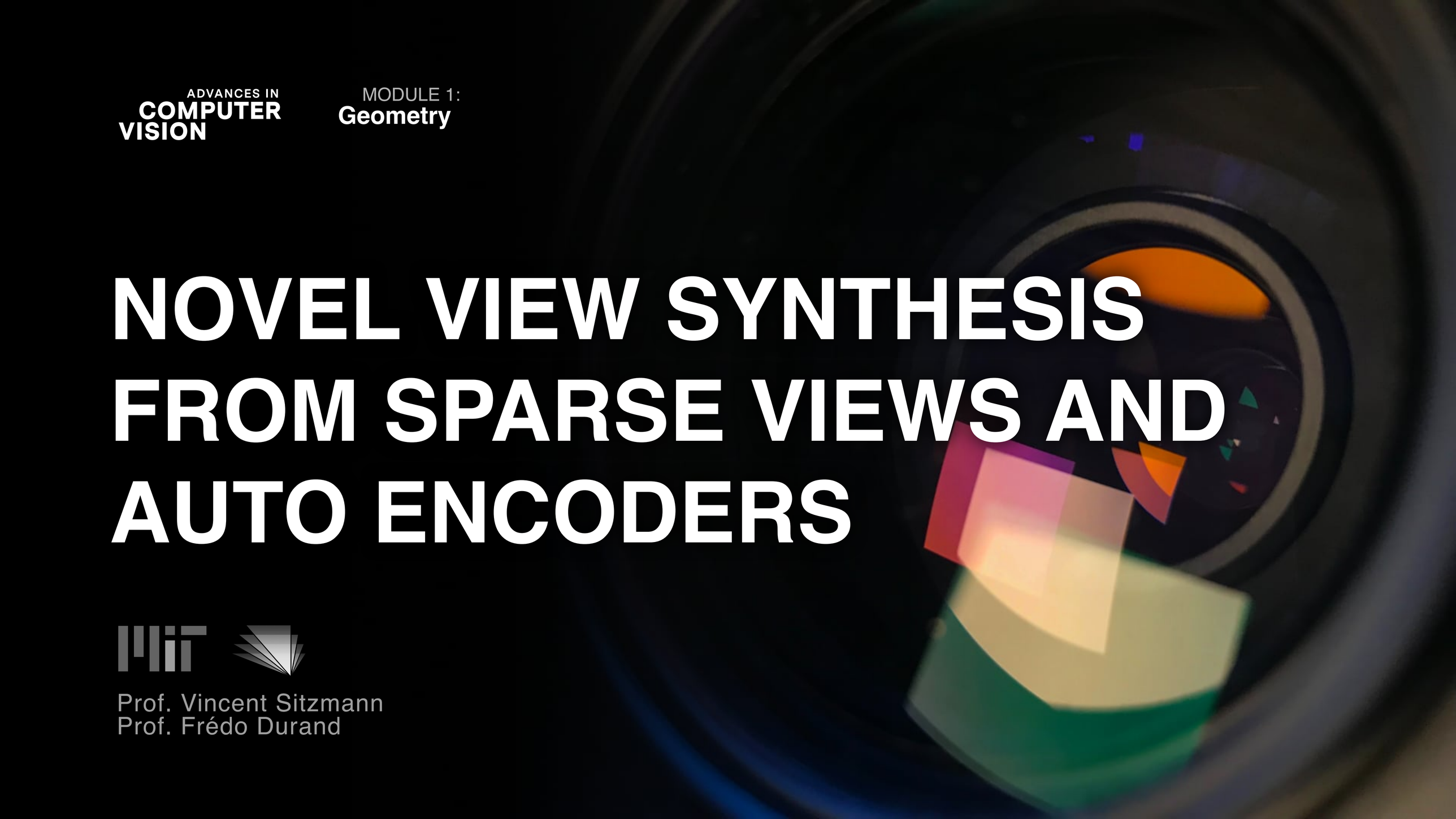
ADVANCES IN COMPUTER VISION

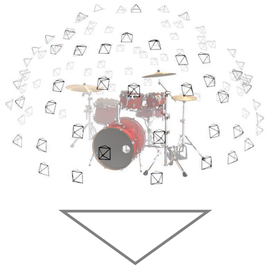
MODULE 1: Geometry

NOVEL VIEW SYNTHESIS FROM SPARSE VIEWS AND AUTO ENCODERS



Prof. Vincent Sitzmann
Prof. Frédo Durand

Recap: Differentiable/Inverse Rendering
Overfitting to a single scene



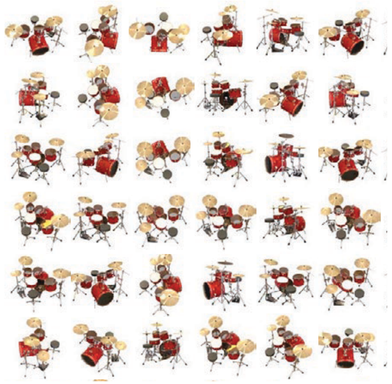
Scene Representation

Neural Renderer

Optimization via SGD



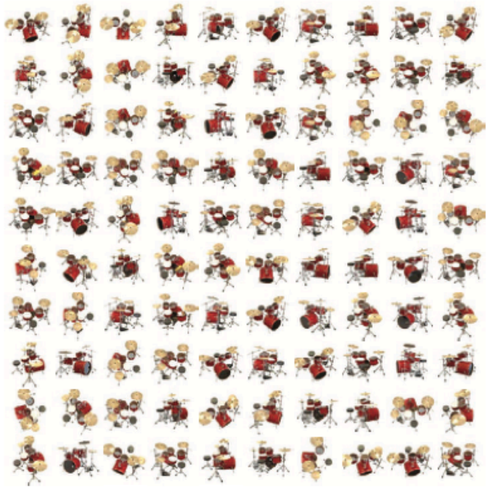
Rendered Images



GT Images

Single scene with many observations.

Input Images



Optimize NeRF

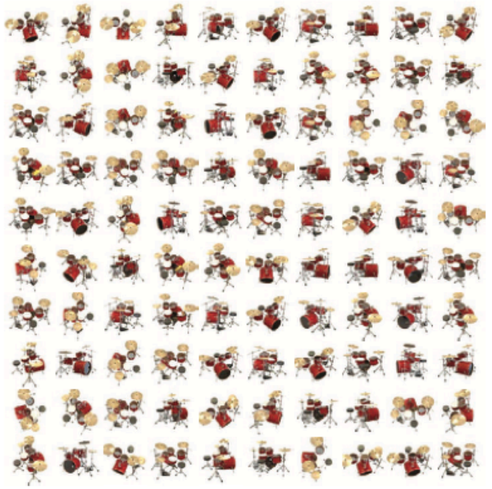


Render new views



Single scene with many observations.

Input Images



Optimize NeRF



Render new views



How about scene with **not so <u>many</u>** observations ??

Input Images



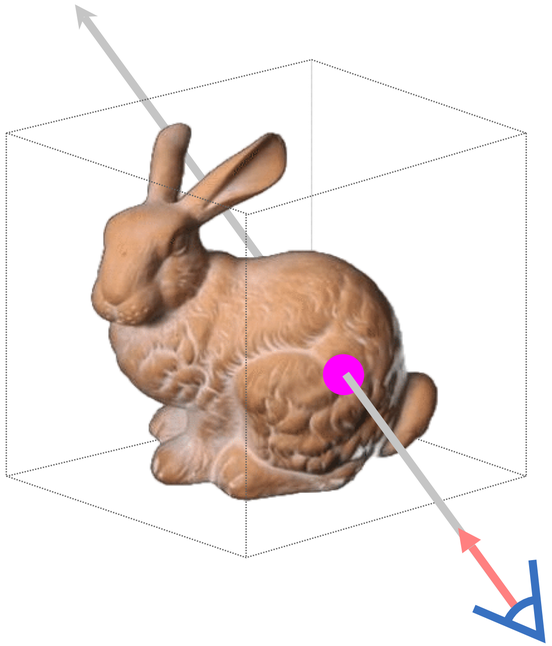
Optimize

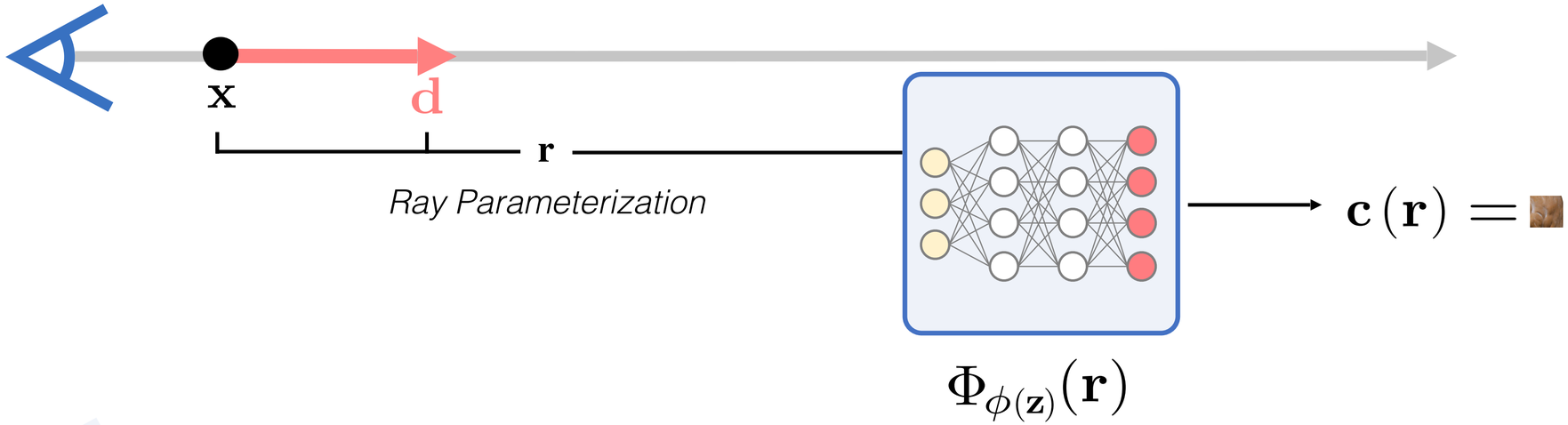
?

Render new views

?

Case study: Light Field Networks





An Alternative Scene Representation

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



Context Views



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



Context Views



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



Context Views



Intermediate Views

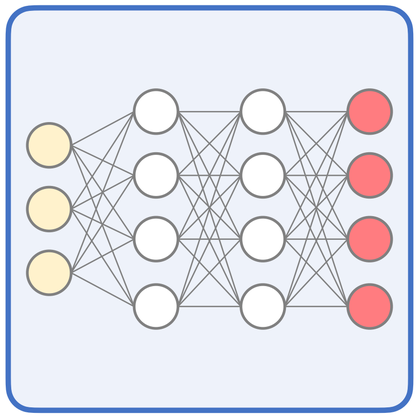
Today: *Learned* inference algorithms!

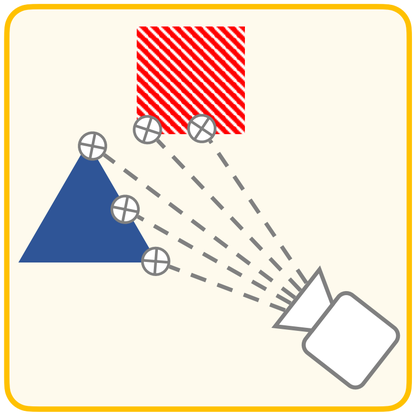
Observations



Inference

?



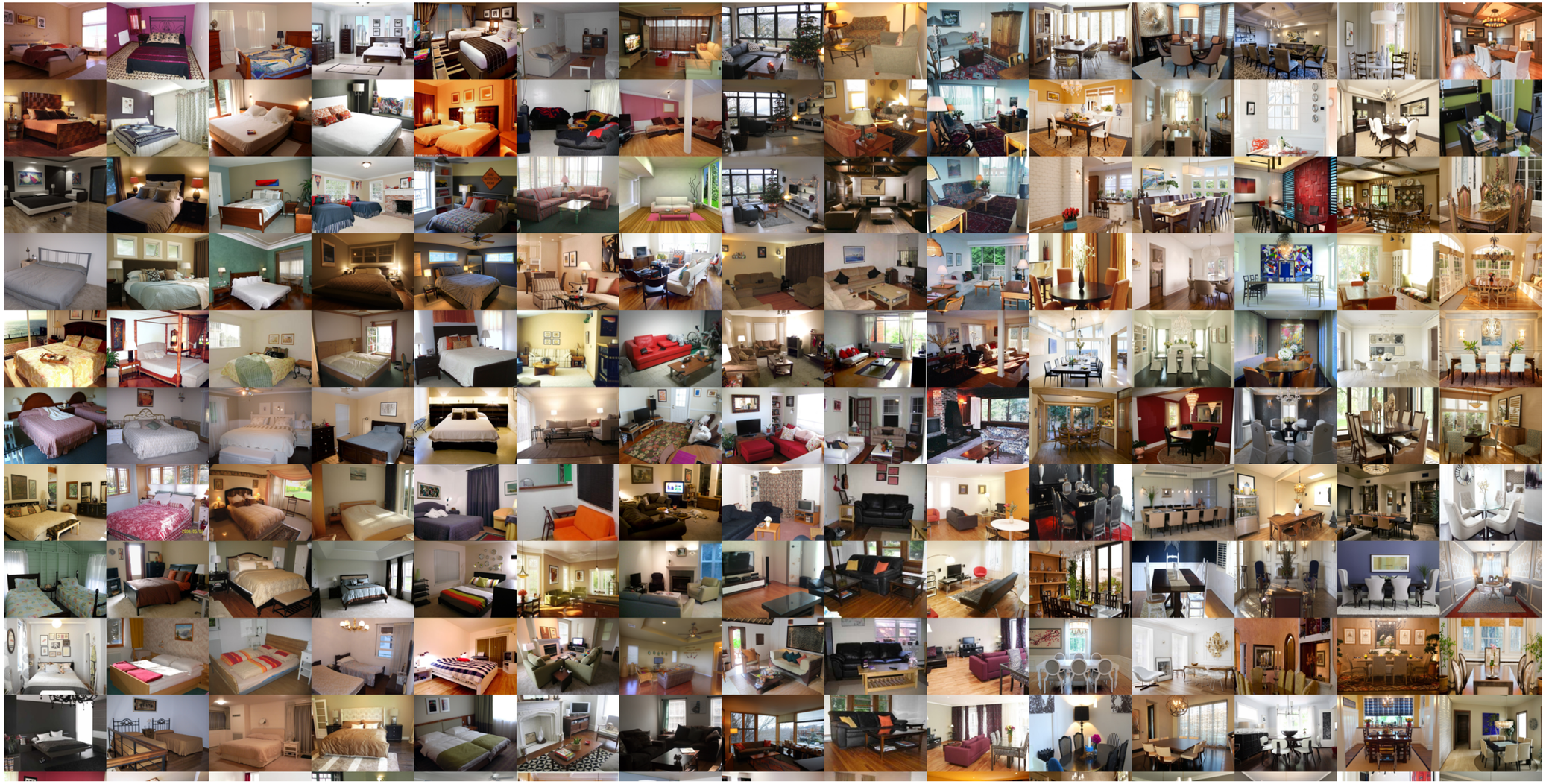


Renderings



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

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

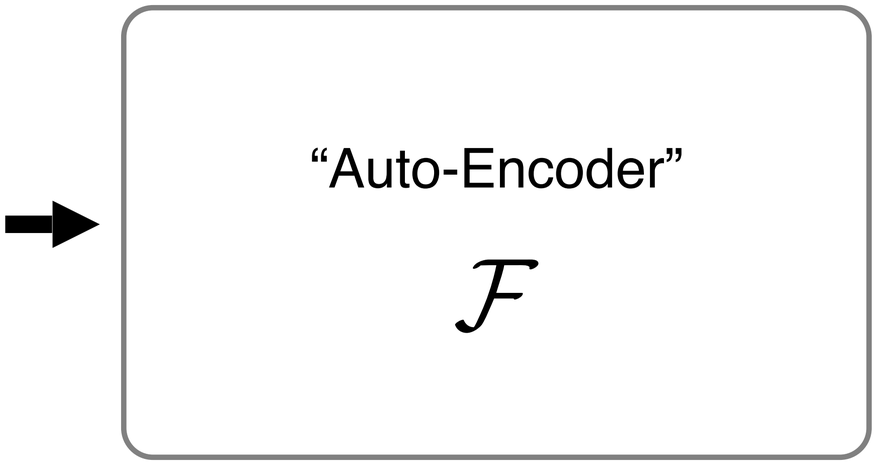


Auto-Encoding

$\mathbf{X}$



Image



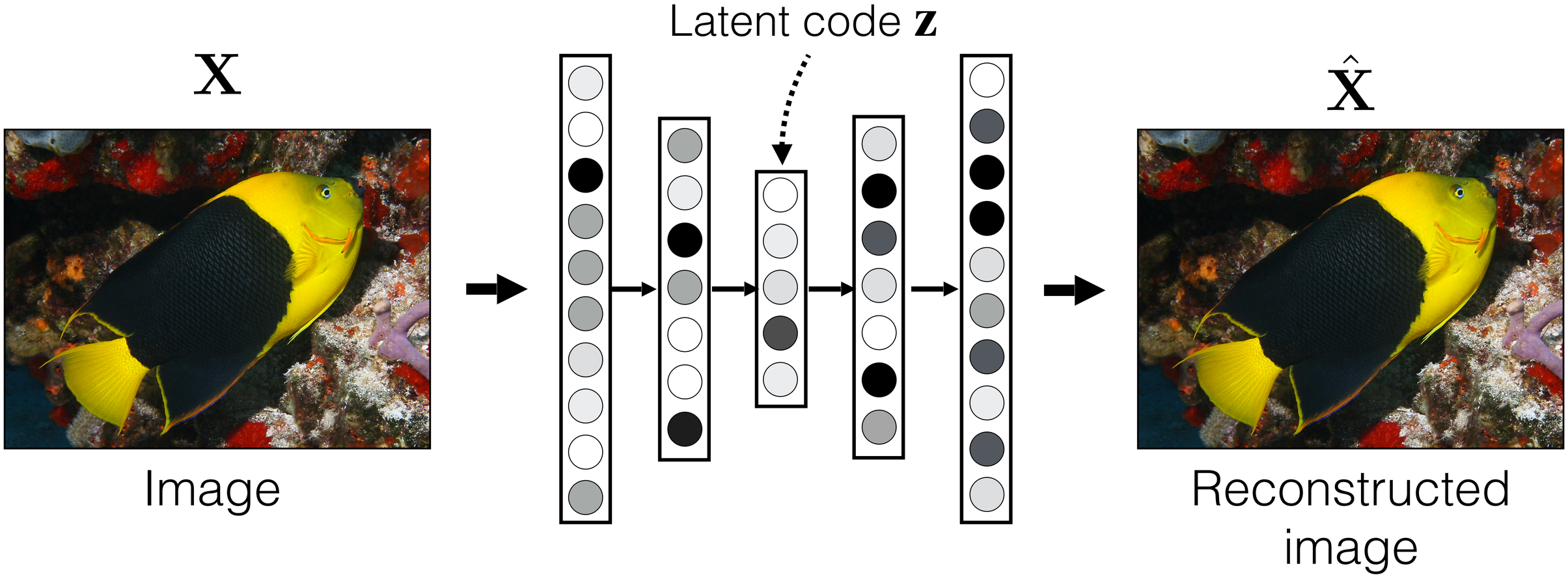
$\hat{\mathbf{X}}$



Reconstructed image

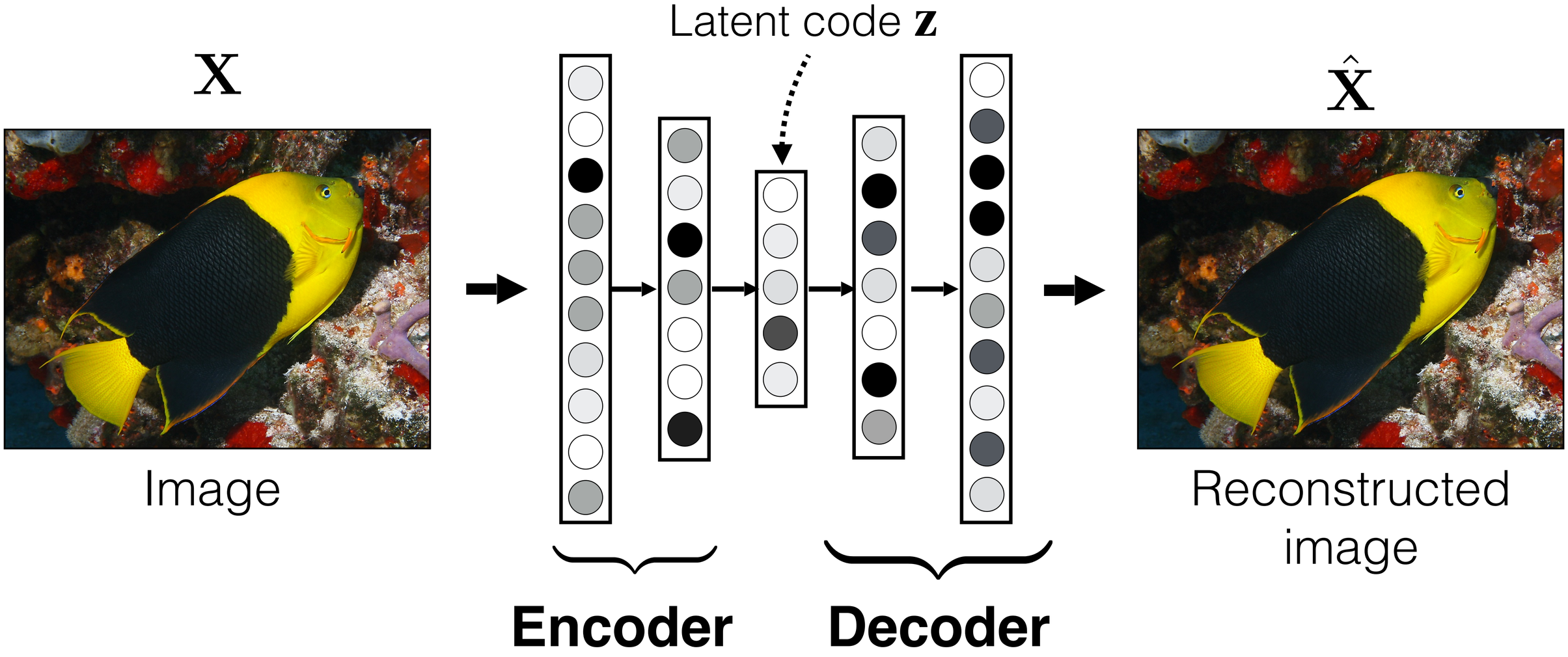
$$\arg\min_{\mathcal{F}} \mathbb{E}_{\mathbf{X}}[||\mathcal{F}(\mathbf{X}) - \mathbf{X}||]$$

Auto-Encoding

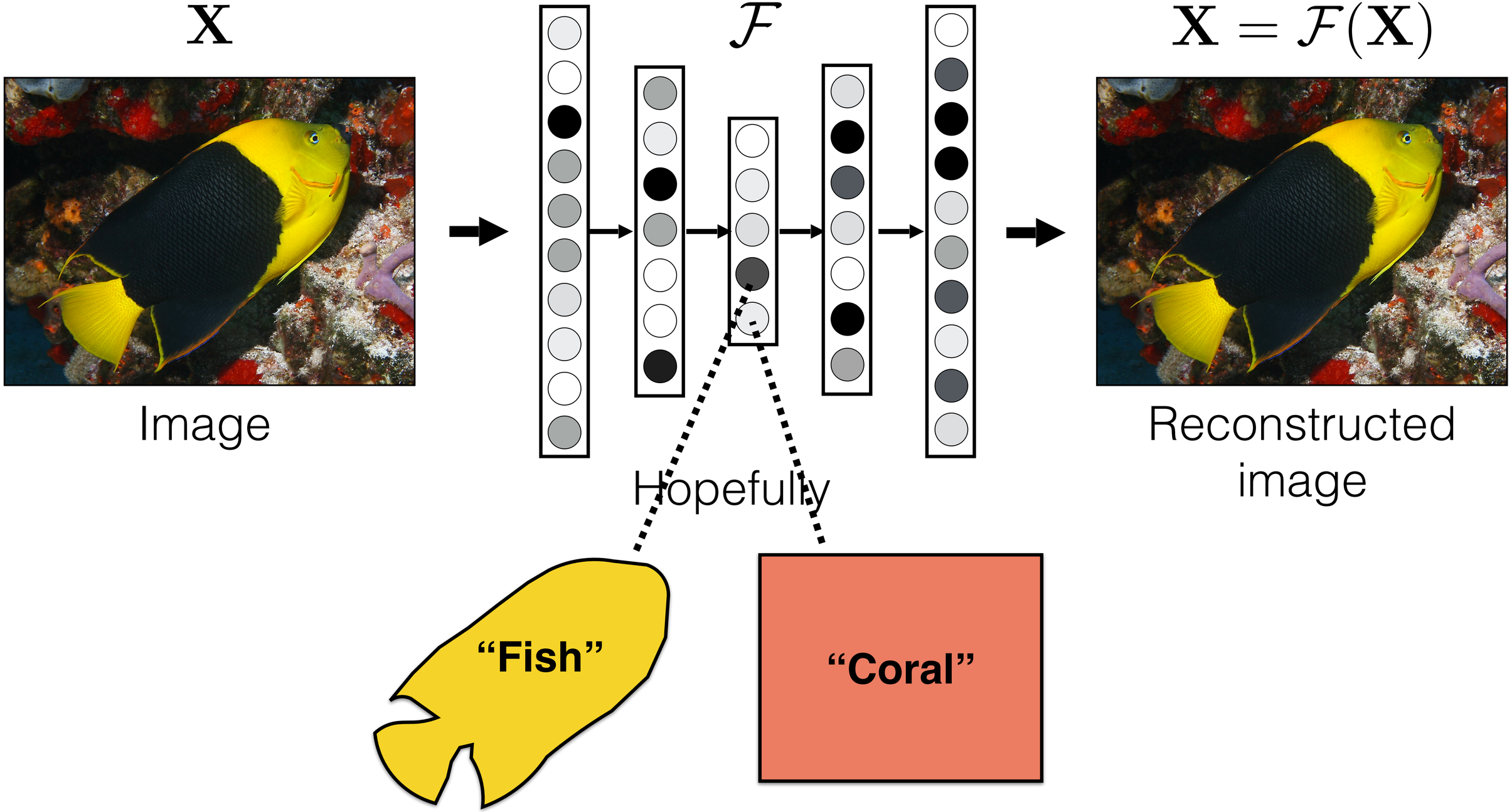


[e.g., Hinton & Salakhutdinov, Science 2006]

Auto-Encoding



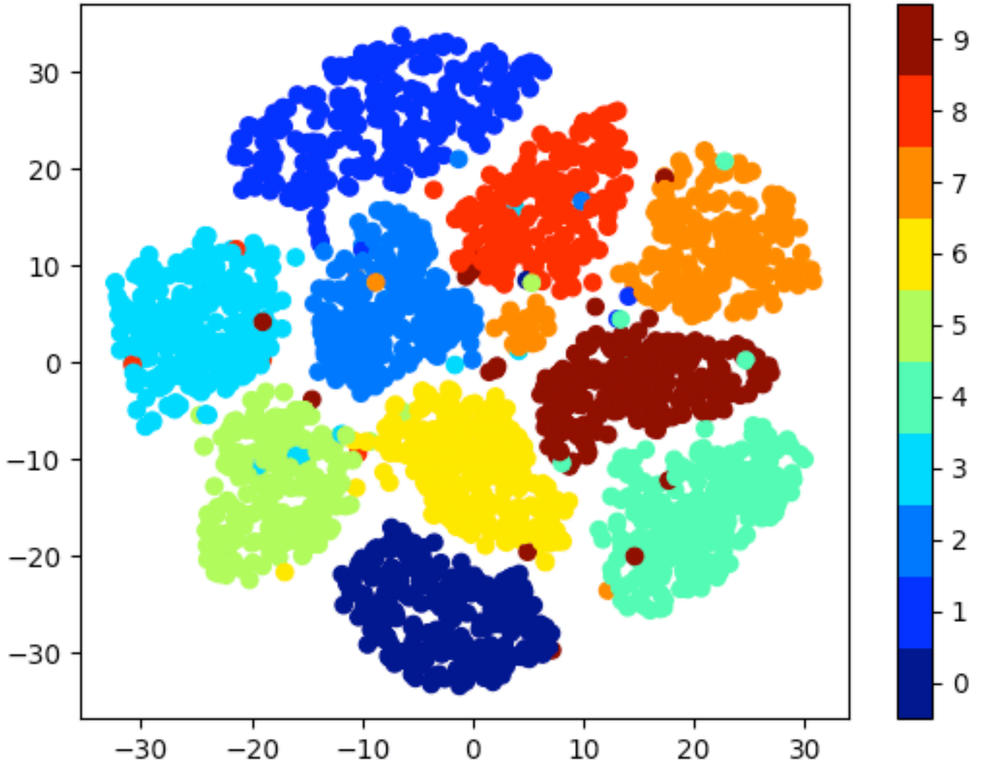
[e.g., Hinton & Salakhutdinov, Science 2006]



Learning Signal: **Forced Compression**

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





Side discussion: self-supervised learning

- General strategy:
  - Train on reconstructing input data
  - Make task hard in some way to force the model to learn something intelligent
    - Autoencoder: make hard by forcing latent to be small
    - Masked autoencoders: make hard by masking part of the input
- DINO a bit different: force to learn same thing regardless of crop.
  You can argue that cropping is the same as masking. But loss is different.

Self-supervised Scene Representation Learning

Latent 3D Scenes

{  ,  , . . . }

Self-supervised Scene Representation Learning

Latent 3D Scenes

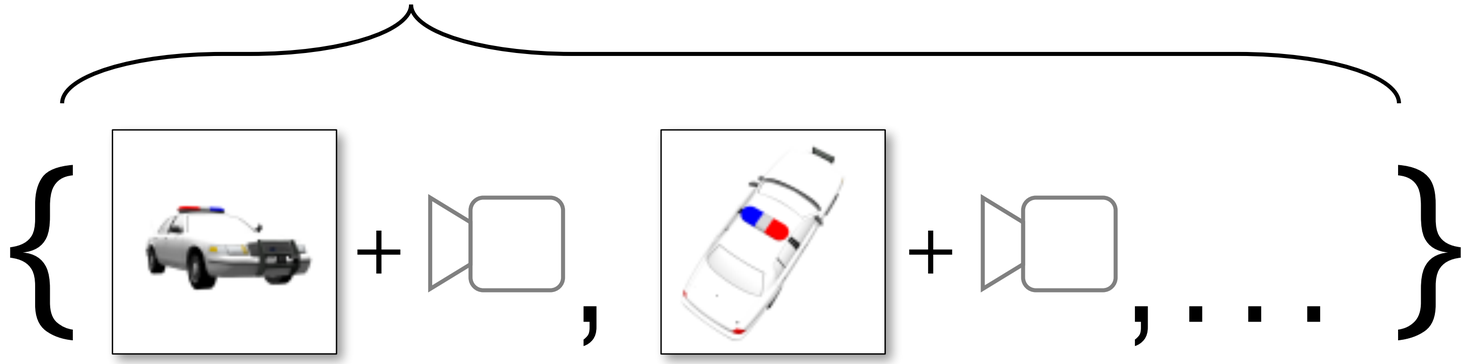
{  ,  , . . . }

Self-supervised Scene Representation Learning

Latent 3D Scenes



Observations
Image + Pose & Intrinsics

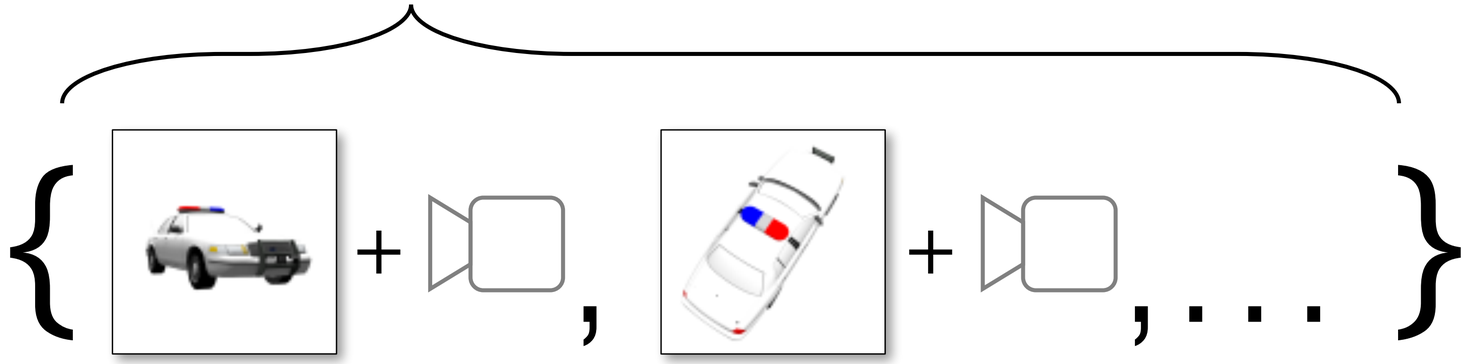


Self-supervised Scene Representation Learning

Latent 3D Scenes



Observations
Image + Pose & Intrinsics



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

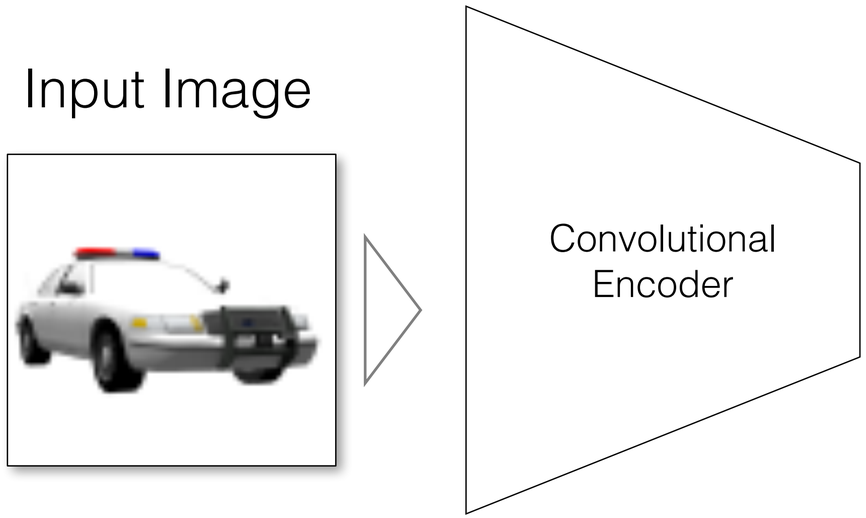
Vision: Learn rich representations just by watching video!

