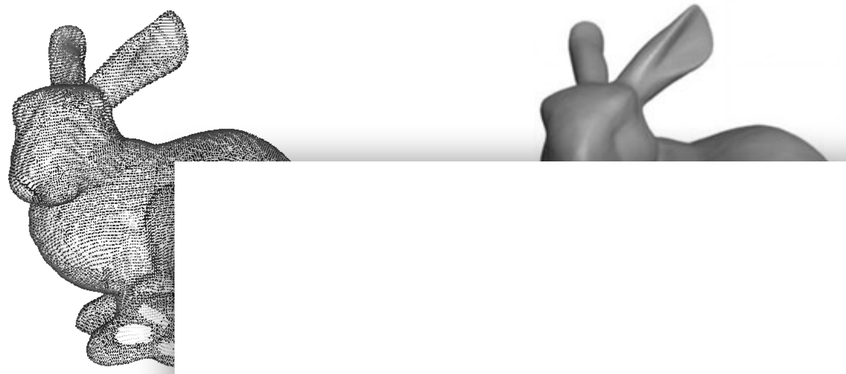


DeepSDF, Occupancy Networks, IM-Net

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

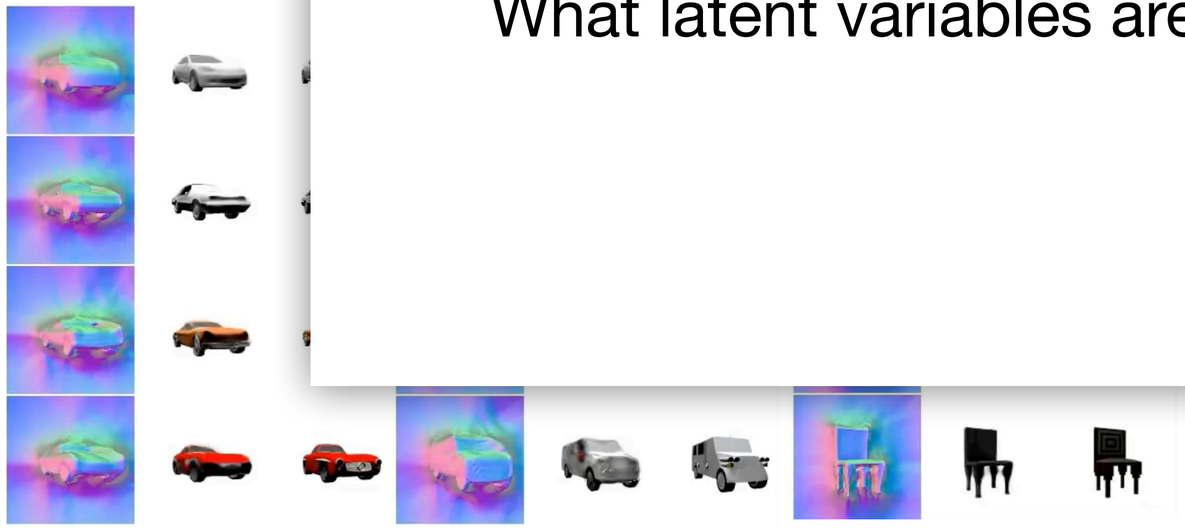
Differential Volumetric Rendering,
Niemeyer et al., CVPR 2020

Global Latent Codes: Enables reconstruction from partial observations!



Dee

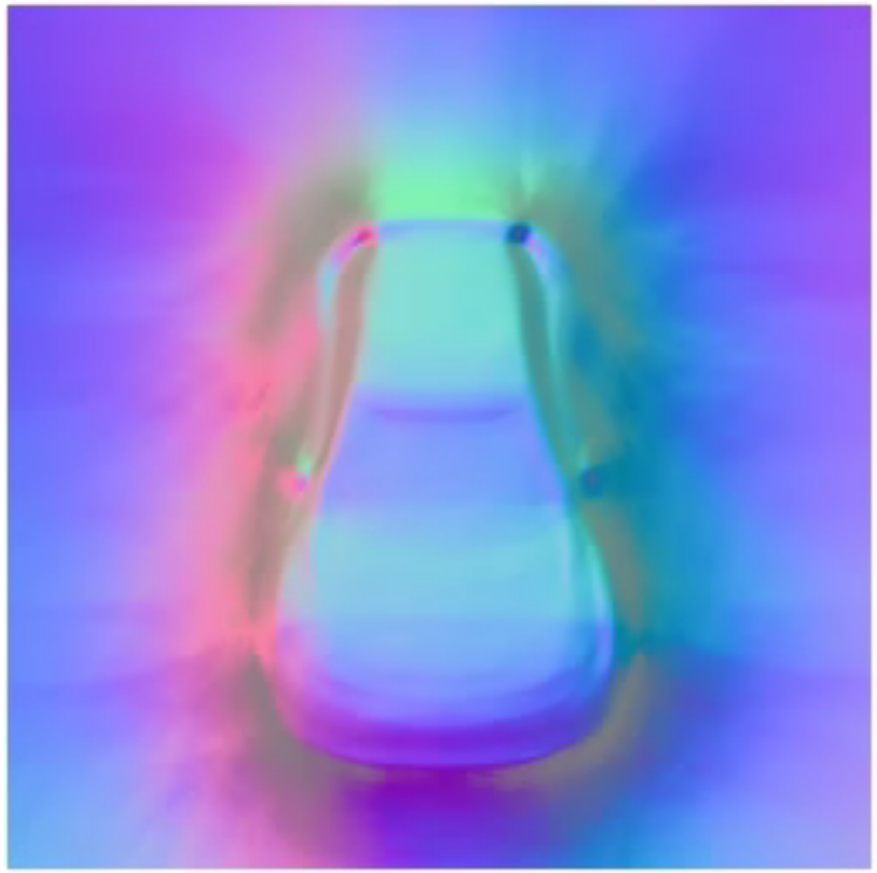
What latent variables are discovered by the model?



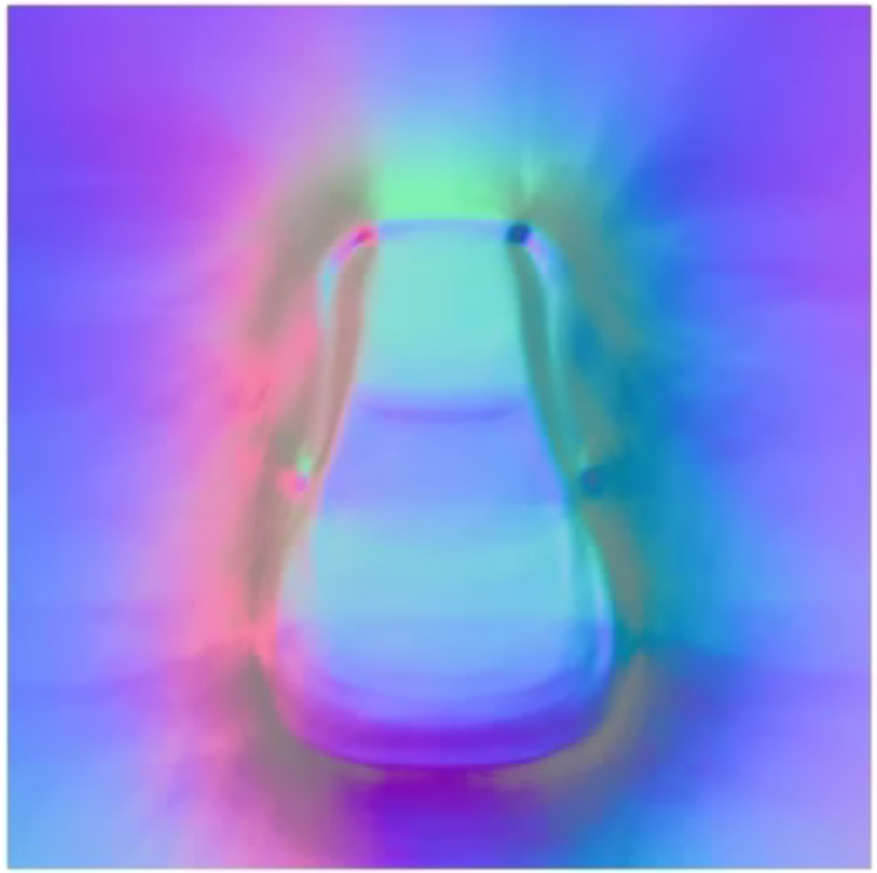
Scene Representation Networks: Continuous 3D-Structure-Aware Neural Scene Representations, NeurIPS 2019.

Differential Volumetric Rendering, Niemeyer et al., CVPR 2020

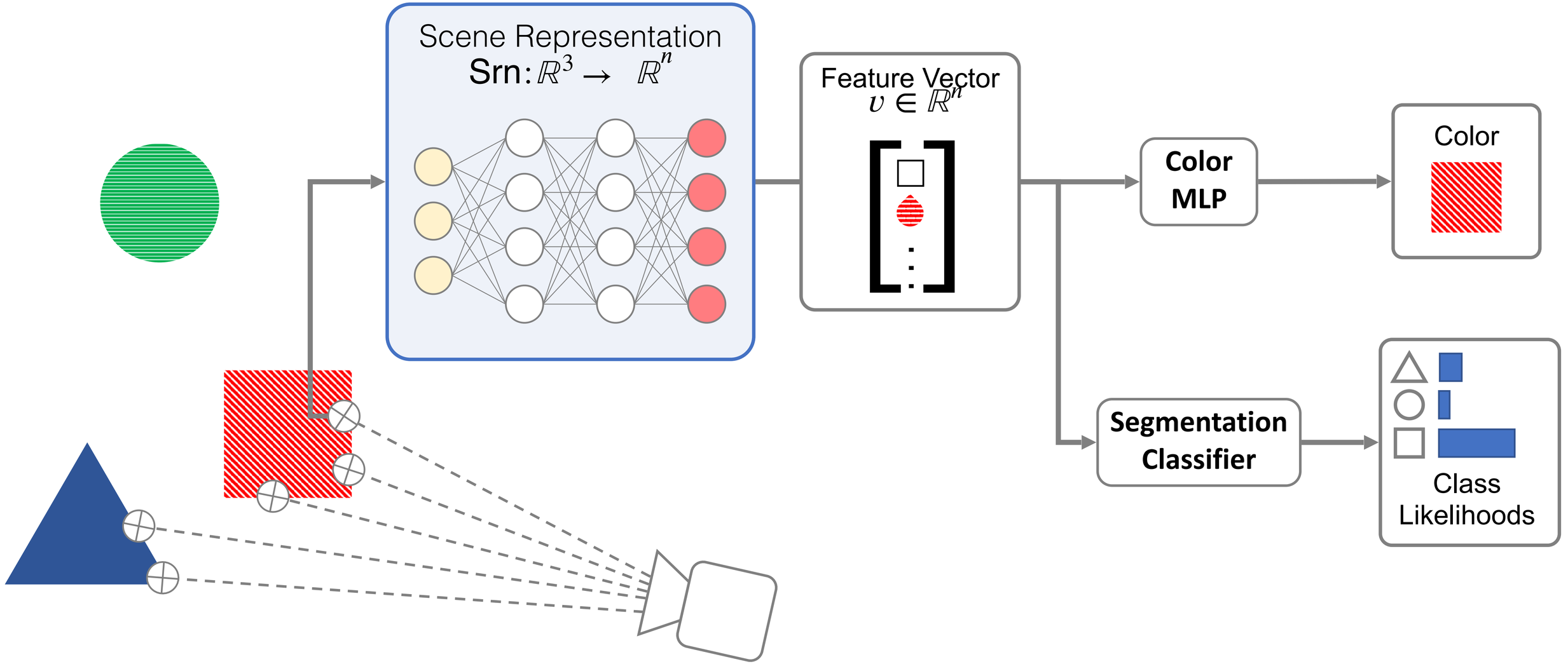
Latent Space Interpolation



Latent Space Interpolation

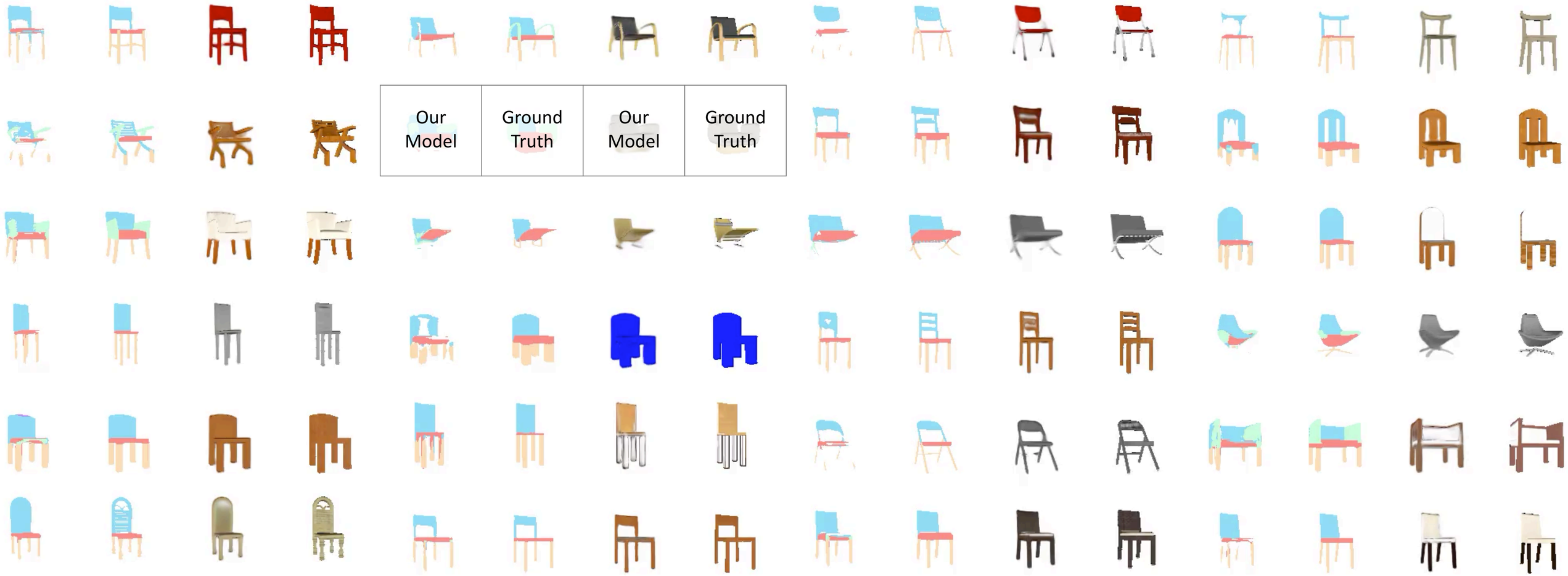


Learned features enable semantic segmentation from few labels.



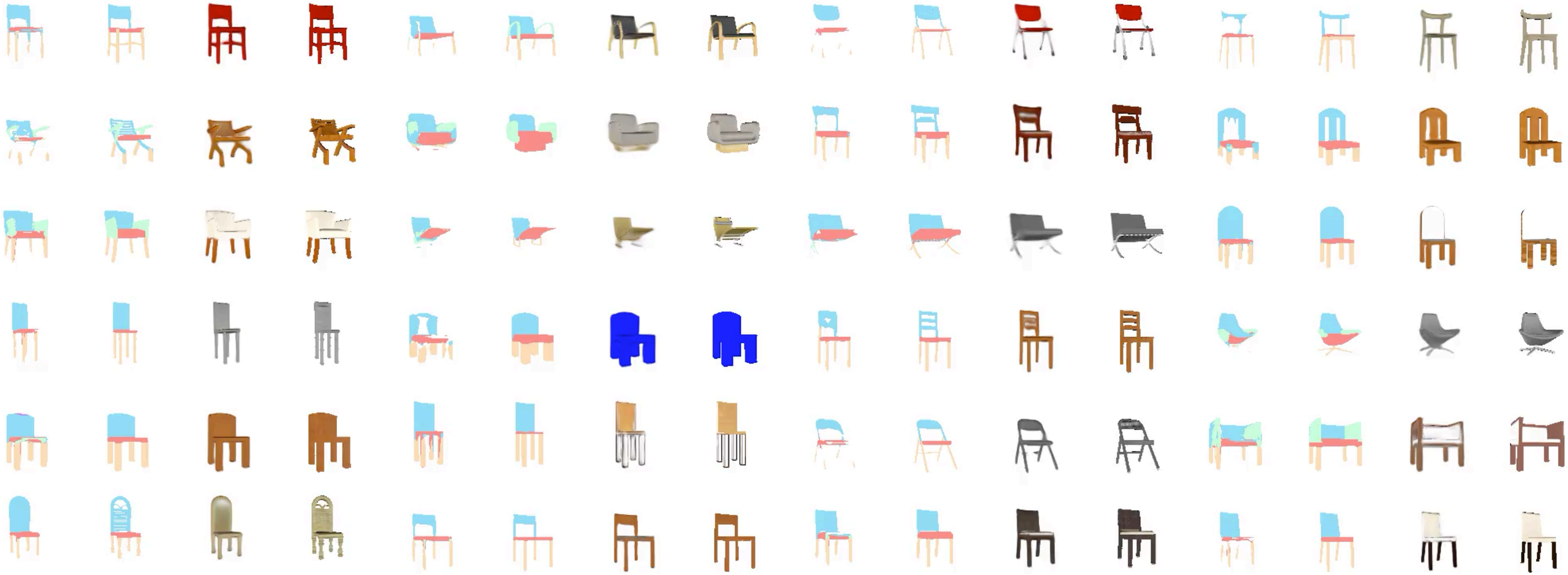
3D Representation Learning for Semantic Segmentation, Kohli et al., 3DV, 2020

Learned features enable semantic segmentation from few labels.



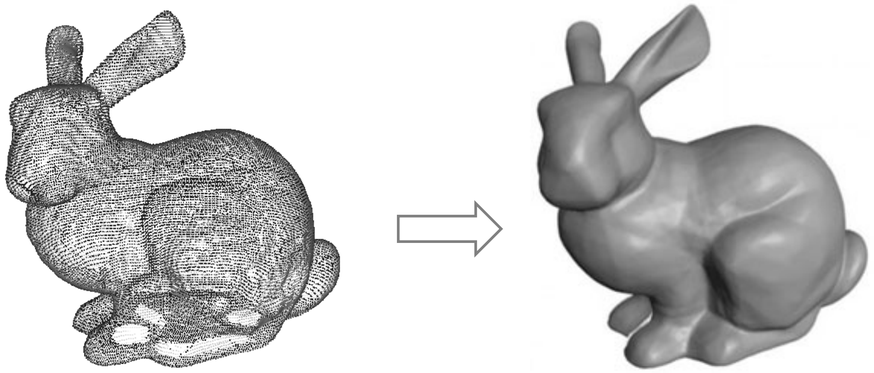
Kohli et al., 3DV, 2020

Learned features enable semantic segmentation from few labels.



Kohli et al., 3DV, 2020

Global Latent Codes: Enables reconstruction from partial observations!

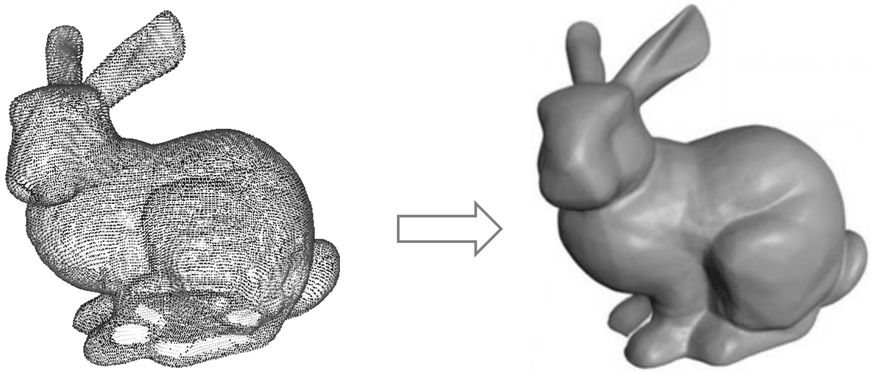


DeepSDF, Occupancy Networks, IM-Net

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!

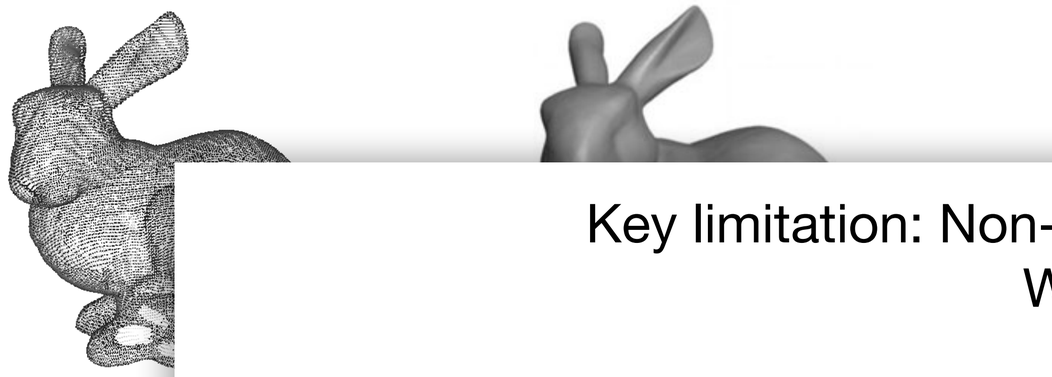


DeepSDF, Occupancy Networks, IM-Net

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?

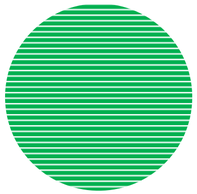
Scene Representation Networks: Continuous 3D-Structure-Aware Neural Scene Representations, NeurIPS 2019.