2D baseline: Auto-Encoder-Like model

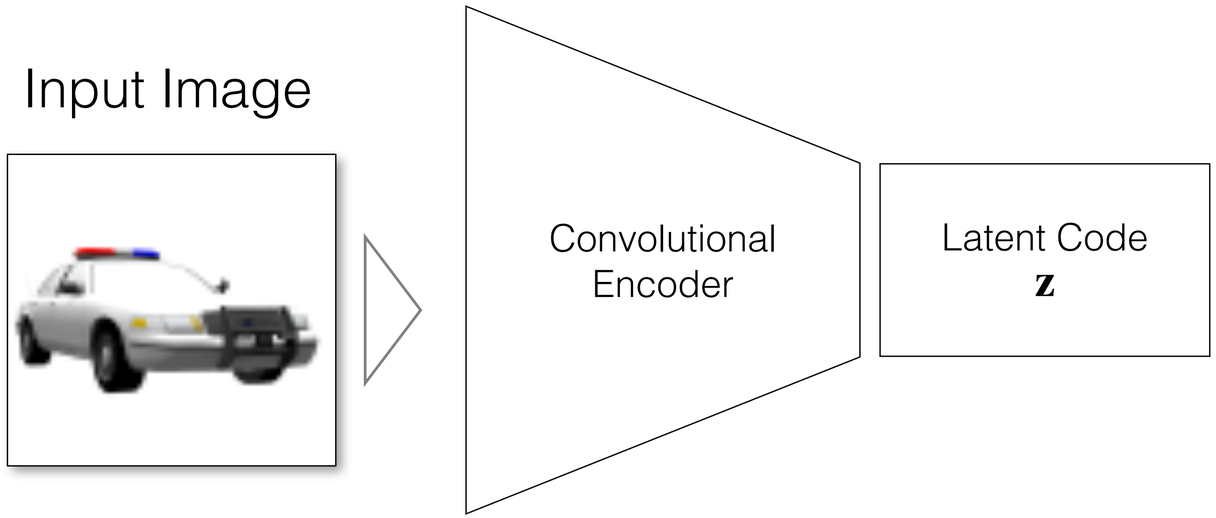
Input Image



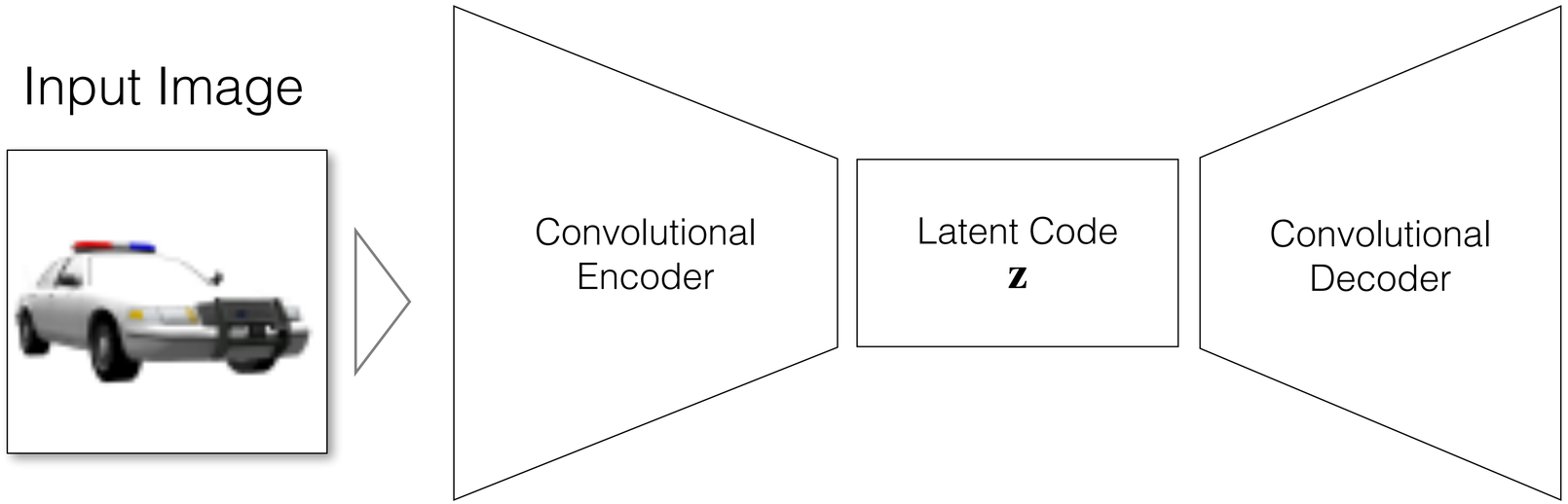
2D baseline: Auto-Encoder-Like model



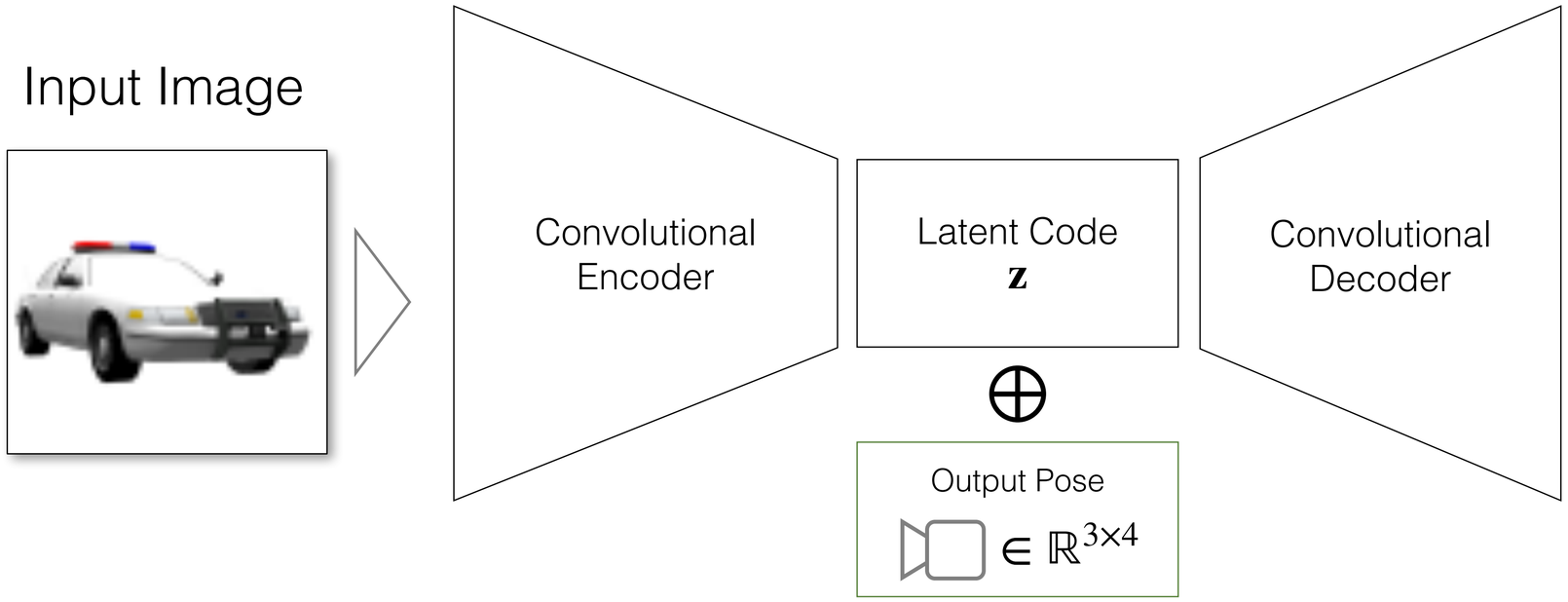
2D baseline: Auto-Encoder-Like model



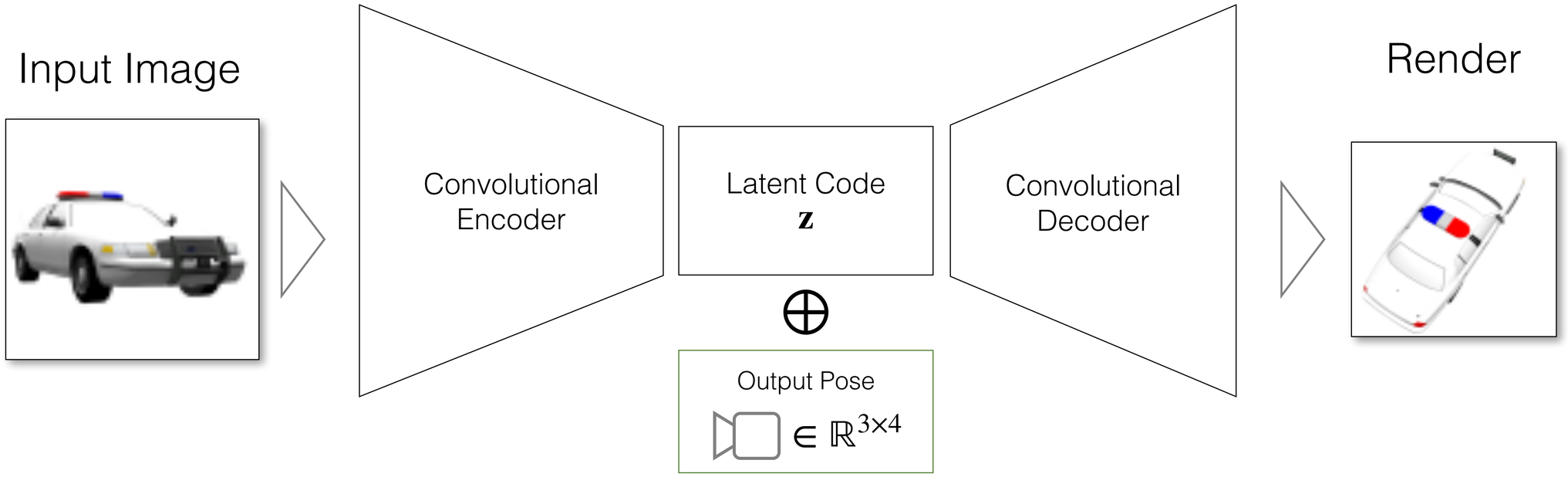
2D baseline: Auto-Encoder-Like model



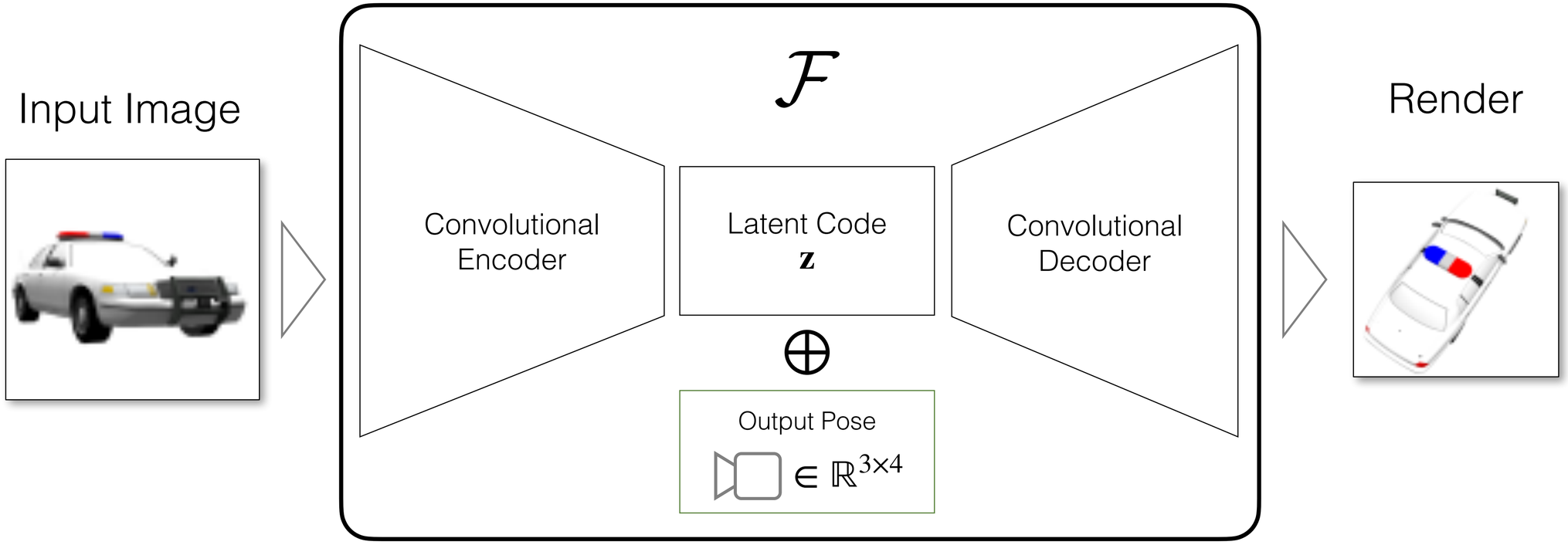
2D baseline: Auto-Encoder-Like model



2D baseline: Auto-Encoder-Like model



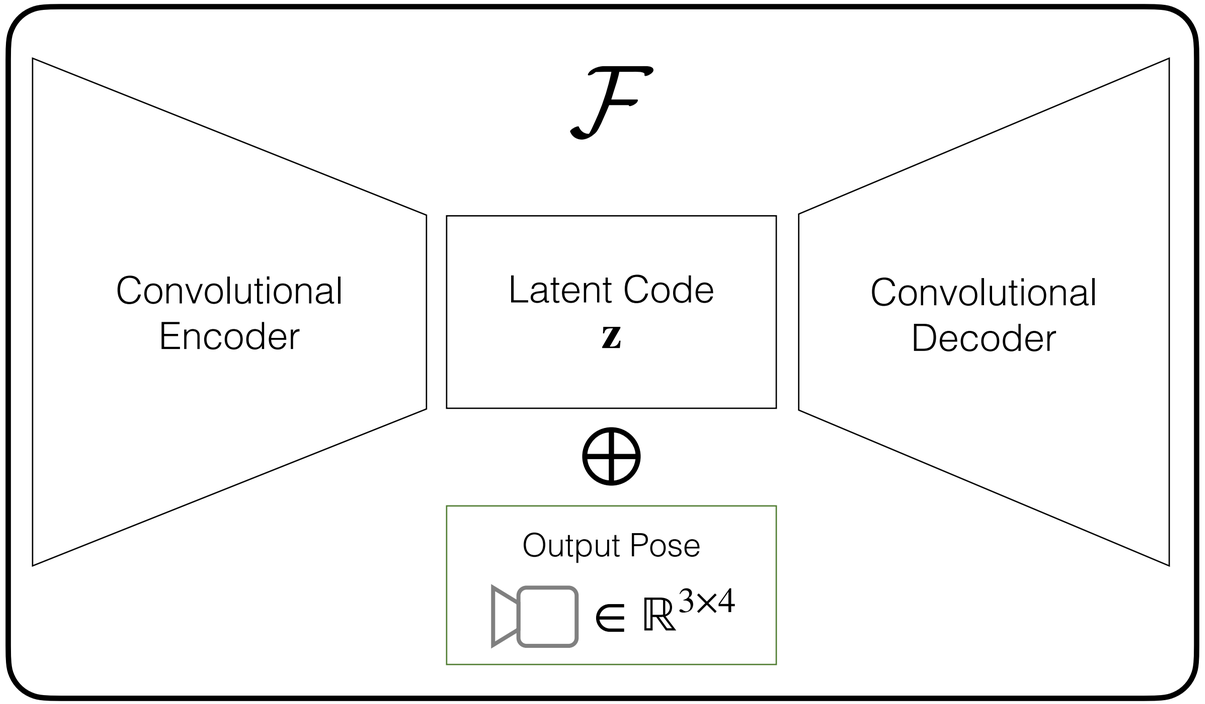
2D baseline: Auto-Encoder-Like model



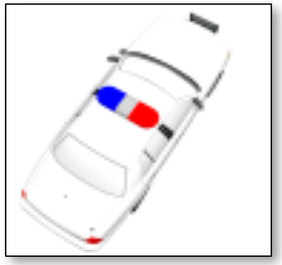
2D baseline: Auto-Encoder-Like model

Input Image





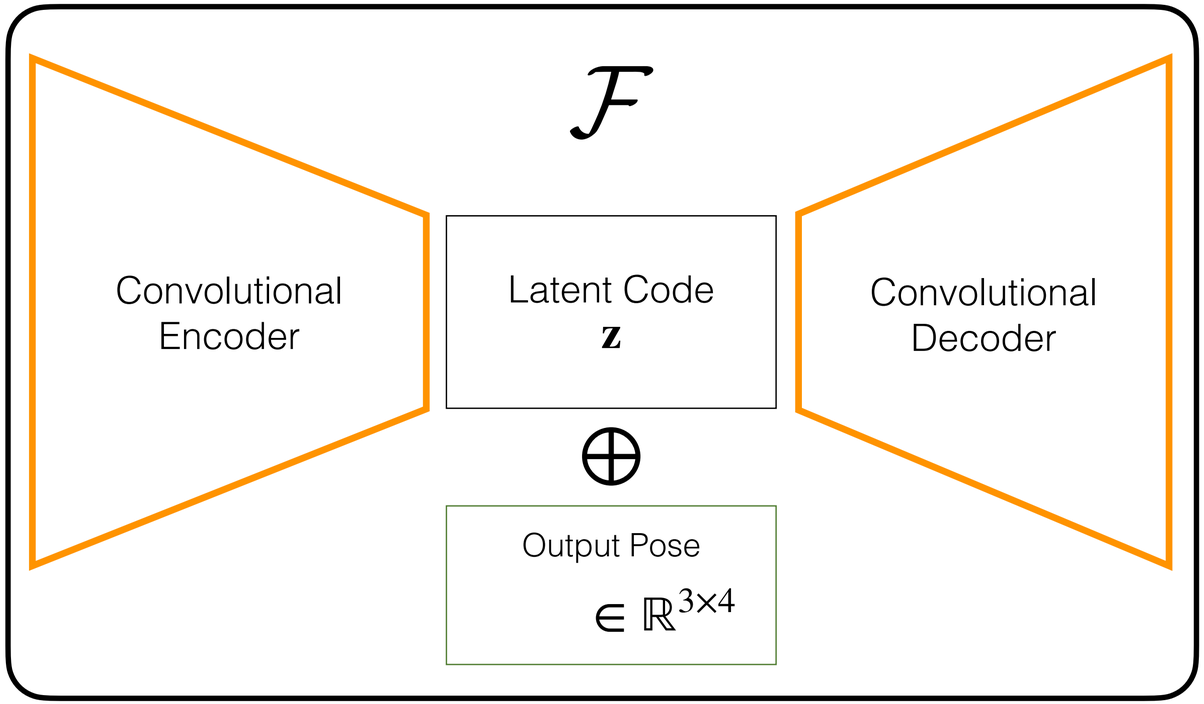
Render



2D baseline: Auto-Encoder-Like model

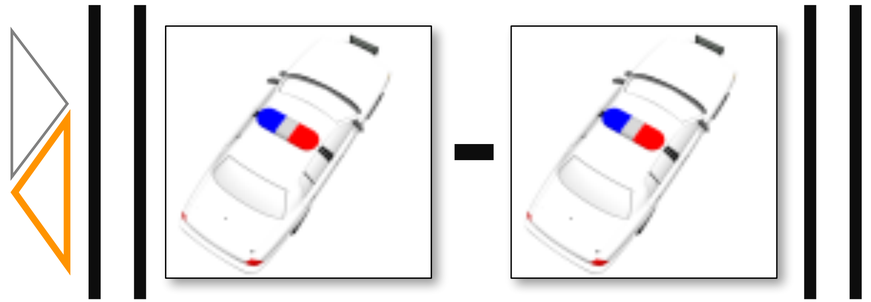
Input Image





Render

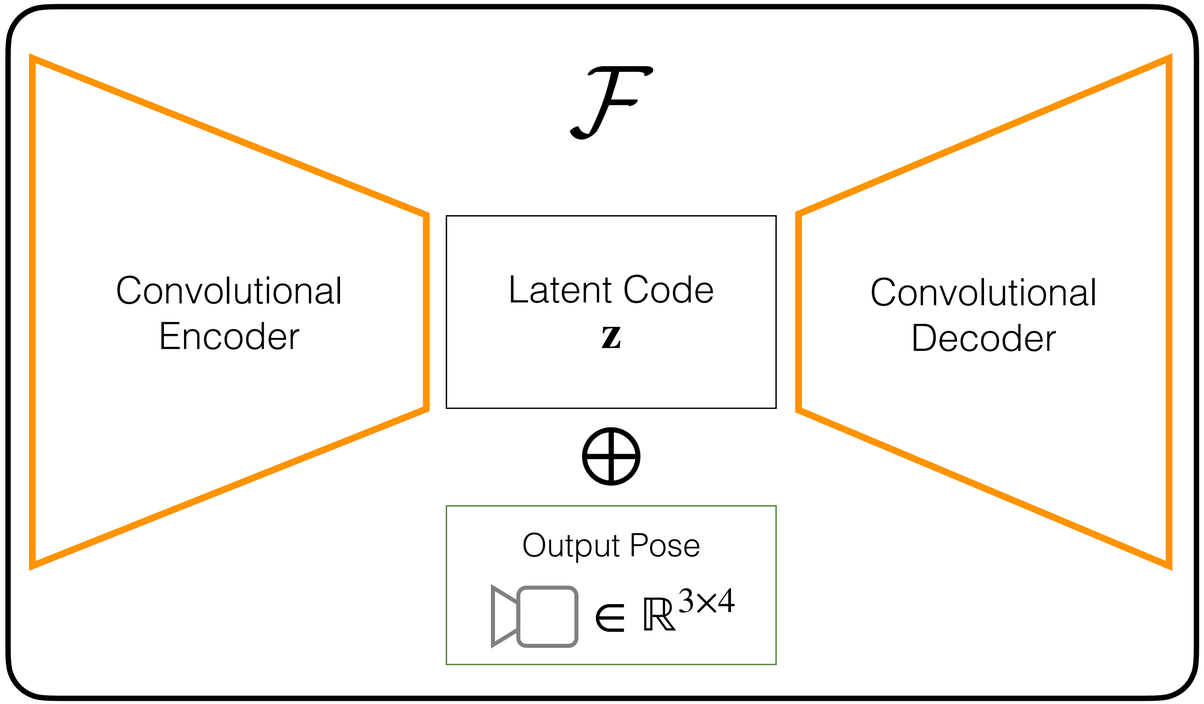
GT



2D baseline: Auto-Encoder-Like model

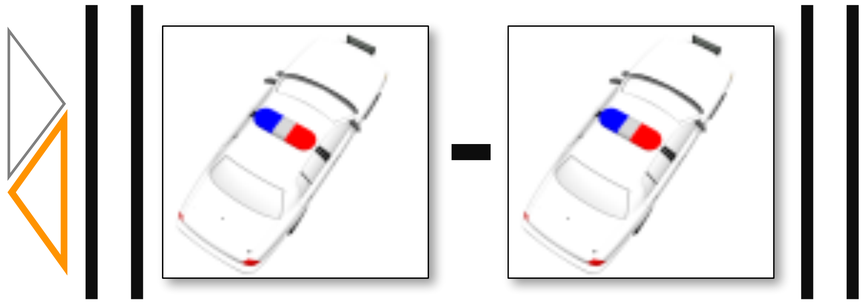
Input Image





Render

GT



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 fully 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.



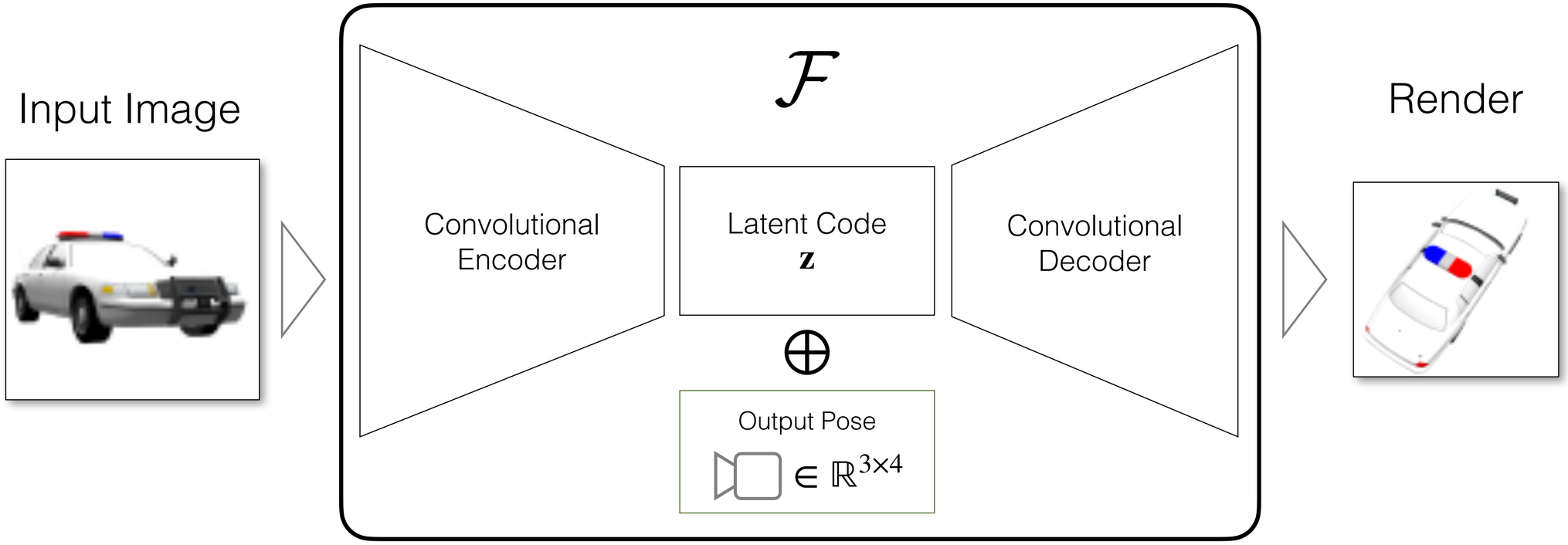
Doesn’t capture 3D properties of scenes.

Trained on ~2500 shapenet cars with 50 observations each.

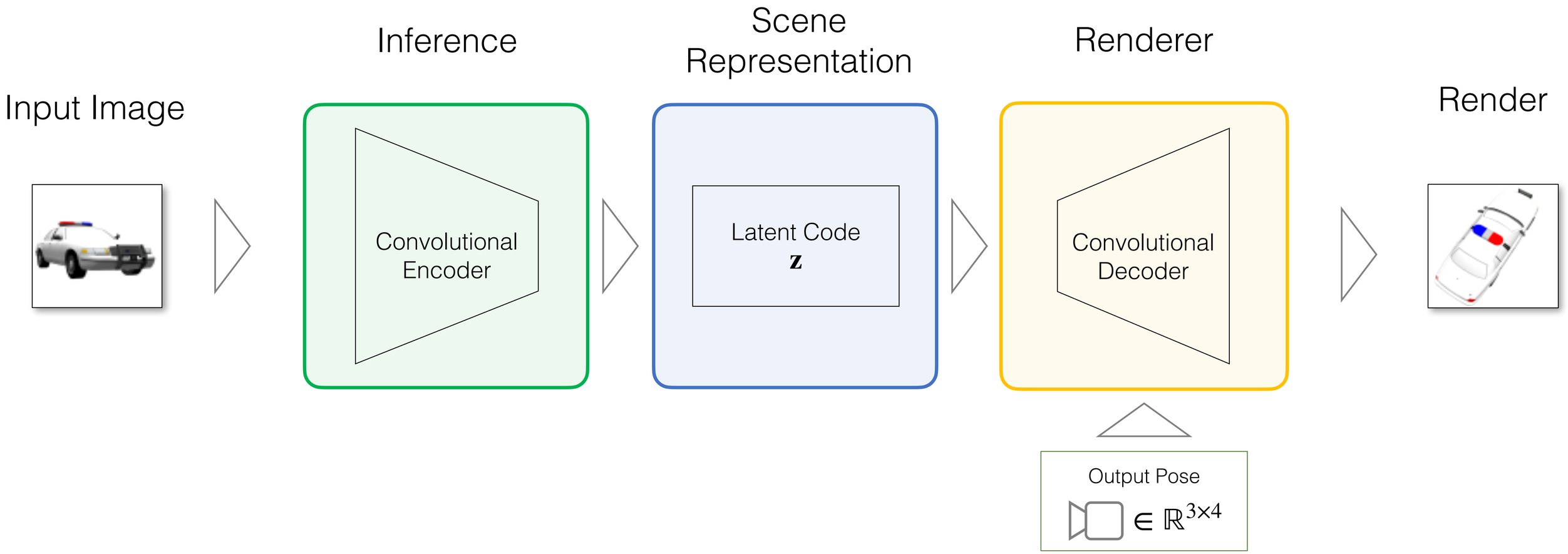


First hypothesis: Doesn’t have 3D structure, therefore, just finds some function that explains the training set that ends up being different from the “true” function.