Differential Volumetric Rendering, Niemeyer et al., CVPR 2020

Global Latent Codes: Enables reconstruction from partial observations!

Dee

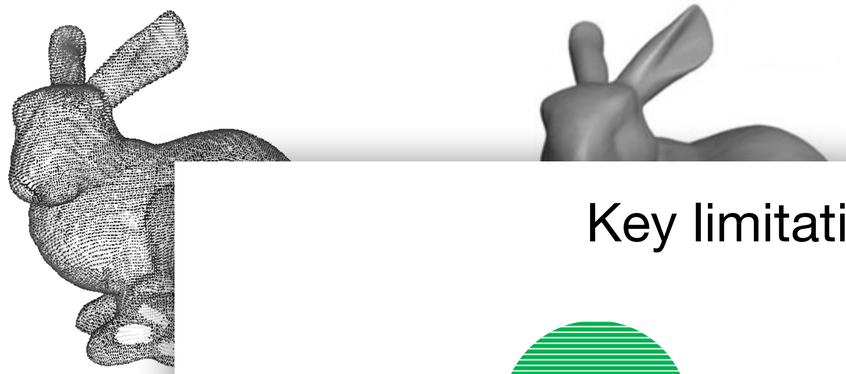
Key limitation: Non-compositional scenes.
Why?



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

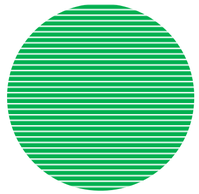
Differential Volumetric Rendering, Niemeyer et al., CVPR 2020

Global Latent Codes: Enables reconstruction from partial observations!



Dee

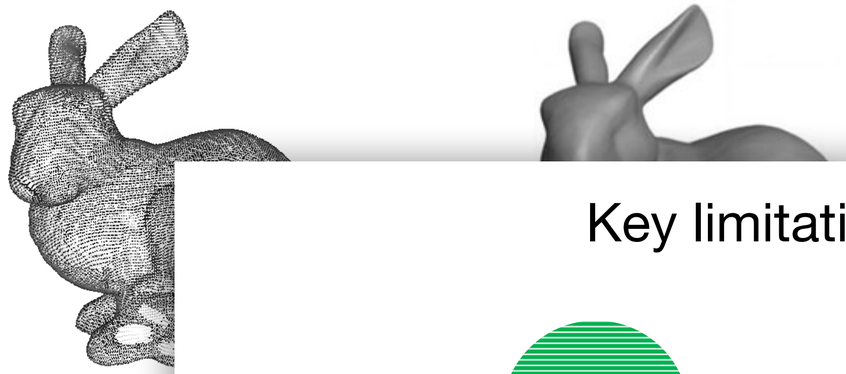
Key limitation: Non-compositional scenes.
Why?



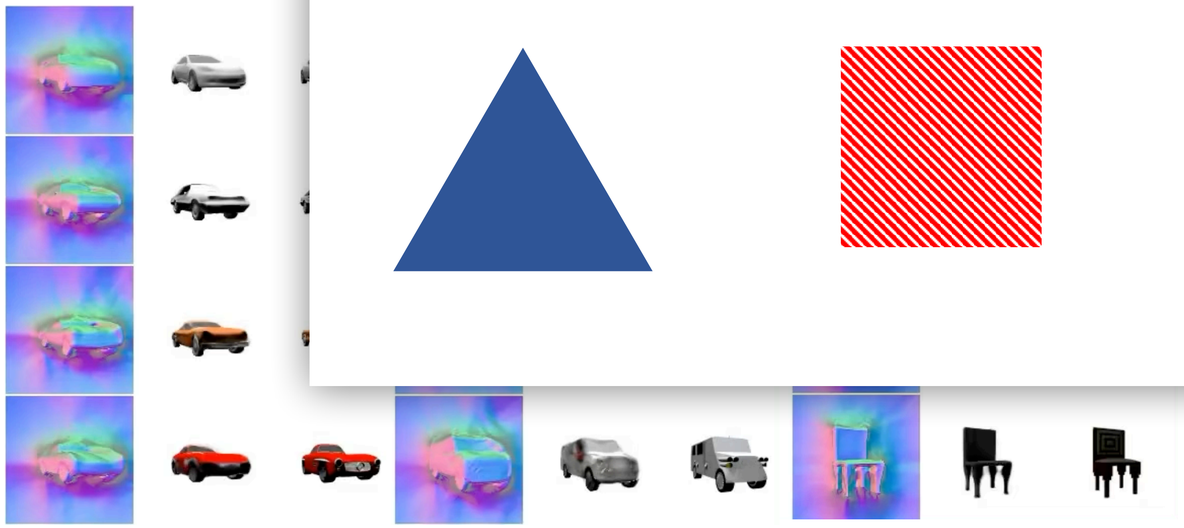
Scene Representation Networks: Continuous 3D-Structure-Aware Neural Scene Representations, NeurIPS 2019.

Differential Volumetric Rendering, Niemeyer et al., CVPR 2020

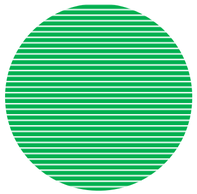
Global Latent Codes: Enables reconstruction from partial observations!



Dee



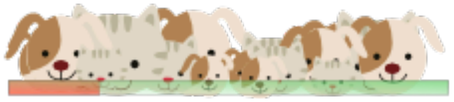
Key limitation: Non-compositional scenes.
Why?



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

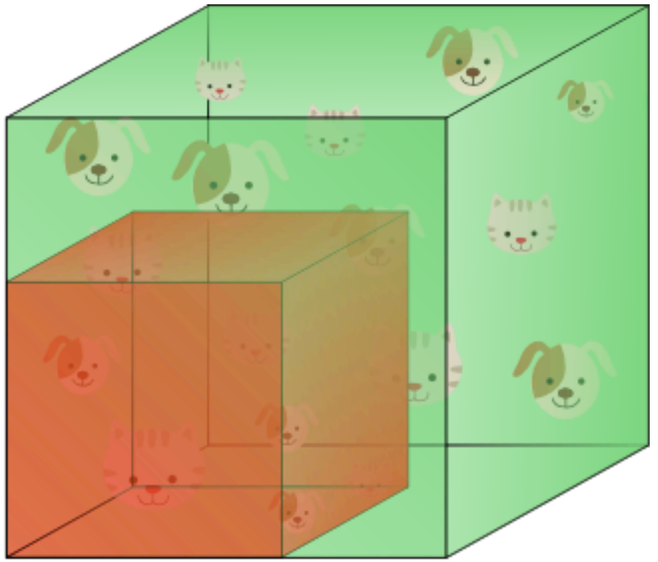
Differential Volumetric Rendering, Niemeyer et al., CVPR 2020

Reminder: Curse of Dimensionality



1D

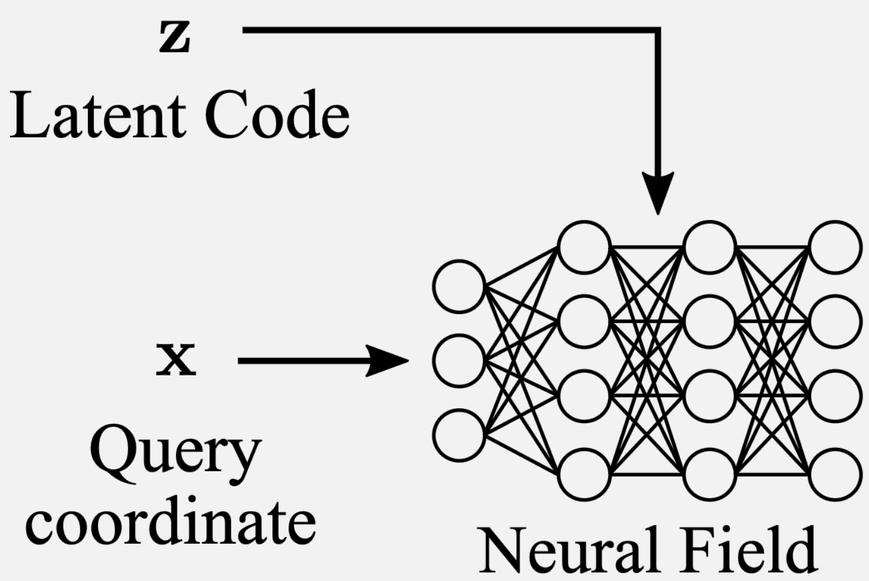
2D



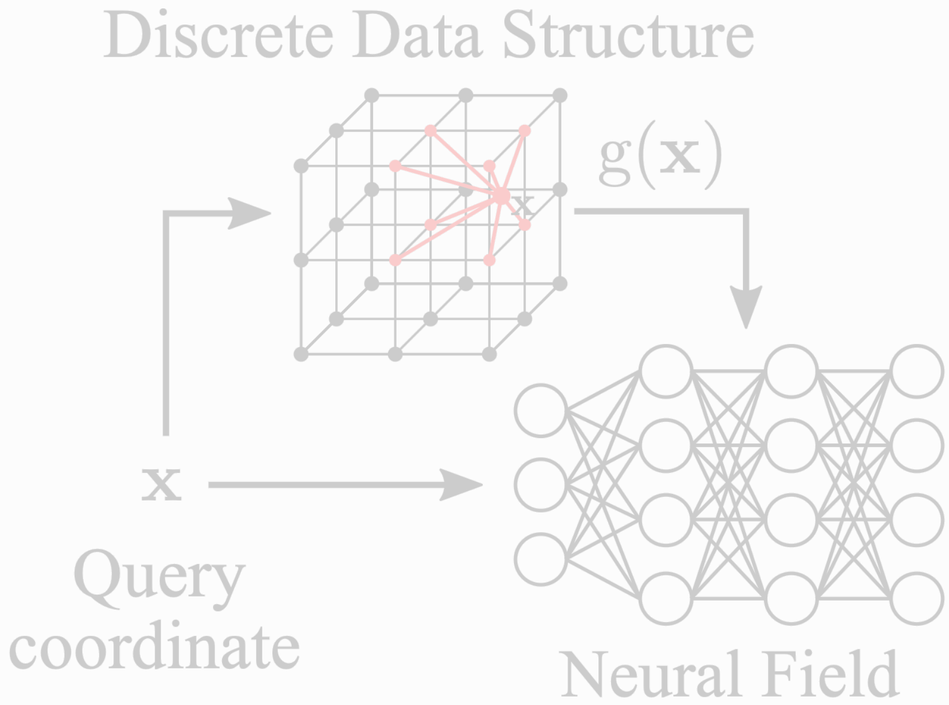
3D

...

Global Latent Codes



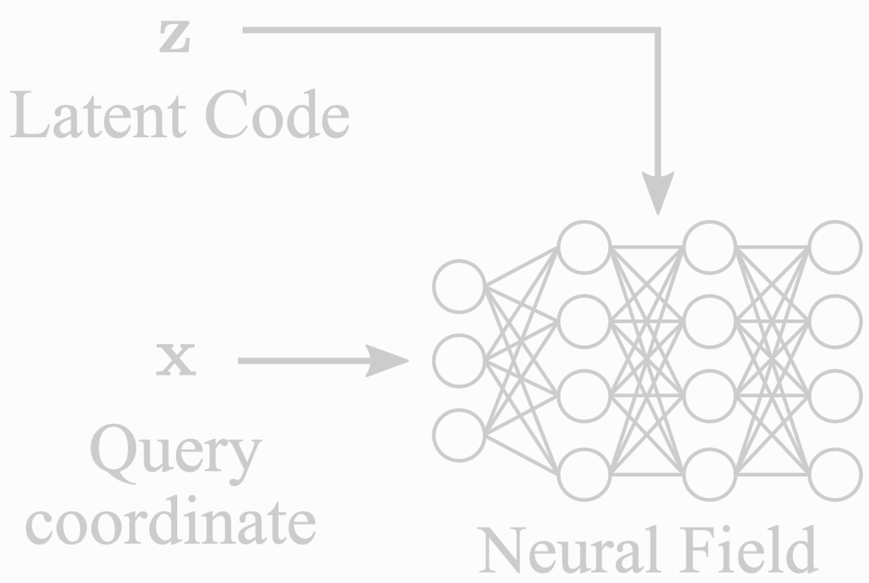
Global Conditioning



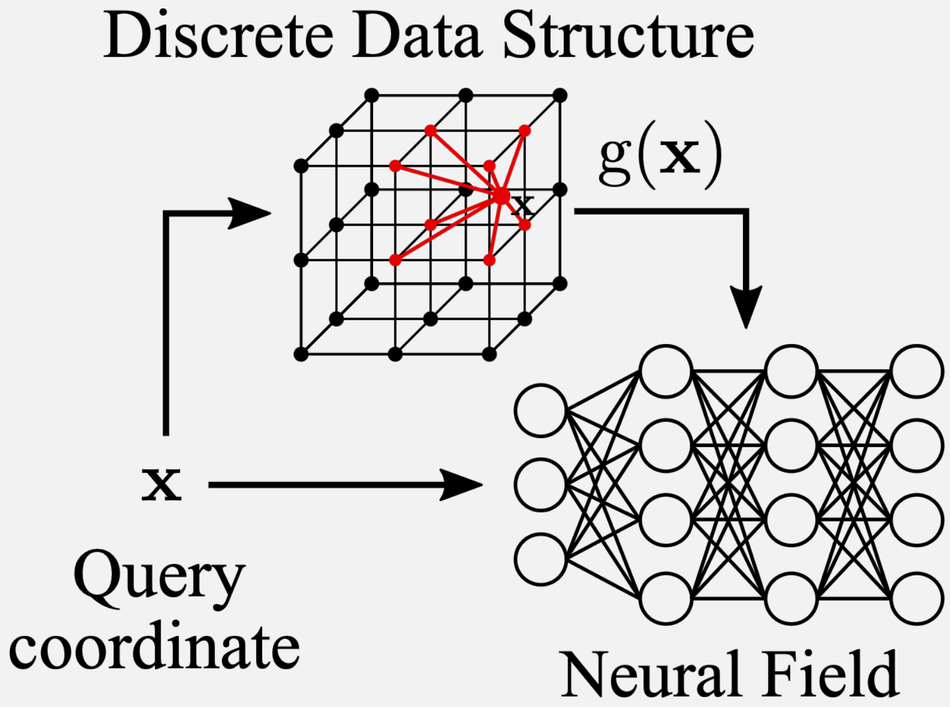
Local Conditioning

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

Local Latent Codes



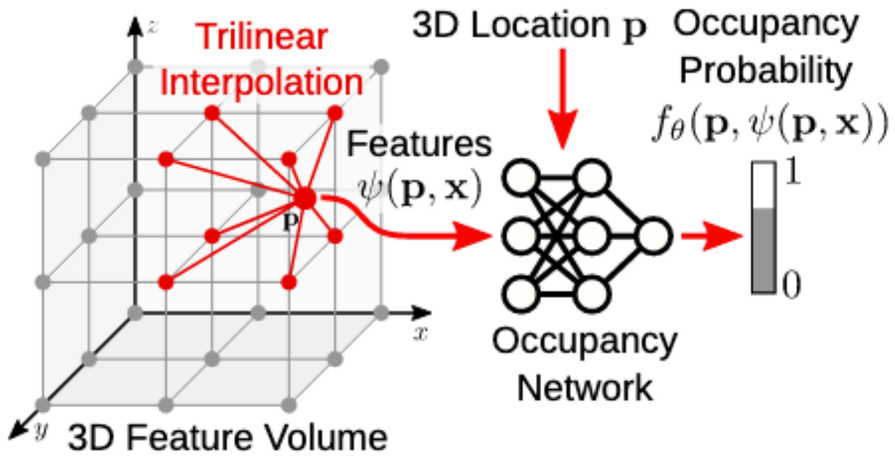
Global Conditioning



Local Conditioning

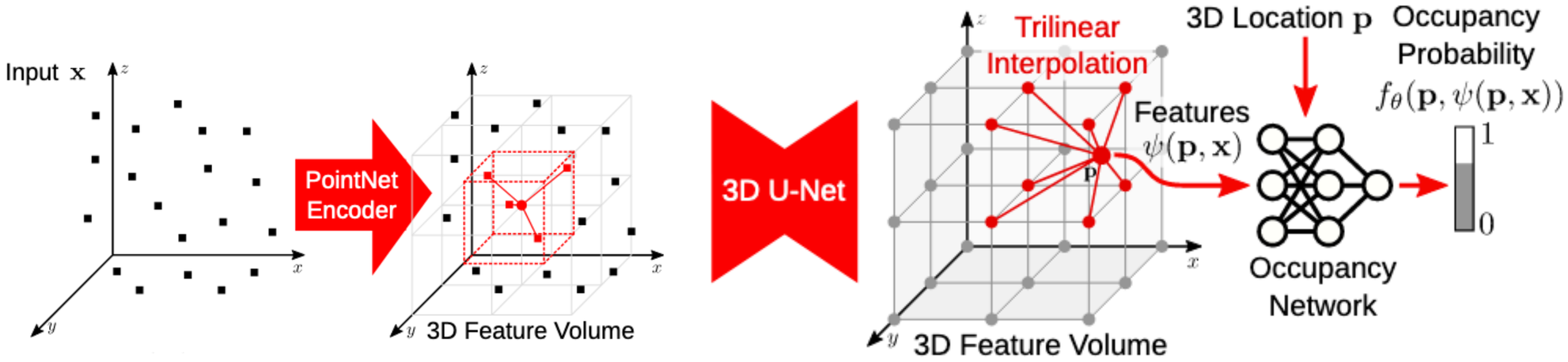
Neural Fields in Visual Computing and Beyond, Xie et al., EG STAR 2022

From point clouds: Conditioning on Feature Voxel grids



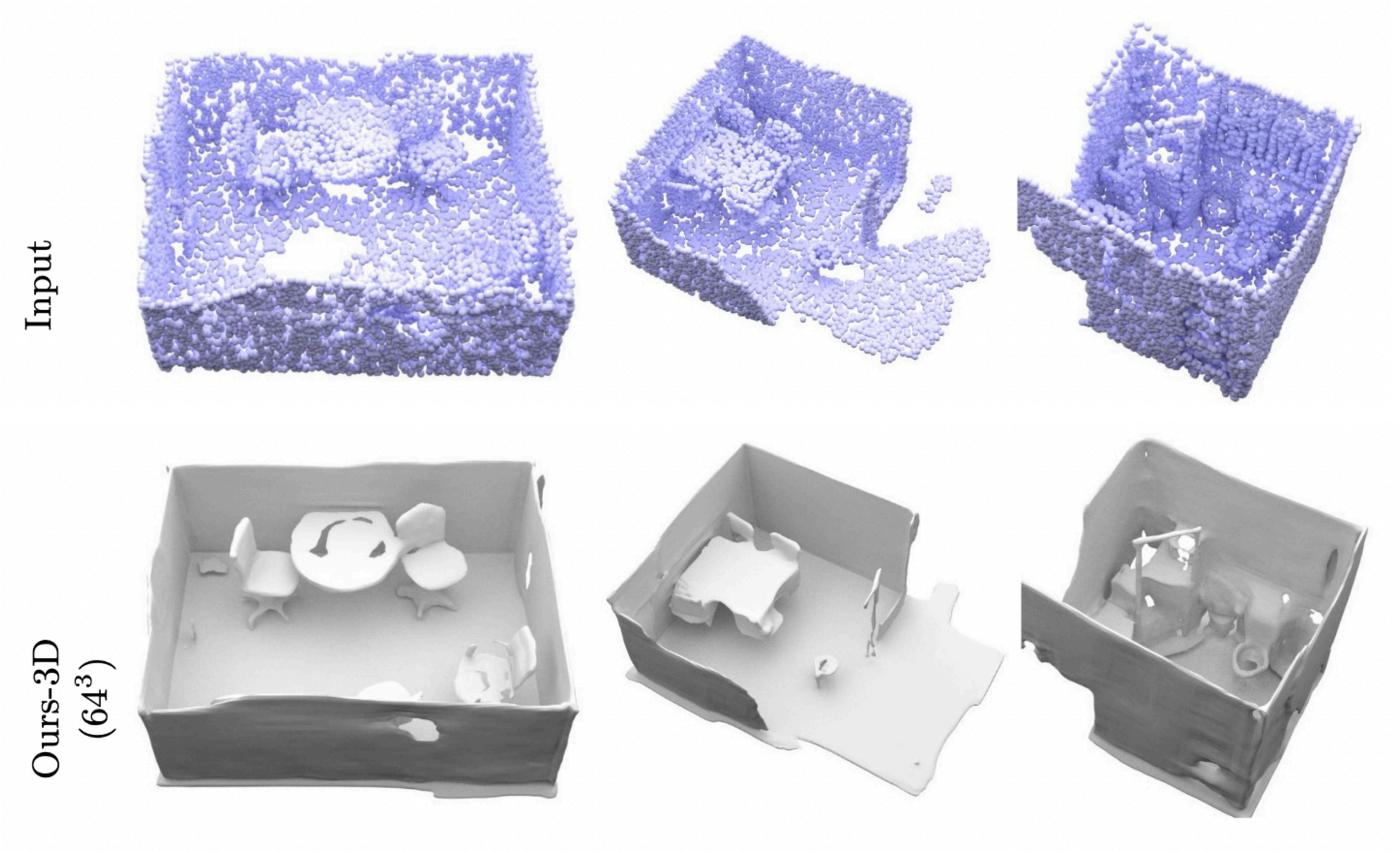
Convolutional Occupancy Networks [Peng et al. 2020]
Local Implicit Grid Representations for 3D Scenes [Jiang et al. 2020]
Implicit Functions in Feature Space for 3D Shape Reconstruction and Completion [Chabra et al. 2020]
Deep Local Shapes: Learning Local SDF Priors for Detailed 3D Reconstruction [Chibane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

From point clouds: Conditioning on Feature Voxel grids

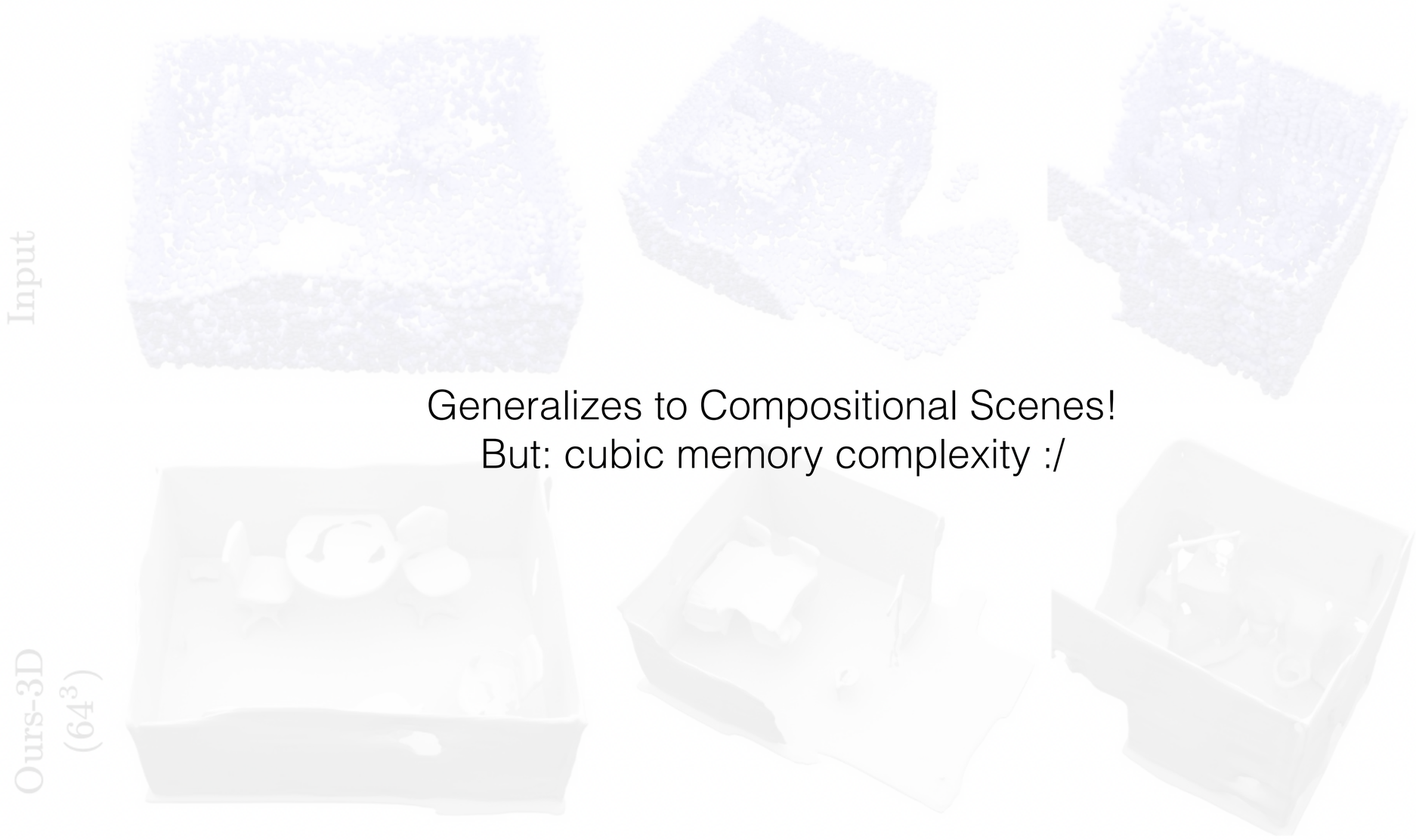


Convolutional Occupancy Networks [Peng et al. 2020]
Local Implicit Grid Representations for 3D Scenes [Jiang et al. 2020]
Implicit Functions in Feature Space for 3D Shape Reconstruction and Completion [Chabra et al. 2020]
Deep Local Shapes: Learning Local SDF Priors for Detailed 3D Reconstruction [Chibane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

From point clouds: Conditioning on Feature Voxel grids

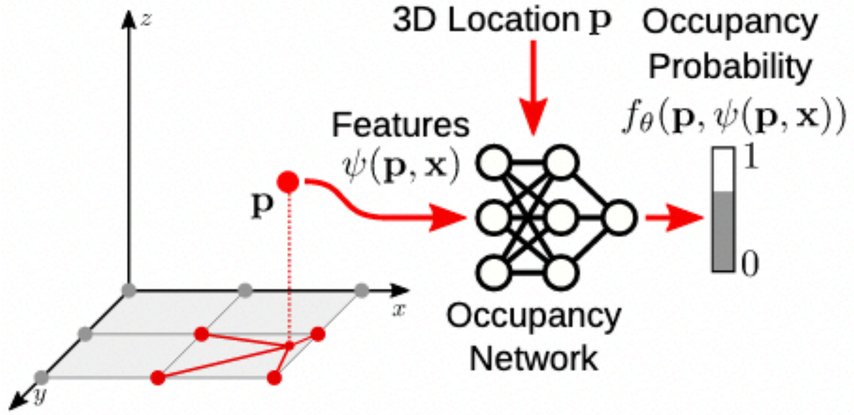


From point clouds: Conditioning on Feature Voxel grids



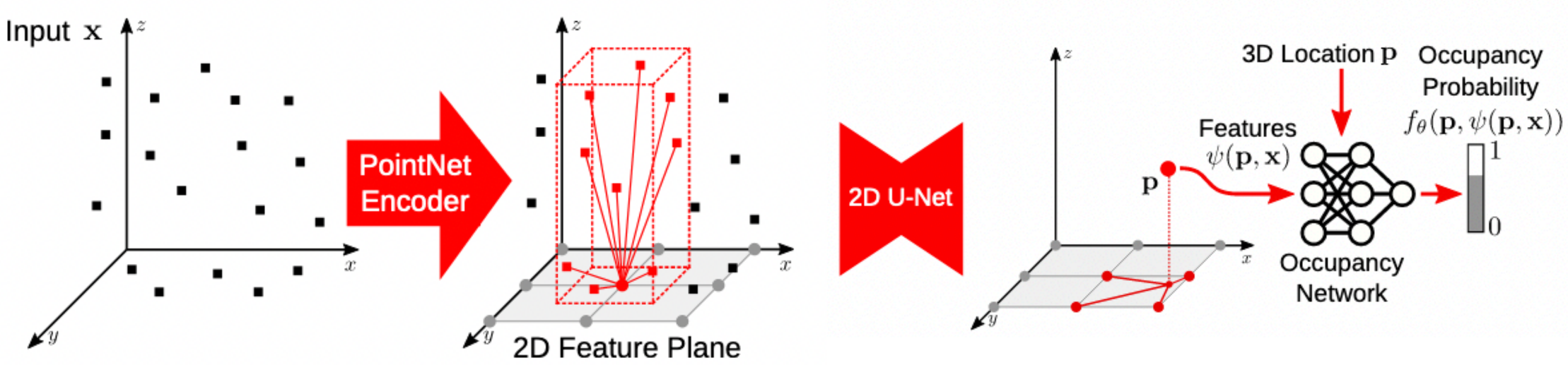
Generalizes to Compositional Scenes!
But: cubic memory complexity :/

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



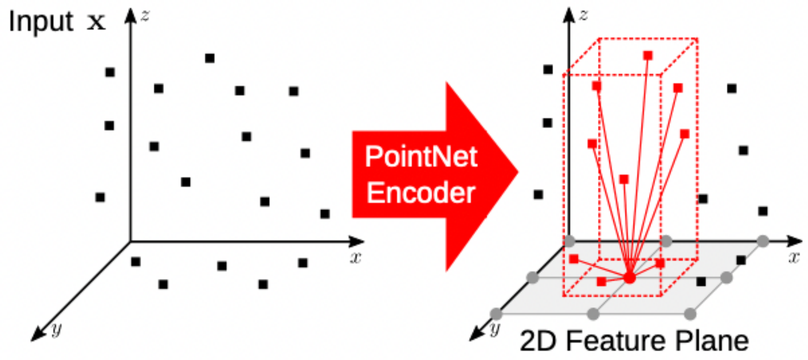
Convolutional Occupancy Networks [Peng et al. 2020]

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



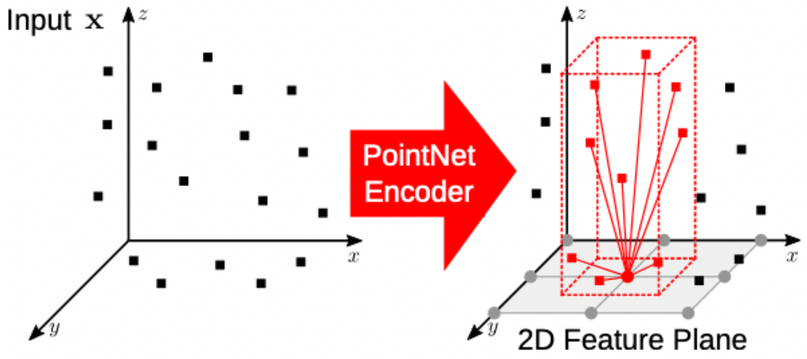
Convolutional Occupancy Networks [Peng et al. 2020]

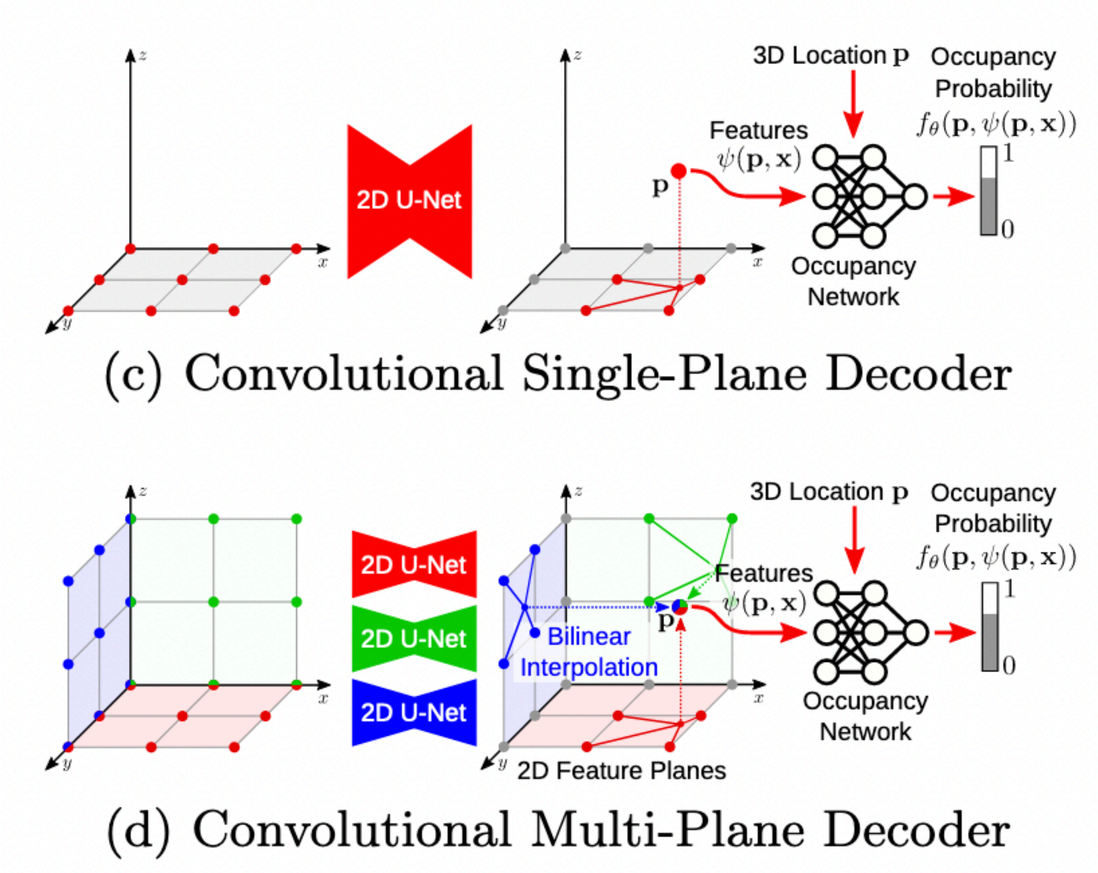
From Point clouds: Ground-plan and Tri-plane factorizations



Convolutional Occupancy Networks [Peng et al. 2020]

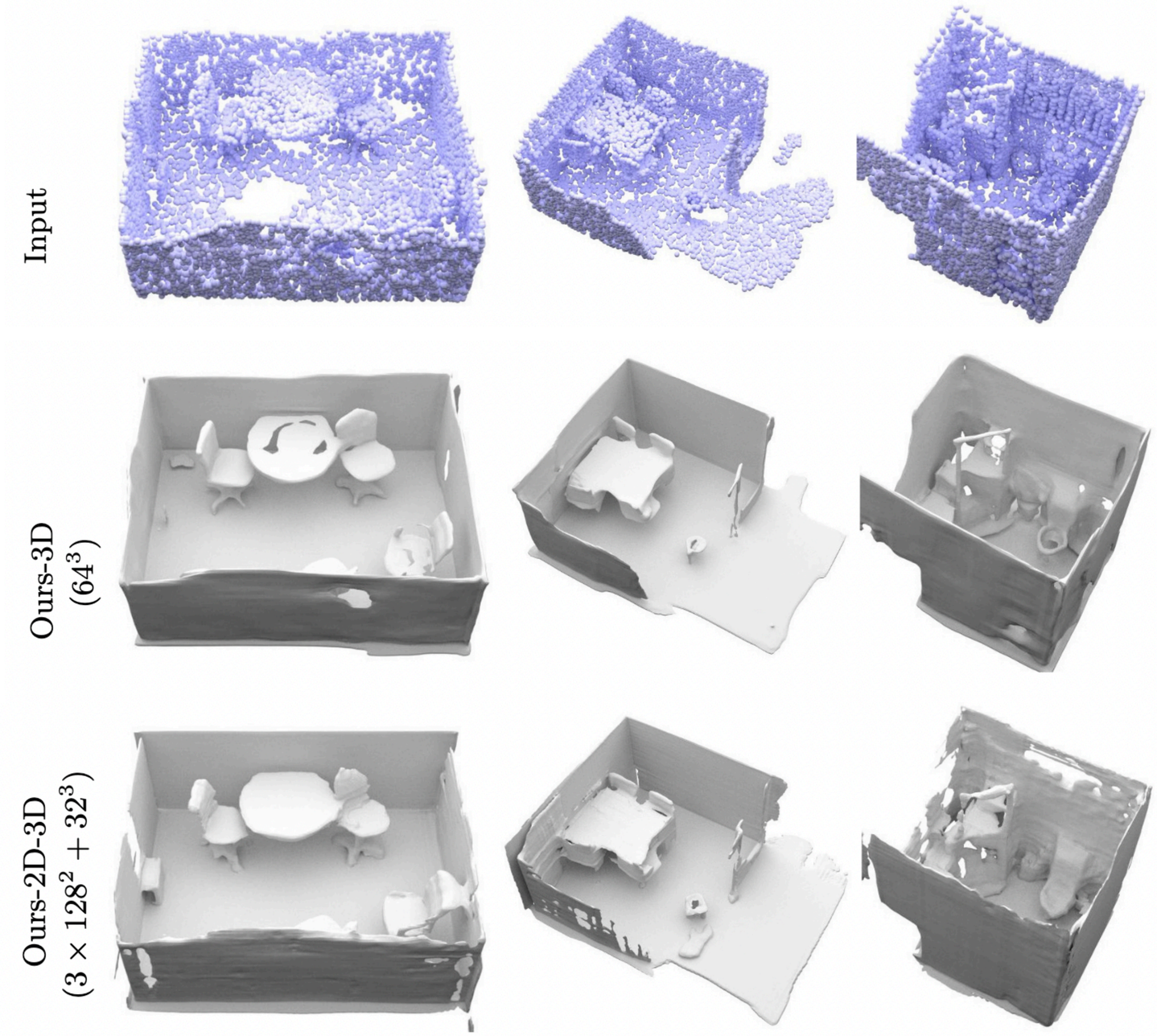
From Point clouds: Ground-plan and Tri-plane factorizations





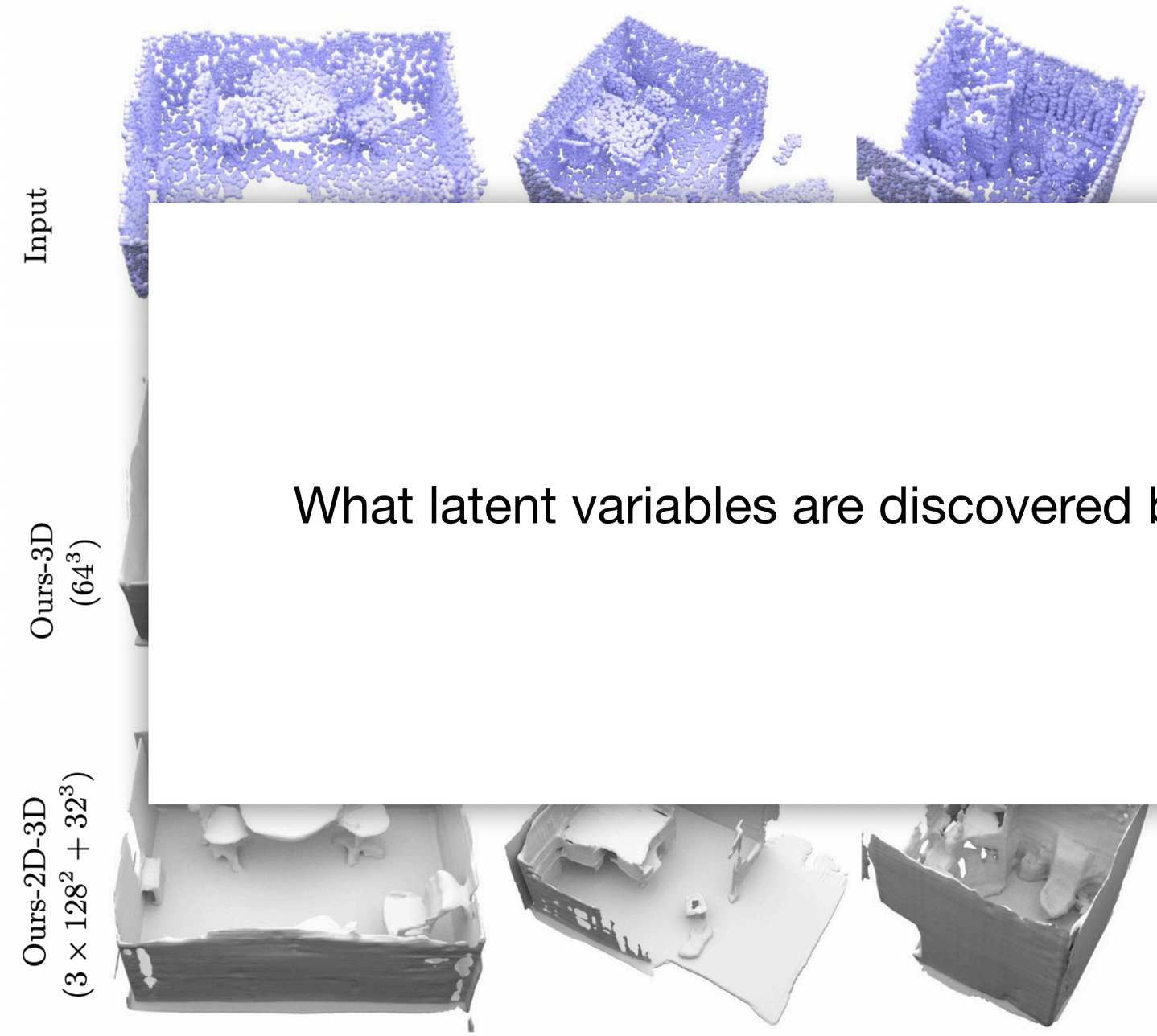
Convolutional Occupancy Networks [Peng et al. 2020]

From point clouds: Conditioning on Reconstructed Voxelgrids



5x less memory!