2D baseline: Auto-Encoder-Like model

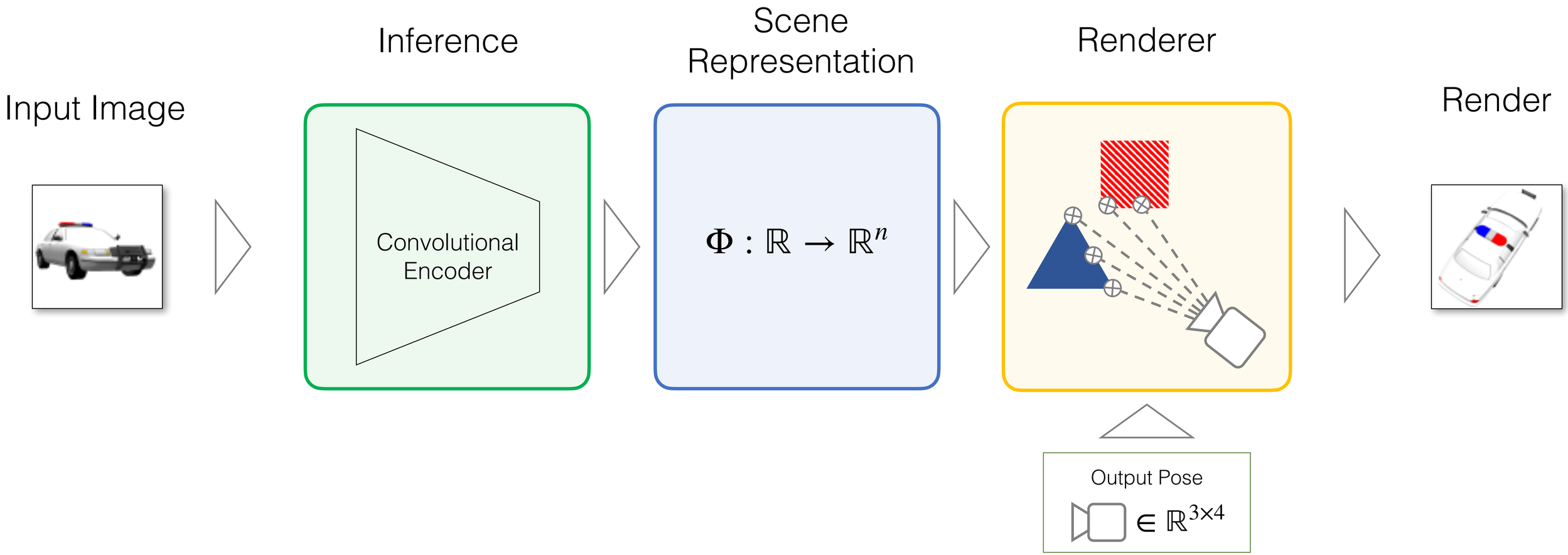


2D baseline: Auto-Encoder-Like model

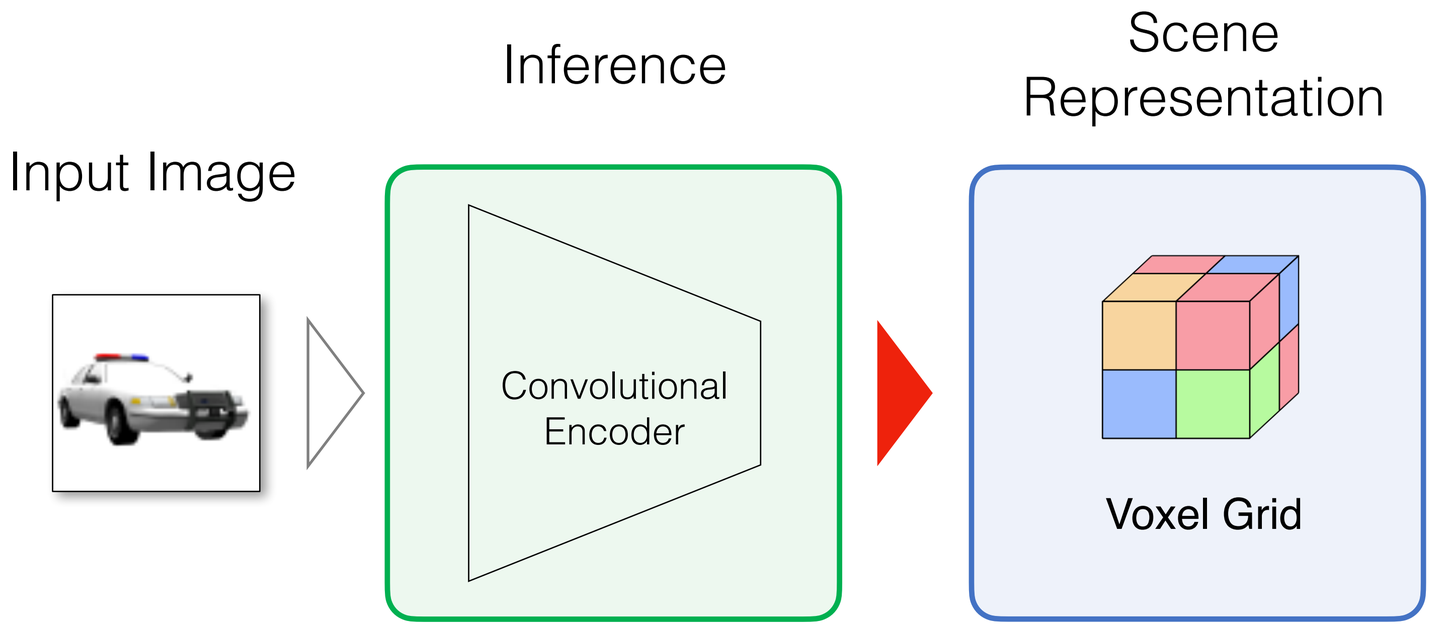


3D-Structured Decoder

Inductive bias

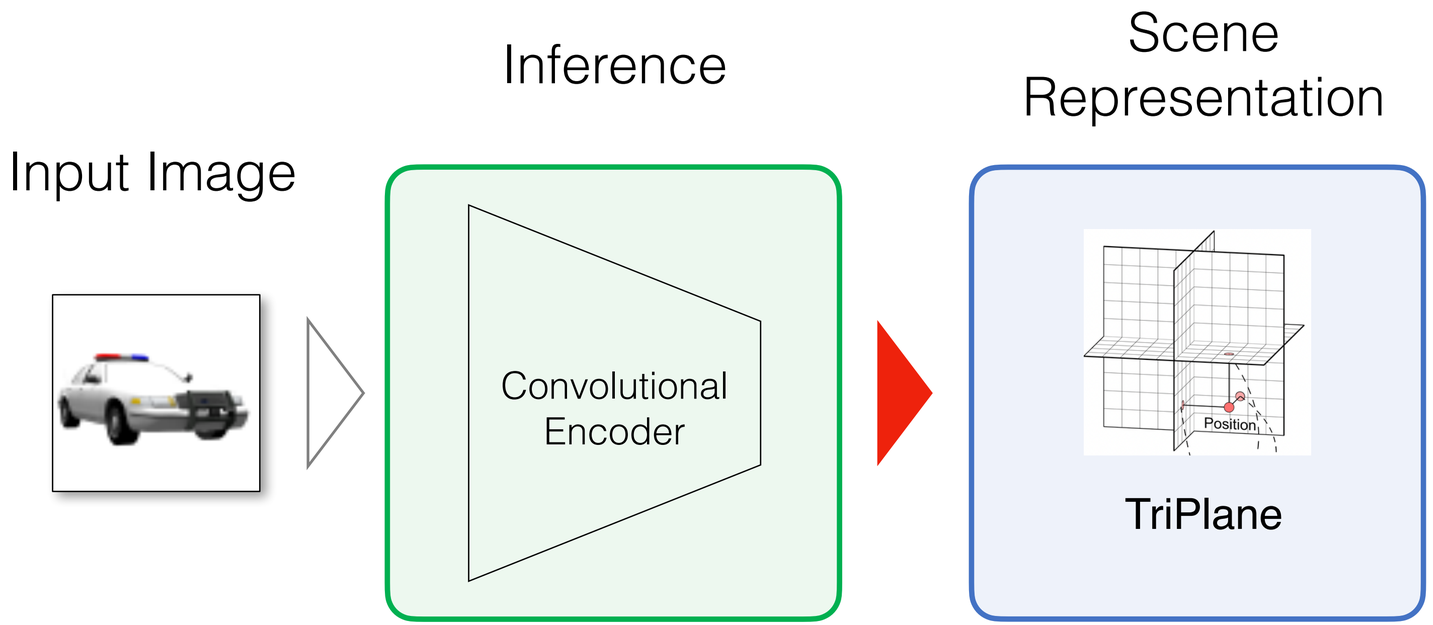


3D-Structured Decoder



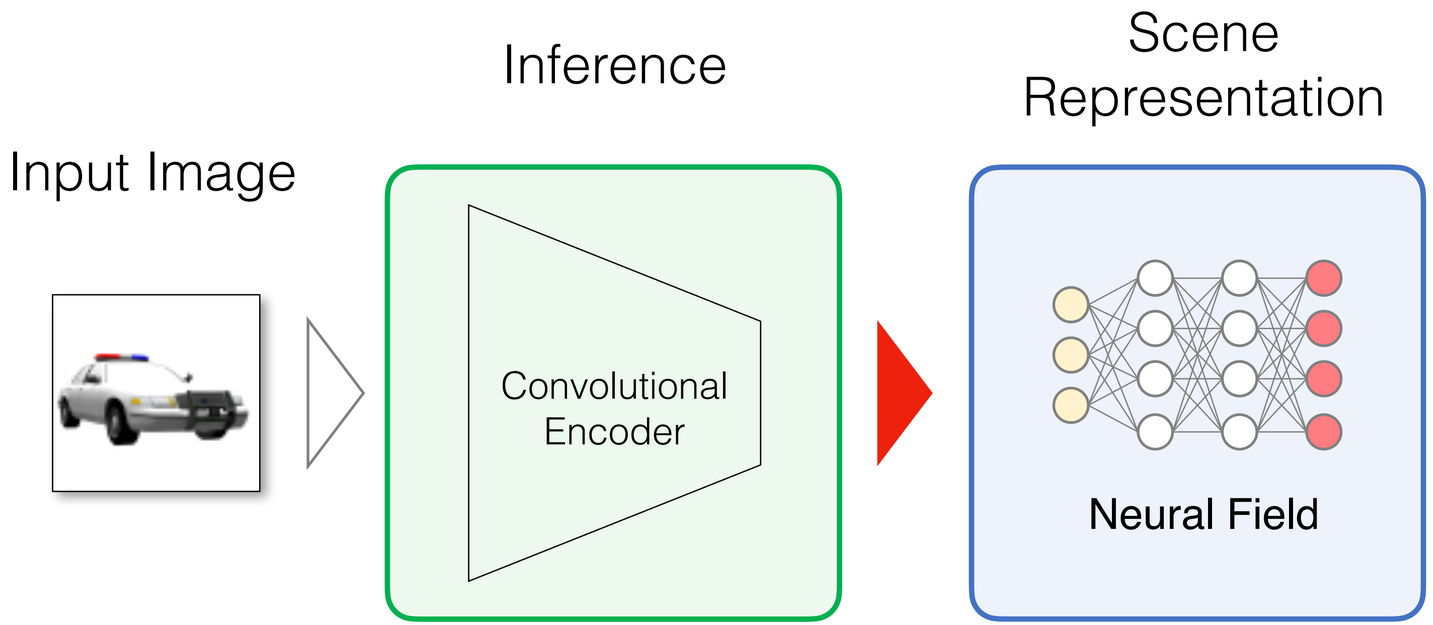
Encoder predict the parameters of the scene representation.

3D-Structured Decoder



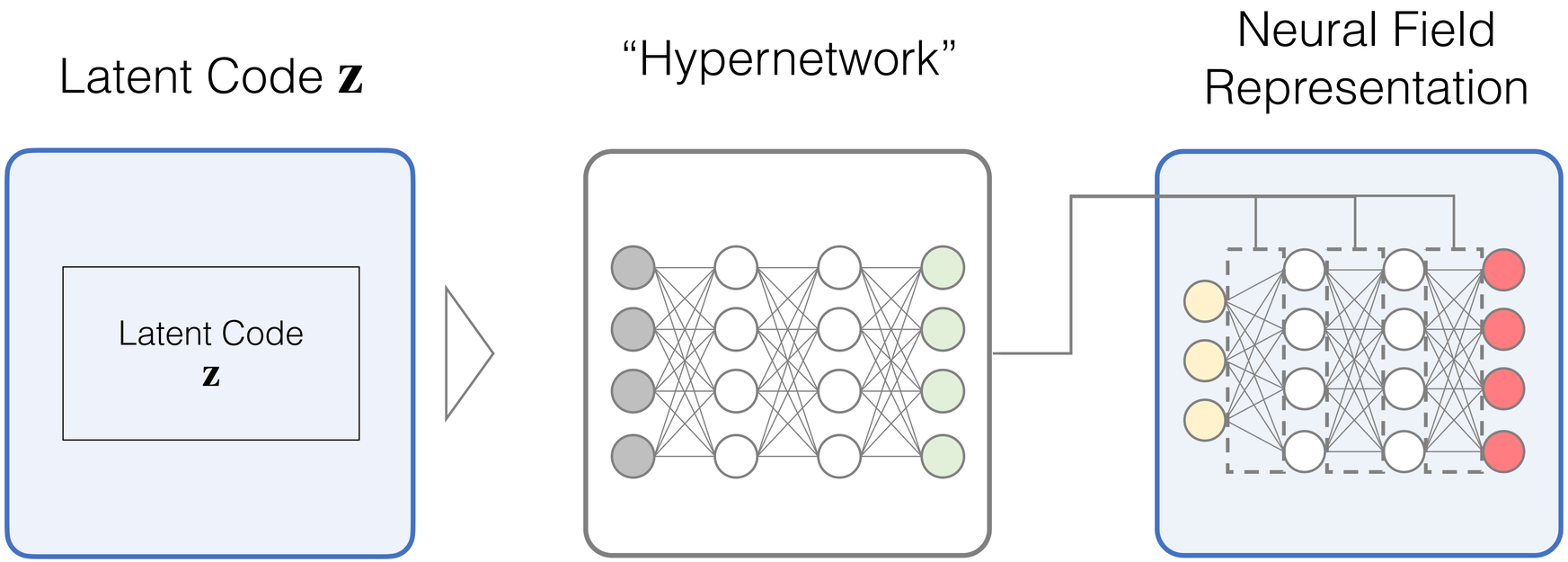
Encoder predict the parameters of the scene representation.

3D-Structured Decoder



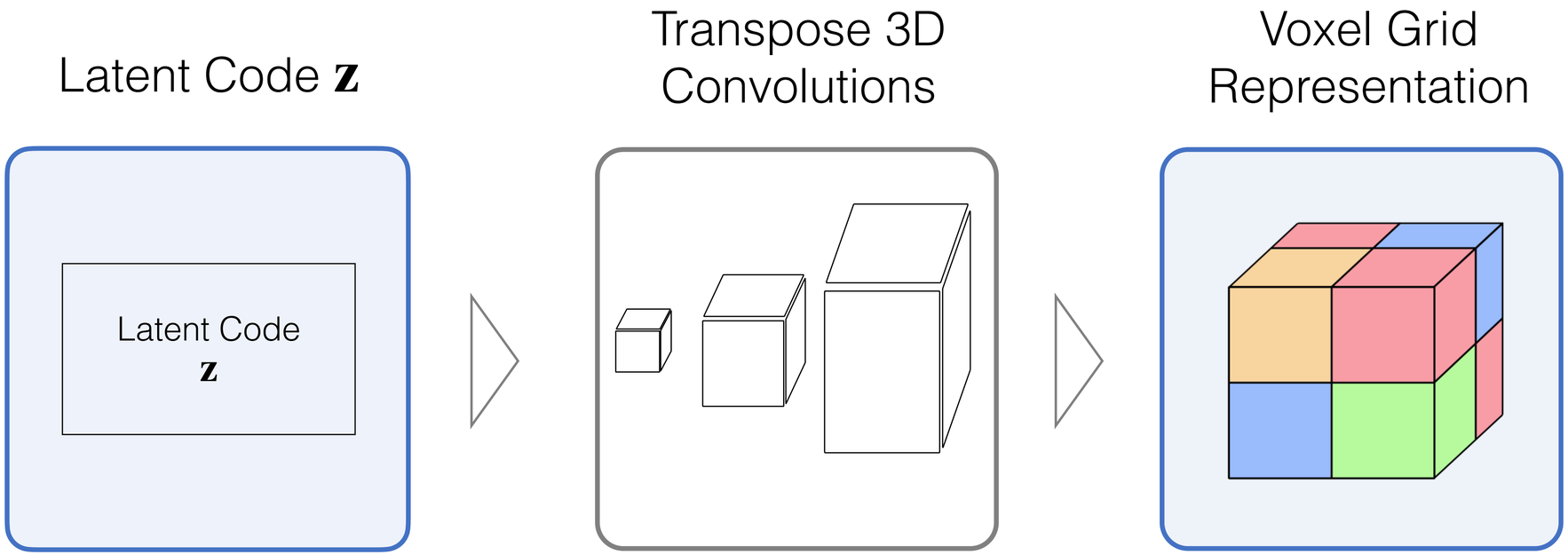
Encoder predict the parameters of the scene representation.

When representation is a neural network: Conditioning Neural Fields by predicting parameters



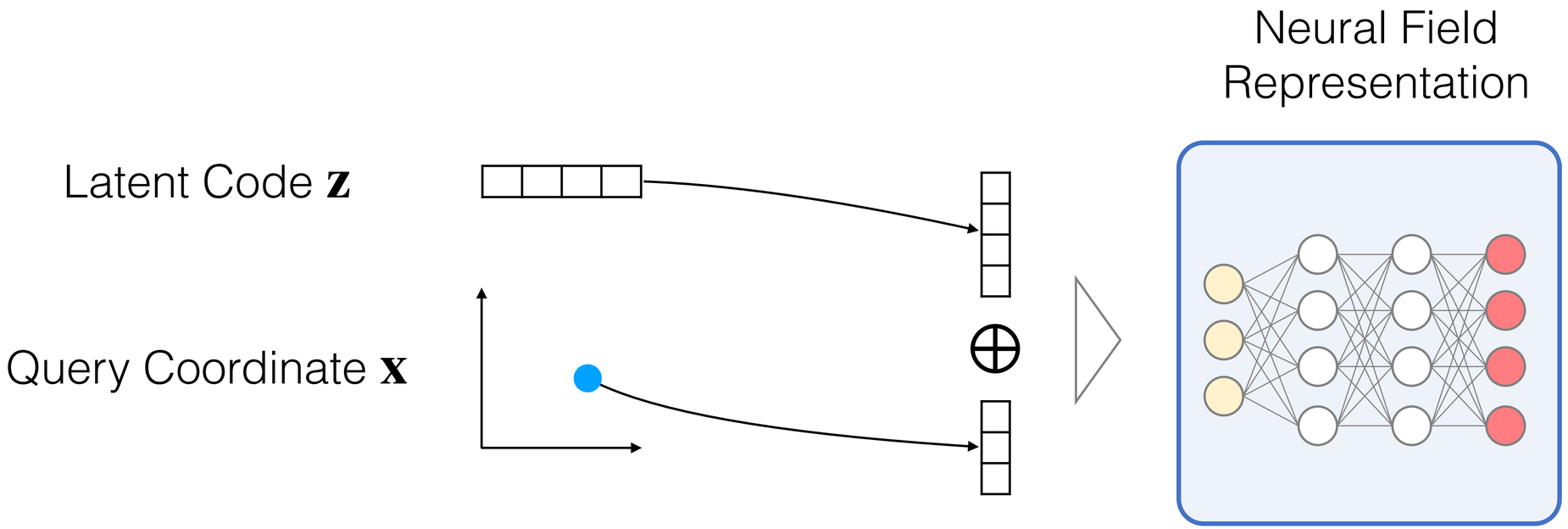
To condition neural field, can literally have MLP that takes as input $\mathbf{z}$ and outputs *all* of the parameters (weights, biases) of the neural field.

Conditioning Voxel Grids by predicting parameters



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 $N \times N \times N \times ch$.

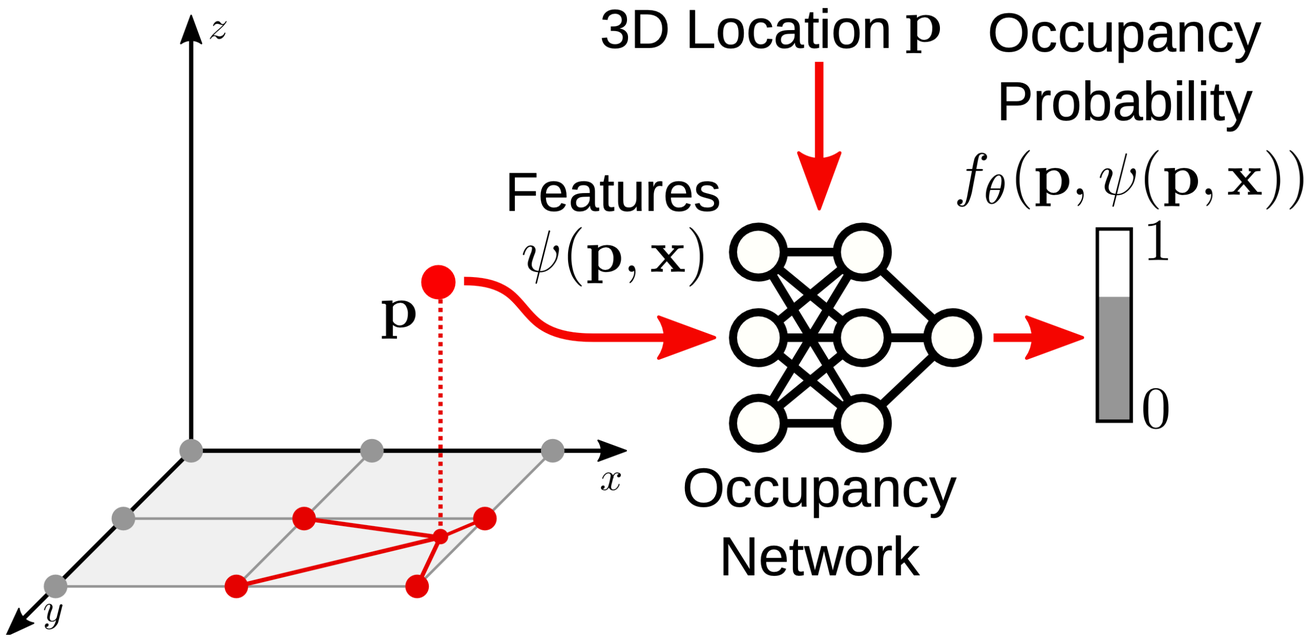
Conditioning Neural Fields: Conditioning via Concatenation



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

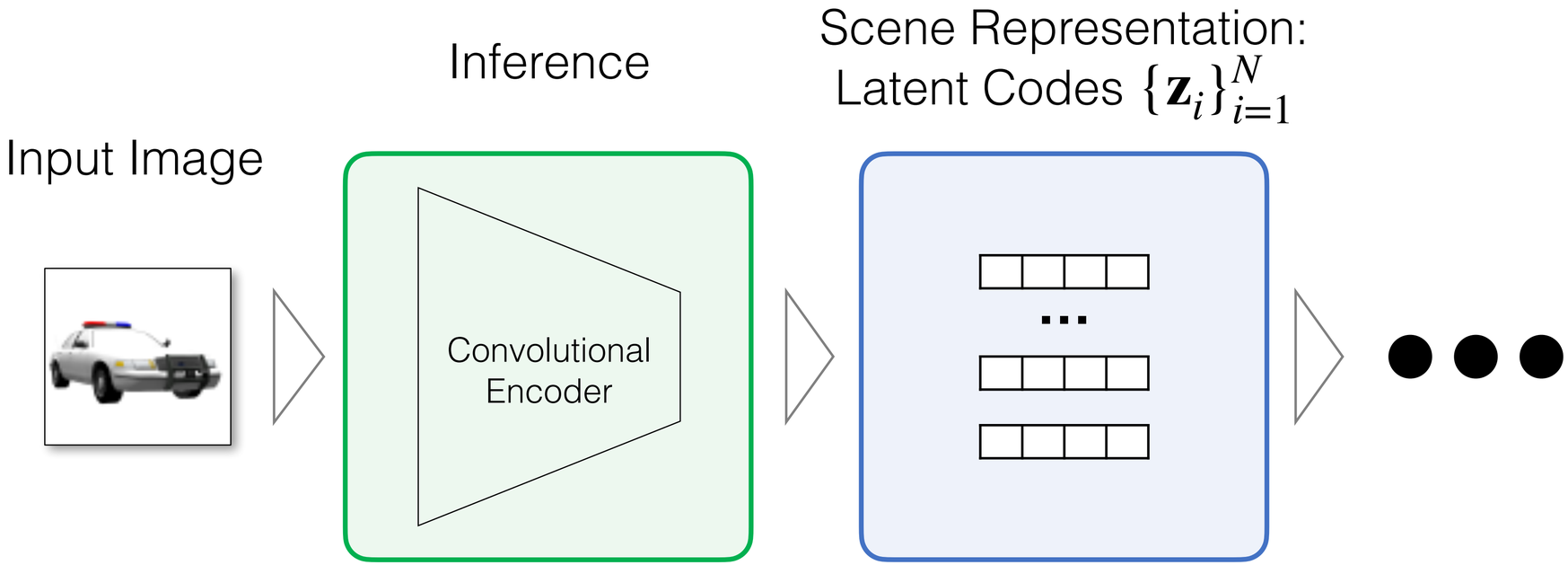
Less expressive than hyper net, but also often fine - for more alternatives, see section 2.1.3. in Neural Fields in Visual Computing and Beyond, Xie et al.

Conditioning Neural Fields: Conditioning via Concatenation



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

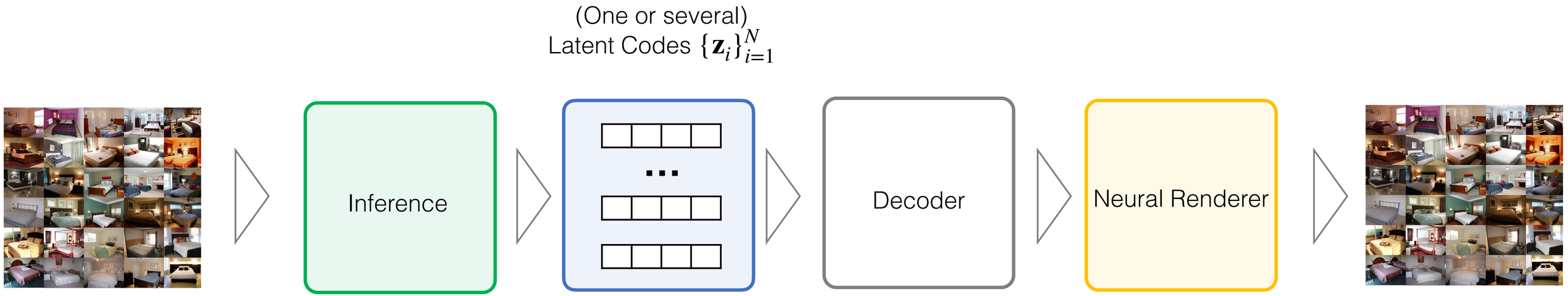
Note on Terminology



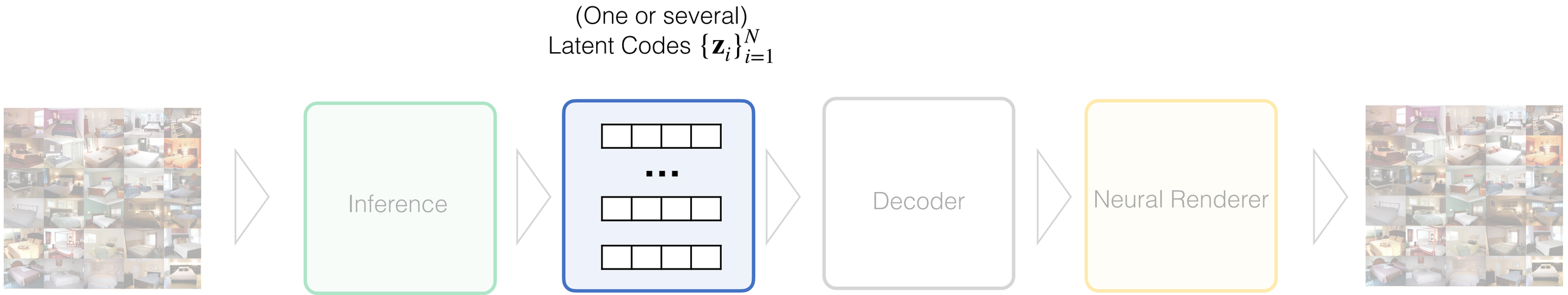
The **Latent Codes** are what's actually encoding the scene!

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

Summary: General Framework

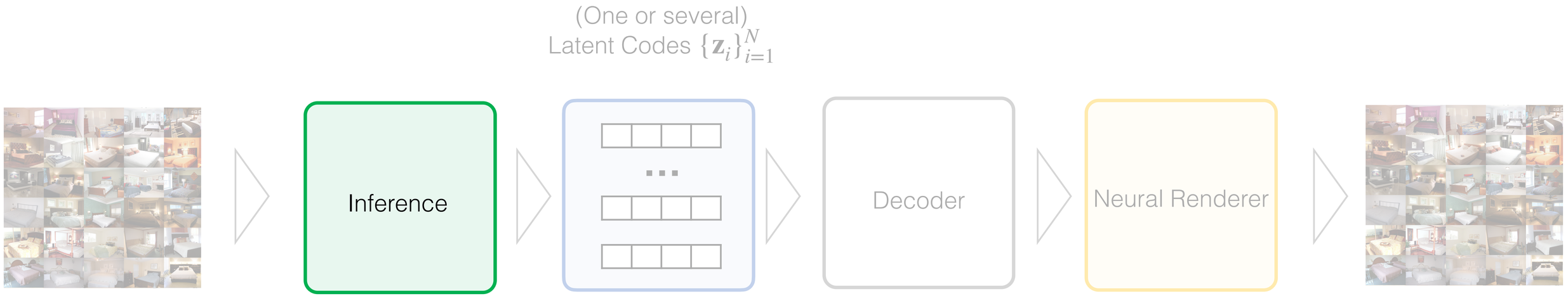


Summary: General Framework



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

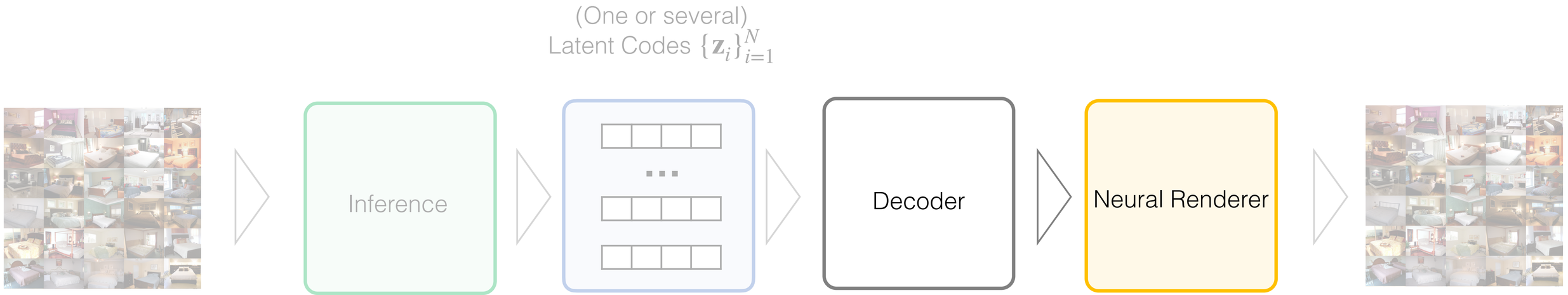
Summary: General Framework



How to get the latent codes from images?

How to guarantee generalization?

Summary: General Framework

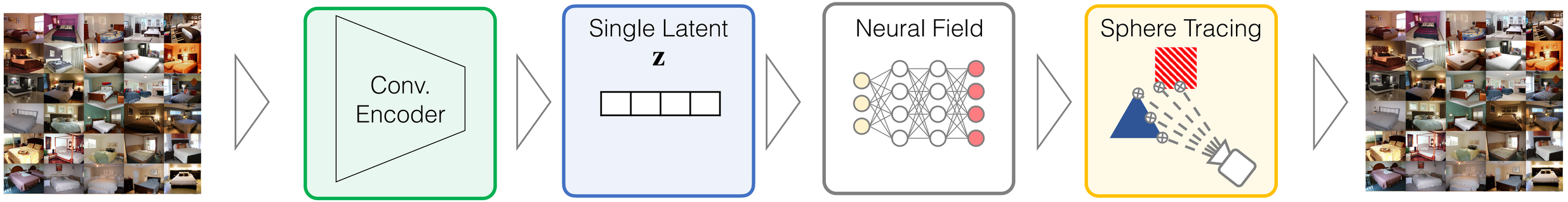


What forms & shapes can the decoder take?

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

Do we need 3D volumetric renderers?

Specific Model

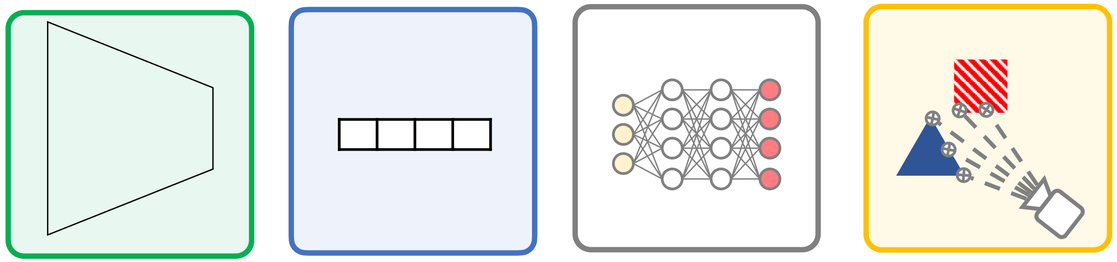


Learn inference model from many images of many scenes.