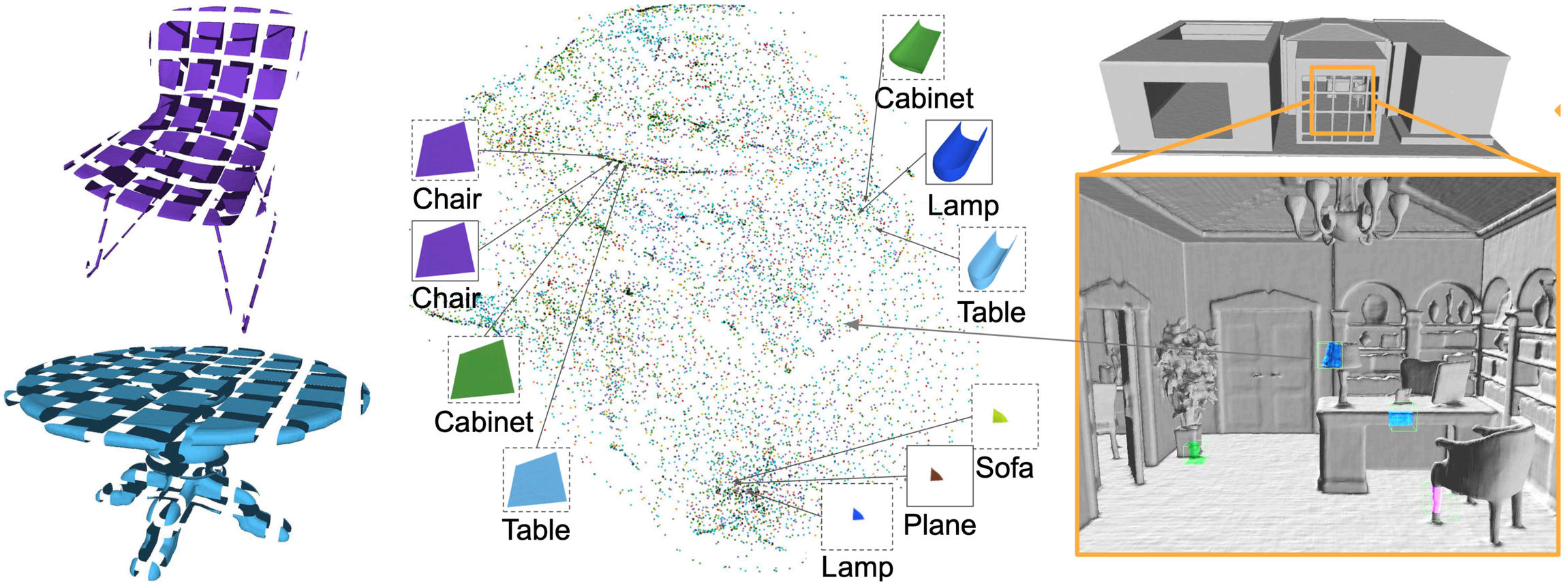
From point clouds: Conditioning on Reconstructed Voxelgrids



What latent variables are discovered by the model?

5x less memory!

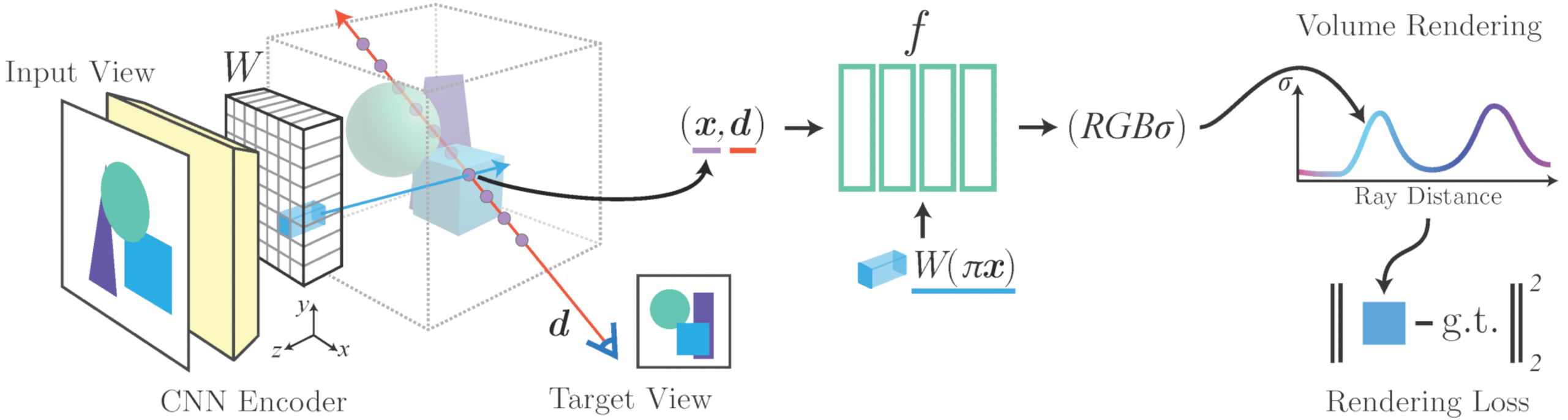
Locally Conditioned Latent Space



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

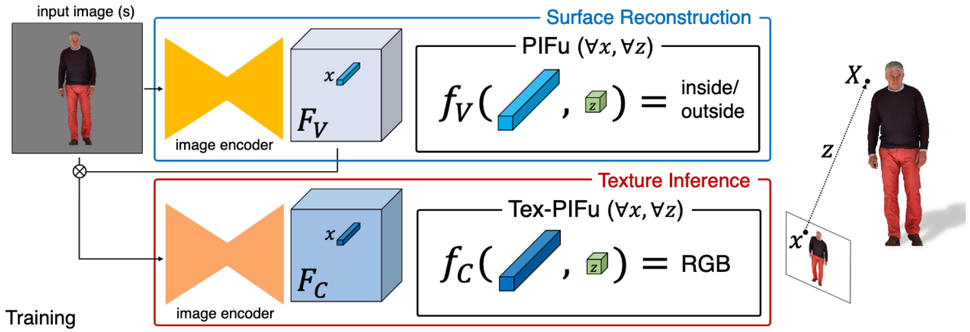
How to locally condition if sensor domain different than field domain?

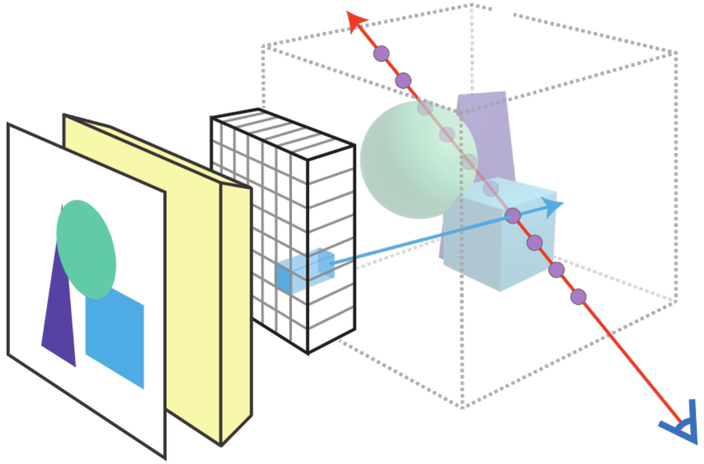
Local Conditioning: Pixel-Aligned Features.



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.

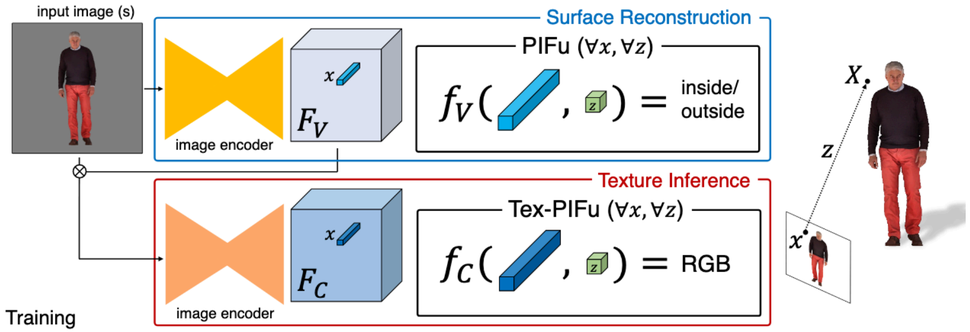


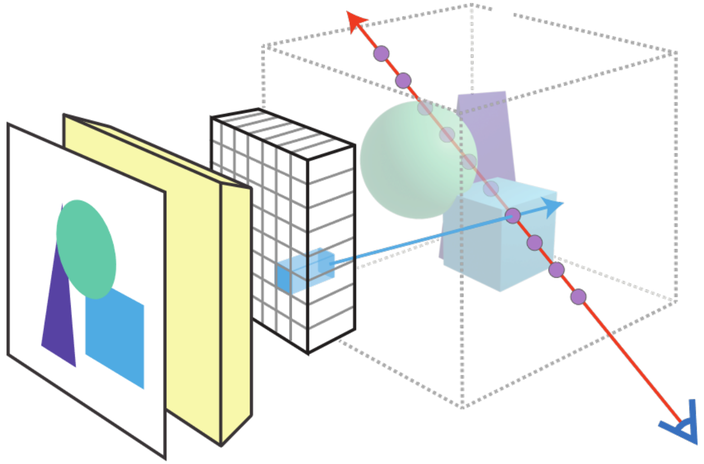


PiFU, Saito et al., ICCV 2019.
PixelNeRF, Yu et al., CVPR 2021
Grf: Learning a general radiance field…, Trevithick et al.

SRNs PixelNeRF

Local Conditioning: Pixel-Aligned Features.



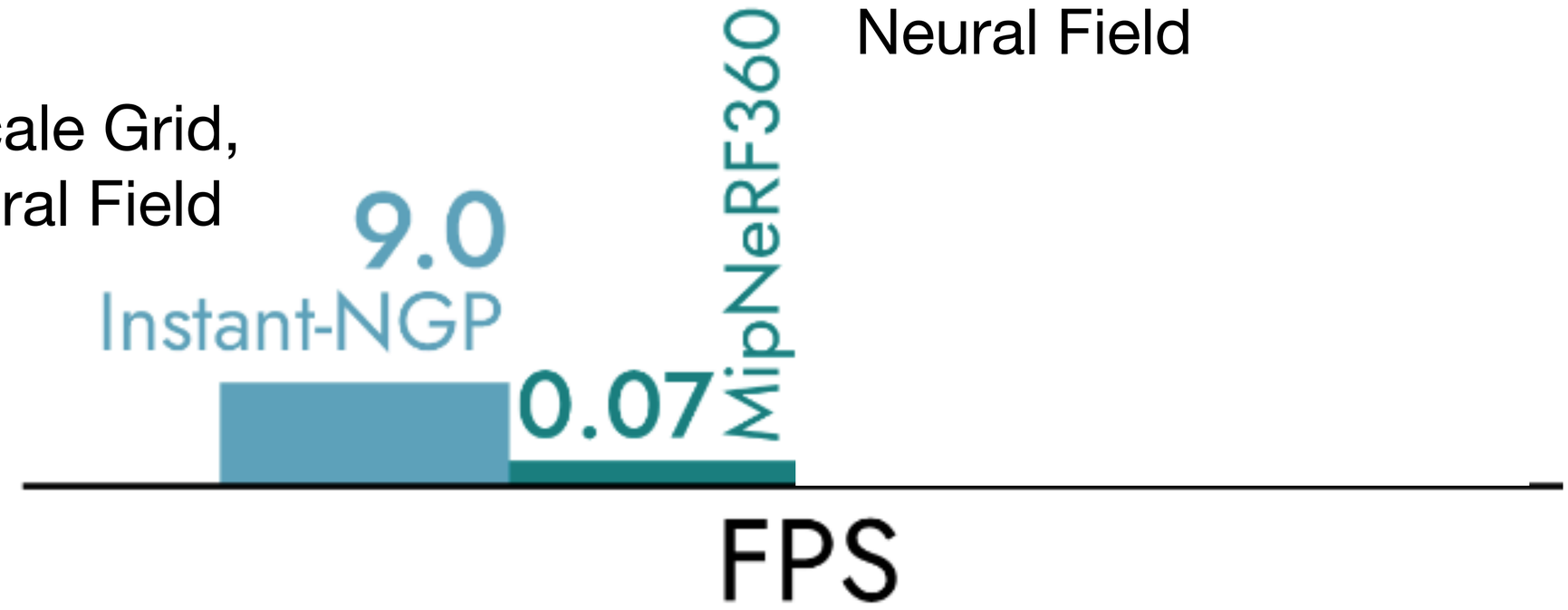


PiFU, Saito et al., ICCV 2019.
PixelNeRF, Yu et al., CVPR 2021
Grf: Learning a general radiance field…, Trevithick et al.

SRNs PixelNeRF

To scale, need *fast & cheap* rendering!

Hybrid Multi-Scale Grid,
HashMap, Neural Field



3D Gaussian Splatting for Real-Time Radiance Field Rendering

BERNHARD KERBL*, Inria, Université Côte d'Azur, France
GEORGIOS KOPANAS*, Inria, Université Côte d'Azur, France
THOMAS LEIMKÜHLER, Max-Planck-Institut für Informatik, Germany
GEORGE DRETTAKIS, Inria, Université Côte d'Azur, France

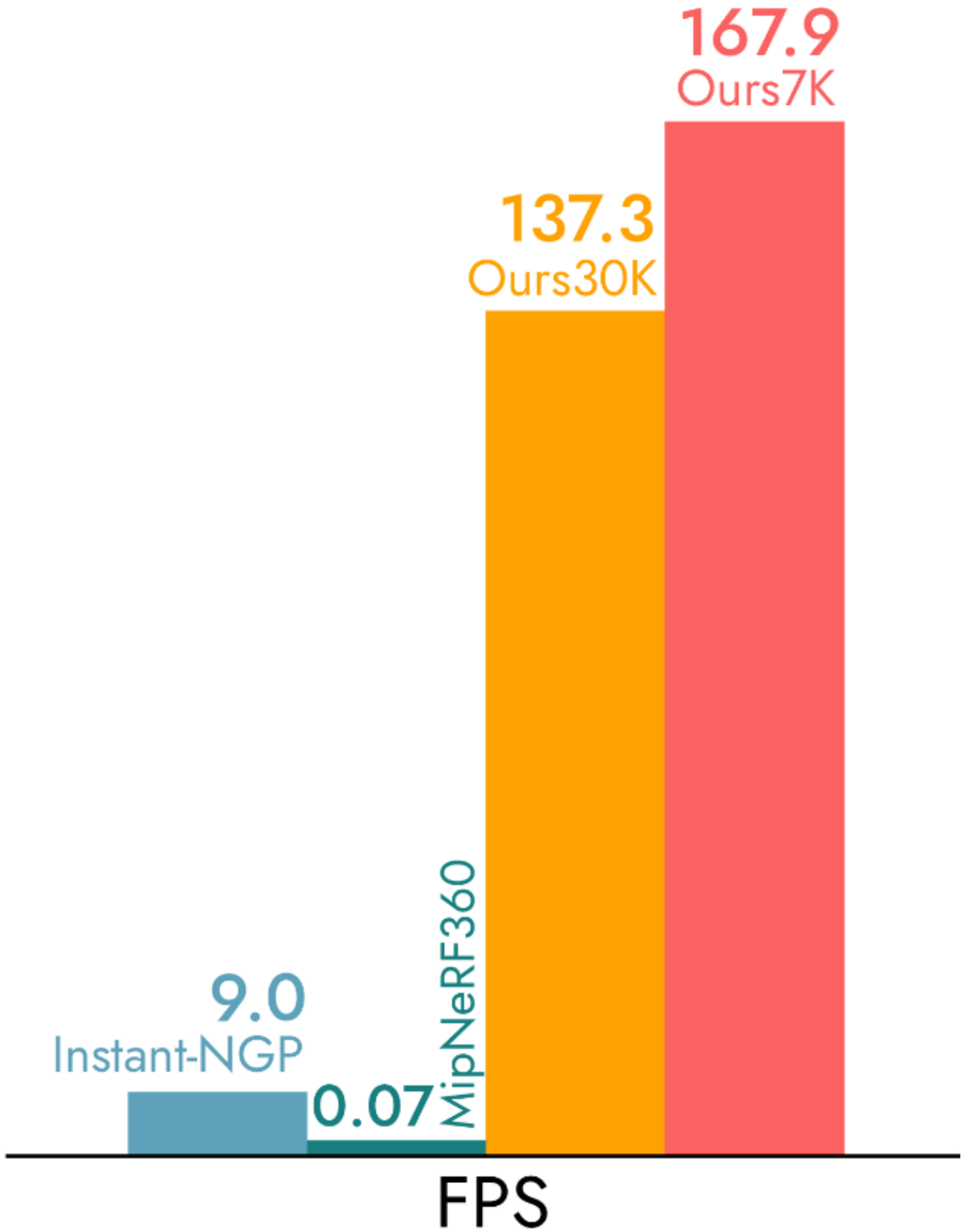
Fig. 1. Our method achieves real-time rendering of radiance fields while only requiring optimization times competitive with the faste performance is a novel 3D Gaussian scene representation coupled optimization and novel view synthesis. Note that for comparable trai this is the maximum quality they reach, by training for 51min we acl

EWA Splatting

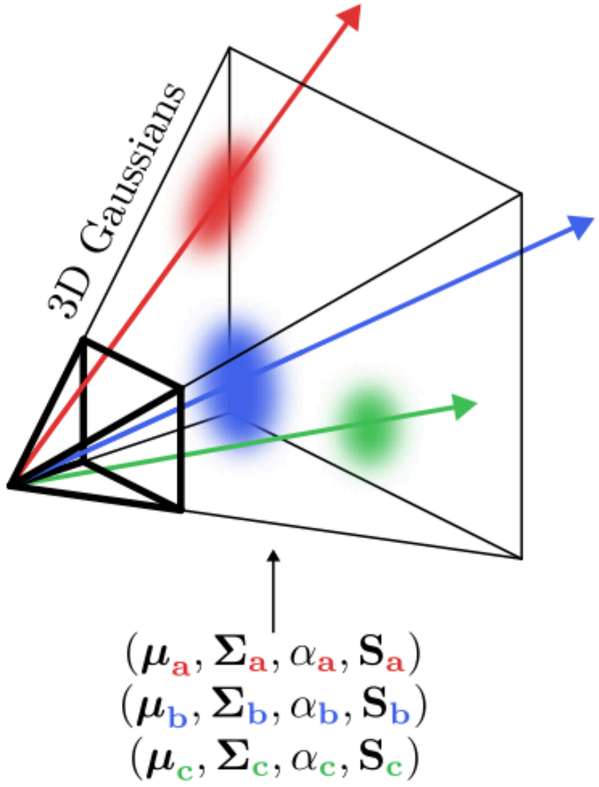
Matthias Zwicker, Hanspeter Pfister, *Member*, *IEEE*,
Jeroen van Baar, and Markus Gross, *Member*, *IEEE*

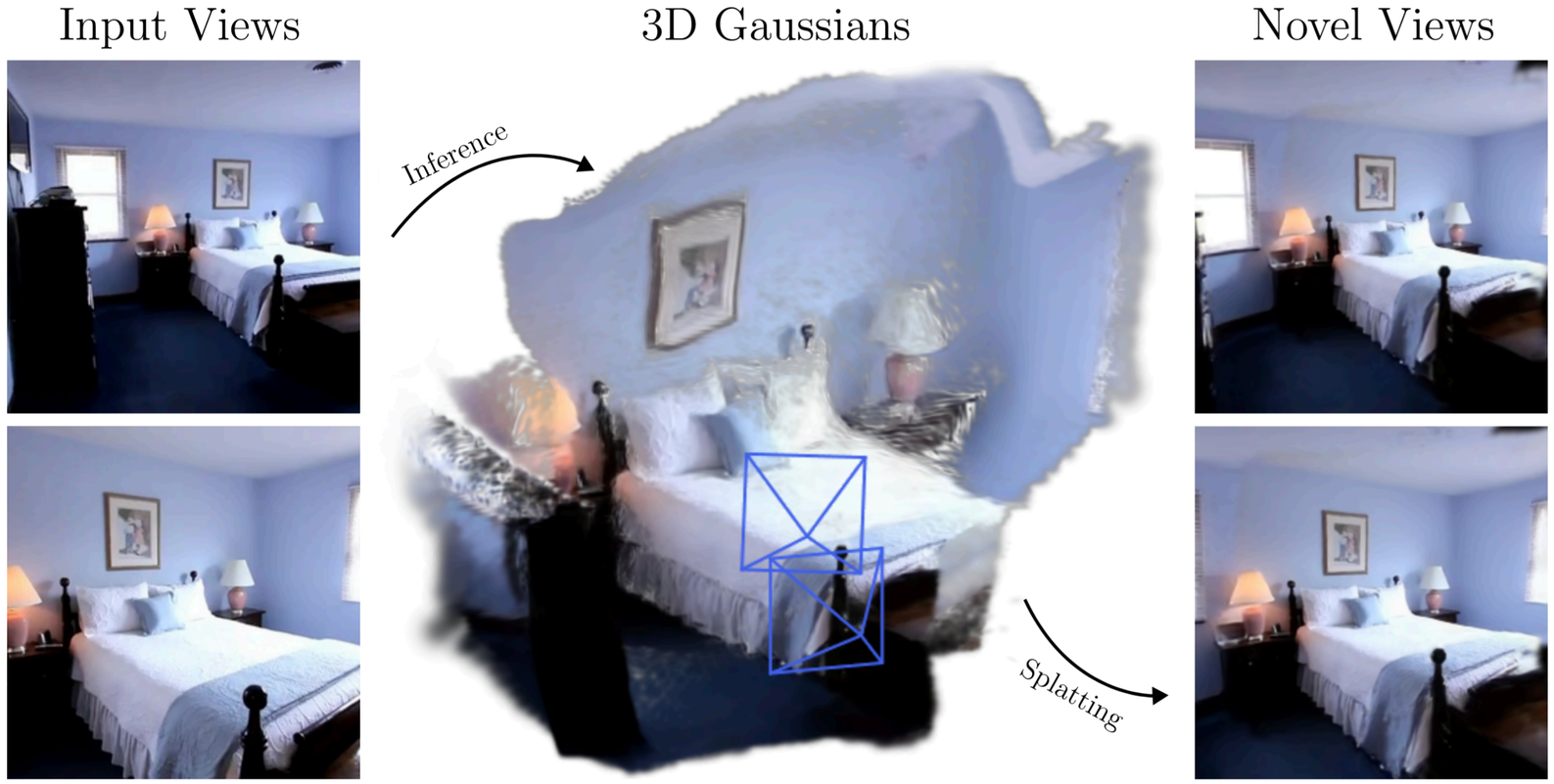
Abstract—In this paper, we present a framework for high quality splatting based on elliptical Gaussian kernels. To avoid aliasing artifacts, we introduce the concept of a resampling filter, combining a reconstruction kernel with a low-pass filter. Because of the similarity to Heckbert's EWA (elliptical weighted average) filter for texture mapping, we call our technique EWA splatting. Our framework allows us to derive EWA splat primitives for volume data and for point-sampled surface data. It provides high image quality without aliasing artifacts or excessive blurring for volume data and, additionally, features anisotropic texture filtering for point-sampled surfaces. It also handles nonspherical volume kernels efficiently; hence, it is suitable for regular, rectilinear, and irregular volume datasets. Moreover, our framework introduces a novel approach to compute the footprint function, facilitating efficient perspective projection of arbitrary elliptical kernels at very little additional cost. Finally, we show that EWA volume reconstruction kernels can be reduced to surface reconstruction kernels. This makes our splat primitive universal in rendering surface and volume data.

Index Terms—Rendering systems, volume rendering, texture mapping, splatting, antialiasing.

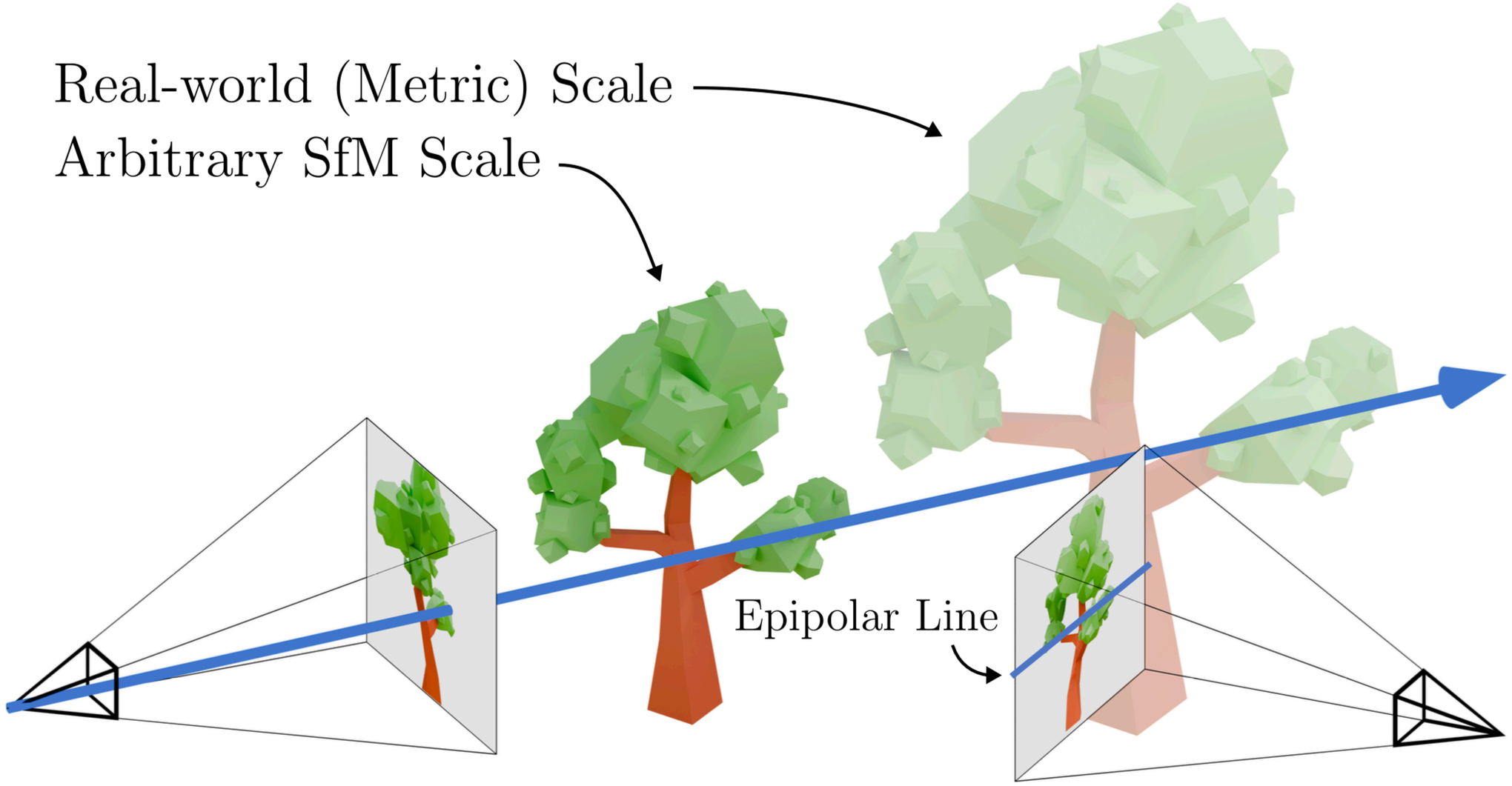

167.9
Ours7K
137.3
Ours30K
9.0
Instant-NGP
0.07
MipNeRF360
FPS

Idea: Predict 1 Gaussian per pixel (pixel-aligned Gaussians)

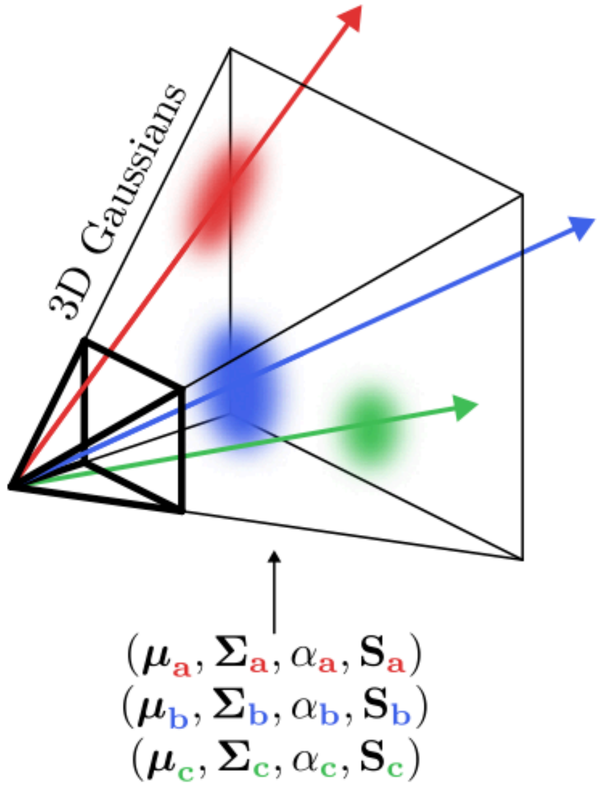




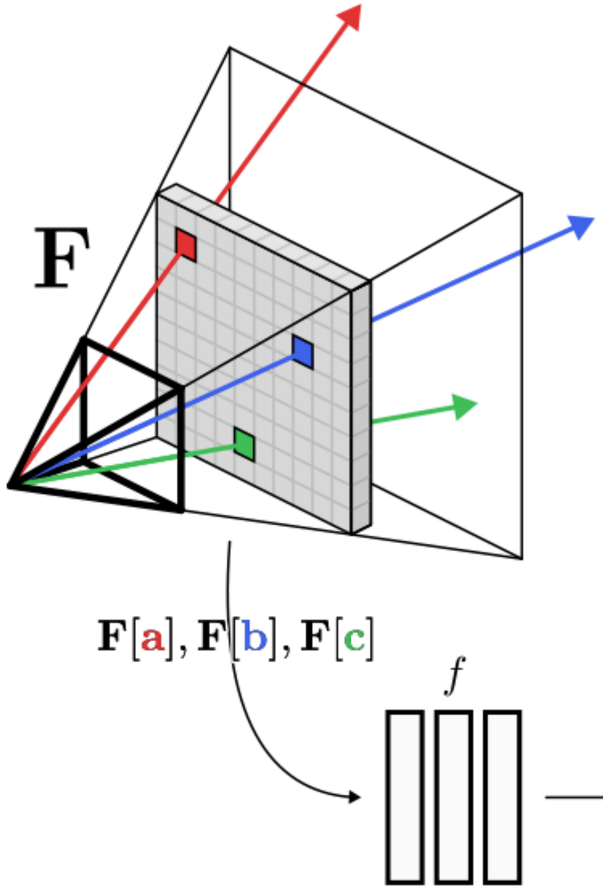
Core Challenge 1: Scale Ambiguity

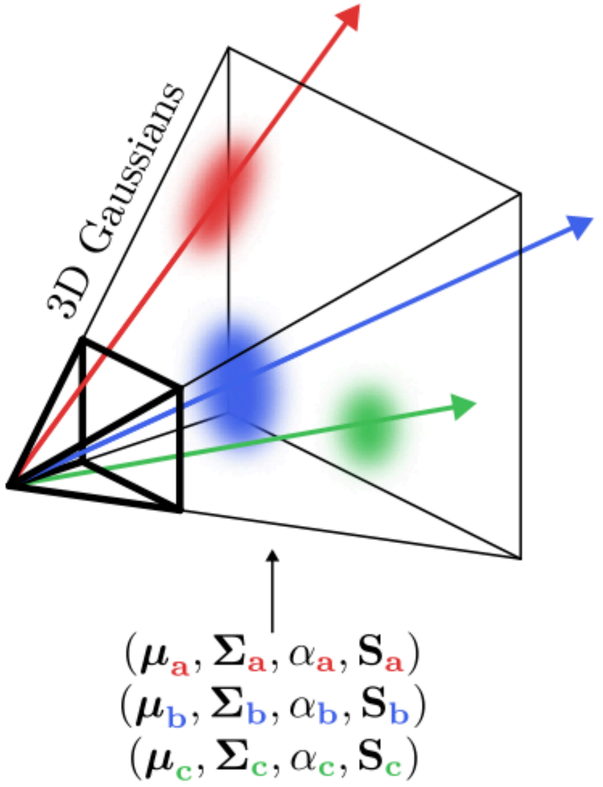


Core Challenge 2: Local Minima

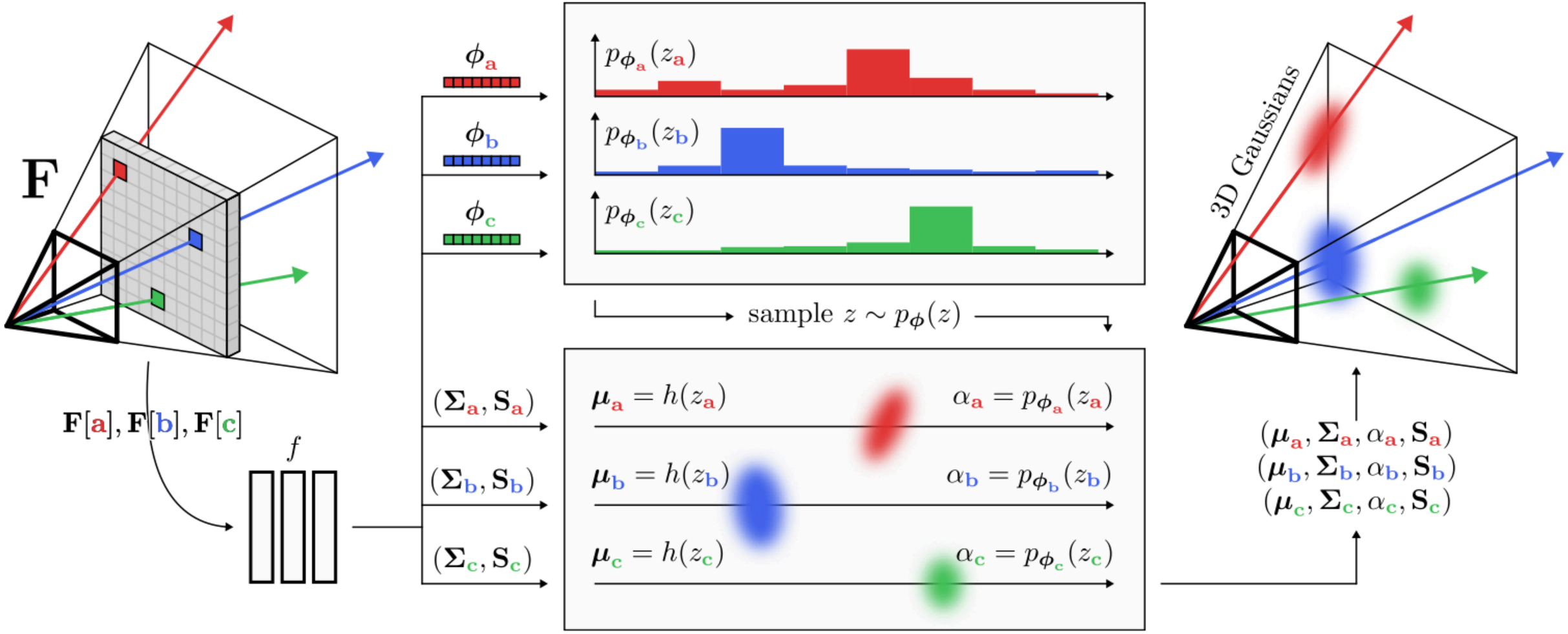


Core Challenge 2: Local Minima



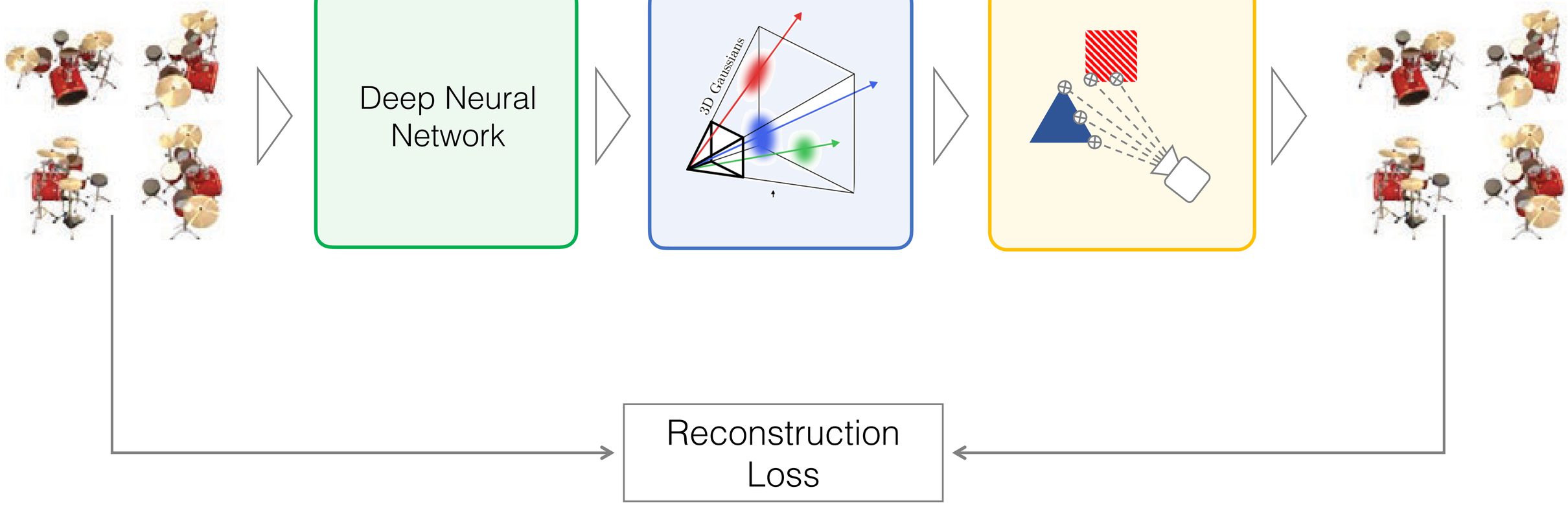


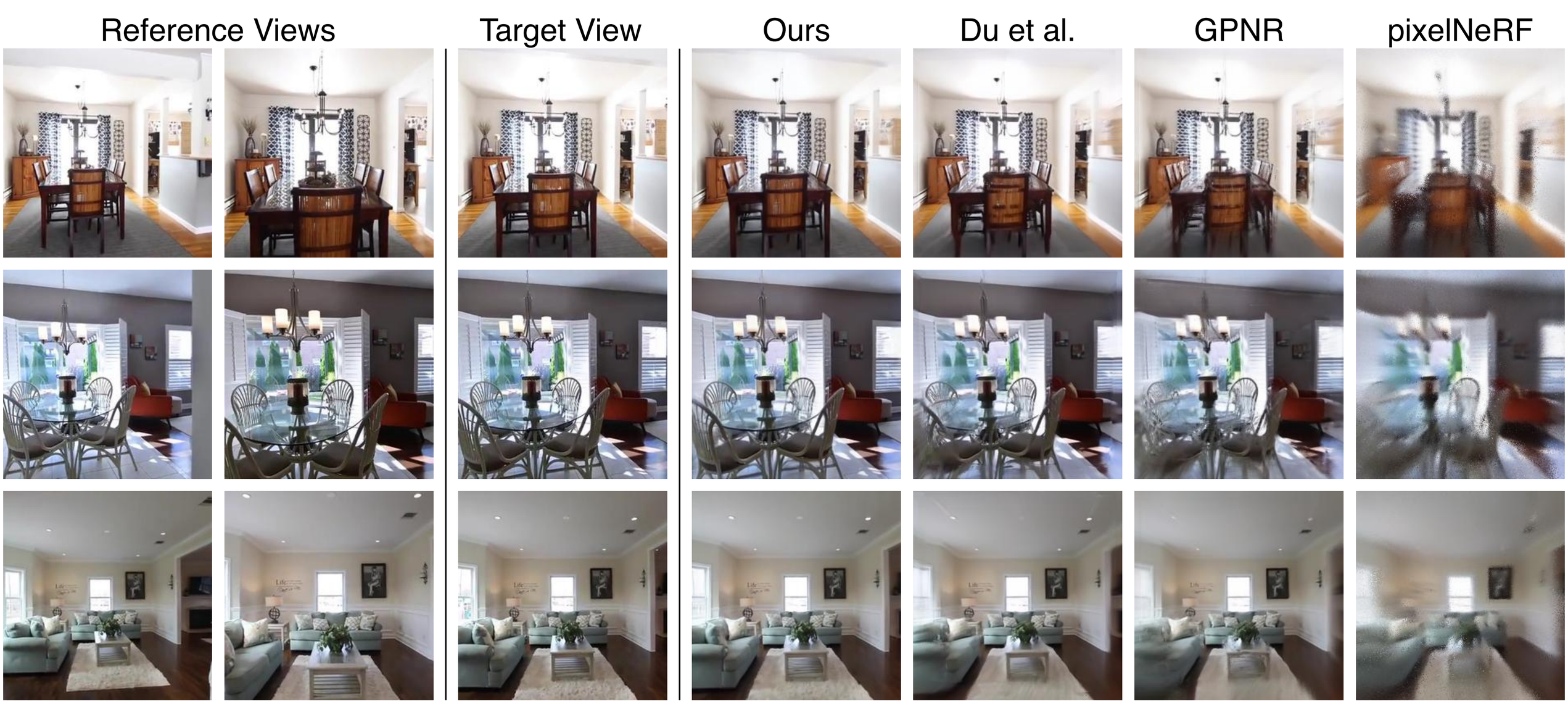
Core Challenge 2: Local Minima

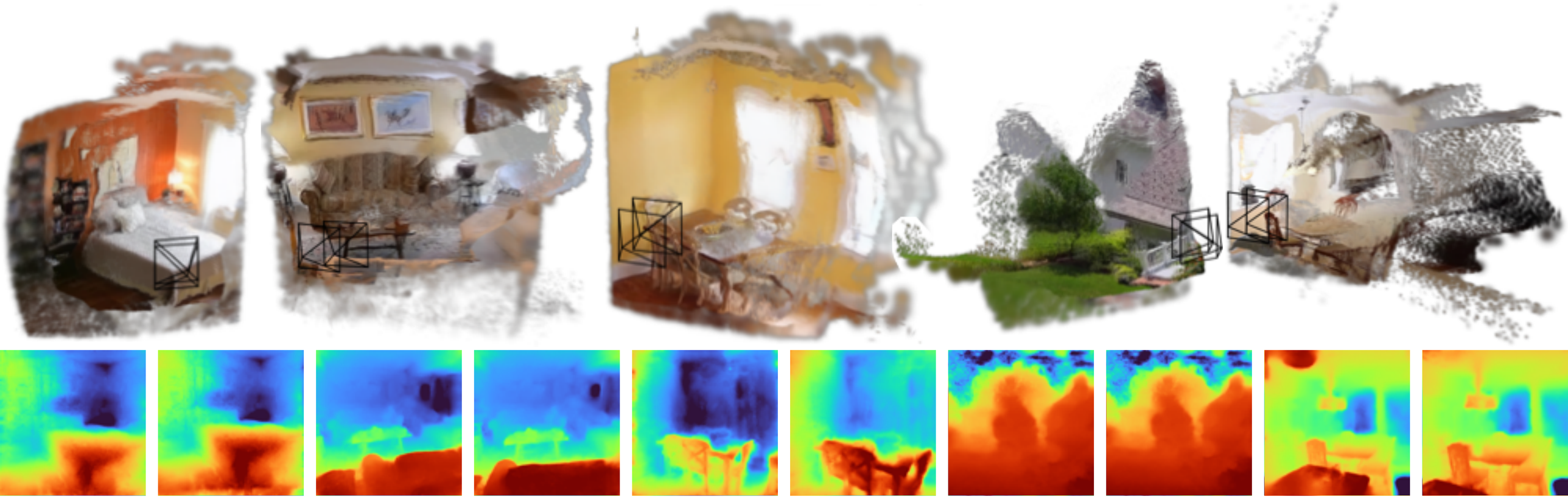


Amortized (=feedforward, generalizable) 3D Reconstruction

Input Images




Reference Views
Target View
Ours
Du et al.
GPNR
pixelNeRF



Video results - RealEstate10k

Scene cdf439b17a6a98d4 (frames 31 to 77)

Video results - RealEstate10k

Scene cdf439b17a6a98d4 (frames 31 to 77)

Video results - Acid

Scene 3fcb7b6b398b4064 (frames 105 to 220)

Video results - Acid

Scene 3fcb7b6b398b4064 (frames 105 to 220)

Splatter Image: Ultra-Fast Single-View 3D Reconstruction

Stanislaw Szymanowicz Christian Rupprecht Andrea Vedaldi
Visual Geometry Group — University of Oxford
{stan,chrisr,vedaldi}@robots.ox.ac.uk

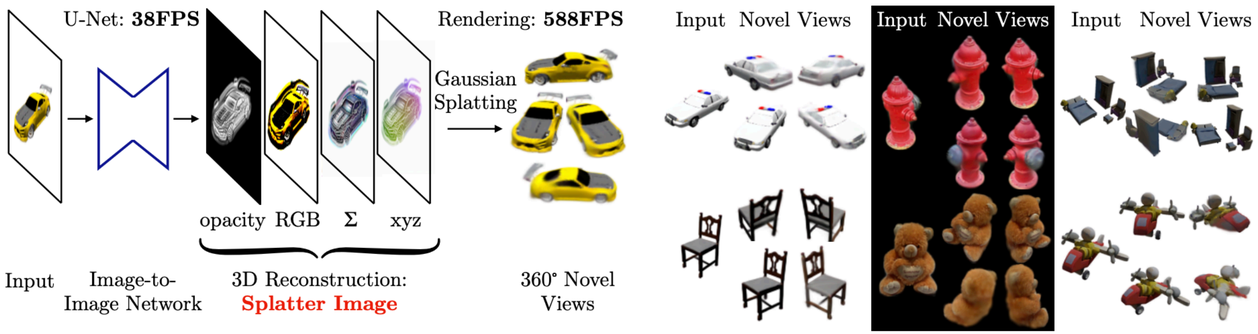


Figure 1. The **Splatter Image** is an ultra-efficient method for single- and few-view 3D reconstruction. It uses an image-to-image neural network to map the input image to another image that holds the parameters of one coloured 3D Gaussian per pixel. Splatter Image achieves excellent 3D reconstruction quality on synthetic, real and large-scale datasets while using a single GPU for training.