Output

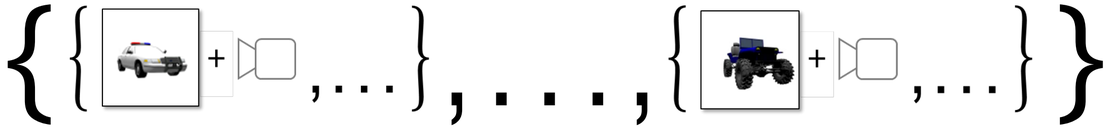
Input view | Normal map | RGB

Model





Dataset

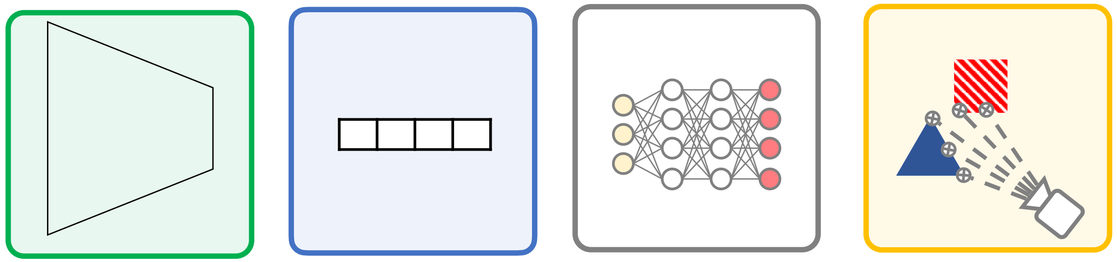


Learn inference model from many images of many scenes.

Output

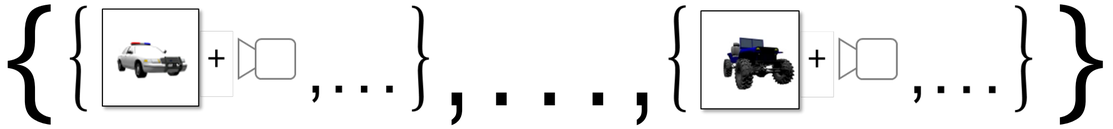
Input view | Normal map | RGB

Model



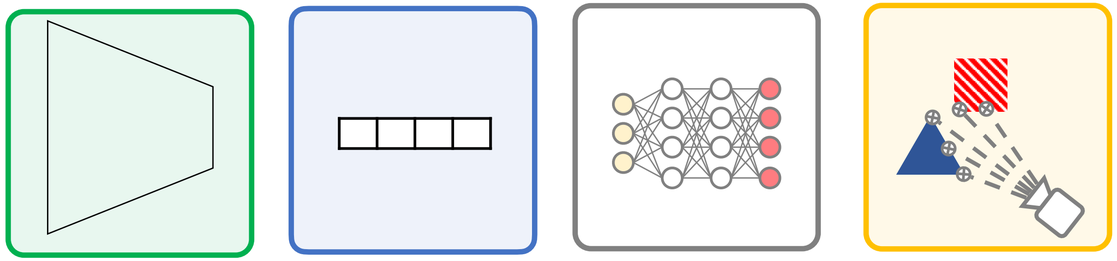


Dataset



Learn inference model from many images of many scenes.

Model

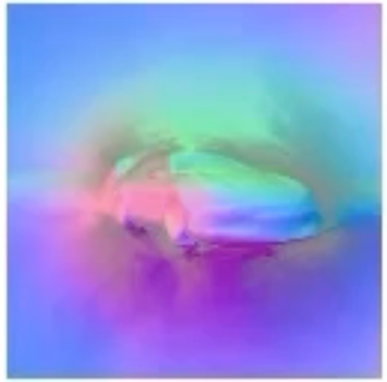


Output

Input view



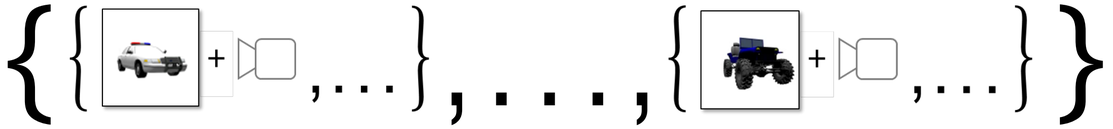
Normal map



RGB

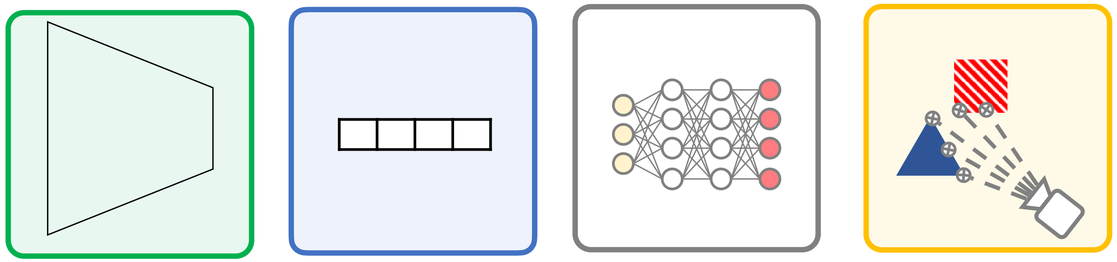


Dataset

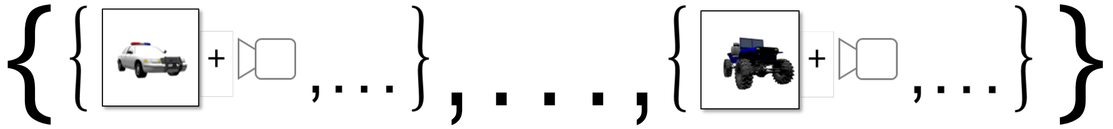


Learn inference model from many images of many scenes.

Model



Dataset

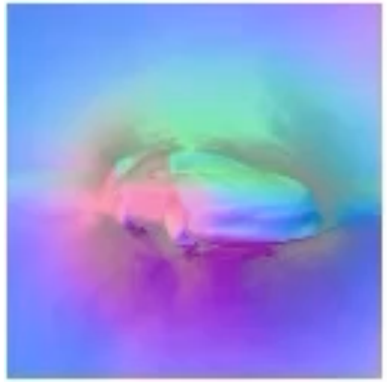


Output

Input view



Normal map



RGB



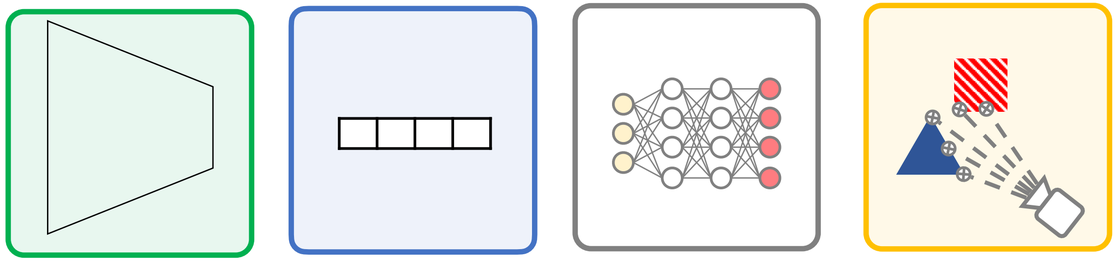
Out-of-distribution
Input view

Normal map

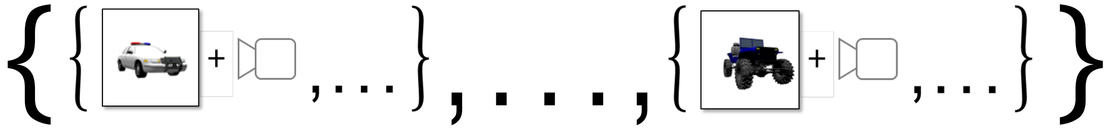
RGB

Learn inference model from many images of many scenes.

Model



Dataset

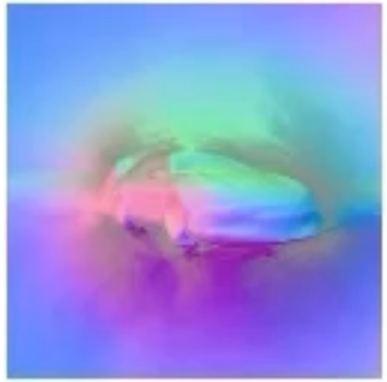


Output

Input view



Normal map



RGB



Out-of-distribution
Input view

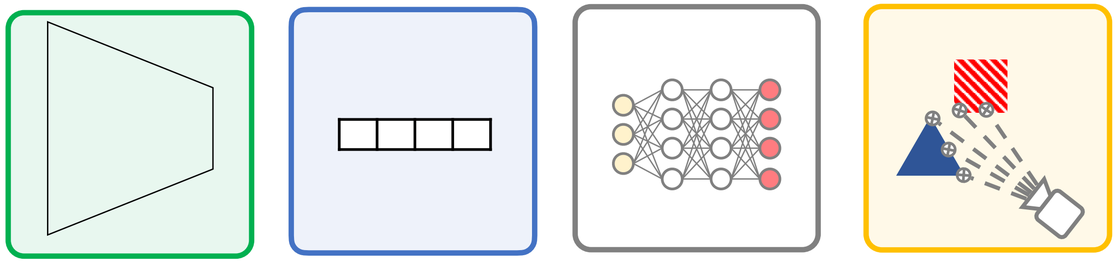


Normal map

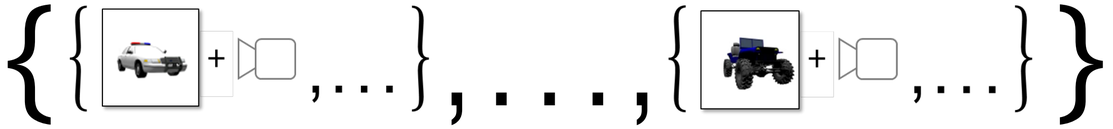
RGB

Learn inference model from many images of many scenes.

Model



Dataset

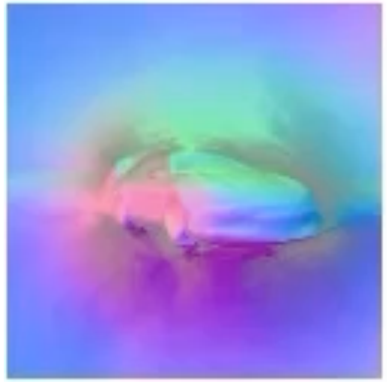


Output

Input view



Normal map



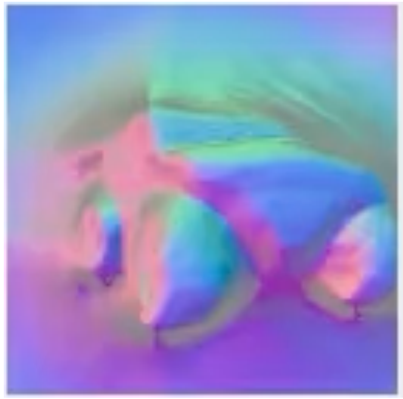
RGB



Out-of-distribution
Input view



Normal map

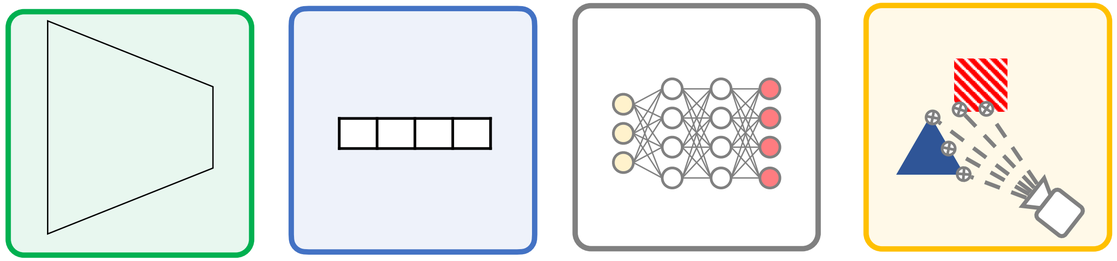


RGB

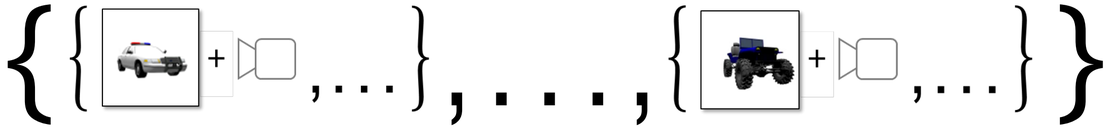


Learn inference model from many images of many scenes.

Model



Dataset



Output

Input view



Normal map



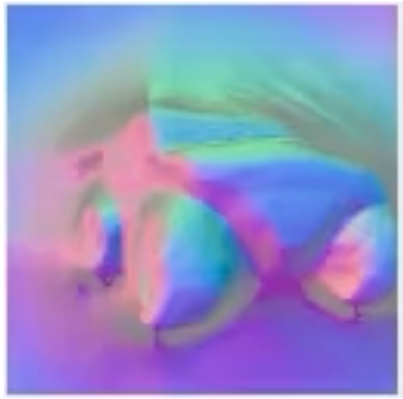
RGB



Out-of-distribution
Input view



Normal map

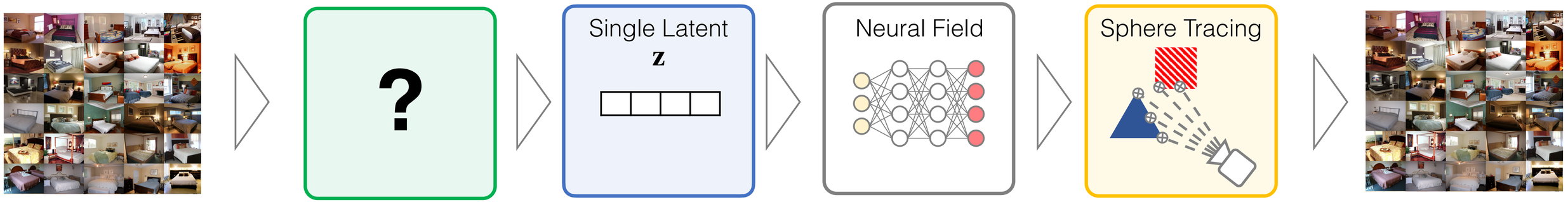


RGB



Wrong window color, wrong tires, wrong shape…

Specific Model



Can we think of an alternative inference method?

Training: Initialize one embedding per scene

embedding z_0

embedding z_1



+

○ ○ ○

○ ○ ○

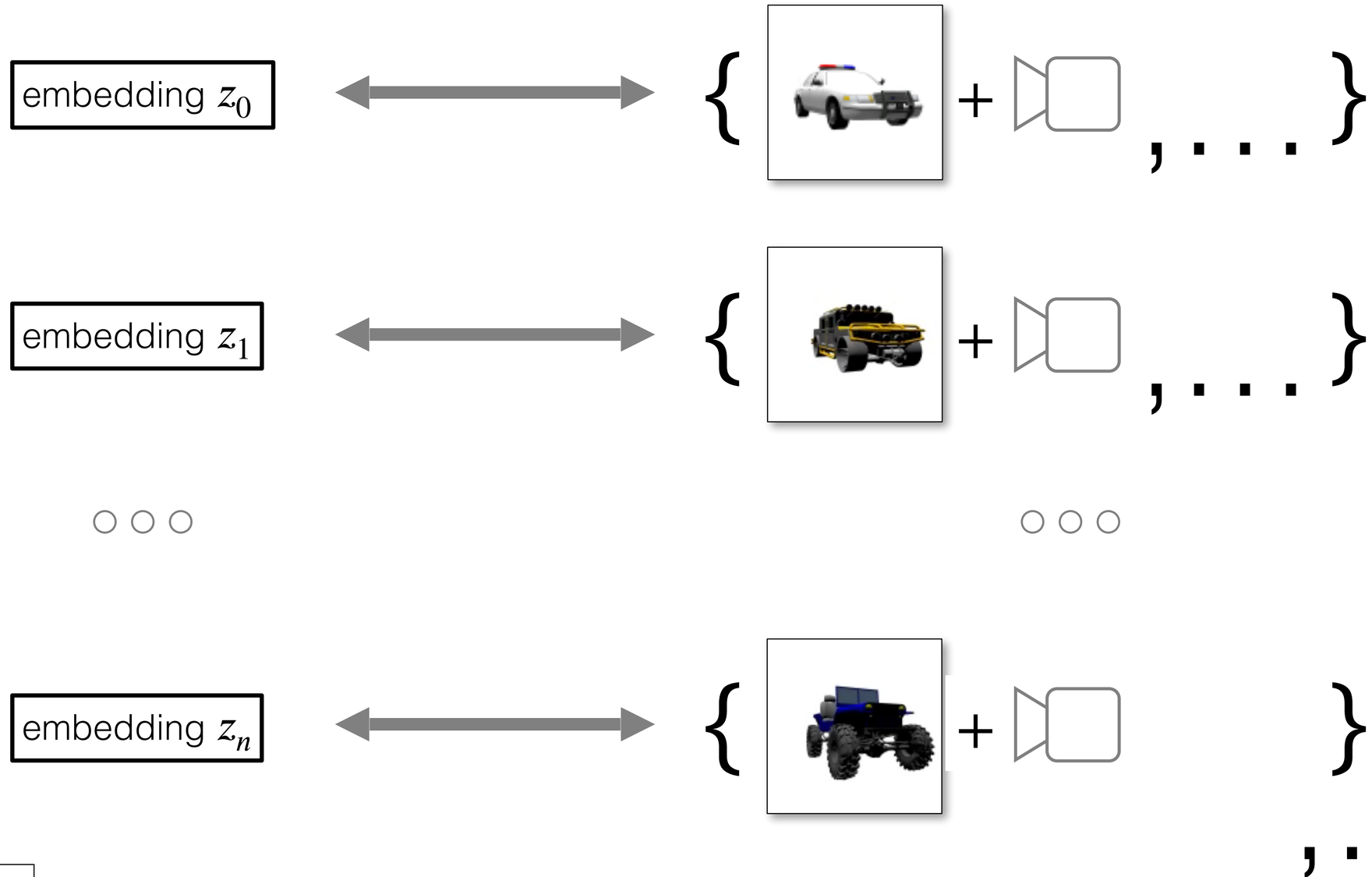
embedding z_n



+

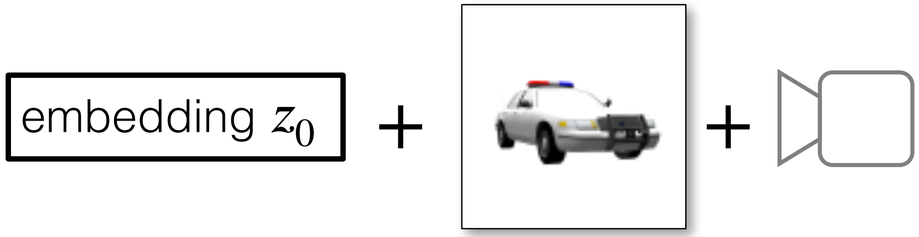
$$z_{j=0,\ldots,n} \sim \mathcal{N}(0, \sigma^2)$$

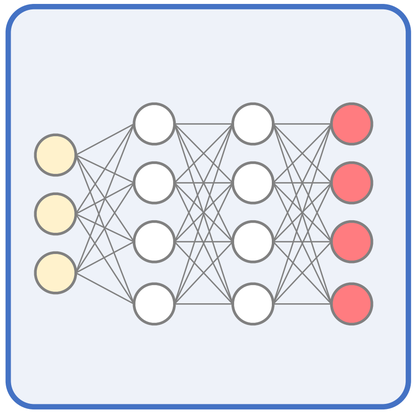
Training: Initialize one embedding per scene



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

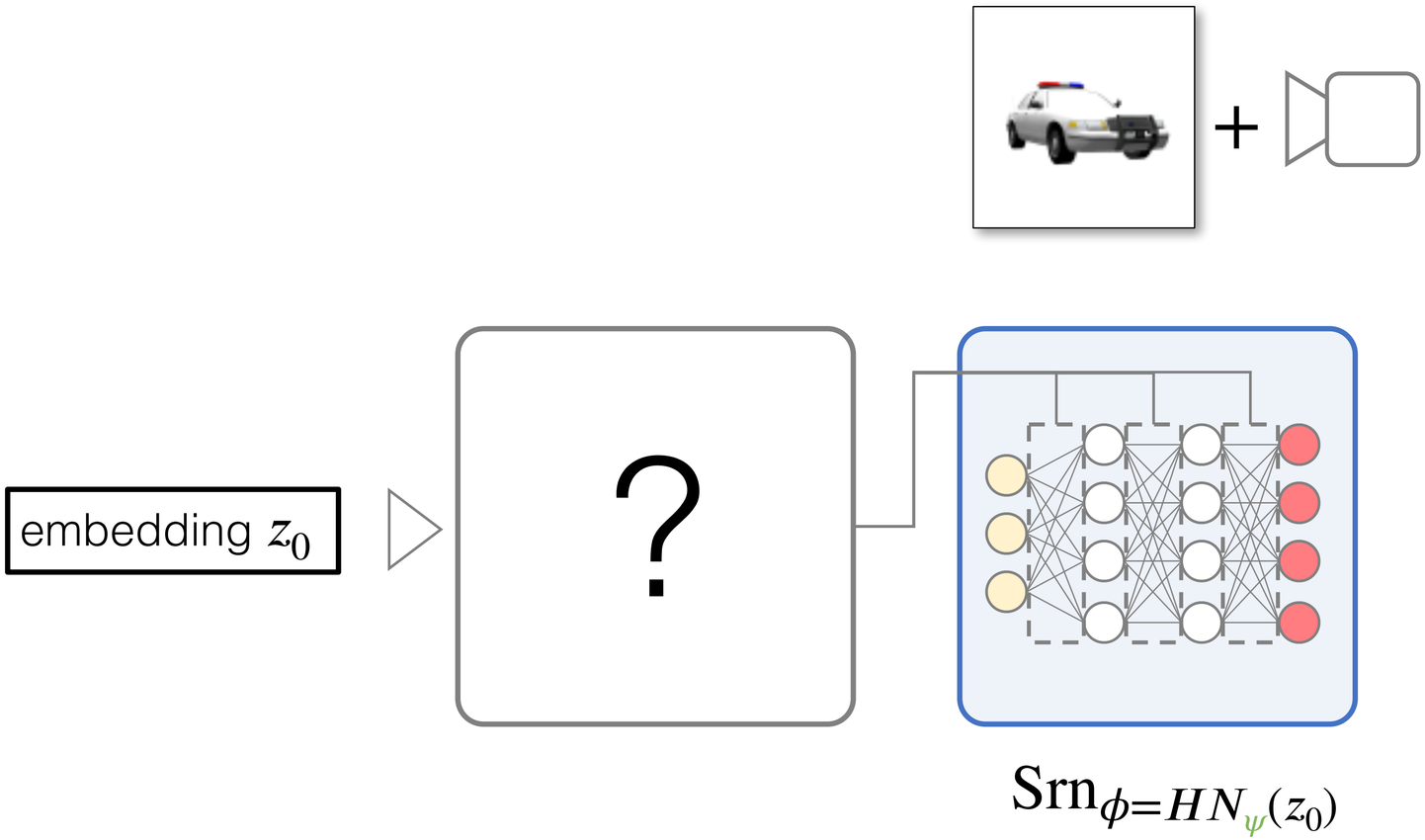
Decode embedding into scene representation



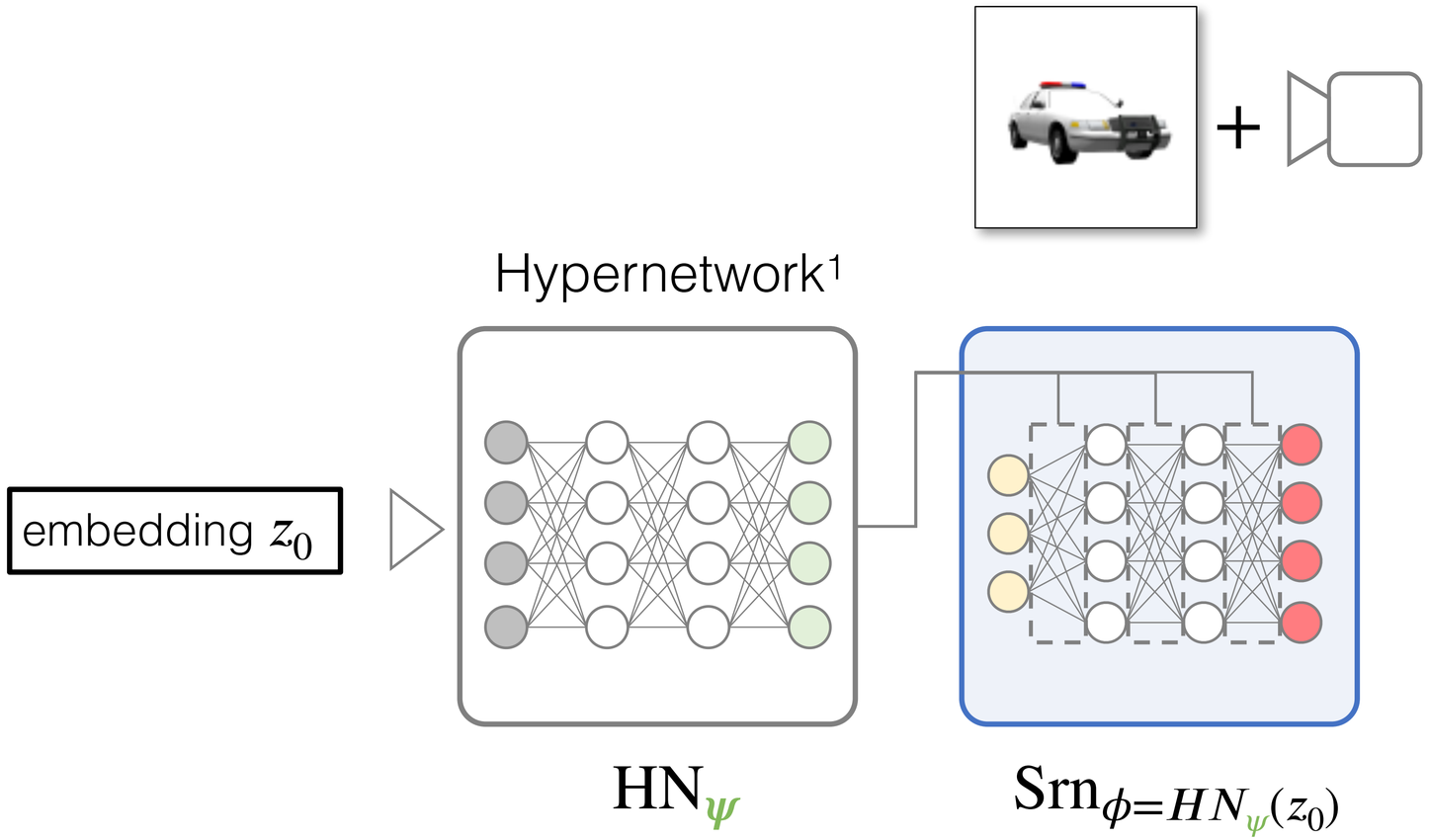


$$\text{Srn}_{\phi=HN_{\psi}(z_0)}$$

Decode embedding into scene representation

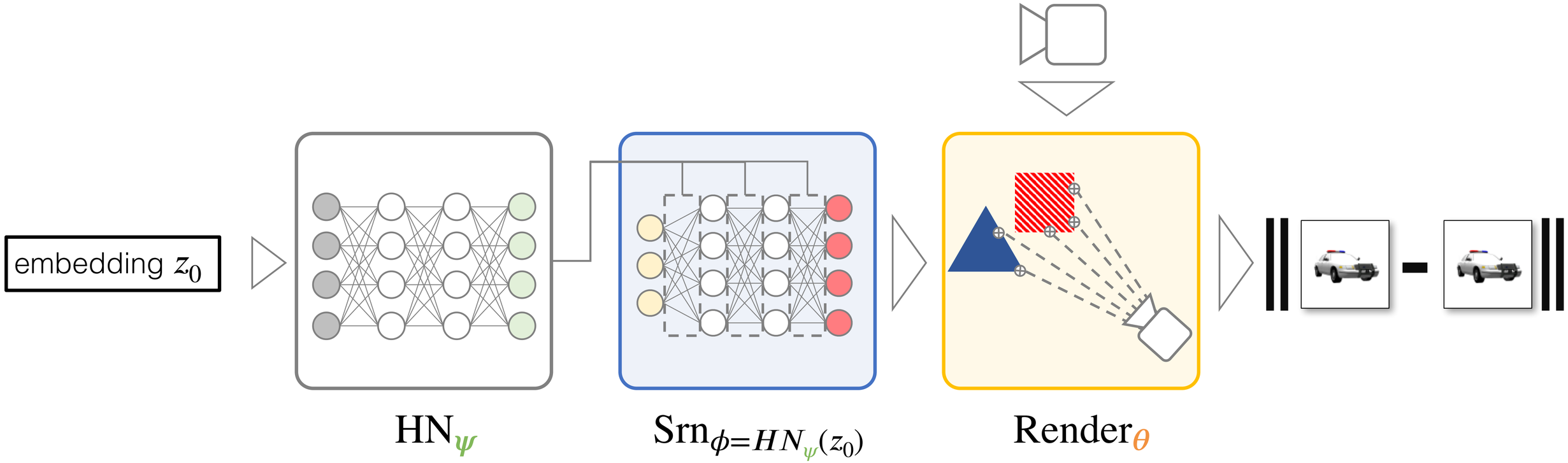


Decode embedding into scene representation

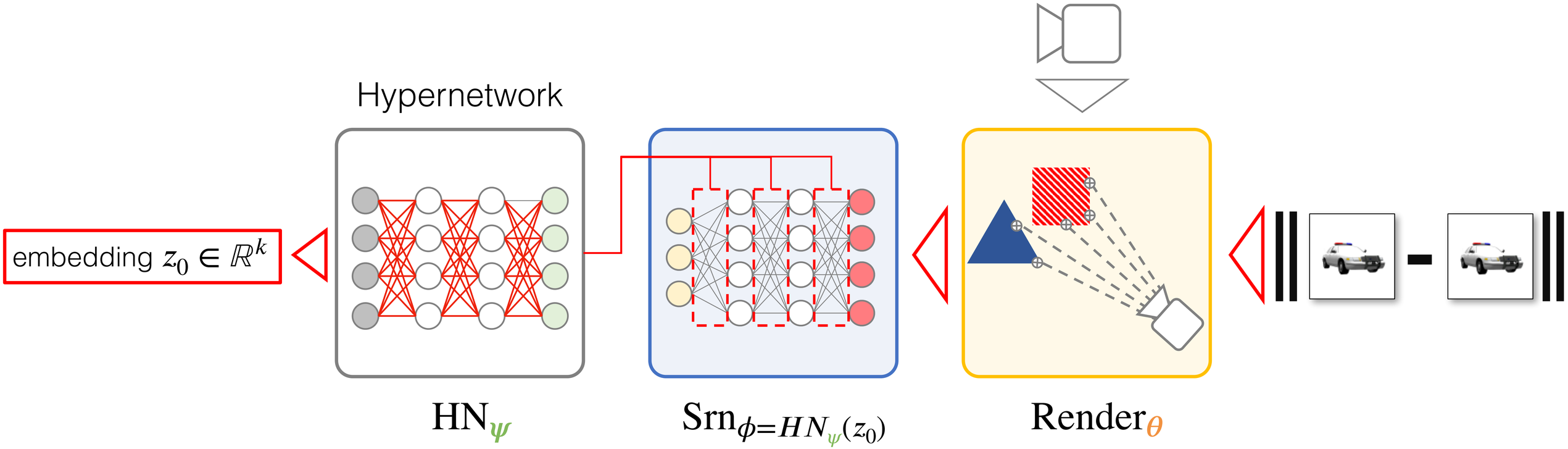


[1][Schmidhuber et al. 1992, Schmidhuber et al. 1993, Stanley et al. 2009, Ha et al., 2016]

Render training view

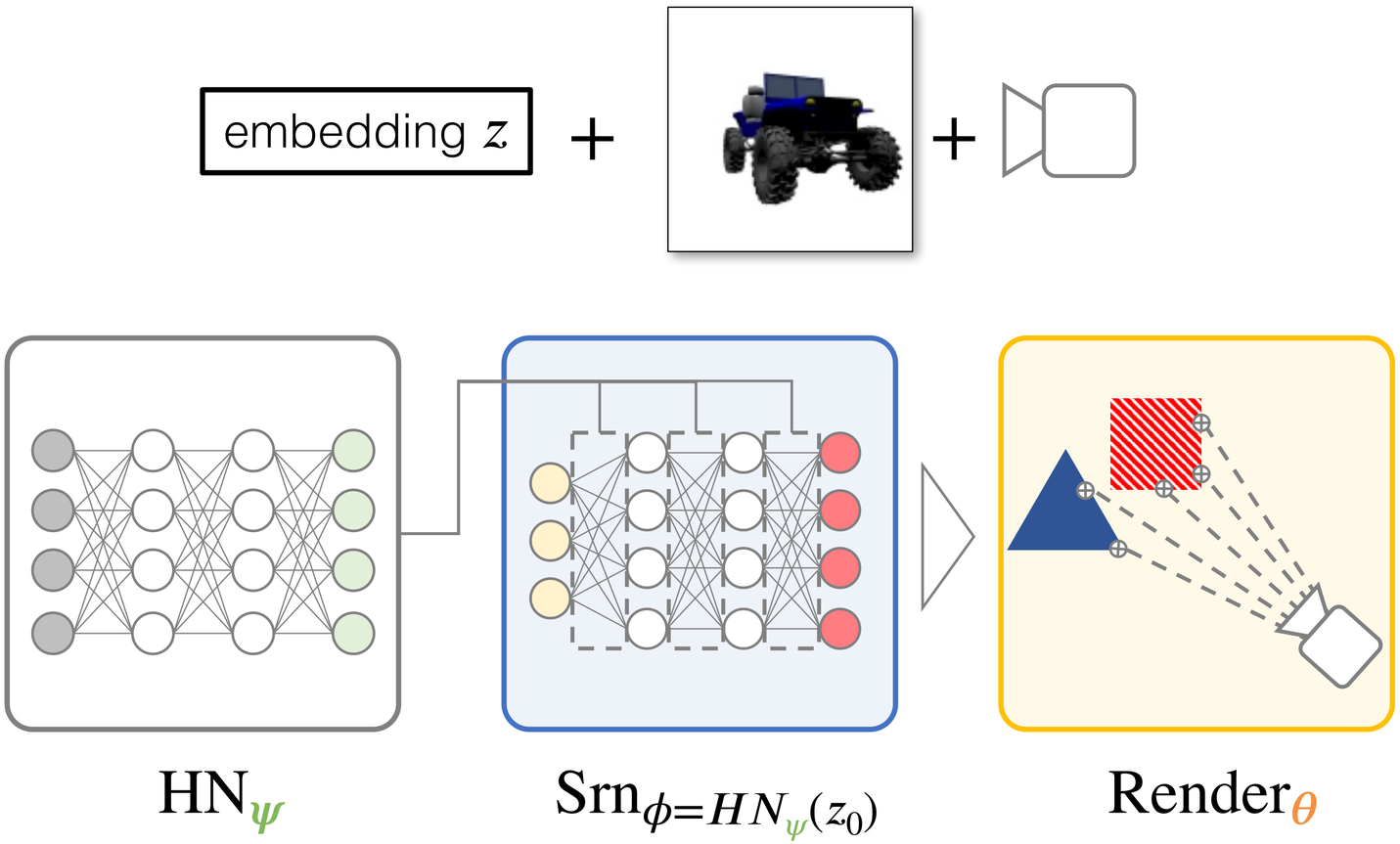


Co-optimize embeddings and model weights!



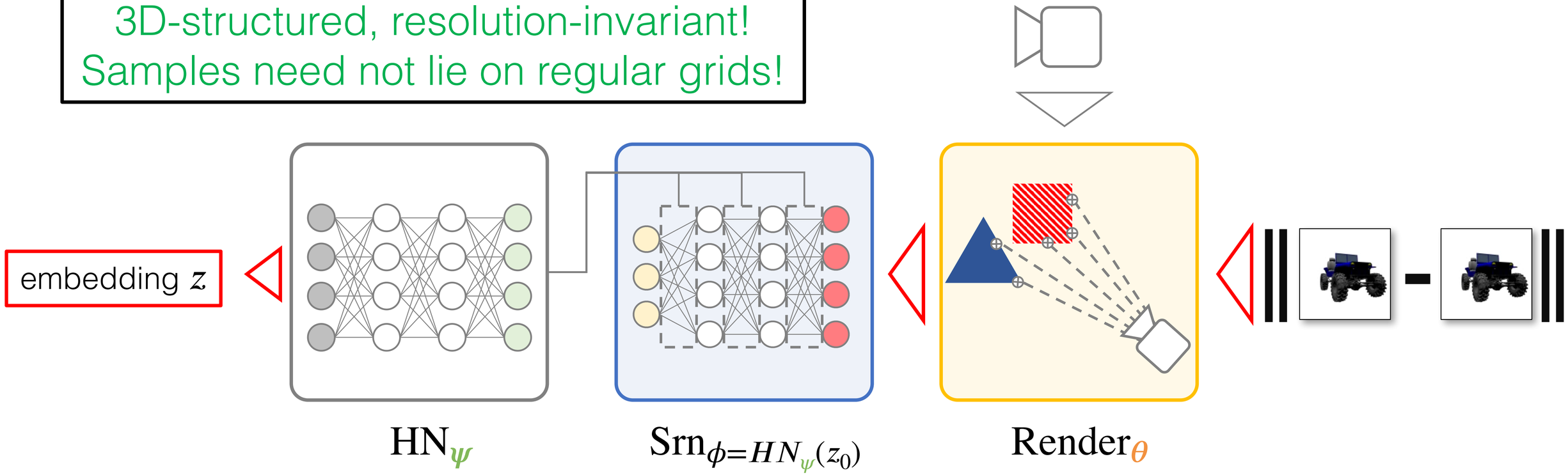
$$\arg \min_{\{z_j\}_{j=1}^{M},\psi,\theta} \sum_j \sum_i \left\| \text{Render}_{\theta}(\text{Srn}_{\phi=HN_{\psi}(z_j)}, \xi_i) - \mathcal{I}_i^j \right\|$$

Test time: initialize new embedding



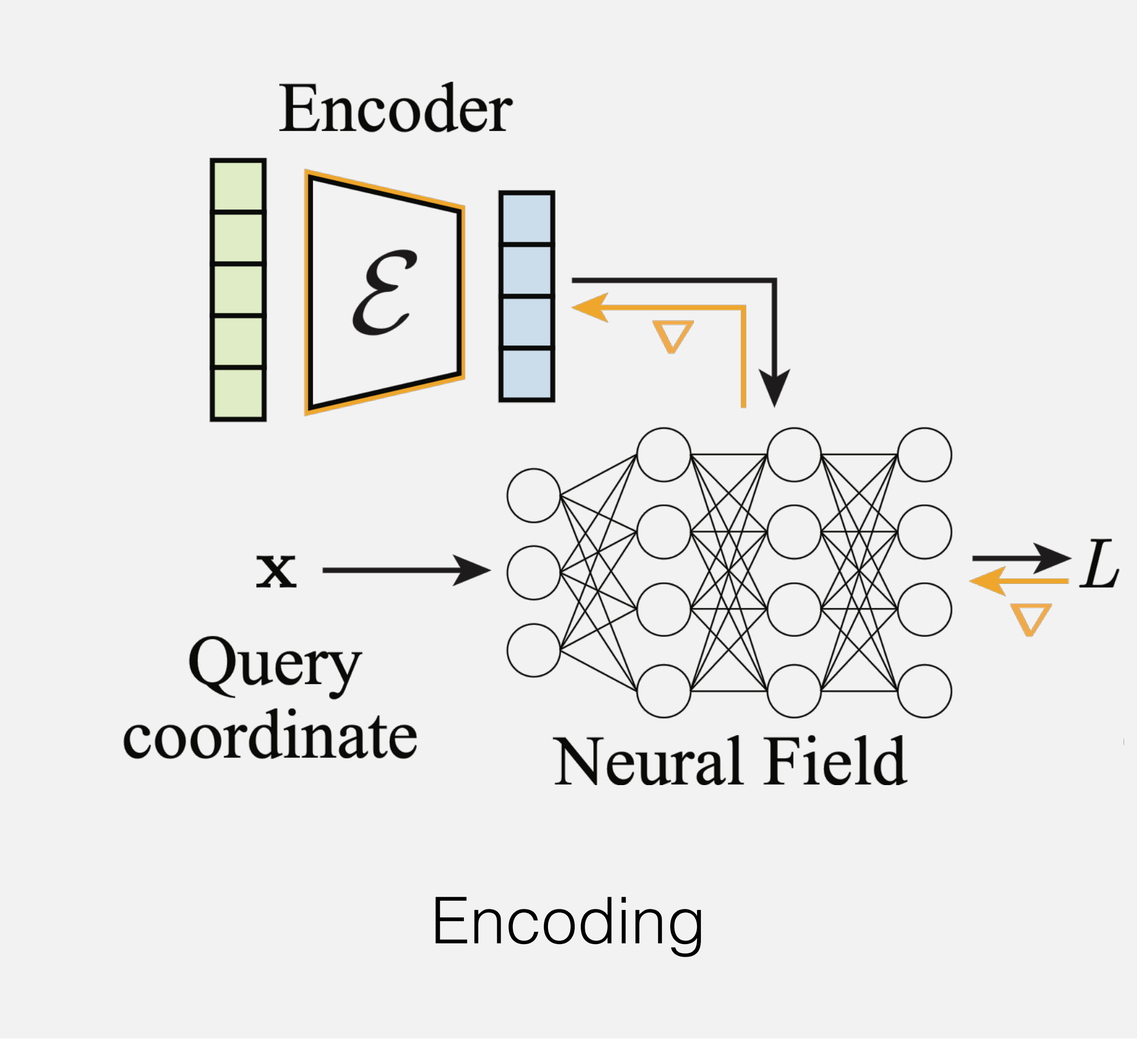
Inference: Freeze model weights, backprop into embedding only!

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



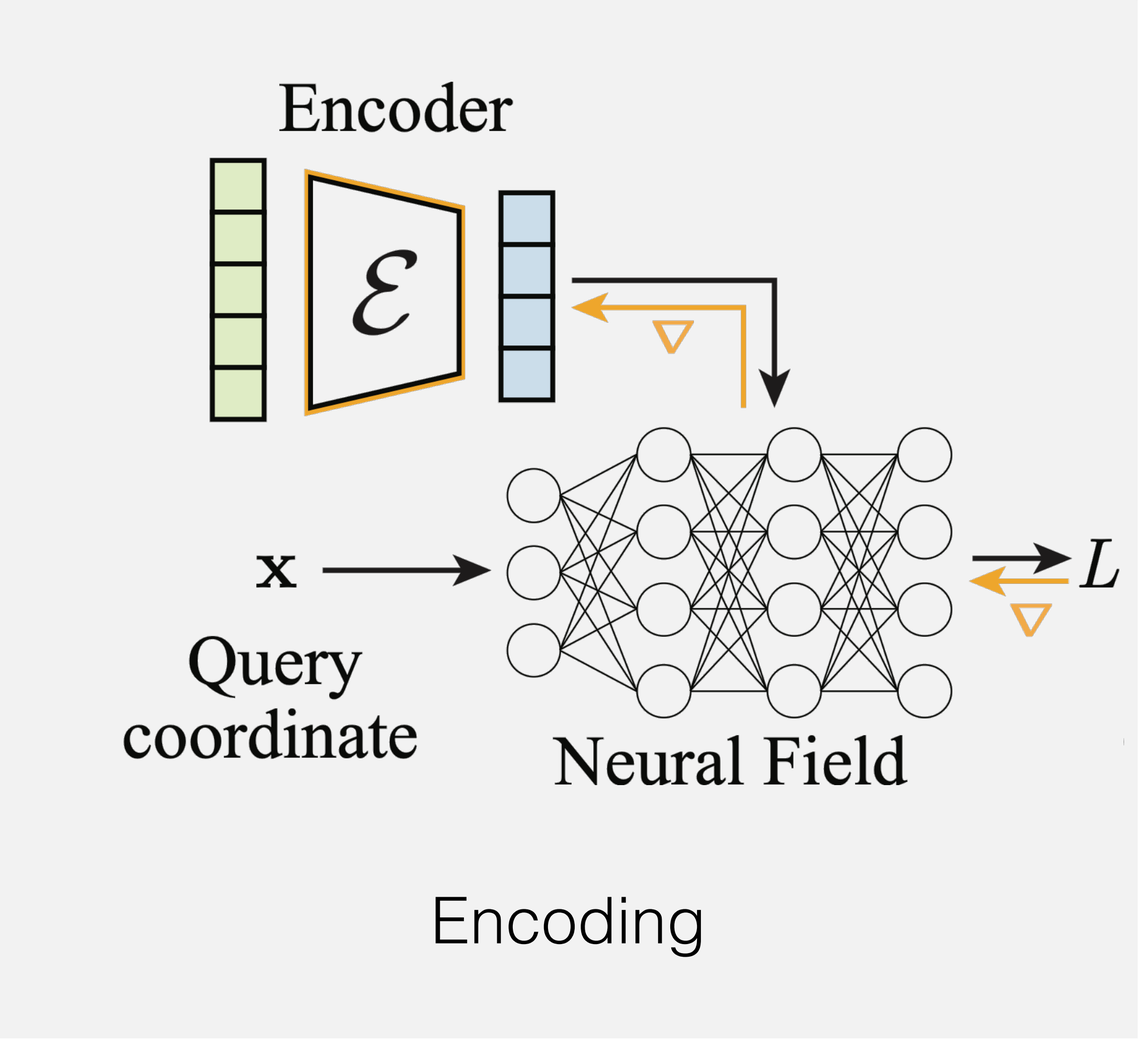
$$\hat{z} = \underset{z}{\text{argmin}} \left\| \text{Render}_{\theta}(\text{Srn}_{\phi=HN_{\psi}(z)}, \xi) - \mathcal{I} \right\|$$

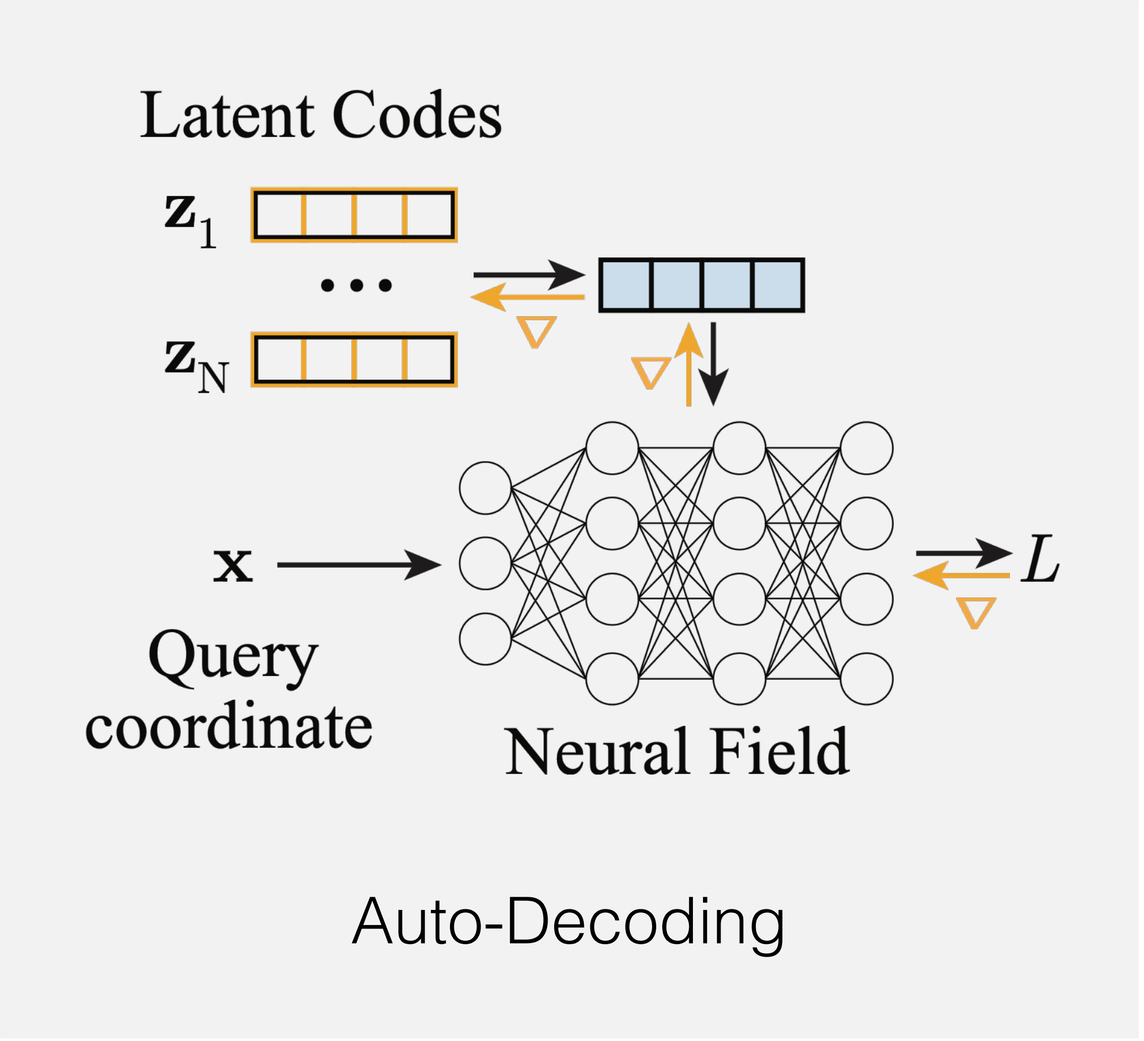
Encoding & Auto-Decoding



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

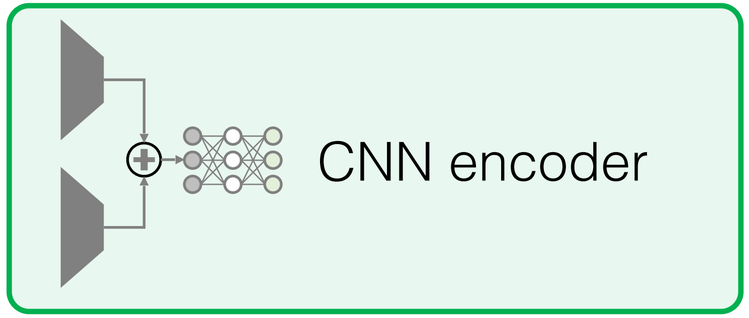
Encoding & Auto-Decoding





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

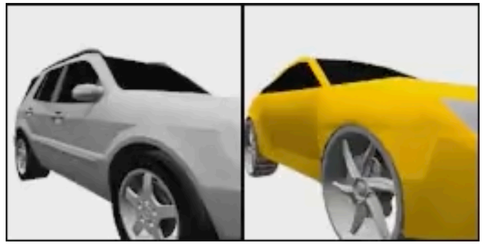
Out-of-distribution generalization



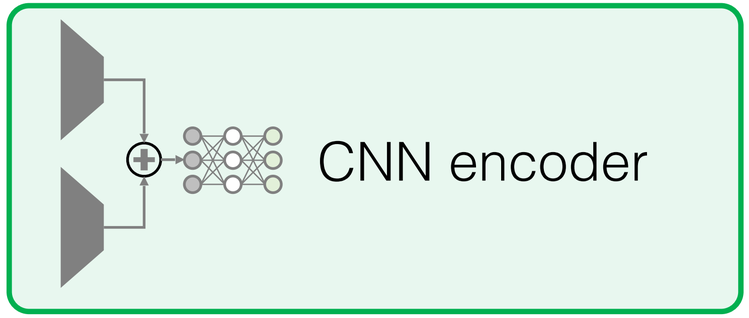
Input

Reconstructions



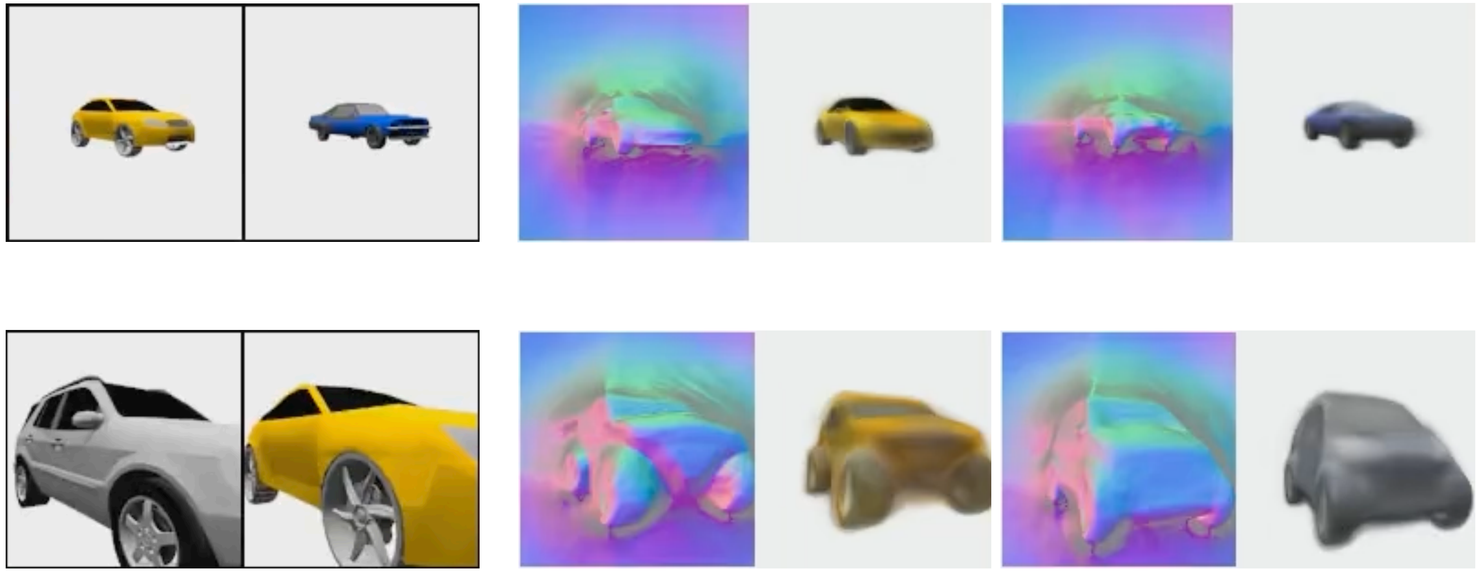


Out-of-distribution generalization

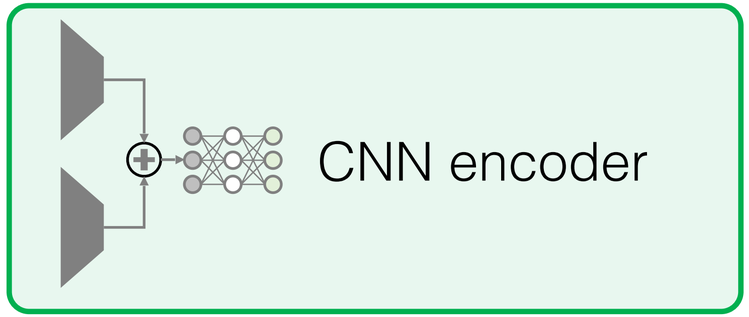


Input

Reconstructions



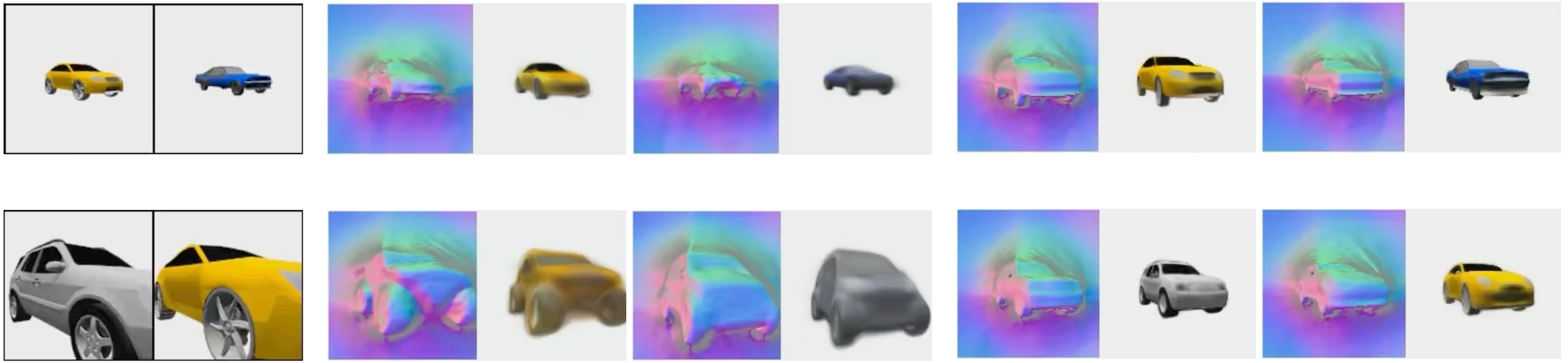
Out-of-distribution generalization



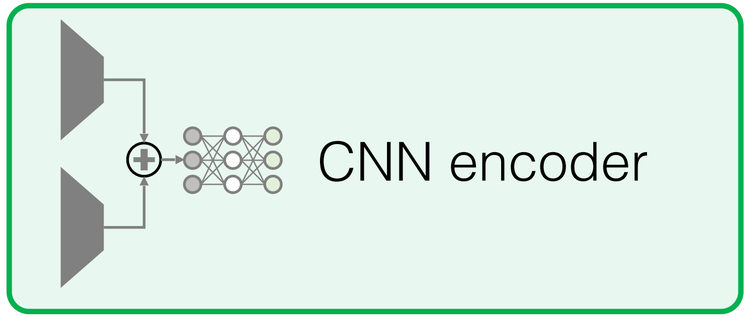
$$argmin_z ||Render(SRN, \xi) - \mathcal{I}||$$

Input

Reconstructions



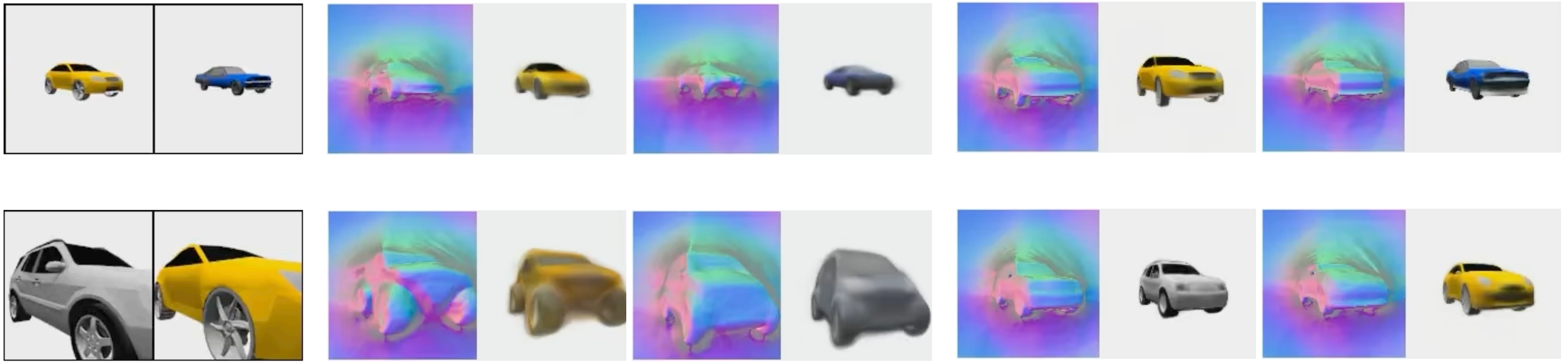
Out-of-distribution generalization



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

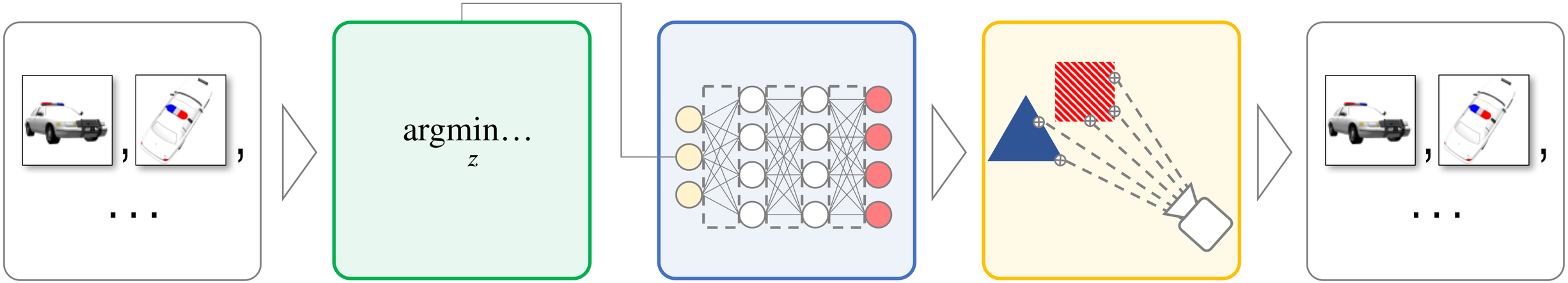
Input

Reconstructions

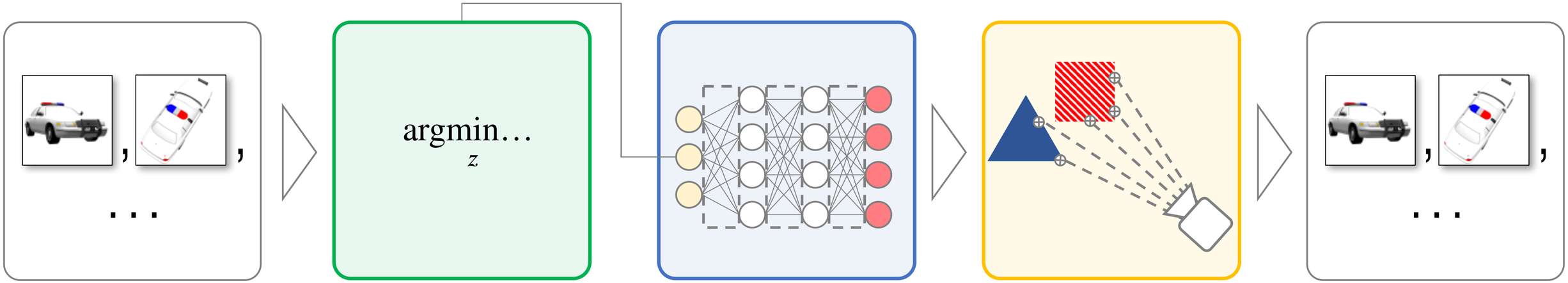


3D structure enables generalization to out-of-distribution camera poses!

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

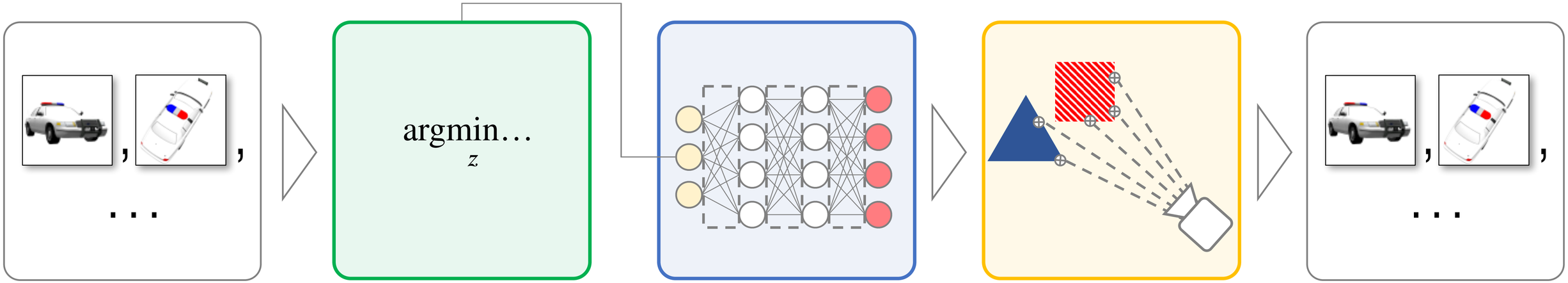


Scene Representation Networks (Sitzmann et al., Neurips 2019)



Enable reconstruction from incomplete observations!

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



Enable reconstruction from incomplete observations!

Scene understanding!