Abstract

We introduce the Splatter Image, an ultra-efficient approach for monocular 3D object reconstruction. Splatter Image is based on Gaussian Splatting, which allows fast and high-quality reconstruction of 3D scenes from multiple images. We apply Gaussian Splatting to monocular reconstruction by learning a neural network that, at test time, performs reconstruction in a feed-forward manner, at 38 FPS. Our main innovation is the surprisingly straightforward design of this network, which, using 2D operators, maps the input image to one 3D Gaussian per pixel. The re-

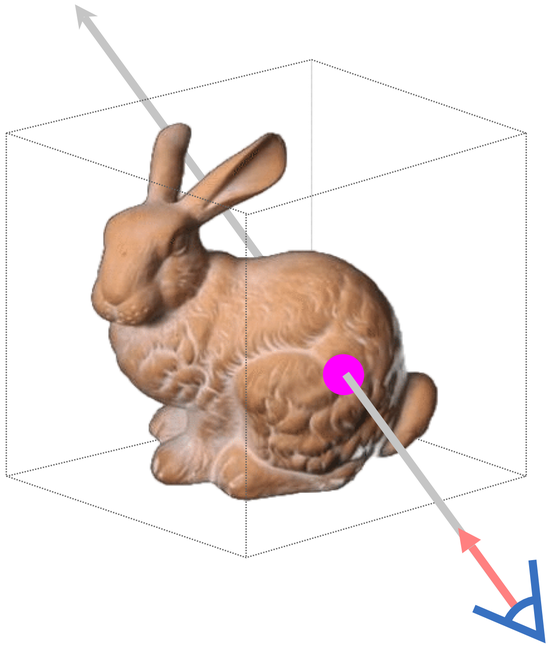
1. Introduction

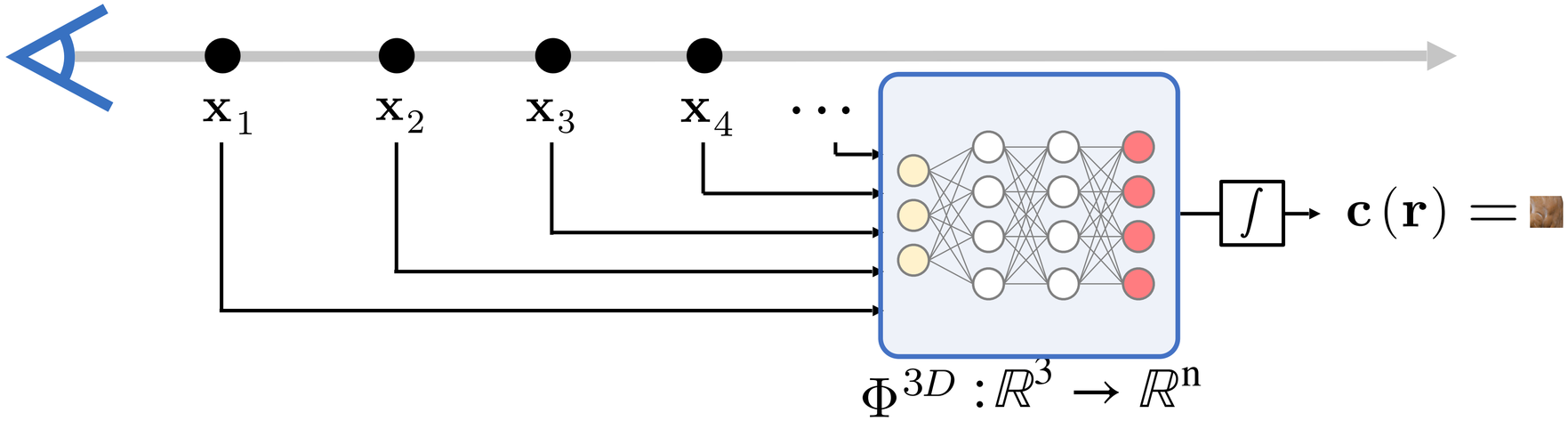
We contribute Splatter Image, a method that achieves ultra-fast single-view reconstruction of the 3D shape and appearance of objects. Splatter Image uses a set of 3D Gaussians as the 3D representation, taking advantage of the rendering quality and speed of Gaussian Splatting [22]. Splatter Image works by predicting a 3D Gaussian for each of the input image pixels, using an image-to-image neural network. Remarkably, the predicted 3D Gaussians provide 360° reconstructions of quality comparable or superior to much slower methods (Fig. 1).

We formulate monocular 3D reconstruction as the prob-

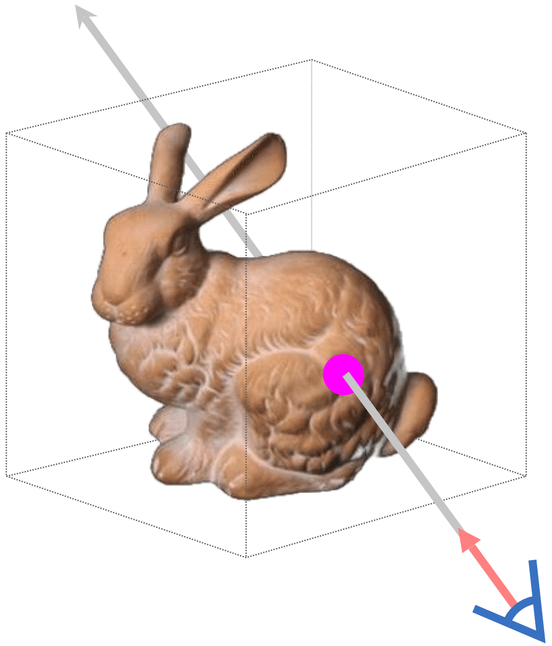
Concurrent work “SplatterImage” suggests that local minima not a huge deal

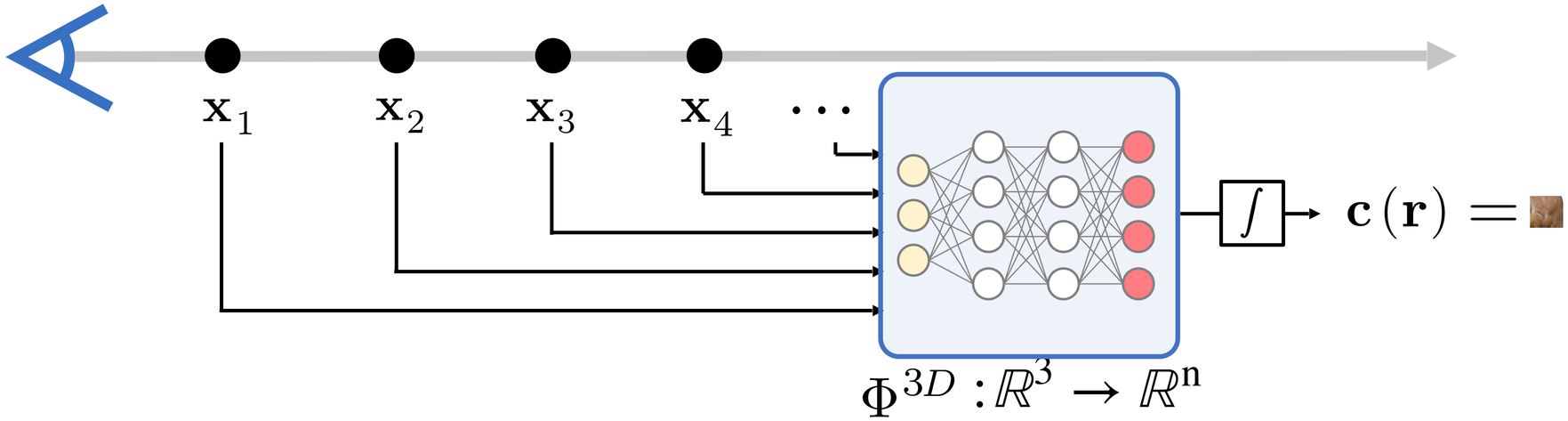
3D-structured Neural Scene Representations





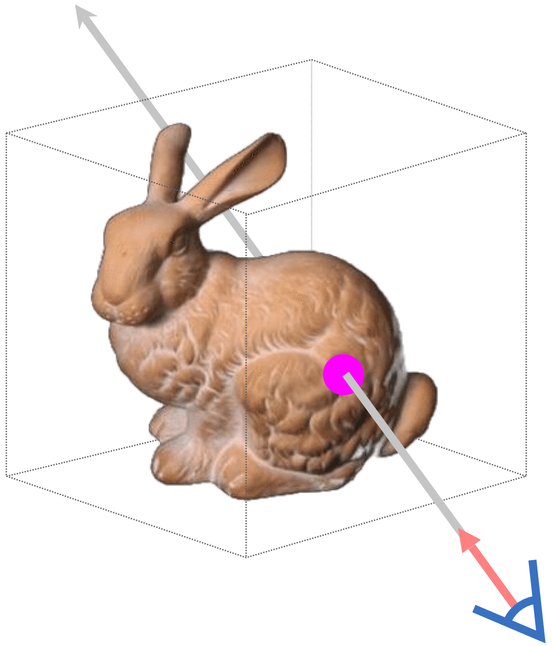
3D-structured Neural Scene Representations

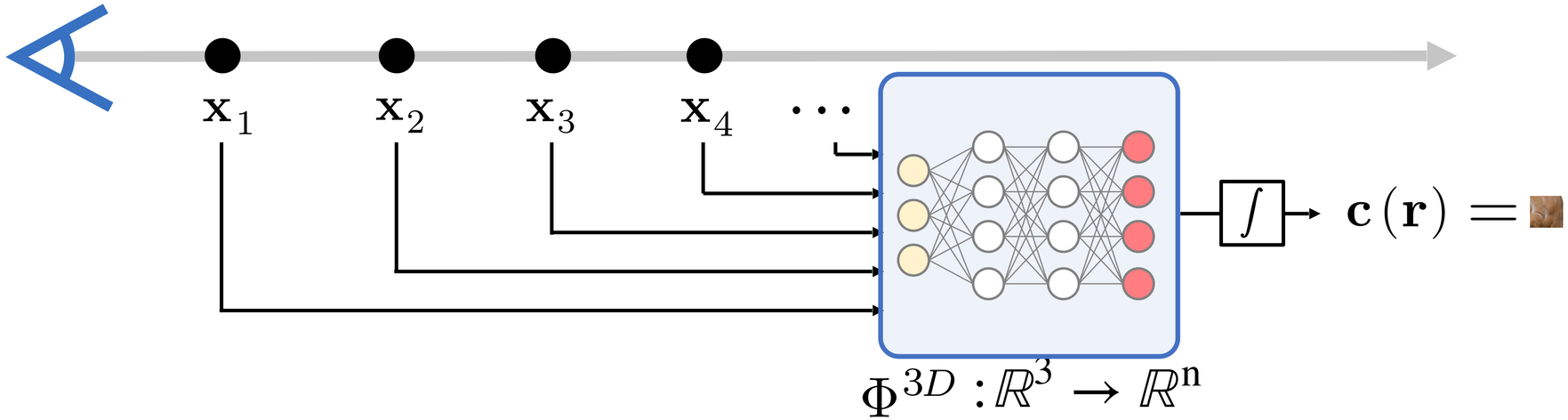




Hundreds of samples per ray.

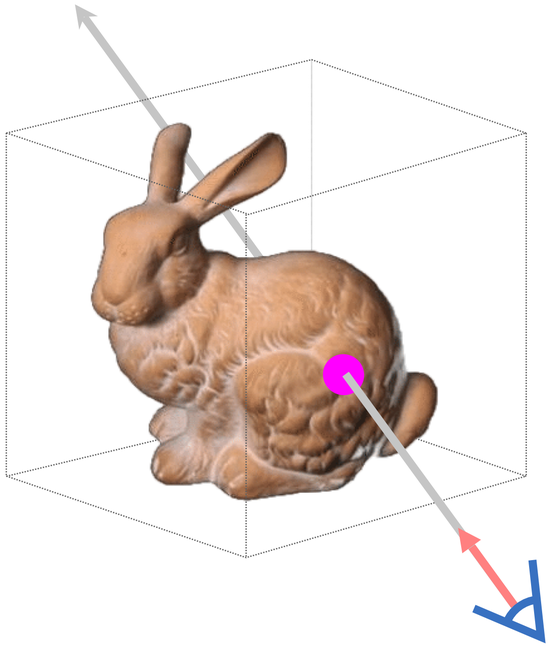
3D-structured Neural Scene Representations





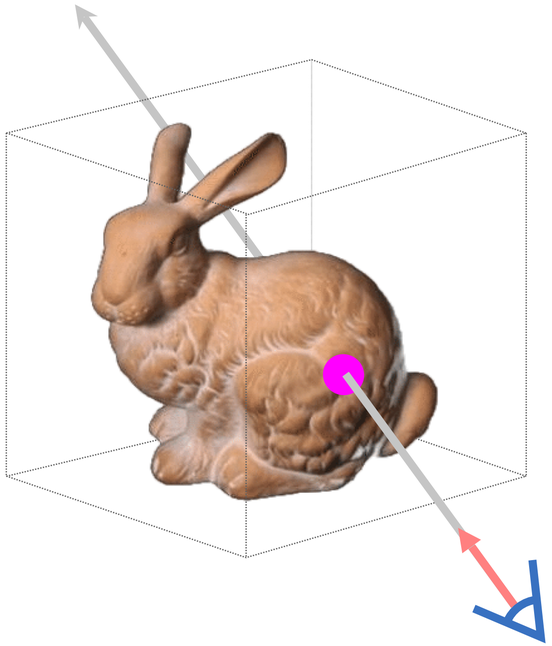
Hundreds of samples per ray.

Time- and memory-intensive.



Light Field

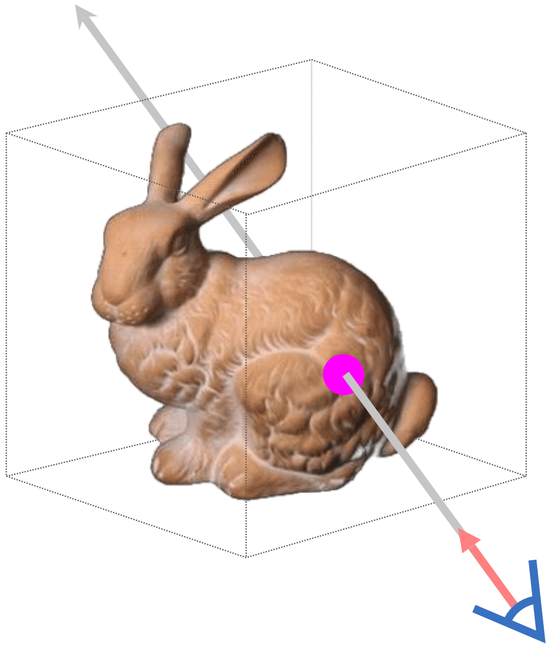
[Adelson et al. 1991, Levoy et al. 1996, Gortler et al. 1996]