Single-shot reconstruction of held-out test objects

Input observation



SRNs (Ours)



Tatarchenko et al. 2015



Worrall et al. 2017



Deterministic GQN, adapted Eslami et al. 2018



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

Single-shot reconstruction of held-out test objects

Input observation



SRNs (Ours)



Tatarchenko et al. 2015



Worrall et al. 2017

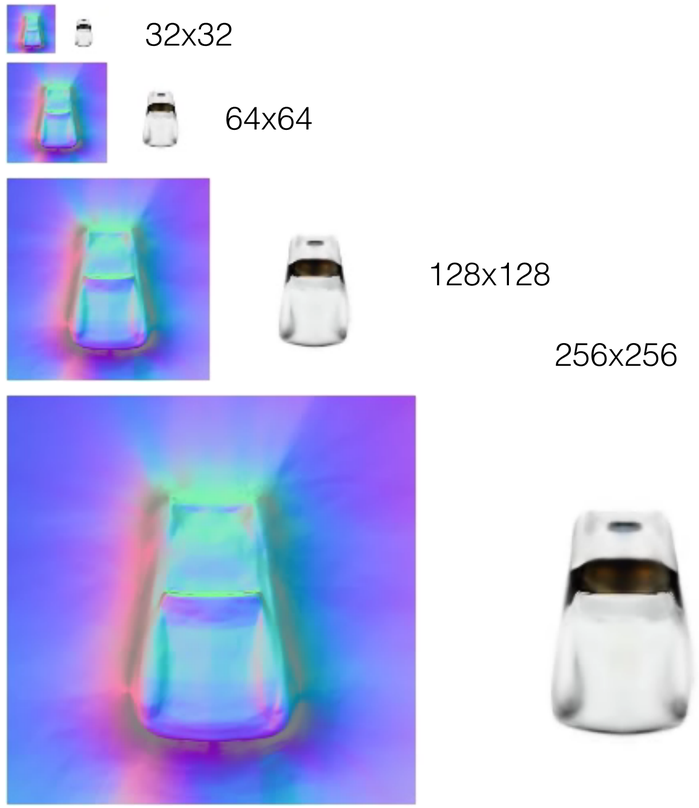


Deterministic GQN, adapted Eslami et al. 2018

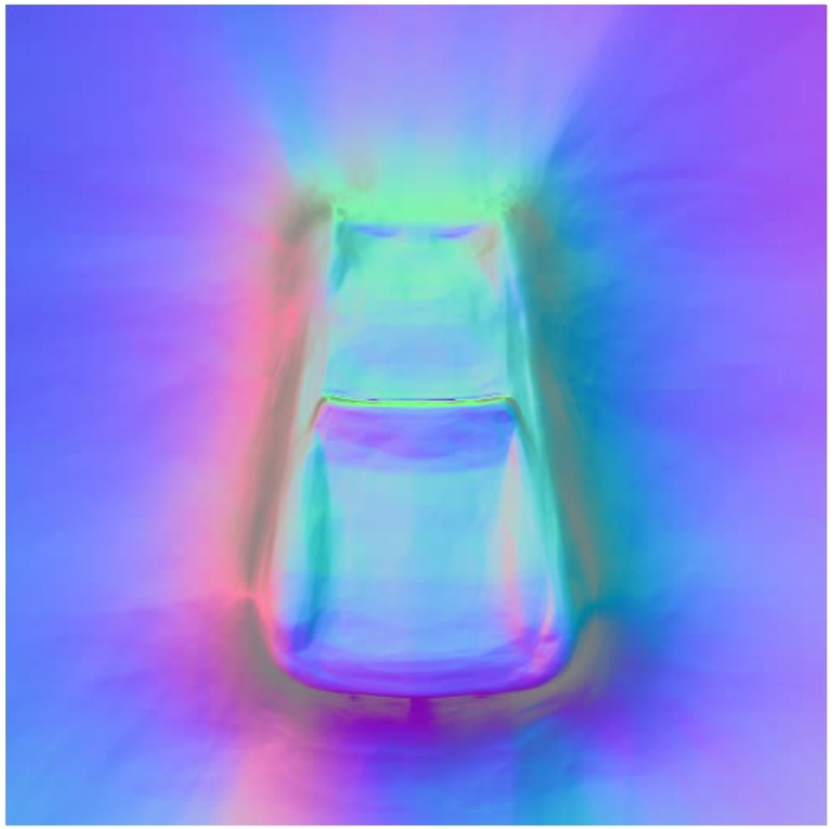


Scene Representation Networks: Continuous 3D-structure Aware Neural Scene Representations, Sitzmann et al., NeurIPS

Sampling at arbitrary resolutions



512x512



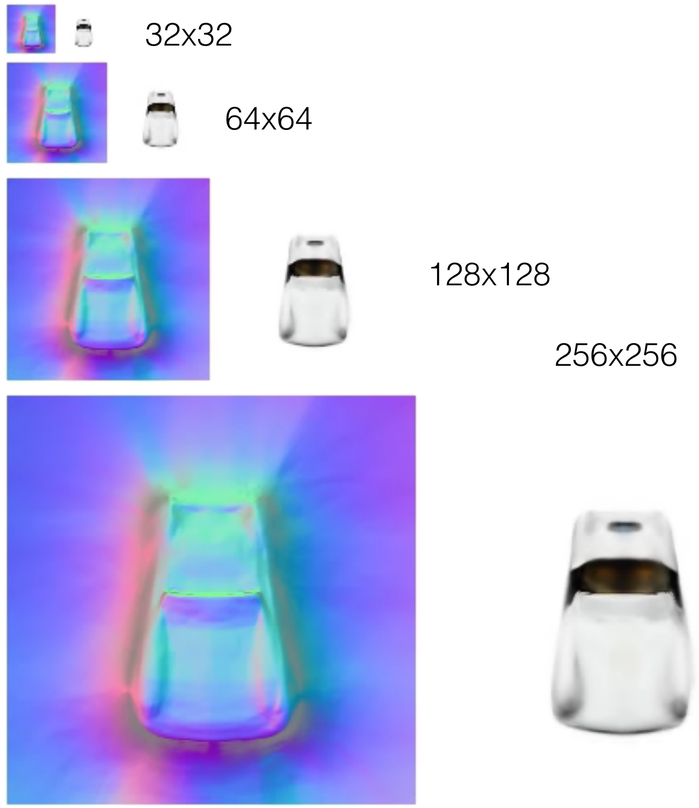
Surface Normals



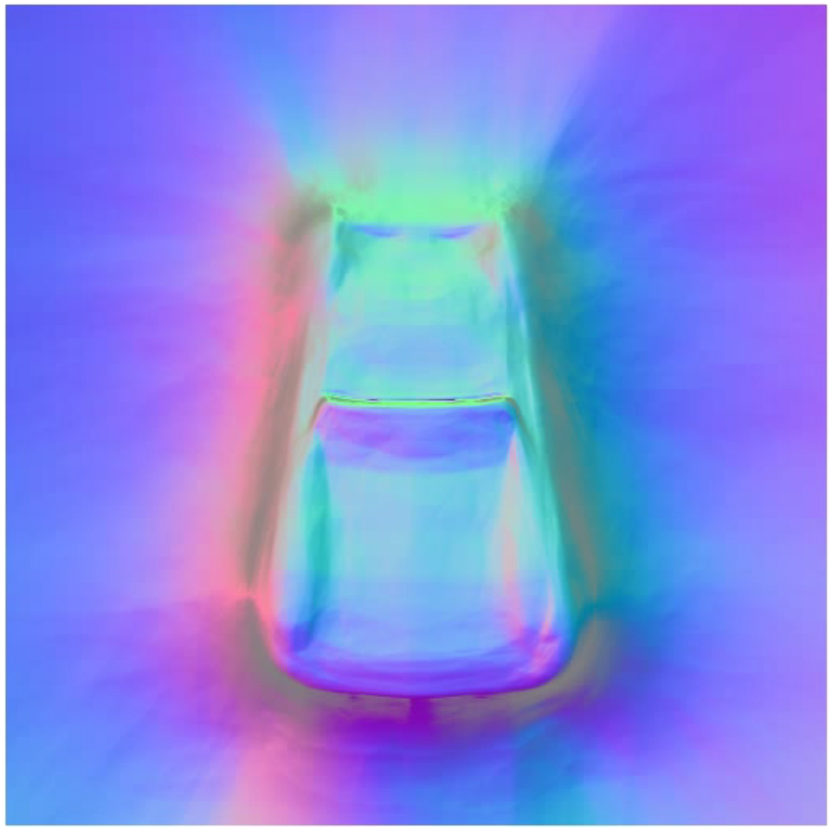
RGB

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

Sampling at arbitrary resolutions



512x512



Surface Normals



RGB

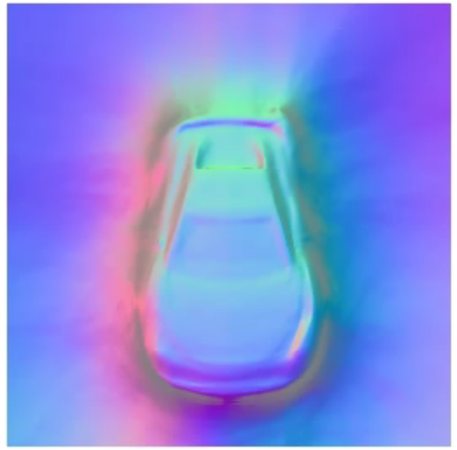
Scene Representation Networks: Continuous 3D-structure Aware Neural Scene Representations, Sitzmann et al., NeurIPS

Generalization to unseen camera poses

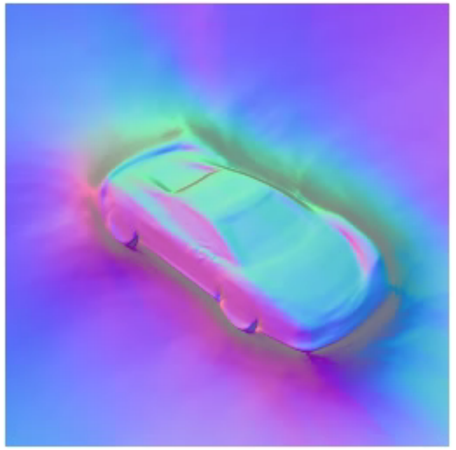
Camera close-up

Camera Roll

SRNs









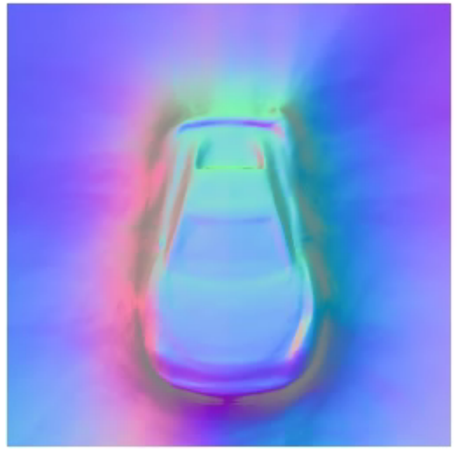
Scene Representation Networks: Continuous 3D-structure Aware Neural Scene Representations, Sitzmann et al., NeurIPS

Generalization to unseen camera poses

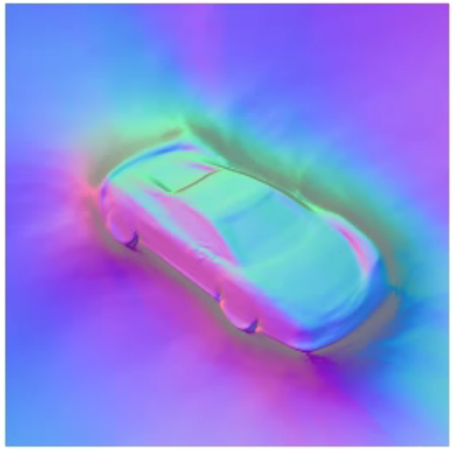
Camera close-up

Camera Roll

SRNs









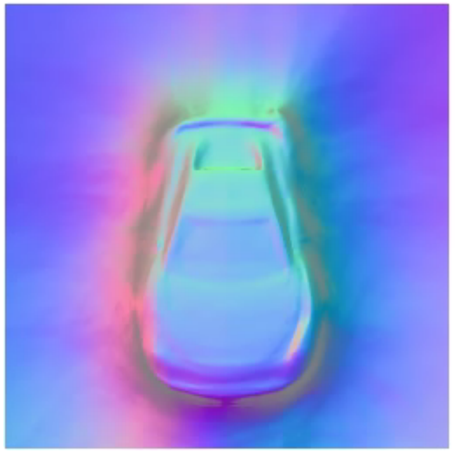
Scene Representation Networks: Continuous 3D-structure Aware Neural Scene Representations, Sitzmann et al., NeurIPS

Generalization to unseen camera poses

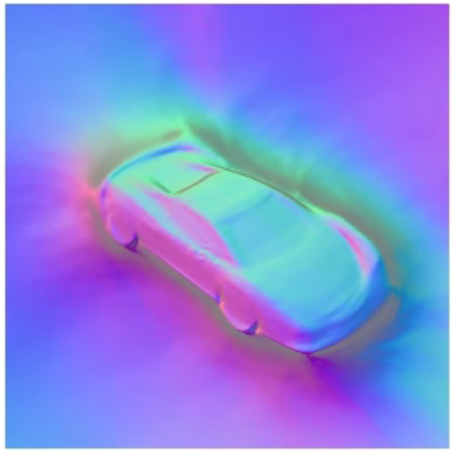
Camera close-up

Camera Roll

SRNs









Tatarchenko et al.

Doesn’t reconstruct geometry



Doesn’t reconstruct geometry



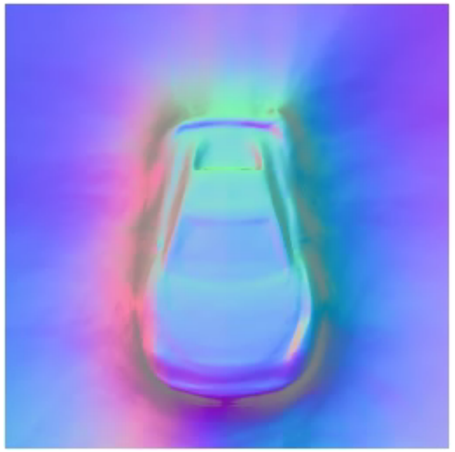
Scene Representation Networks: Continuous 3D-structure Aware Neural Scene Representations, Sitzmann et al., NeurIPS

Generalization to unseen camera poses

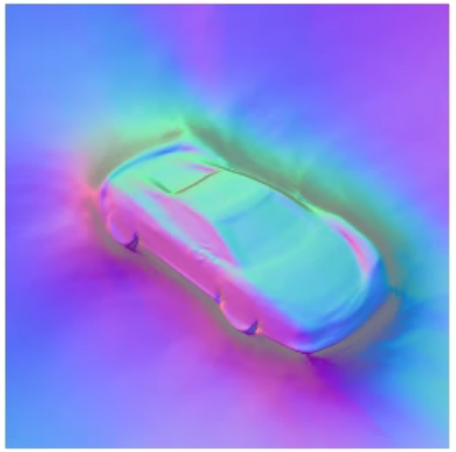
Camera close-up

Camera Roll

SRNs









Tatarchenko et al.

Doesn’t reconstruct geometry

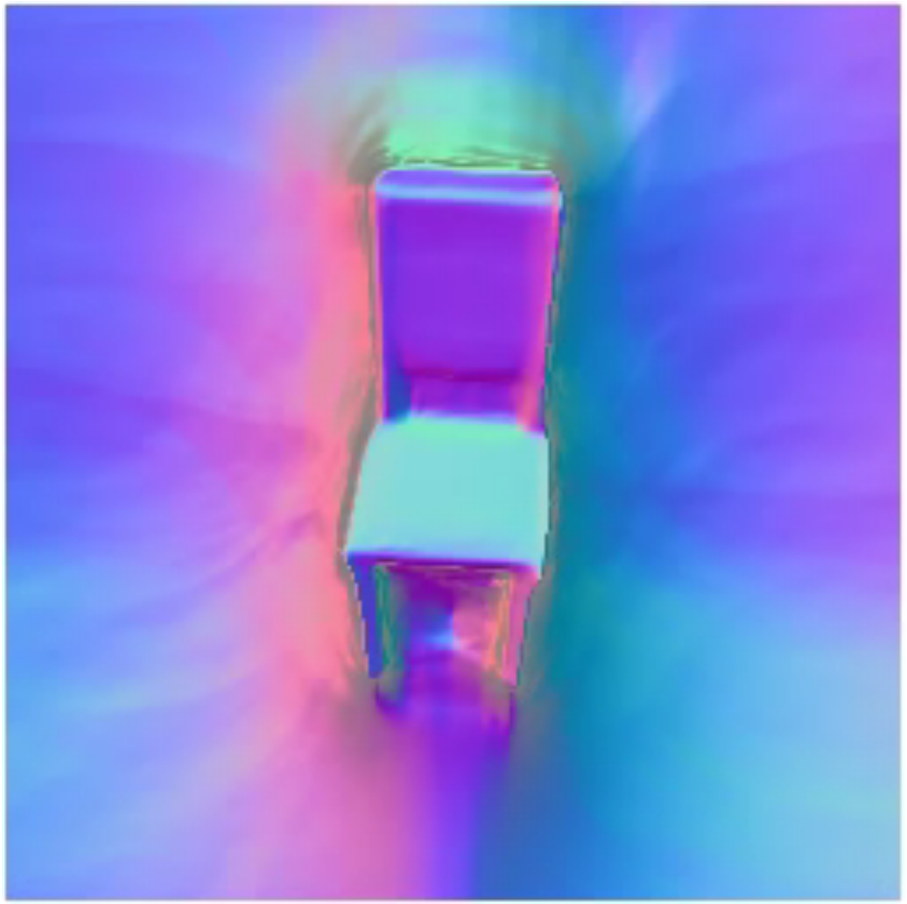


Doesn’t reconstruct geometry



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

Latent code interpolation



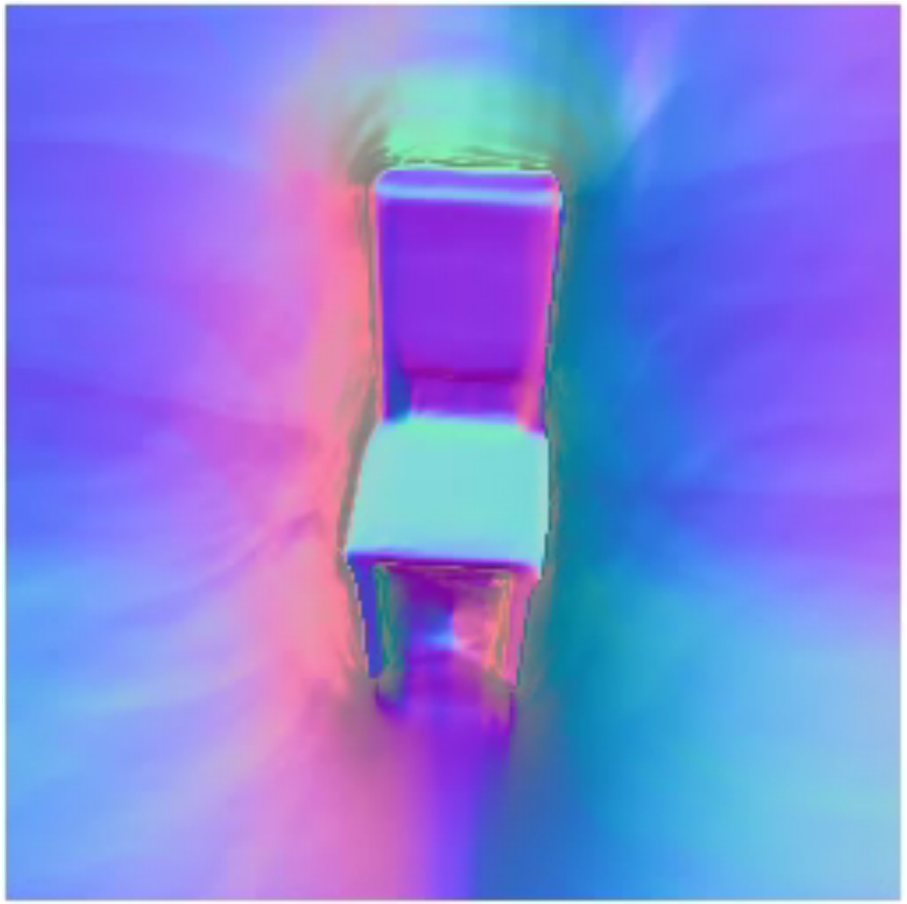
Surface Normals



RGB

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

Latent code interpolation



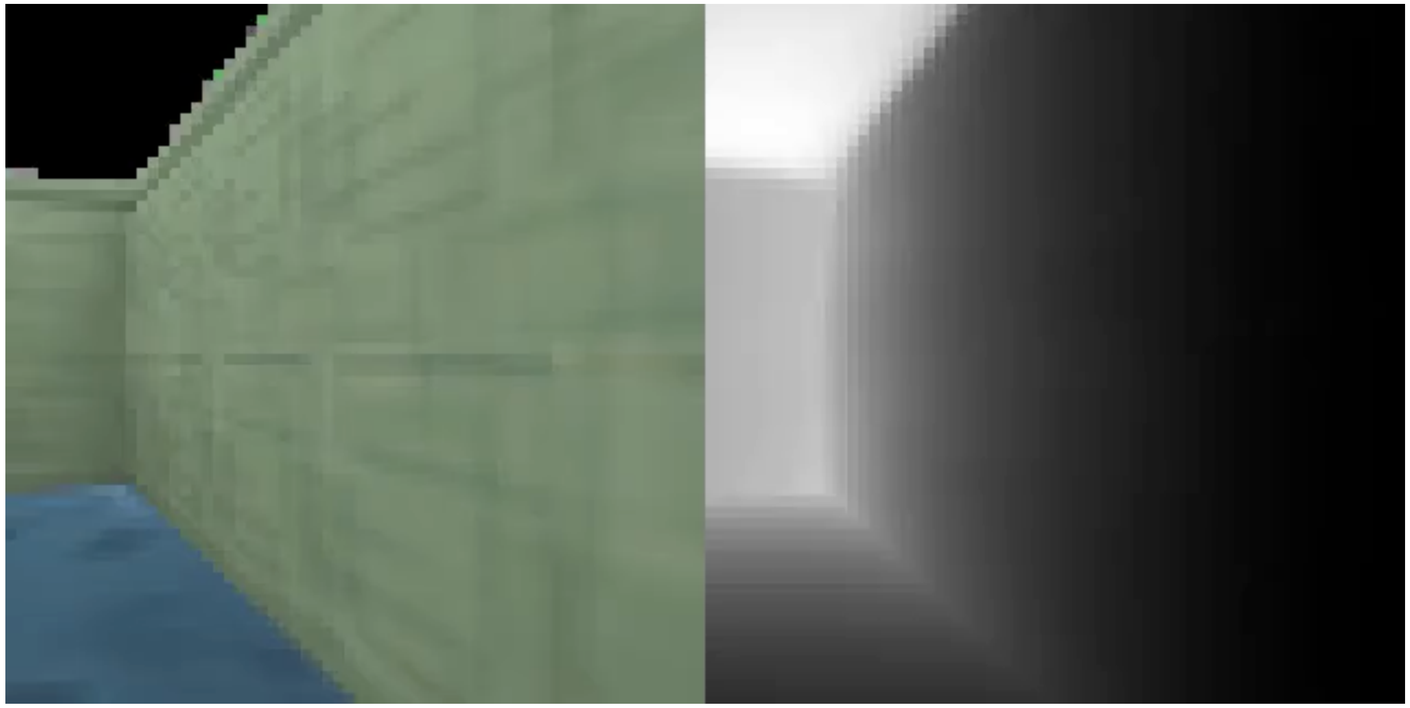
Surface Normals



RGB

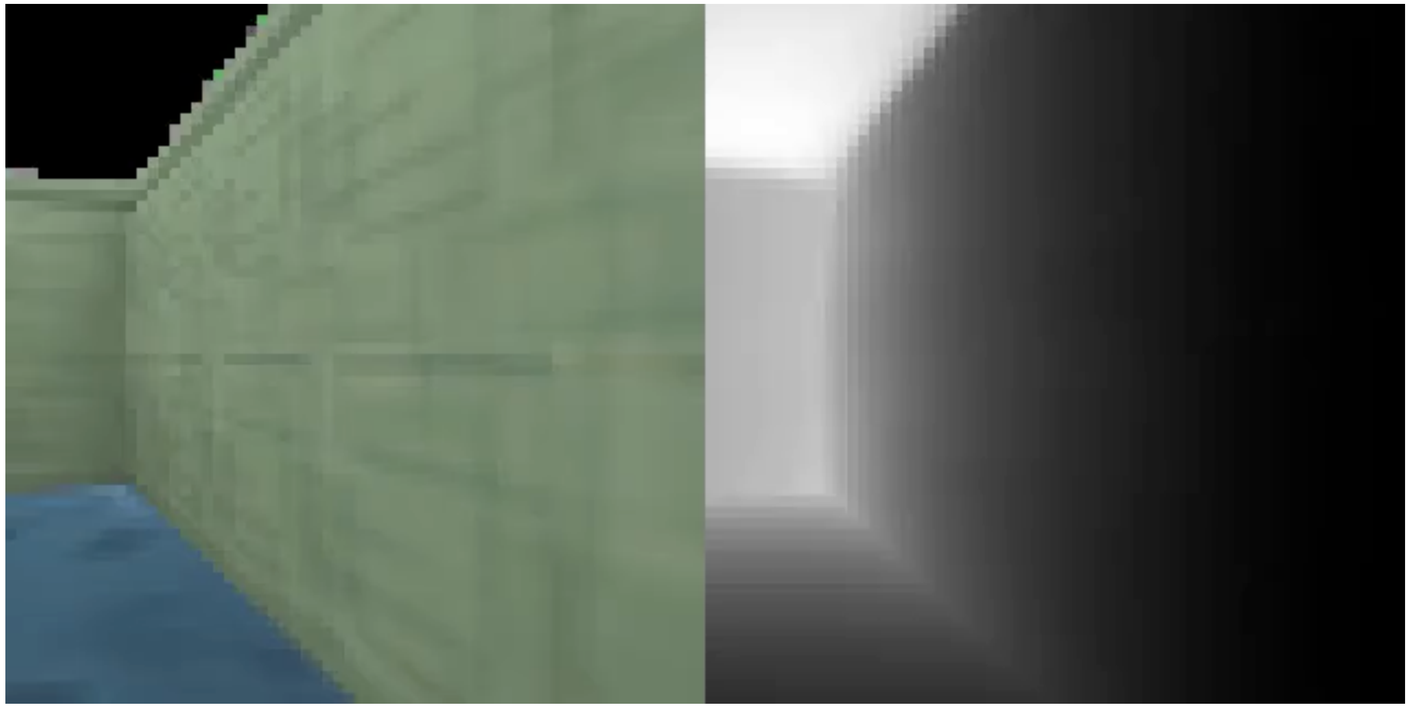
Scene Representation Networks: Continuous 3D-structure Aware Neural Scene Representations, Sitzmann et al., NeurIPS

Can generalize across simple room-scale scenes.



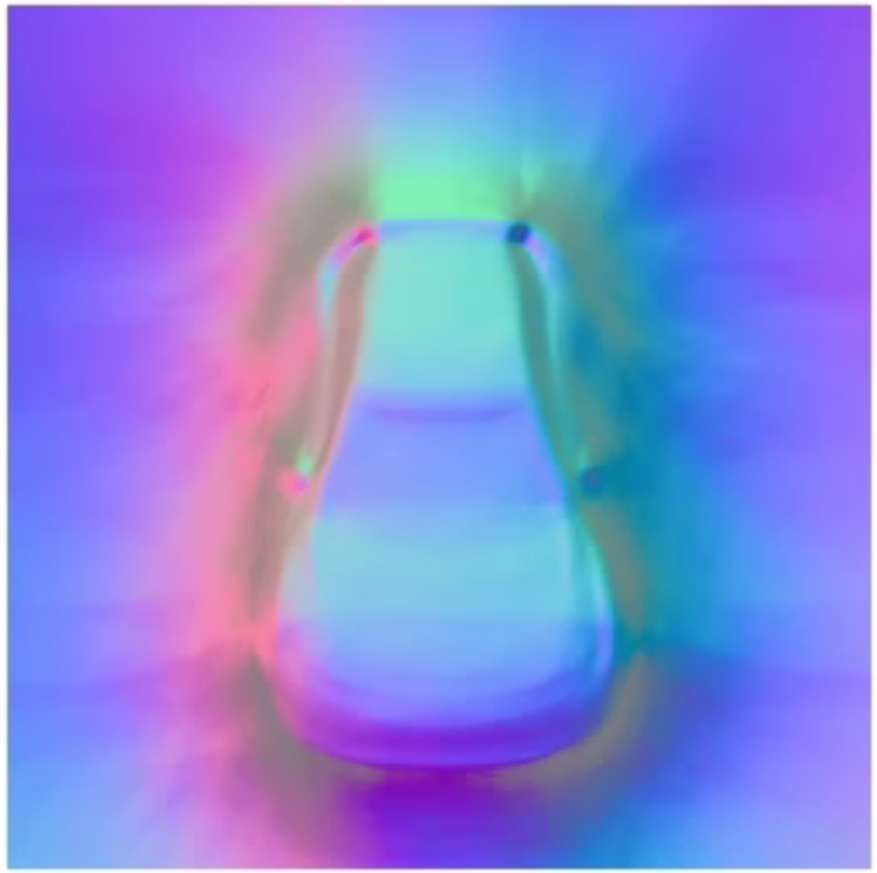
Scene Representation Networks: Continuous 3D-structure Aware Neural Scene Representations, Sitzmann et al., NeurIPS

Can generalize across simple room-scale scenes.



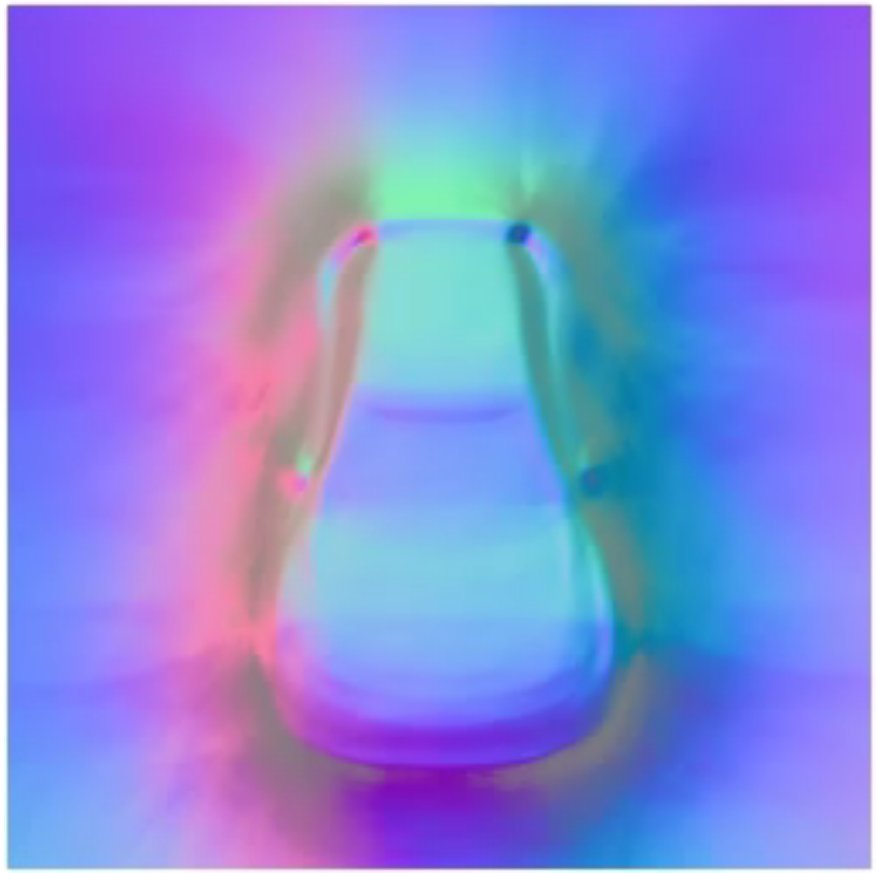
Scene Representation Networks: Continuous 3D-structure Aware Neural Scene Representations, Sitzmann et al., NeurIPS

Latent Space Interpolation



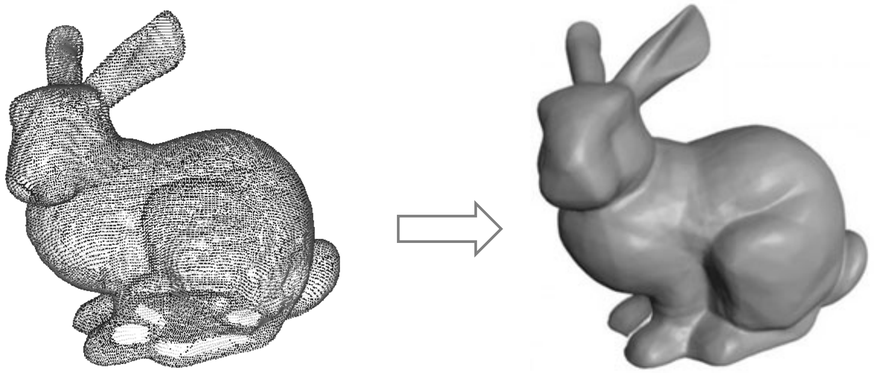


Latent Space Interpolation





Global Latent Codes: Enables reconstruction from partial observations!



DeepSDF, Occupancy Networks, IM-Net



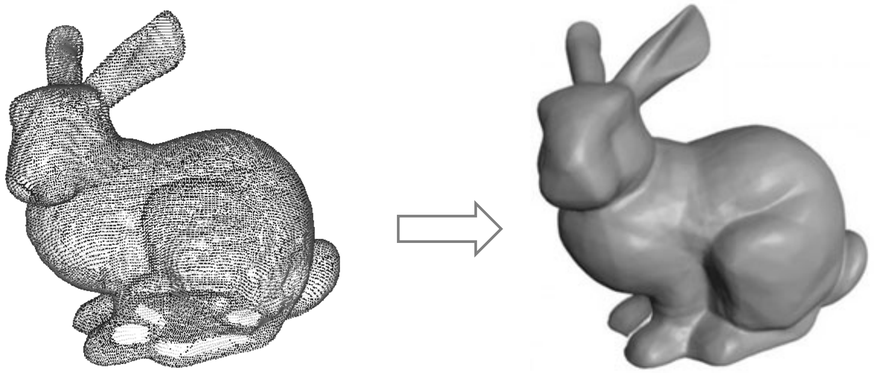


Scene Representation Networks: Continuous
3D-Structure-Aware Neural Scene Representations, NeurIPS 2019.



Differential Volumetric Rendering,
Niemeyer et al., CVPR 2020

Global Latent Codes: Enables reconstruction from partial observations!



DeepSDF, Occupancy Networks, IM-Net



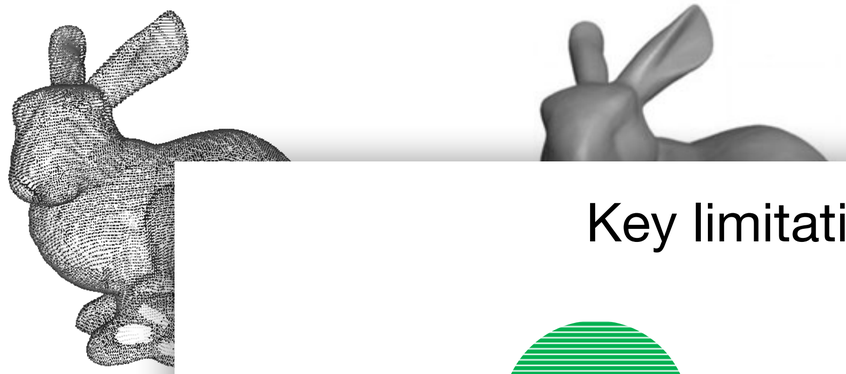


Scene Representation Networks: Continuous
3D-Structure-Aware Neural Scene Representations, NeurIPS 2019.



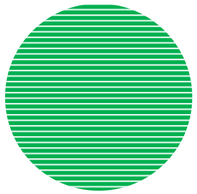
Differential Volumetric Rendering,
Niemeyer et al., CVPR 2020

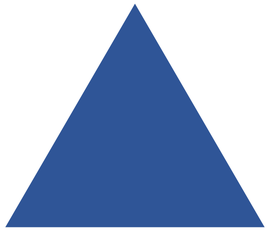
Global Latent Codes: Enables reconstruction from partial observations!

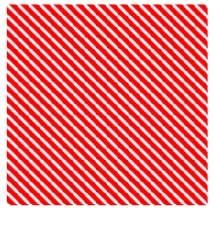


Dee

Key limitation: Non-compositional scenes.



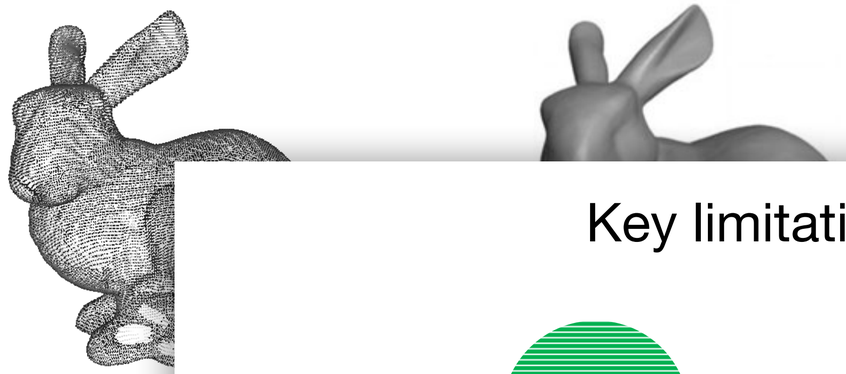




Scene Representation Networks: Continuous 3D-Structure-Aware Neural Scene Representations, NeurIPS 2019.

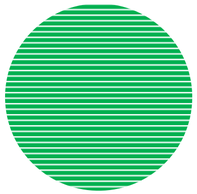
Differential Volumetric Rendering, Niemeyer et al., CVPR 2020

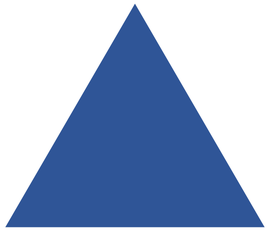
Global Latent Codes: Enables reconstruction from partial observations!

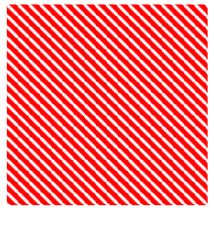


Dee

Key limitation: Non-compositional scenes.



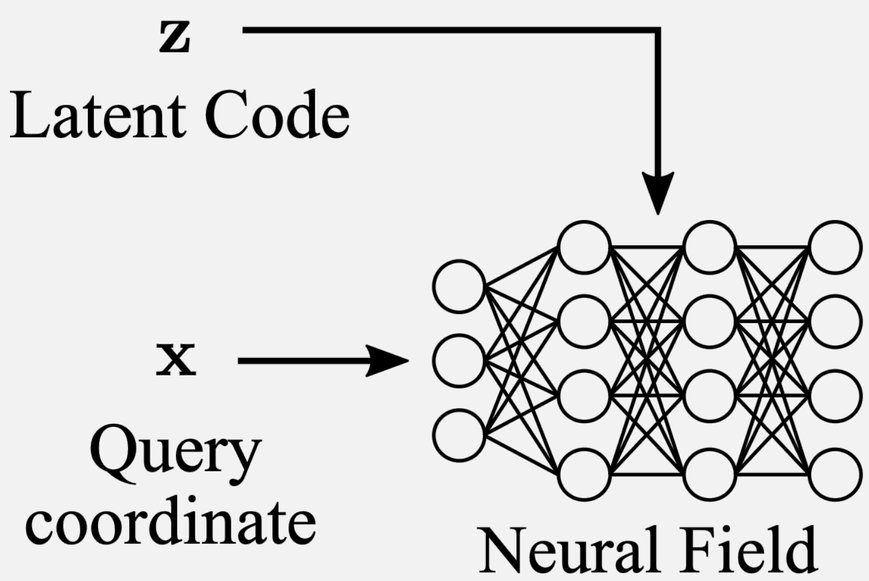




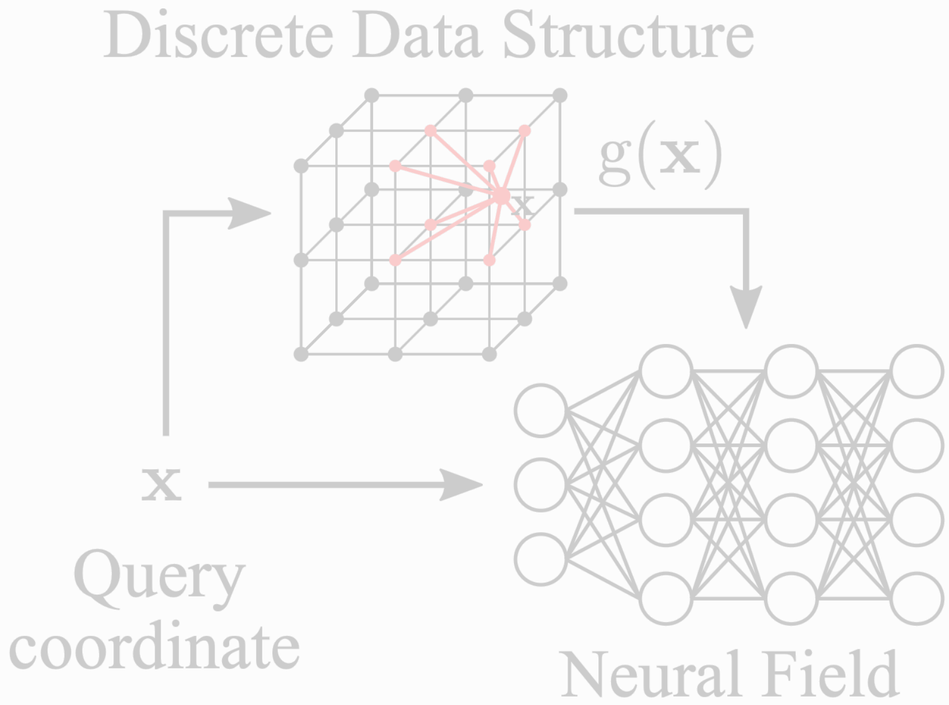
Scene Representation Networks: Continuous 3D-Structure-Aware Neural Scene Representations, NeurIPS 2019.

Differential Volumetric Rendering, Niemeyer et al., CVPR 2020

Global Latent Codes



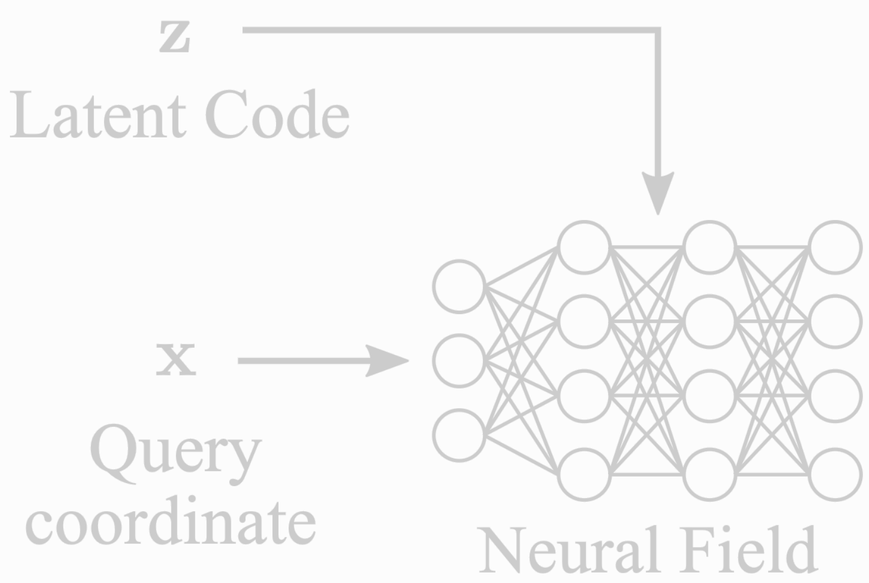
Global Conditioning



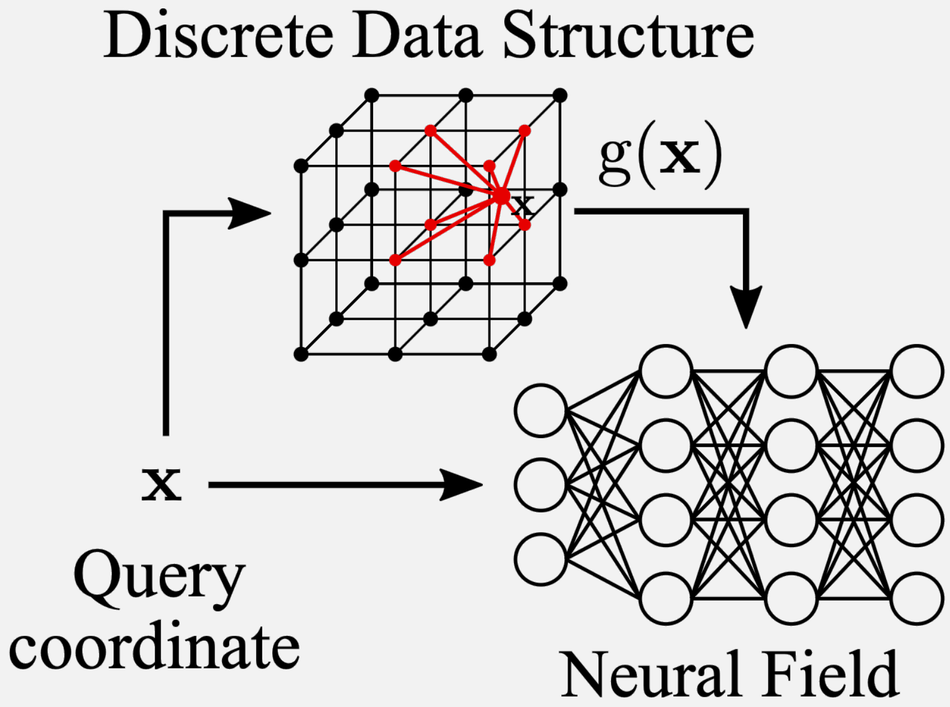
Local Conditioning

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

Local Latent Codes



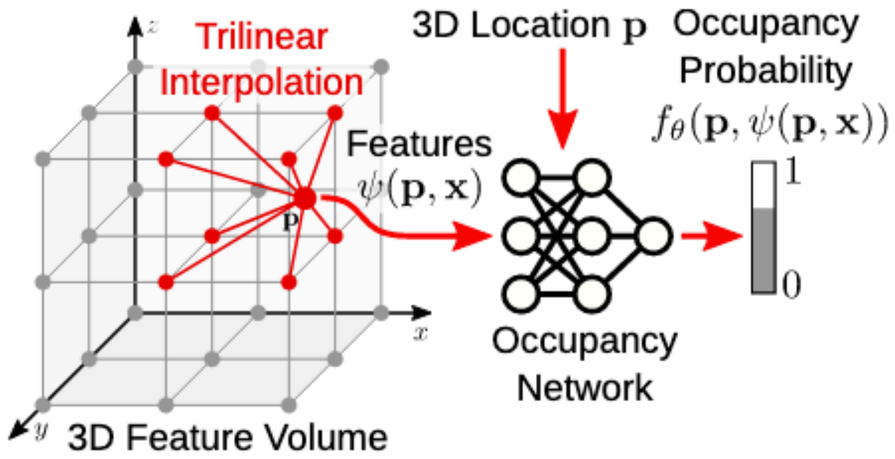
Global Conditioning



Local Conditioning

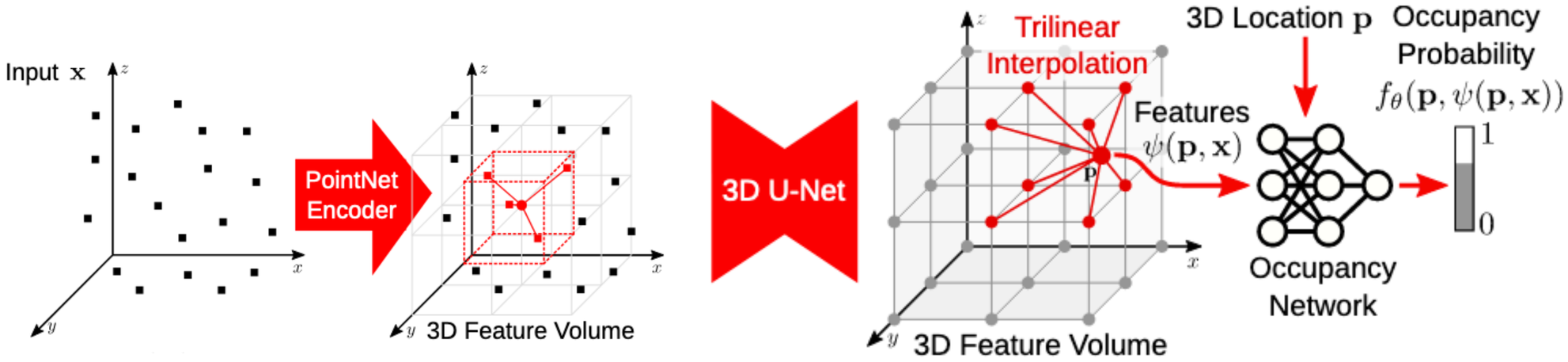
Neural Fields in Visual Computing and Beyond, Xie et al., EG STAR 2022

From point clouds: Conditioning on Feature Voxel grids



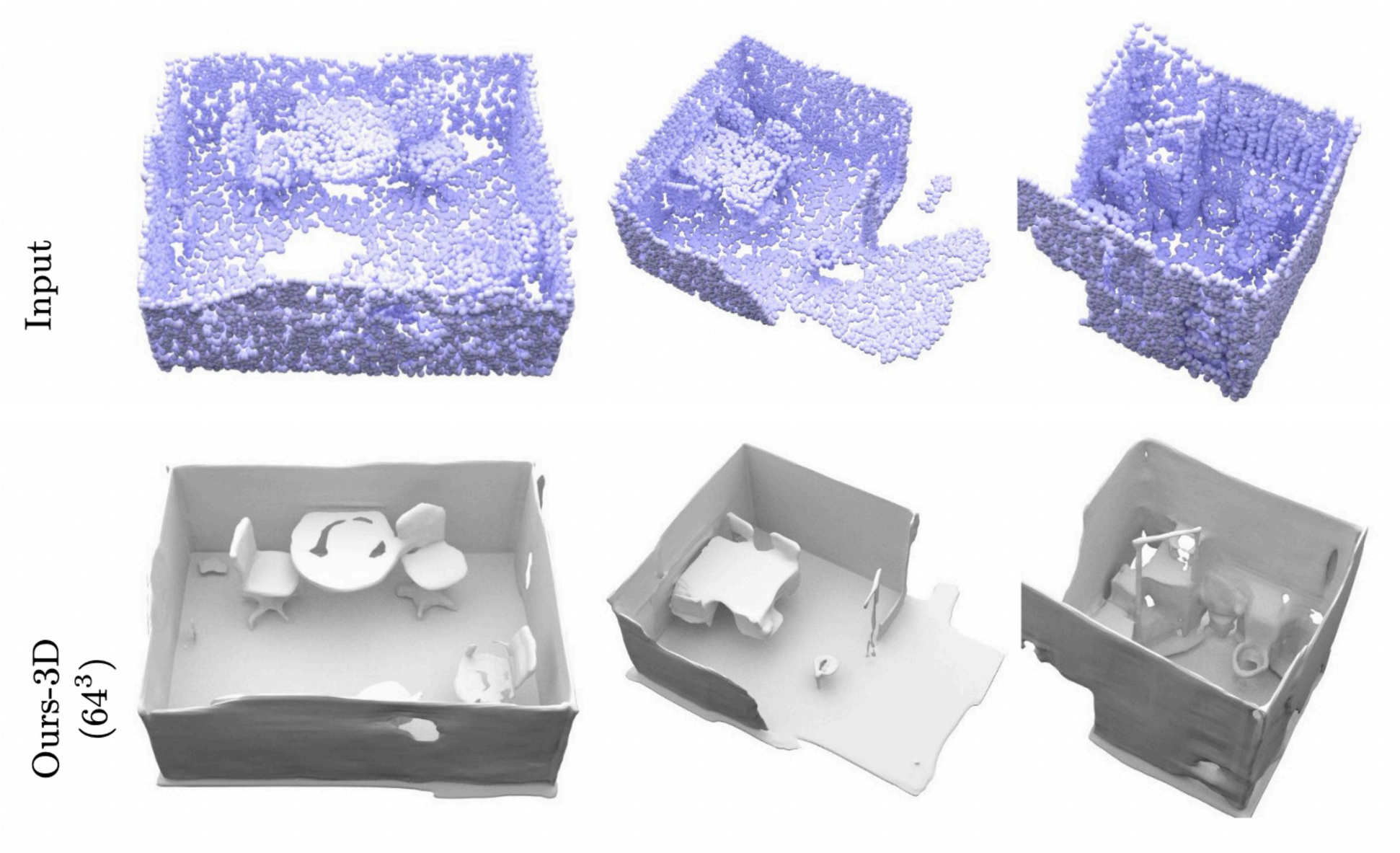
Convolutional Occupancy Networks [Peng et al. 2020]
Local Implicit Grid Representations for 3D Scenes [Jiang et al. 2020]
Implicit Functions in Feature Space for 3D Shape Reconstruction and Completion [Chabra et al. 2020]
Deep Local Shapes: Learning Local SDF Priors for Detailed 3D Reconstruction [Chibane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

From point clouds: Conditioning on Feature Voxel grids

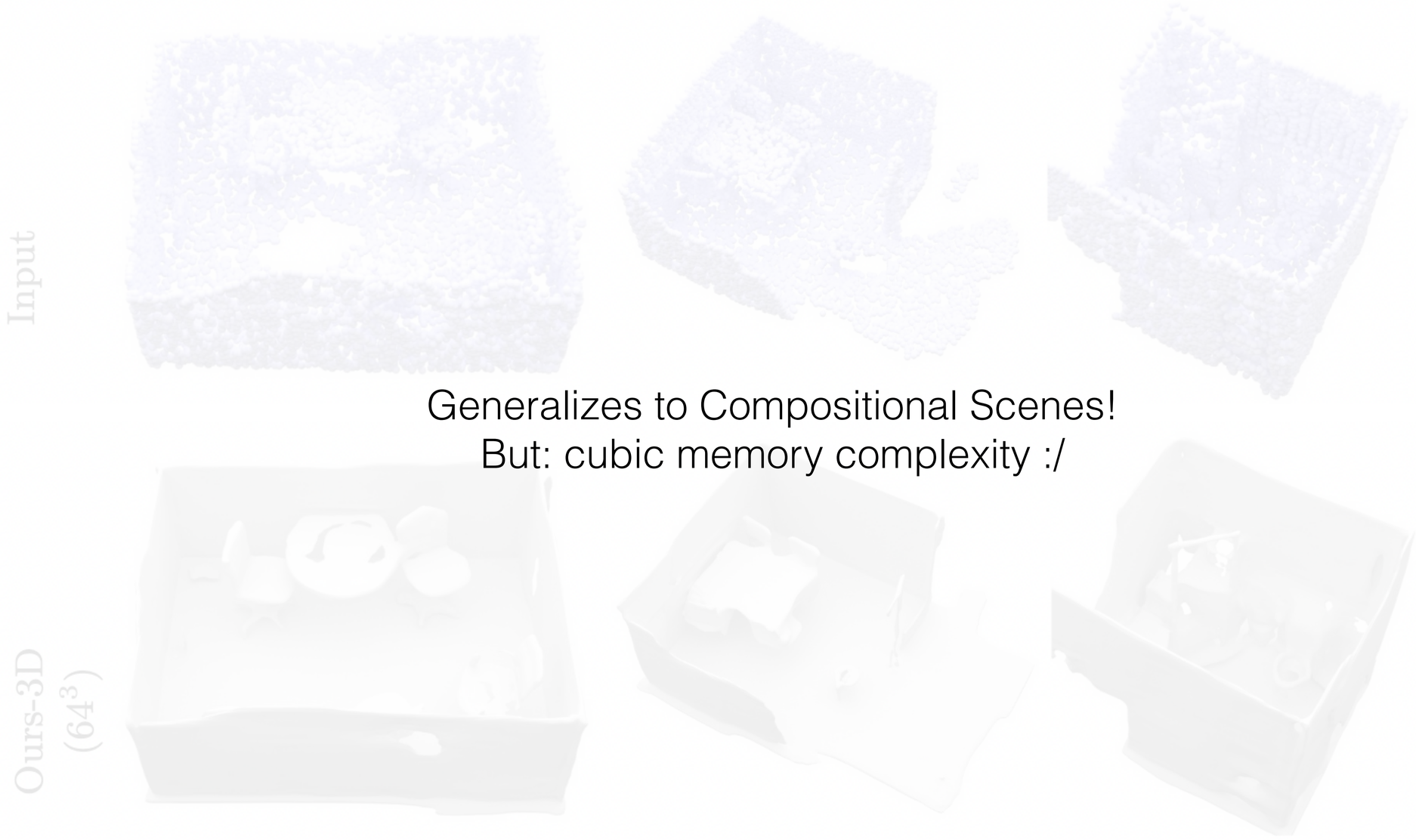


Convolutional Occupancy Networks [Peng et al. 2020]
Local Implicit Grid Representations for 3D Scenes [Jiang et al. 2020]
Implicit Functions in Feature Space for 3D Shape Reconstruction and Completion [Chabra et al. 2020]
Deep Local Shapes: Learning Local SDF Priors for Detailed 3D Reconstruction [Chibane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

From point clouds: Conditioning on Feature Voxel grids

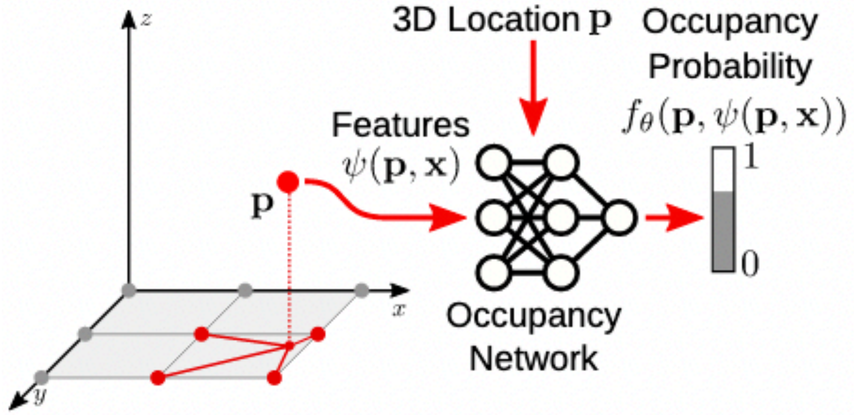


From point clouds: Conditioning on Feature Voxel grids



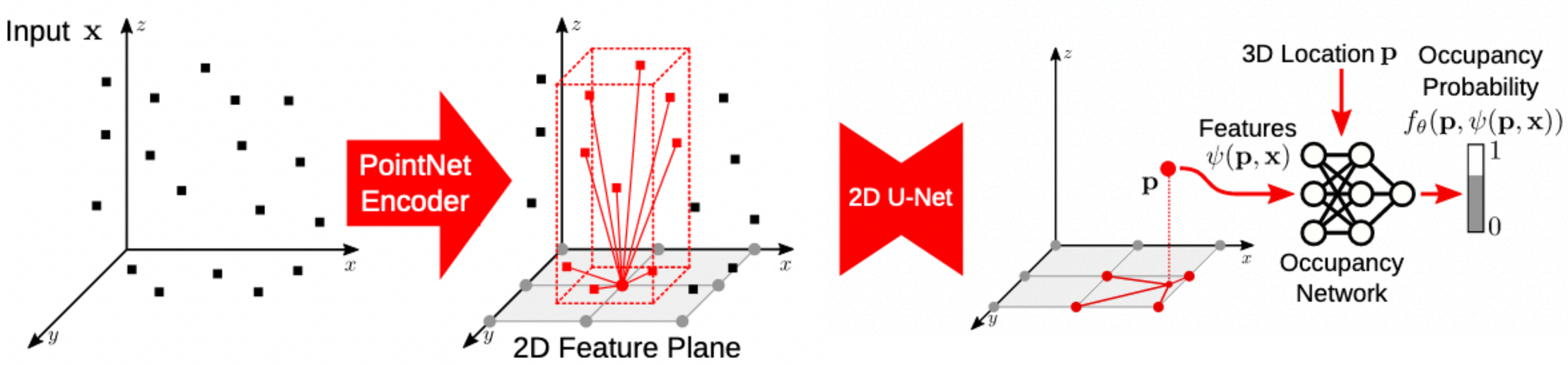
Generalizes to Compositional Scenes!
But: cubic memory complexity :/

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



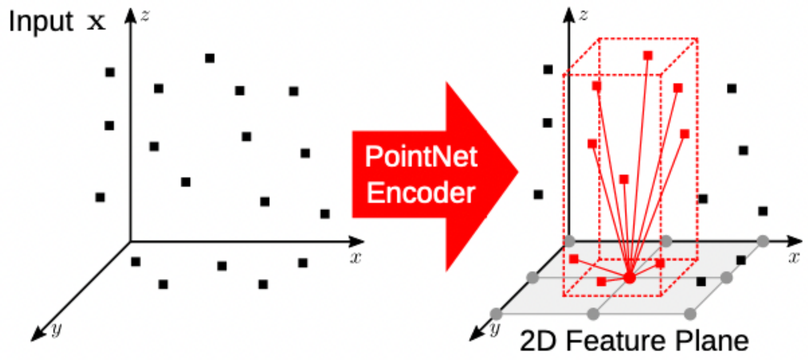
Convolutional Occupancy Networks [Peng et al. 2020]

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



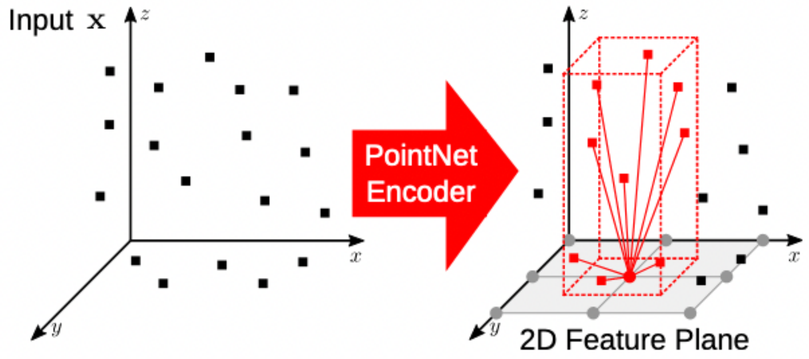
Convolutional Occupancy Networks [Peng et al. 2020]

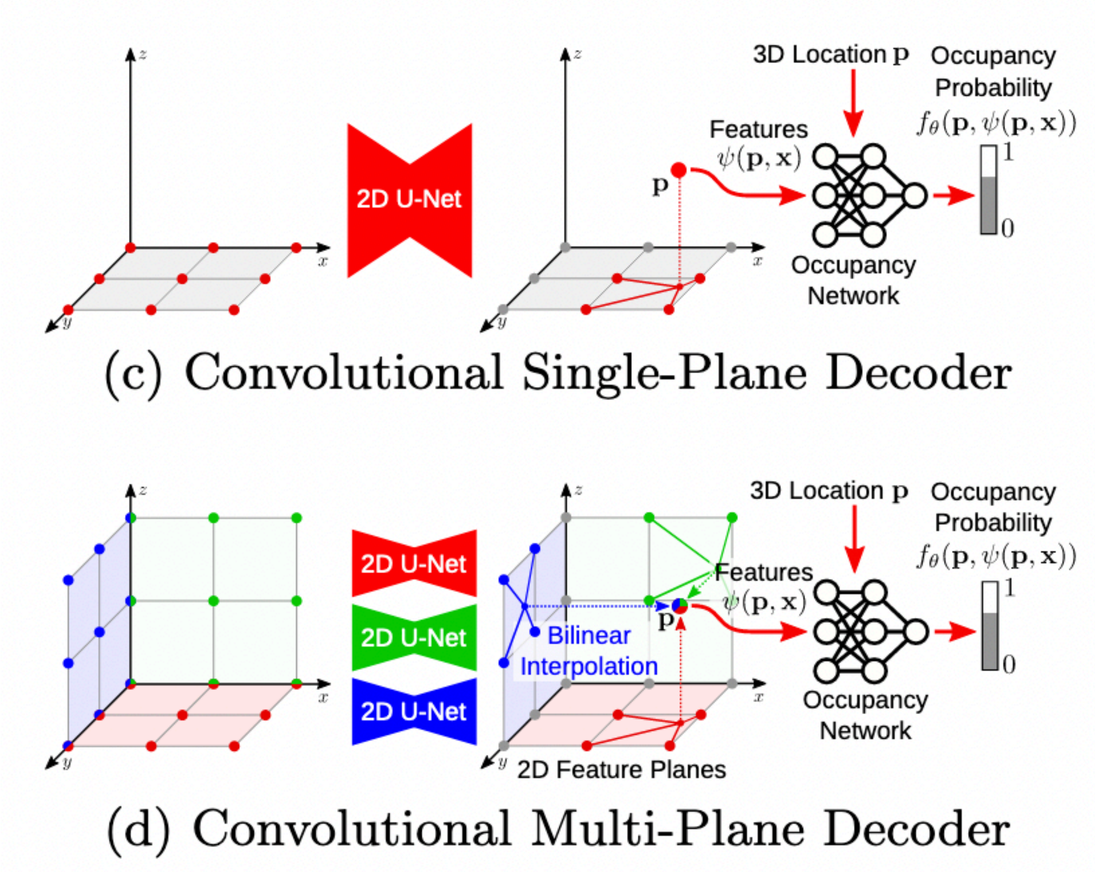
From Point clouds: Ground-plan and Tri-plane factorizations