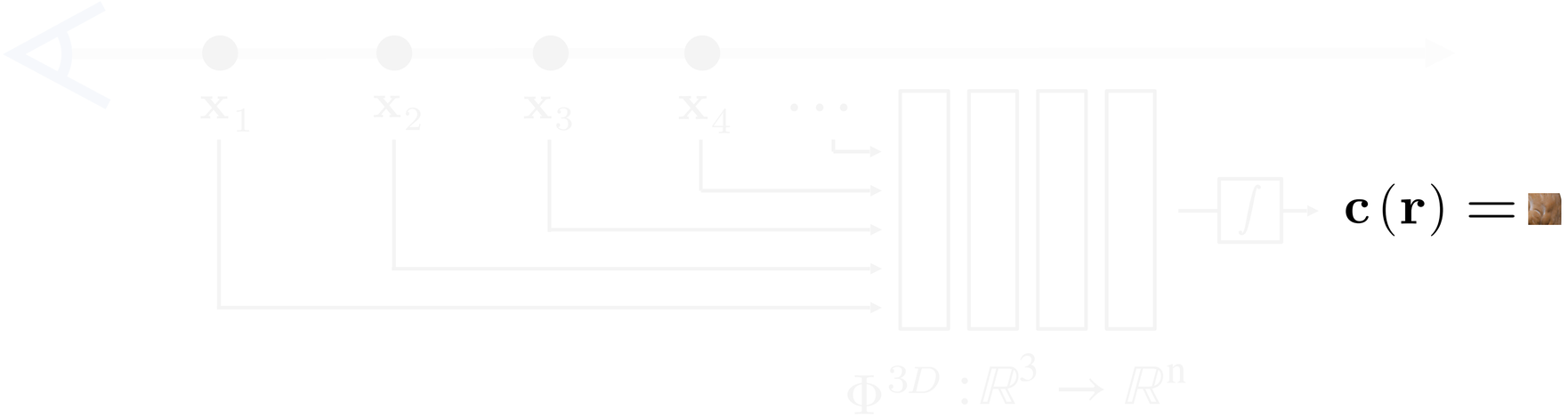


Light Field Networks

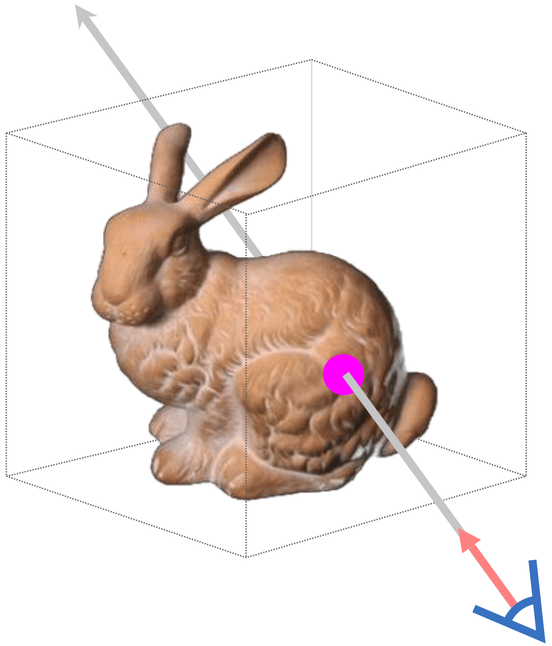
$\mathbf{c}(\mathbf{r}) =$

Light Field Networks





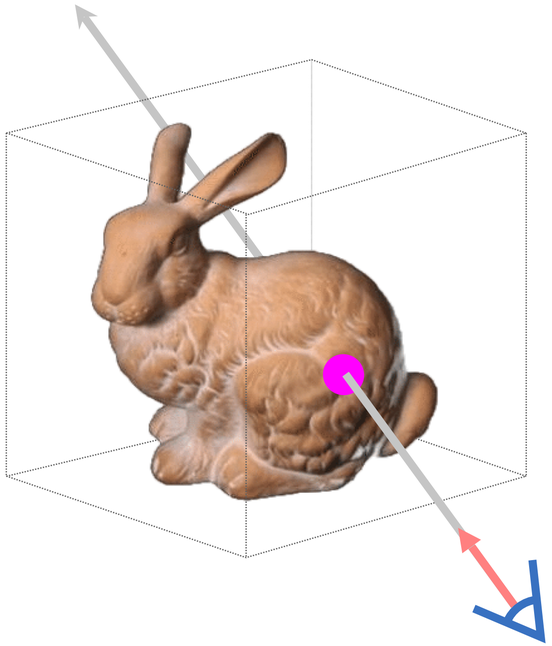
Light Field Networks

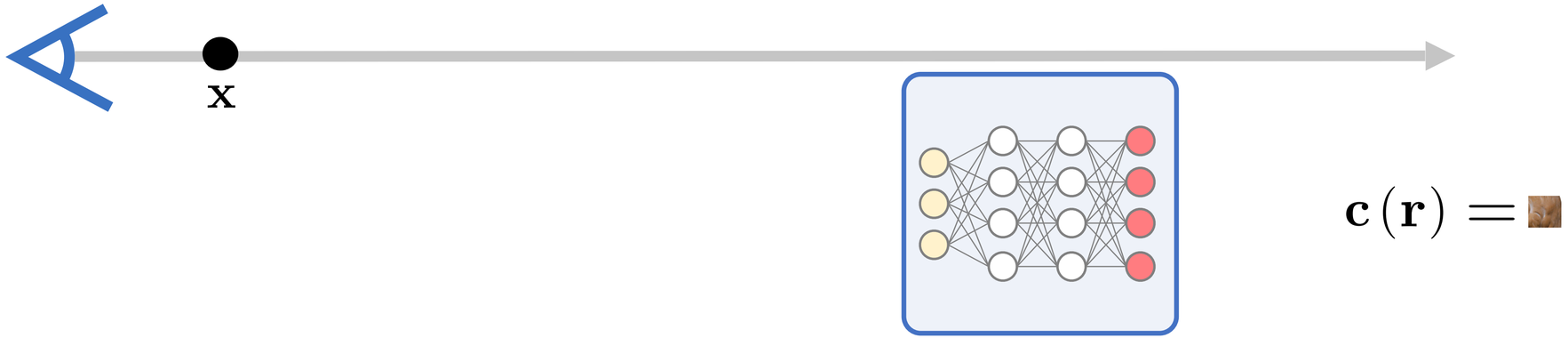


$\mathbf{c}(\mathbf{r}) =$

An Alternative Scene Representation

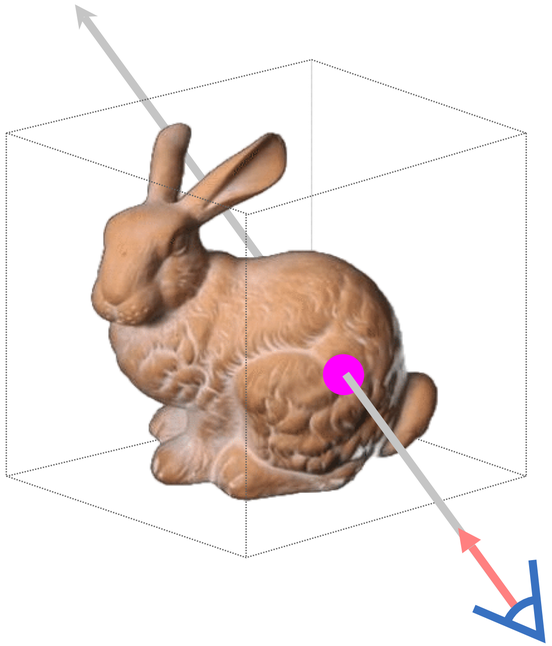
Light Field Networks





An Alternative Scene Representation

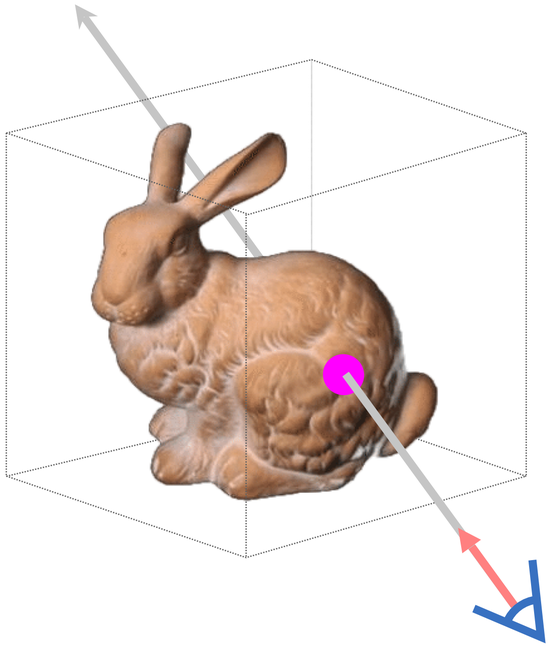
Light Field Networks

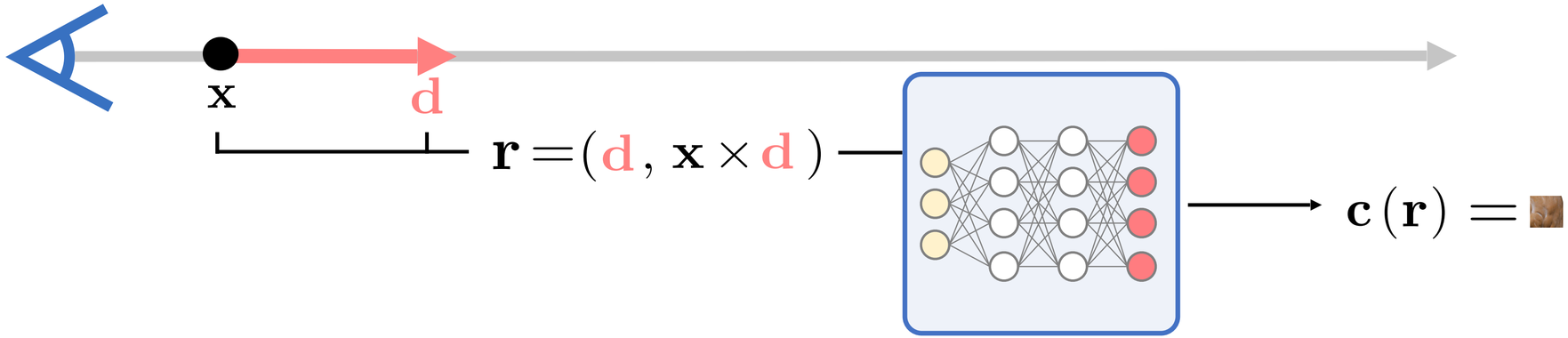


$$\mathbf{c}(\mathbf{r}) =$$

An Alternative Scene Representation

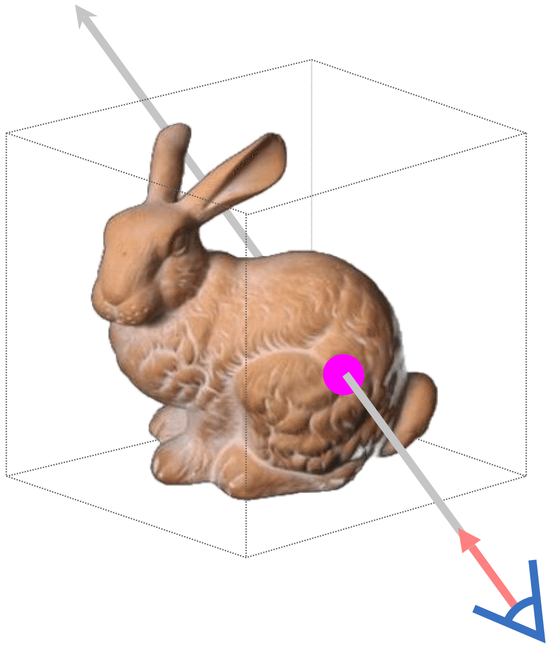
Light Field Networks

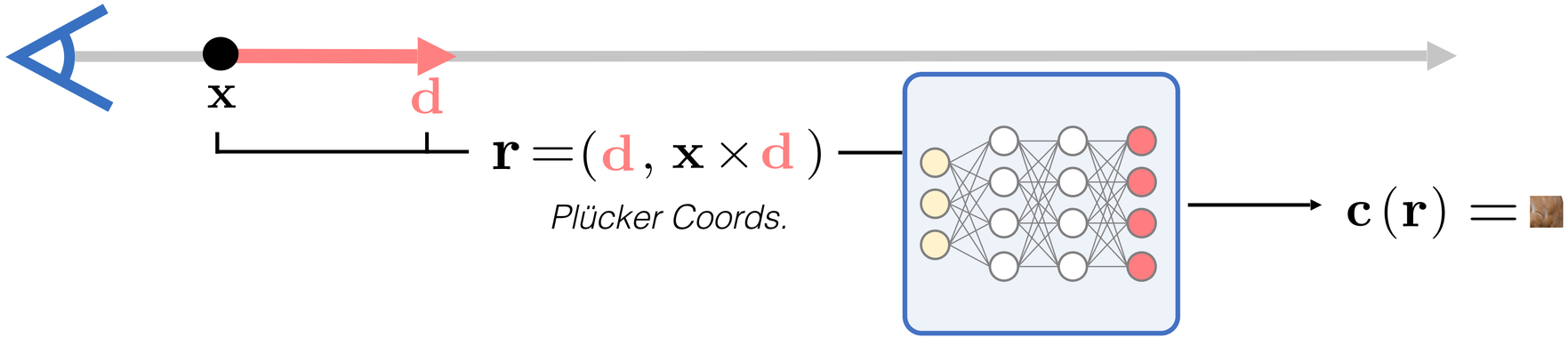




An Alternative Scene Representation

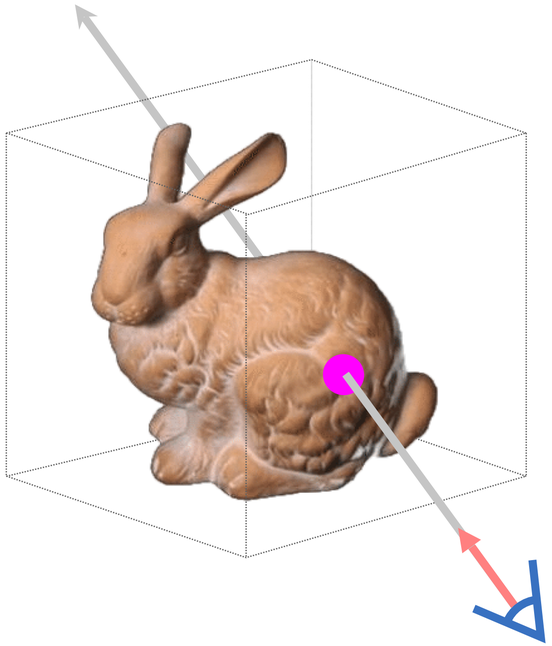
Light Field Networks

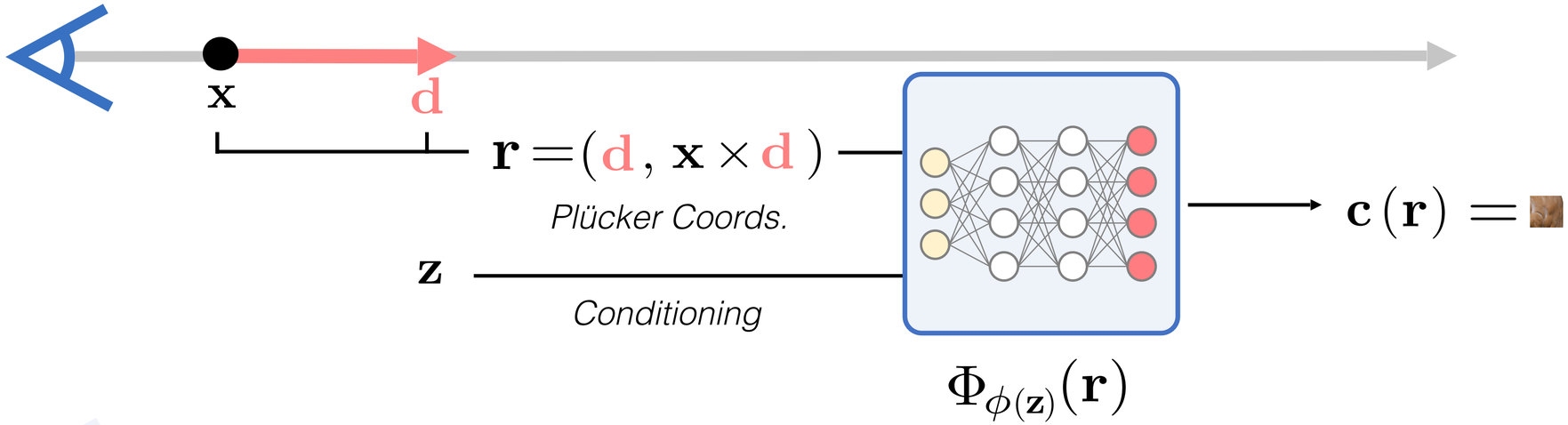




An Alternative Scene Representation

Light Field Networks





An Alternative Scene Representation

Light Field Networks
500 FPS
1 evaluation per ray

Volumetric Rendering (pixelNeRF)
0.033 FPS
196 evaluations per ray

100x speed

Real-time.
>100x reduction in memory: Can be trained on small GPUs!

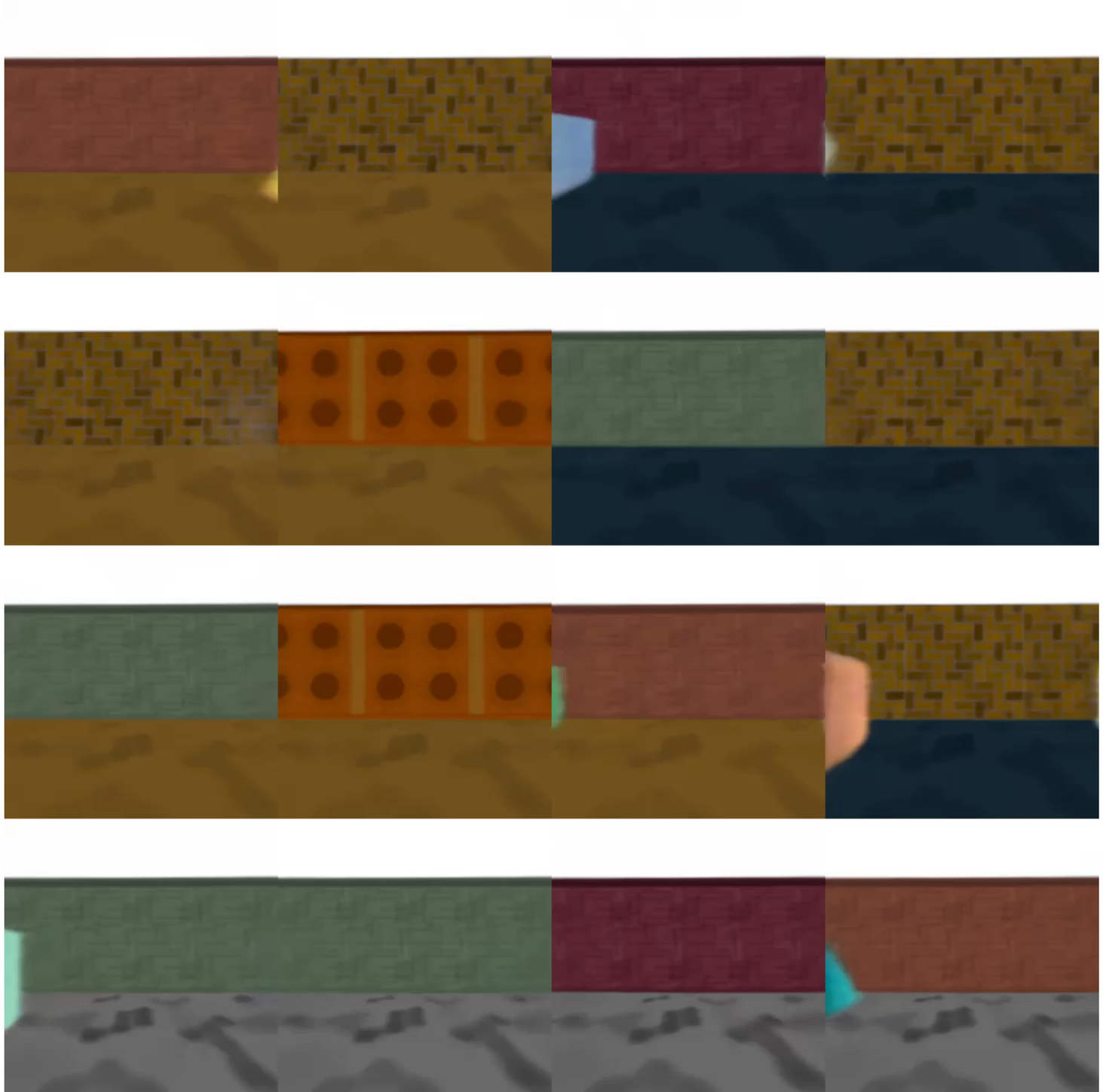
Light Field Networks
500 FPS
1 evaluation per ray

Volumetric Rendering (pixelNeRF)
0.033 FPS
196 evaluations per ray

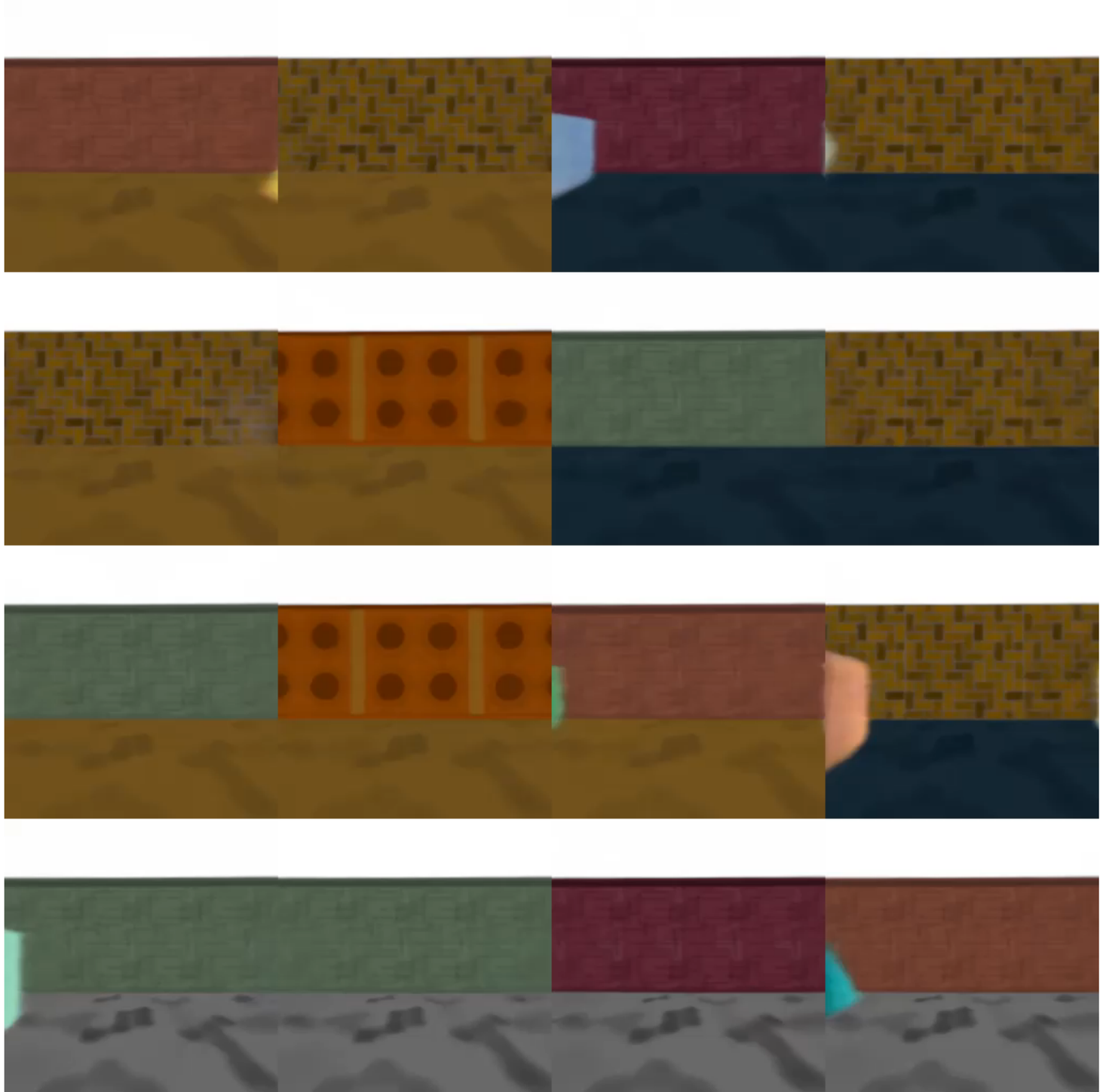
100x speed

Real-time.
>100x reduction in memory: Can be trained on small GPUs!

Light Field Networks
500 FPS
1 evaluation per ray



Light Field Networks
500 FPS
1 evaluation per ray



LVSM: A Large View Synthesis Model with Minimal 3D Inductive Bias

Haian Jin[1]* **Hanwen Jiang**[2] **Hao Tan**[3] **Kai Zhang**[3] **Sai Bi**[3] **Tianyuan Zhang**[4]
Fujun Luan[3] **Noah Snavely**[1] **Zexiang Xu**[3]
[1]Cornell University [2]The University of Texas at Austin
[3]Adobe Research [4]Massachusetts Institute of Technology

Abstract

We propose the Large View Synthesis Model (LVSM), a novel transformer-based approach for scalable and generalizable novel view synthesis from sparse-view inputs. We introduce two architectures: (1) an encoder-decoder LVSM, which encodes input image tokens into a fixed number of 1D latent tokens, functioning as a fully learned scene representation, and decodes novel-view images from them; and (2) a decoder-only LVSM, which directly maps input images to novel-view outputs, completely eliminating intermediate scene representations. Both models bypass the 3D inductive biases used in previous methods—from 3D representations (e.g., NeRF, 3DGS) to network designs (e.g., epipolar projections, plane sweeps)—addressing novel view synthesis with a fully data-driven approach. While the encoder-decoder model offers faster inference due to its independent latent representation, the decoder-only LVSM achieves superior quality, scalability, and zero-shot generalization, outperforming previous state-of-the-art methods by 1.5 to 3.5 dB PSNR. Comprehensive evaluations across multiple datasets demonstrate that both LVSM variants achieve state-of-the-art novel view synthesis quality. Notably, our models surpass all previous methods even with reduced computational resources (1-2 GPUs). Please see our website for more details: https://haian-jin.github.io/projects/LVSM/.

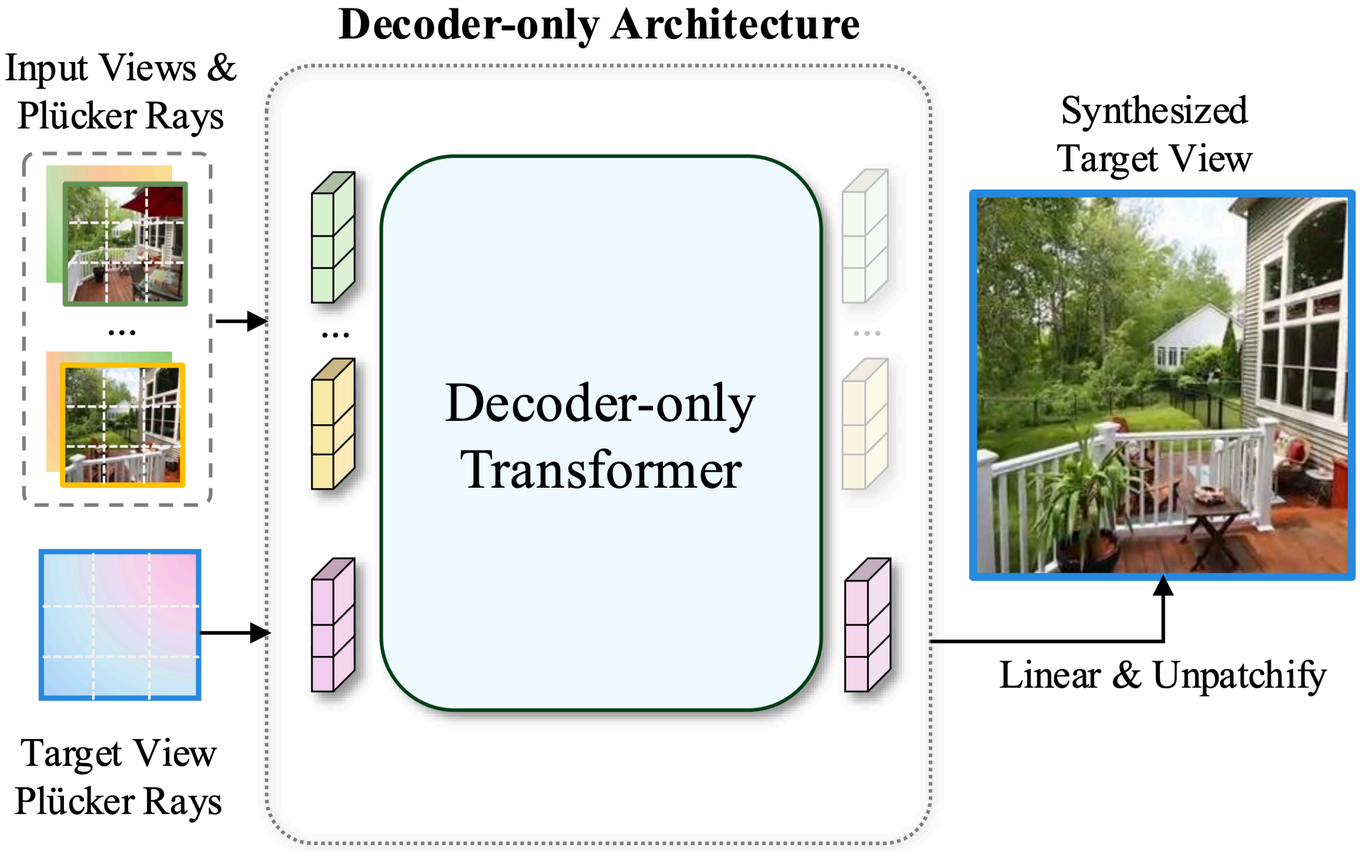
1 Introduction

Novel view synthesis is a long-standing challenge in vision and graphics. For decades, the community has generally relied on various 3D inductive biases, incorporating 3D priors and handcrafted structures to simplify the task and improve synthesis quality. Recently, NeRF, 3D Gaussian Splatting (3DGS), and their variants (Mildenhall et al., 2020; Barron et al., 2021; Müller et al., 2022; Chen et al., 2022; Xu et al., 2022; Kerbl et al., 2023; Yu et al., 2024) have significantly advanced the field by introducing new inductive biases through carefully designed 3D representations (e.g., continuous volumetric fields and Gaussian primitives) and rendering equations (e.g., ray marching and splatting with alpha blending), reframing view synthesis as the optimization of the representations using rendering losses on a per-scene basis. Other methods have also built generalizable networks to estimate these representations or directly generate novel-view images in a feed-forward manner, often incorporating additional 3D inductive biases, such as projective epipolar lines or plane-sweep volumes, in their architecture designs (Wang et al., 2021a; Yu et al., 2021; Chen et al., 2021; Suhail et al., 2022b; Charatan et al., 2024; Chen et al., 2024).

While effective, these 3D inductive biases inherently limit model flexibility, constraining their adaptability to more diverse and complex scenarios that do not align with predefined priors or handcrafted structures. Recent large reconstruction models (LRMs) (Hong et al., 2024; Li et al., 2023; Wei et al., 2024; Zhang et al., 2024) have made notable progress in removing architecture-level biases by leveraging large transformers without relying on epipolar projections or plane-sweep volumes, achieving state-of-the-art novel view synthesis quality. However, despite these advances, LRMs still rely on representation-level biases—such as NeRFs, meshes, or 3DGS, along with their respective rendering equations—that limit their potential generalization and scalability.

*This work was done when Haian Jin, Hanwen Jiang, and Tianyuan Zhang were interns at Adobe Research.

SOTA today: Light Field Transformers



LVSM: A Large View Synthesis Model with Minimal 3D Inductive Bias

Haian Jin[1*] **Hanwen Jiang**[2] **Hao Tan**[3] **Kai Zhang**[3] **Sai Bi**[3] **Tianyuan Zhang**[4]
Fujun Luan[3] **Noah Snavely**[1] **Zexiang Xu**[3]
[1]Cornell University [2]The University of Texas at Austin
[3]Adobe Research [4]Massachusetts Institute of Technology

Abstract

We propose the Large View Synthesis Model (LVSM), a novel transformer-based approach for scalable and generalizable novel view synthesis from sparse-view inputs. We introduce two architectures: (1) an encoder-decoder LVSM, which encodes input image tokens into a fixed number of 1D latent tokens, functioning as a fully learned scene representation, and decodes novel-view images from them; and (2) a decoder-only LVSM, which directly maps input images to novel-view outputs, completely eliminating intermediate scene representations. Both models bypass the 3D inductive biases used in previous methods—from 3D representations (e.g., NeRF, 3DGS) to network designs (e.g., epipolar projections, plane sweeps)—addressing novel view synthesis with a fully data-driven approach. While the encoder-decoder model offers faster inference due to its independent latent representation, the decoder-only LVSM achieves superior quality, scalability, and zero-shot generalization, outperforming previous state-of-the-art methods by 1.5 to 3.5 dB PSNR. Comprehensive evaluations across multiple datasets demonstrate that both LVSM variants achieve state-of-the-art novel view synthesis quality. Notably, our models surpass all previous methods even with reduced computational resources (1-2 GPUs). Please see our website for more details: https://haian-jin.github.io/projects/LVSM/.

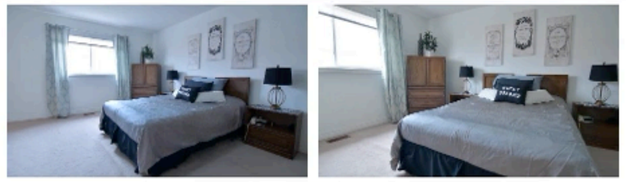
1 Introduction

Novel view synthesis is a long-standing challenge in vision and graphics. For decades, the community has generally relied on various 3D inductive biases, incorporating 3D priors and handcrafted structures to simplify the task and improve synthesis quality. Recently, NeRF, 3D Gaussian Splatting (3DGS), and their variants (Mildenhall et al., 2020; Barron et al., 2021; Müller et al., 2022; Chen et al., 2022; Xu et al., 2022; Kerbl et al., 2023; Yu et al., 2024) have significantly advanced the field by introducing new inductive biases through carefully designed 3D representations (e.g., continuous volumetric fields and Gaussian primitives) and rendering equations (e.g., ray marching and splatting with alpha blending), reframing view synthesis as the optimization of the representations using rendering losses on a per-scene basis. Other methods have also built generalizable networks to estimate these representations or directly generate novel-view images in a feed-forward manner, often incorporating additional 3D inductive biases, such as projective epipolar lines or plane-sweep volumes, in their architecture designs (Wang et al., 2021a; Yu et al., 2021; Chen et al., 2021; Suhail et al., 2022b; Charatan et al., 2024; Chen et al., 2024).

While effective, these 3D inductive biases inherently limit model flexibility, constraining their adaptability to more diverse and complex scenarios that do not align with predefined priors or handcrafted structures. Recent large reconstruction models (LRMs) (Hong et al., 2024; Li et al., 2023; Wei et al., 2024; Zhang et al., 2024) have made notable progress in removing architecture-level biases by leveraging large transformers without relying on epipolar projections or plane-sweep volumes, achieving state-of-the-art novel view synthesis quality. However, despite these advances, LRMs still rely on representation-level biases—such as NeRFs, meshes, or 3DGS, along with their respective rendering equations—that limit their potential generalization and scalability.

*This work was done when Haian Jin, Hanwen Jiang, and Tianyuan Zhang were interns at Adobe Research.

SOTA today: Light Field Transformers



2 Input Views

LVSM: A Large View Synthesis Model with Minimal 3D Inductive Bias

Haian Jin[1*] **Hanwen Jiang**[2] **Hao Tan**[3] **Kai Zhang**[3] **Sai Bi**[3] **Tianyuan Zhang**[4]
Fujun Luan[3] **Noah Snavely**[1] **Zexiang Xu**[3]
[1]Cornell University [2]The University of Texas at Austin
[3]Adobe Research [4]Massachusetts Institute of Technology

Abstract

We propose the Large View Synthesis Model (LVSM), a novel transformer-based approach for scalable and generalizable novel view synthesis from sparse-view inputs. We introduce two architectures: (1) an encoder-decoder LVSM, which encodes input image tokens into a fixed number of 1D latent tokens, functioning as a fully learned scene representation, and decodes novel-view images from them; and (2) a decoder-only LVSM, which directly maps input images to novel-view outputs, completely eliminating intermediate scene representations. Both models bypass the 3D inductive biases used in previous methods—from 3D representations (e.g., NeRF, 3DGS) to network designs (e.g., epipolar projections, plane sweeps)—addressing novel view synthesis with a fully data-driven approach. While the encoder-decoder model offers faster inference due to its independent latent representation, the decoder-only LVSM achieves superior quality, scalability, and zero-shot generalization, outperforming previous state-of-the-art methods by 1.5 to 3.5 dB PSNR. Comprehensive evaluations across multiple datasets demonstrate that both LVSM variants achieve state-of-the-art novel view synthesis quality. Notably, our models surpass all previous methods even with reduced computational resources (1-2 GPUs). Please see our website for more details: https://haian-jin.github.io/projects/LVSM/.

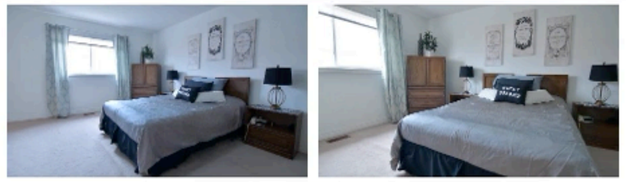
1 Introduction

Novel view synthesis is a long-standing challenge in vision and graphics. For decades, the community has generally relied on various 3D inductive biases, incorporating 3D priors and handcrafted structures to simplify the task and improve synthesis quality. Recently, NeRF, 3D Gaussian Splatting (3DGS), and their variants (Mildenhall et al., 2020; Barron et al., 2021; Müller et al., 2022; Chen et al., 2022; Xu et al., 2022; Kerbl et al., 2023; Yu et al., 2024) have significantly advanced the field by introducing new inductive biases through carefully designed 3D representations (e.g., continuous volumetric fields and Gaussian primitives) and rendering equations (e.g., ray marching and splatting with alpha blending), reframing view synthesis as the optimization of the representations using rendering losses on a per-scene basis. Other methods have also built generalizable networks to estimate these representations or directly generate novel-view images in a feed-forward manner, often incorporating additional 3D inductive biases, such as projective epipolar lines or plane-sweep volumes, in their architecture designs (Wang et al., 2021a; Yu et al., 2021; Chen et al., 2021; Suhail et al., 2022b; Charatan et al., 2024; Chen et al., 2024).

While effective, these 3D inductive biases inherently limit model flexibility, constraining their adaptability to more diverse and complex scenarios that do not align with predefined priors or handcrafted structures. Recent large reconstruction models (LRMs) (Hong et al., 2024; Li et al., 2023; Wei et al., 2024; Zhang et al., 2024) have made notable progress in removing architecture-level biases by leveraging large transformers without relying on epipolar projections or plane-sweep volumes, achieving state-of-the-art novel view synthesis quality. However, despite these advances, LRMs still rely on representation-level biases—such as NeRFs, meshes, or 3DGS, along with their respective rendering equations—that limit their potential generalization and scalability.

*This work was done when Haian Jin, Hanwen Jiang, and Tianyuan Zhang were interns at Adobe Research.

SOTA today: Light Field Transformers



2 Input Views

History-Guided Video Diffusion

Kiwhan Song* [1] **Boyuan Chen*** [1] **Max Simchowitz** [1] **Yilun Du** [1] **Russ Tedrake** [1] **Vincent Sitzmann** [1]

Abstract

Classifier-free guidance (CFG) is a key technique for improving conditional generation in diffusion models, enabling more accurate control while enhancing sample quality. It is natural to extend this technique to video diffusion, which generates video conditioned on a variable number of context frames, collectively referred to as history. However, we find two key challenges to guiding with variable-length history: architectures that only support fixed-size conditioning, and the empirical observation that CFG-style history dropout performs poorly. To address this, we propose the Diffusion Forcing Transformer (DFoT), a video diffusion architecture and theoretically grounded training objective that jointly enable conditioning on a flexible number of history frames. We then introduce *History Guidance*, a family of guidance methods uniquely enabled by DFoT. We show that its simplest form, *vanilla history guidance*, already significantly improves video generation quality and temporal consistency. A more advanced method, *history guidance across time and frequency* further enhances motion dynamics, enables compositional generalization to out-of-distribution history, and can stably roll out extremely long videos. Website: this URL

1 Introduction

Diffusion models are effective generative models in domains such as image, sound, and video. Critical to their success is classifier-free guidance (CFG) (Ho & Salimans, 2022), which trades off between sample quality and diversity by jointly training a conditional and an unconditional diffusion model and combining their score estimates when sampling.

In the realm of video generative models, CFG commonly relies on either text or image prompts as conditioning variables. Yet, another conditioning variable, namely the entire collection of previous video frames, or *history*, deserves further exploration. In this paper, we investigate the following question: Can we use different portions of history - variable lengths, subsets of frames, and even different image-domain frequencies - as a form of guidance for video generation? Importantly, CFG with flexible history is incompatible with existing diffusion model architectures and the most obvious fix significantly degrades sample quality (see Section 3).

To address these limitations, we propose the Diffusion Forcing Transformer (DFoT), a video diffusion framework that enables flexible conditioning on any portion of the input history. Extending the "noising-as-masking" paradigm in Diffusion Forcing (Chen et al., 2024) to non-causal transformers, DFoT trains video diffusion models by applying independent noise levels to each frame. During sampling, portions of the history can be selectively masked with noise, enabling flexible conditioning and guidance. For instance, in CFG, the unconditional score corresponds to our model with the entire history masked out. Notably, DFoT is compatible with existing architectures such as DiT (Peebles & Xie, 2023) and U-ViT (Hoogeboom et al., 2023; 2024) and can be efficiently implemented through fine-tuning of pre-trained video diffusion models.

At sampling time, the DFoT facilitates a family of history-conditioned guidance methods, collectively referred to as *History Guidance* (HG). The simplest of these, *Vanilla History Guidance* (HG-v), uses an arbitrary length of history as the conditioning variable for CFG. Notably, even this simple method significantly enhances video quality. We further introduce two advanced methods enabled by the DFoT: *Temporal History Guidance* (HG-t) and *Fractional History Guidance* (HG-f) . These extend history guidance beyond a special case of CFG. Temporal History Guidance combines scores from different history windows. Fractional History Guidance conditions on history windows corrupted by varying levels of noise, effectively acting as a "low-pass filter" on historical frames. With minor modifications, it can also target specific *frequency bandwidths* to enhance the dynamic degree of generated videos (hence the frequency-based terminology). Together, we compose HG-t and HG-f to create a comprehensive history guidance paradigm, which we term *history guidance across time and frequency* (HG-tf).

The Diffusion Forcing Transformer and associated History Guidance methods dramatically improve the quality and consistency of video generation, enabling the creation of exceptionally long videos through autoregressive extension,

*Equal contribution [1]MIT. Correspondence to: Kiwhan Song <kiwhan@mit.edu>, Boyuan Chen <boyuanc@mit.edu>.

Preprint. Under review.

Increasingly close to camera-conditioned video diffusion models (more not these later!)

History-Guided Video Diffusion

Kiwhan Song* [1] **Boyuan Chen*** [1] **Max Simchowitz** [1] **Yilun Du** [1] **Russ Tedrake** [1] **Vincent Sitzmann** [1]

Abstract

Classifier-free guidance (CFG) is a key technique for improving conditional generation in diffusion models, enabling more accurate control while enhancing sample quality. It is natural to extend this technique to video diffusion, which generates video conditioned on a variable number of context frames, collectively referred to as history. However, we find two key challenges to guiding with variable-length history: architectures that only support fixed-size conditioning, and the empirical observation that CFG-style history dropout performs poorly. To address this, we propose the Diffusion Forcing Transformer (DFoT), a video diffusion architecture and theoretically grounded training objective that jointly enable conditioning on a flexible number of history frames. We then introduce *History Guidance*, a family of guidance methods uniquely enabled by DFoT. We show that its simplest form, *vanilla history guidance*, already significantly improves video generation quality and temporal consistency. A more advanced method, *history guidance across time and frequency* further enhances motion dynamics, enables compositional generalization to out-of-distribution history, and can stably roll out extremely long videos. Website: this URL

1 Introduction

Diffusion models are effective generative models in domains such as image, sound, and video. Critical to their success is classifier-free guidance (CFG) (Ho & Salimans, 2022), which trades off between sample quality and diversity by jointly training a conditional and an unconditional diffusion model and combining their score estimates when sampling.

In the realm of video generative models, CFG commonly relies on either text or image prompts as conditioning variables. Yet, another conditioning variable, namely the entire collection of previous video frames, or *history*, deserves further exploration. In this paper, we investigate the following question: Can we use different portions of history - variable lengths, subsets of frames, and even different image-domain frequencies - as a form of guidance for video generation? Importantly, CFG with flexible history is incompatible with existing diffusion model architectures and the most obvious fix significantly degrades sample quality (see Section 3).

To address these limitations, we propose the Diffusion Forcing Transformer (DFoT), a video diffusion framework that enables flexible conditioning on any portion of the input history. Extending the "noising-as-masking" paradigm in Diffusion Forcing (Chen et al., 2024) to non-causal transformers, DFoT trains video diffusion models by applying independent noise levels to each frame. During sampling, portions of the history can be selectively masked with noise, enabling flexible conditioning and guidance. For instance, in CFG, the unconditional score corresponds to our model with the entire history masked out. Notably, DFoT is compatible with existing architectures such as DiT (Peebles & Xie, 2023) and U-ViT (Hoogeboom et al., 2023; 2024) and can be efficiently implemented through fine-tuning of pre-trained video diffusion models.

At sampling time, the DFoT facilitates a family of history-conditioned guidance methods, collectively referred to as *History Guidance* (HG). The simplest of these, *Vanilla History Guidance* (HG-v), uses an arbitrary length of history as the conditioning variable for CFG. Notably, even this simple method significantly enhances video quality. We further introduce two advanced methods enabled by the DFoT: *Temporal History Guidance* (HG-t) and *Fractional History Guidance* (HG-f) . These extend history guidance beyond a special case of CFG. Temporal History Guidance combines scores from different history windows. Fractional History Guidance conditions on history windows corrupted by varying levels of noise, effectively acting as a "low-pass filter" on historical frames. With minor modifications, it can also target specific *frequency bandwidths* to enhance the dynamic degree of generated videos (hence the frequency-based terminology). Together, we compose HG-t and HG-f to create a comprehensive history guidance paradigm, which we term *history guidance across time and frequency* (HG-tf).

The Diffusion Forcing Transformer and associated History Guidance methods dramatically improve the quality and consistency of video generation, enabling the creation of exceptionally long videos through autoregressive extension,

*Equal contribution [1]MIT. Correspondence to: Kiwhan Song <kiwhan@mit.edu>, Boyuan Chen <boyuanc@mit.edu>.

Preprint. Under review.

Increasingly close to camera-conditioned video diffusion models (more not these later!)