Convolutional Occupancy Networks [Peng et al. 2020]

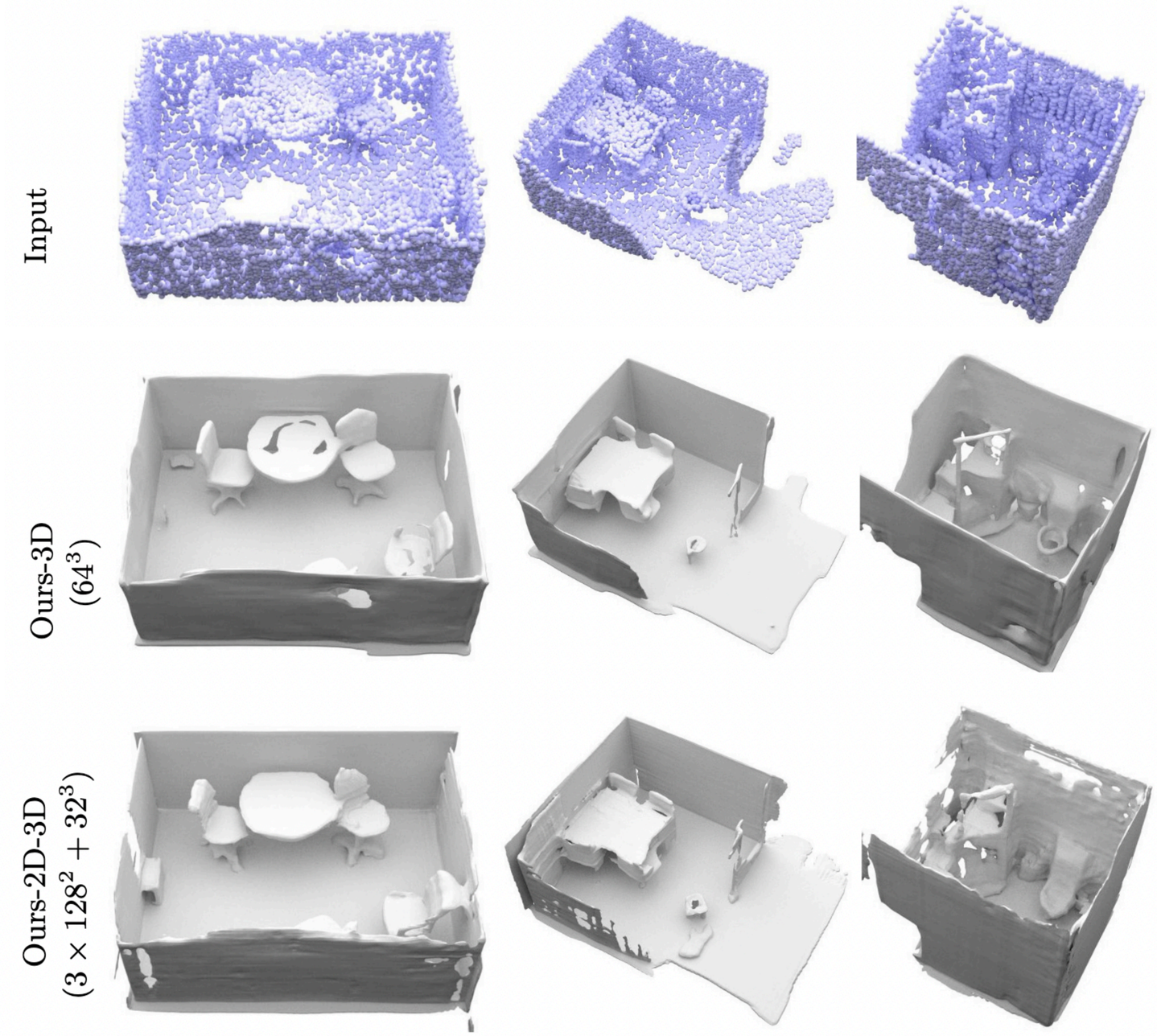
From Point clouds: Ground-plan and Tri-plane factorizations





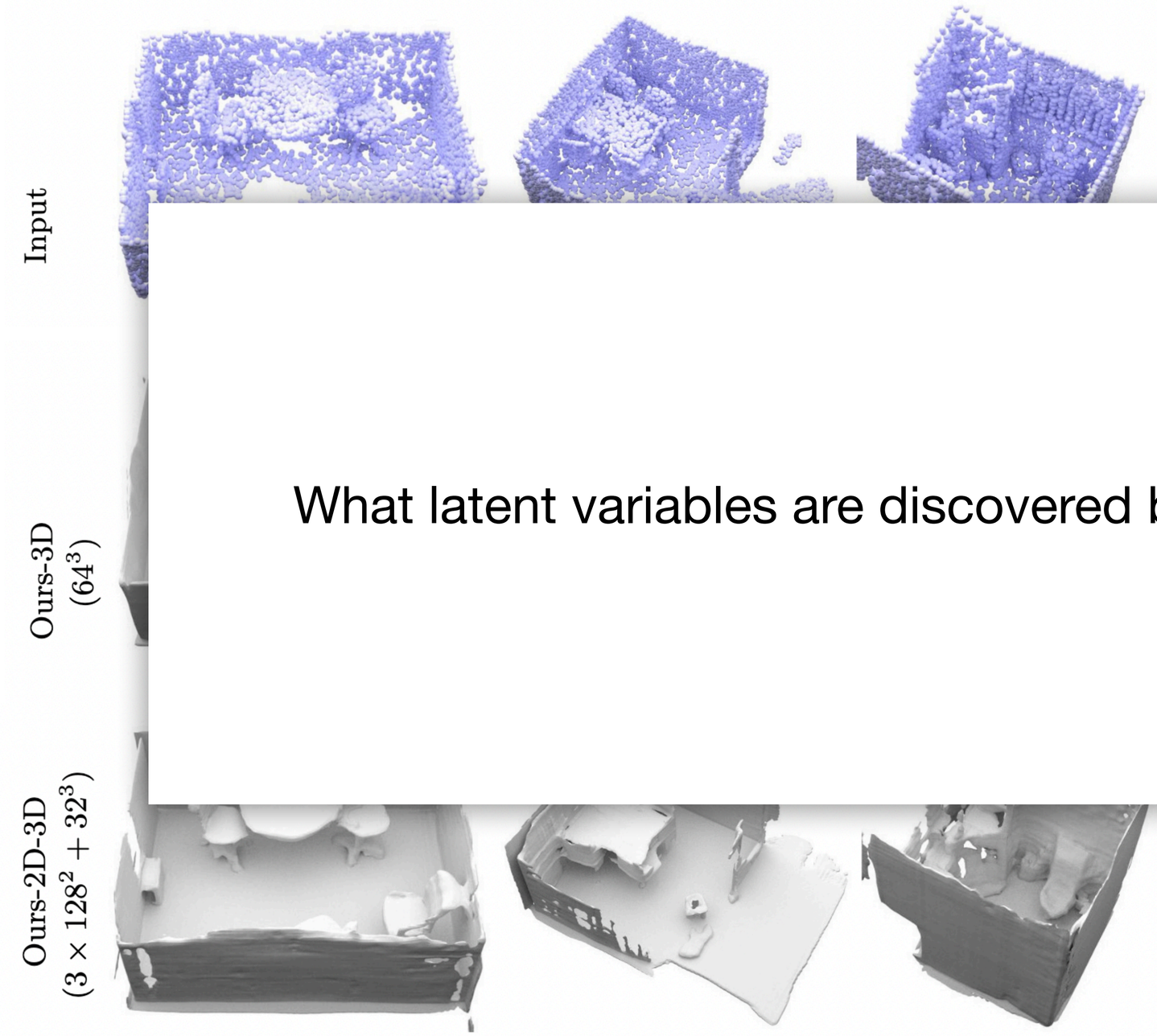
Convolutional Occupancy Networks [Peng et al. 2020]

From point clouds: Conditioning on Reconstructed Voxelgrids



5x less memory!

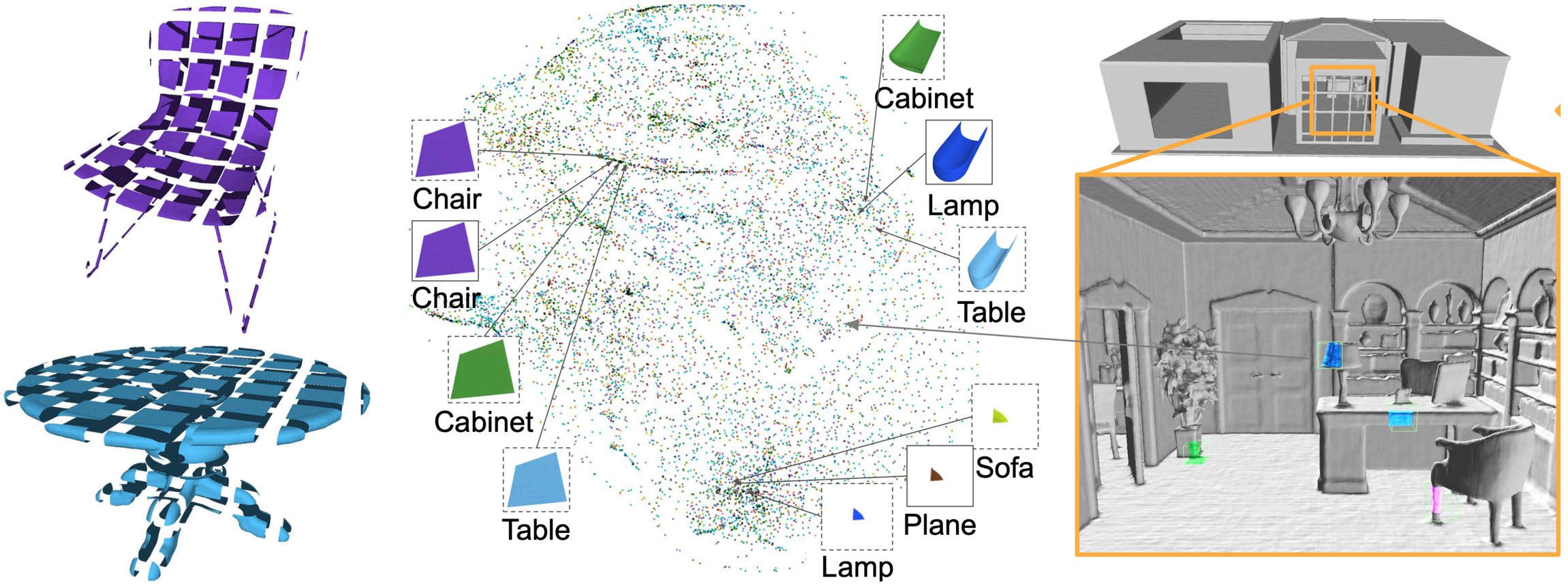
From point clouds: Conditioning on Reconstructed Voxelgrids



What latent variables are discovered by the model?

5x less memory!

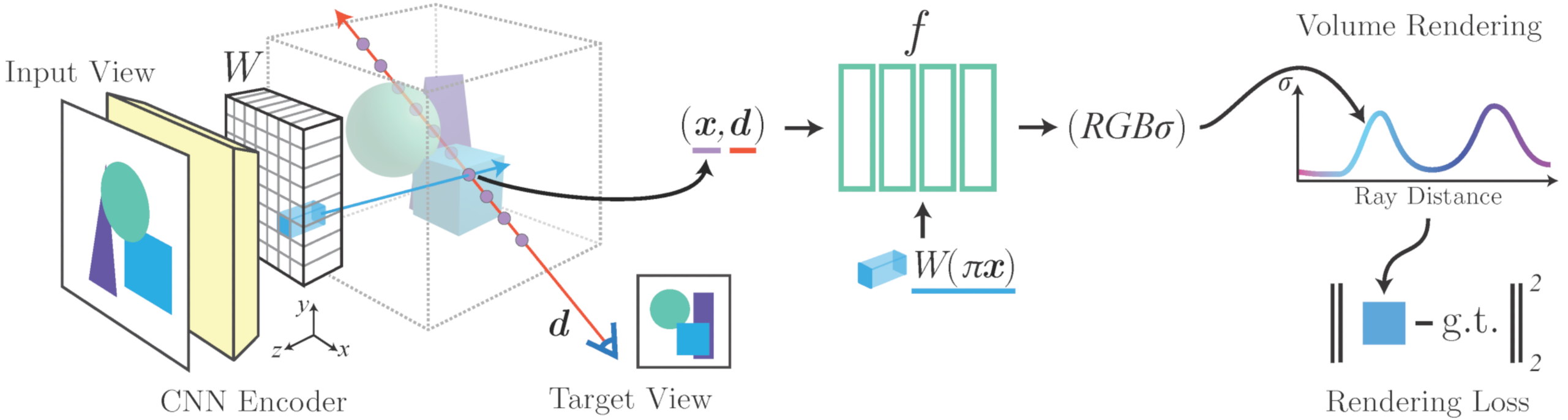
Locally Conditioned Latent Space



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

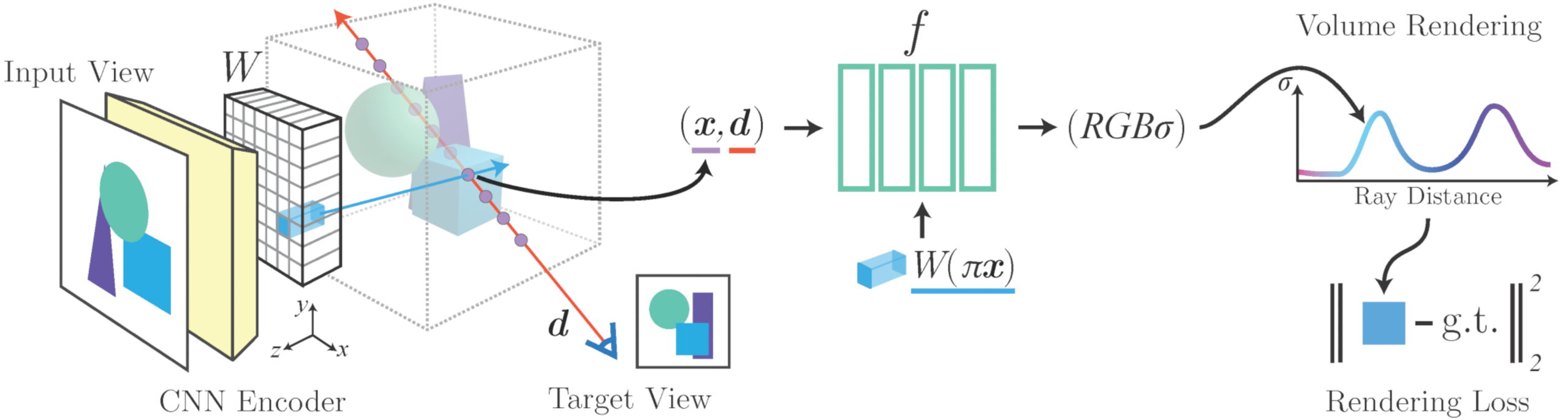
How to locally condition wrt sensor domain

Local Conditioning: Pixel-Aligned Features.



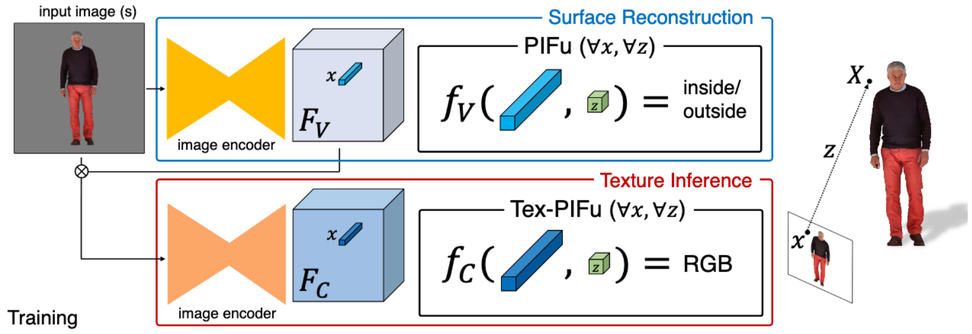
PiFU, Saito et al., ICCV 2019.
PixelNeRF, Yu et al., CVPR 2021
Grf: Learning a general radiance field…, Trevithick et al.

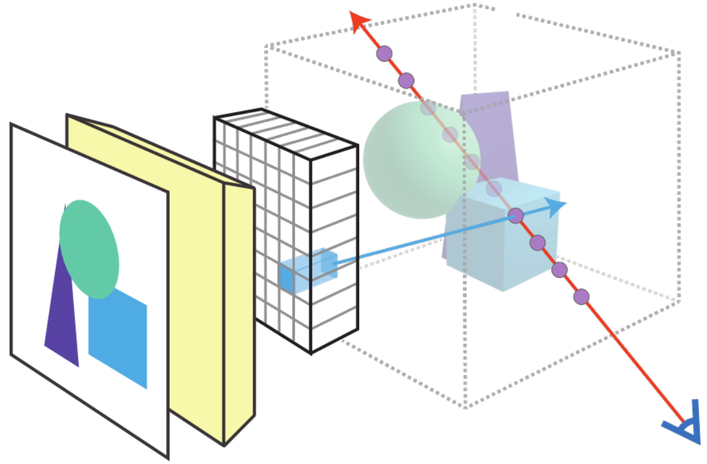
Local Conditioning: Pixel-Aligned Features.



Essentially, for each output ray location:
Project into input view
Predict RGB, alpha based on input view info and 3D location
Implicitly does something similar to monocular depth prediction

Local Conditioning: Pixel-Aligned Features.



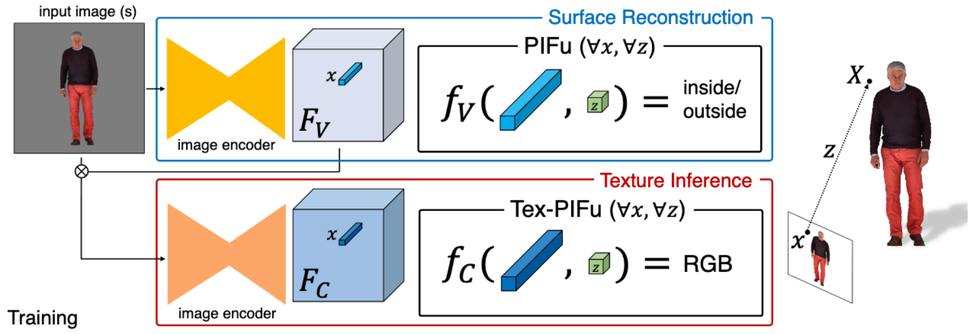


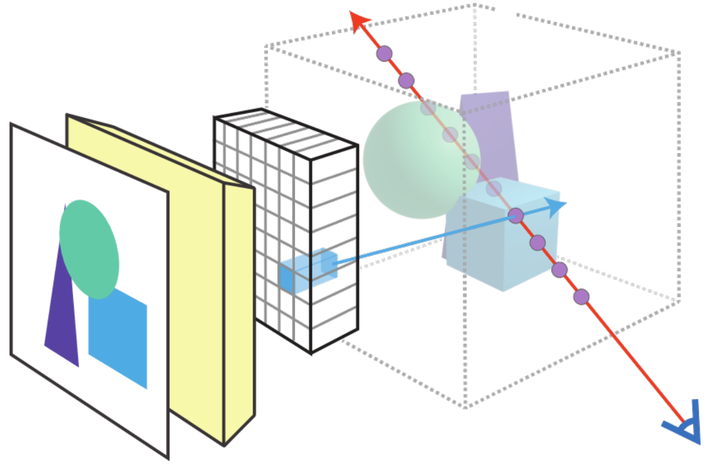
PiFU, Saito et al., ICCV 2019.
PixelNeRF, Yu et al., CVPR 2021
Grf: Learning a general radiance field…, Trevithick et al.

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



arXiv:2311.04400v2 [cs.CV] 9 Mar 2024

LRM: Large Reconstruction Model for Single Image to 3D

Yicong Hong[1,2*] **Kai Zhang**[1] **Jiuxiang Gu**[1] **Sai Bi**[1] **Yang Zhou**[1]
Difan Liu[1] **Feng Liu**[1] **Kalyan Sunkavalli**[1] **Trung Bui**[1] **Hao Tan**[1]
[1]Adobe Research [2]Australian National Univeristy
mr.yiconghong@gmail.com
{kaiz,jigu,sbi,yazhou,diliu,fengl,sunkaval,bui,hatan}@adobe.com

Abstract

We propose the first Large Reconstruction Model (LRM) that predicts the 3D model of an object from a single input image within just 5 seconds. In contrast to many previous methods that are trained on small-scale datasets such as ShapeNet in a category-specific fashion, LRM adopts a highly scalable transformer-based architecture with 500 million learnable parameters to directly predict a neural radiance field (NeRF) from the input image. We train our model in an end-to-end manner on massive multi-view data containing around 1 million objects, including both synthetic renderings from Objaverse and real captures from MVImgNet. This combination of a high-capacity model and large-scale training data empowers our model to be highly generalizable and produce high-quality 3D reconstructions from various testing inputs, including real-world in-the-wild captures and images created by generative models. Video demos and interactable 3D meshes can be found on our LRM project webpage: https://yiconghong.me/LRM.

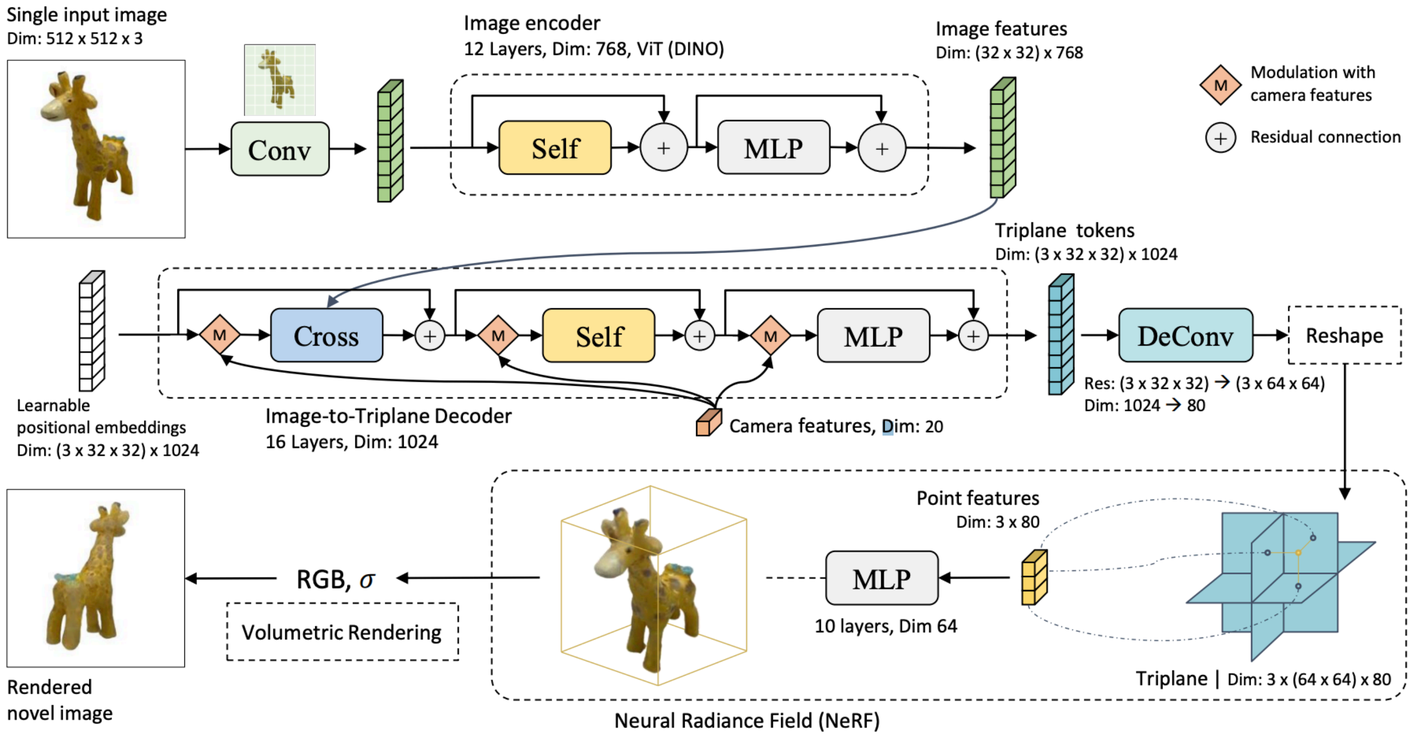
1 Introduction

Imagine if we could instantly create a 3D shape from a single image of an arbitrary object. Broad applications in industrial design, animation, gaming, and AR/VR have strongly motivated relevant research in seeking a generic and efficient approach towards this long-standing goal. Due to the underlying ambiguity of 3D geometry in a single view, early learning-based methods usually perform well on specific categories, utilizing the category data prior to infer the overall shape (Yu et al., 2021). Recently, advances in image generation, such as DALL-E (Ramesh et al., 2021) and Stable Diffusion (Rombach et al., 2022), have inspired research that leverages the remarkable generalization capability of 2D diffusion models to enable multi-view supervision (Liu et al., 2023b; Tang et al., 2023). However, many of these methods require delicate parameter tuning and regularization, and their results are limited by the pre-trained 2D generative models. Meanwhile, there are many approaches that rely on per-shape optimization (*e.g.* optimize a NeRF (Mildenhall et al., 2021; Chan et al., 2022; Chen et al., 2022a; Müller et al., 2022; Sun et al., 2022)) to construct a consistent geometry; this process is often slow and impractical.

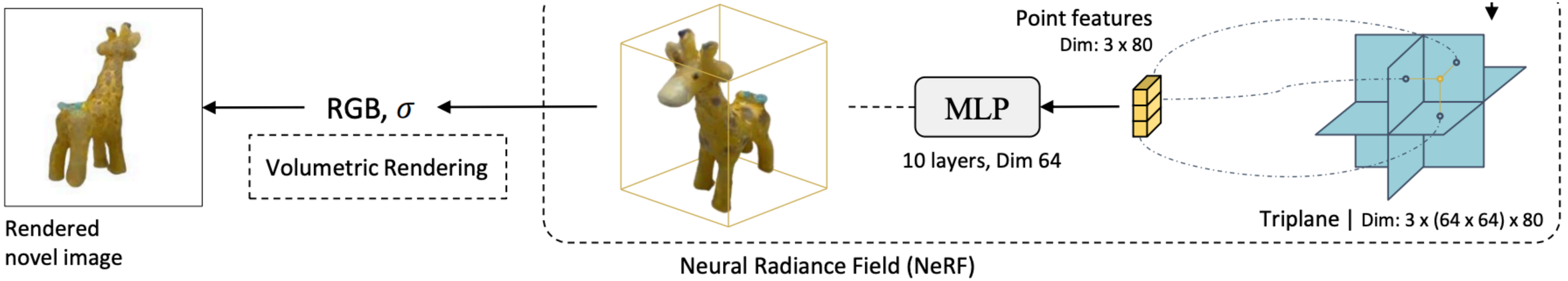
On the other hand, the great success in natural language processing (Devlin et al., 2018; Brown et al., 2020; Chowdhery et al., 2022) and image processing (Caron et al., 2021; Radford et al., 2021; Alayrac et al., 2022; Ramesh et al., 2022) can be largely credited to three critical factors: (1) using highly scalable and effective neural networks, such as the Transformers (Vaswani et al., 2017), for modeling the data distribution, (2) enormous datasets for learning generic priors, as well as (3) self-supervised-like training objectives that encourage the model to discover the underlying data structure while maintaining high scalability. For instance, the GPT (generative pre-trained transformer) series (Radford et al., 2019; Brown et al., 2020; OpenAI, 2023) build large language models with huge transformer networks, large-scale data, and the simple next-word prediction task. In light of this, we pose the same question for 3D: given sufficient 3D data and a large-scale training framework, ***is it possible to learn a generic 3D prior for reconstructing an object from a single image?***

*Intern at Adobe Research.

Trained on big 3D dataset
From image to NERF triplane representation
Builds on DINO image features encoder
Transformer decoder turns DINO into triplane NERF features
Other bells and whistles



Rendering model (right to left)
Same idea as grid but separable to reduce dimensionality
(Three 2D grids instead of 3D grids)
Project x, y z to 3 orthogonal planes, get one set of features per plane, concatenate



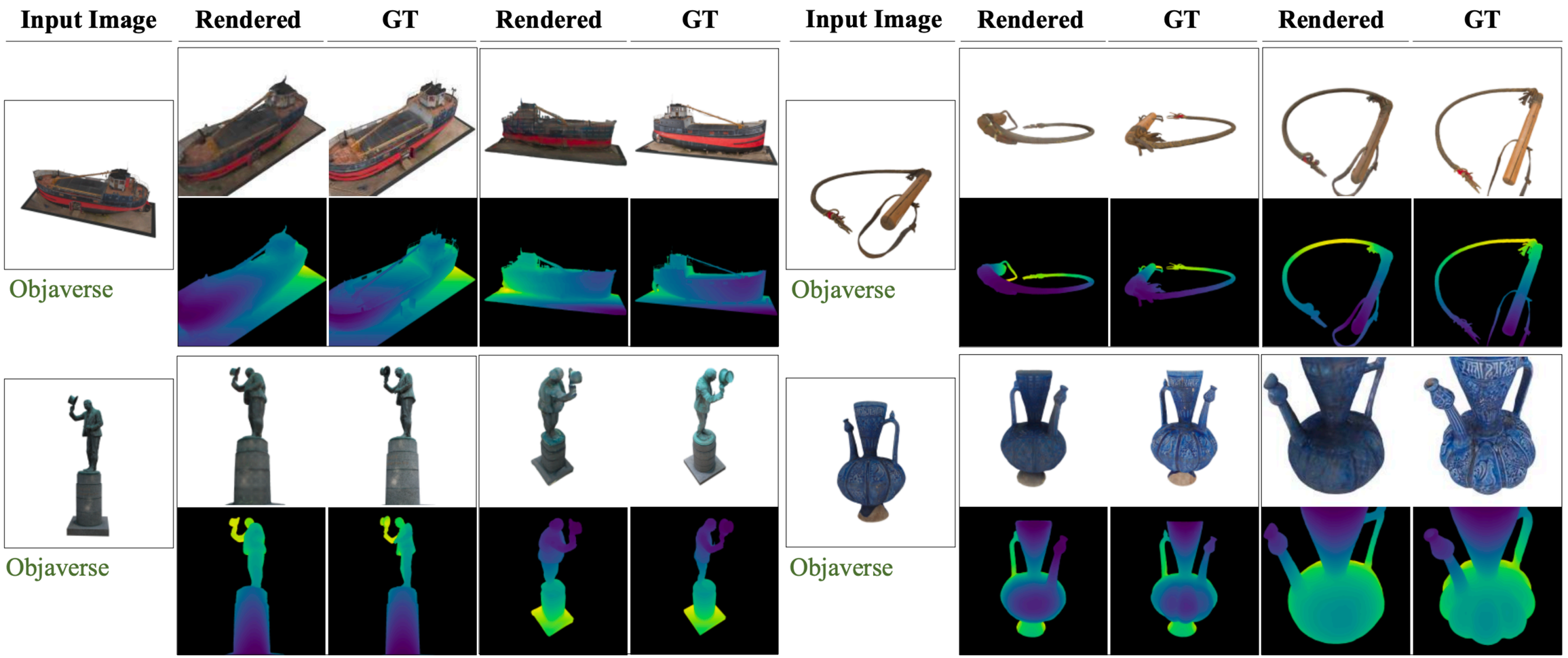
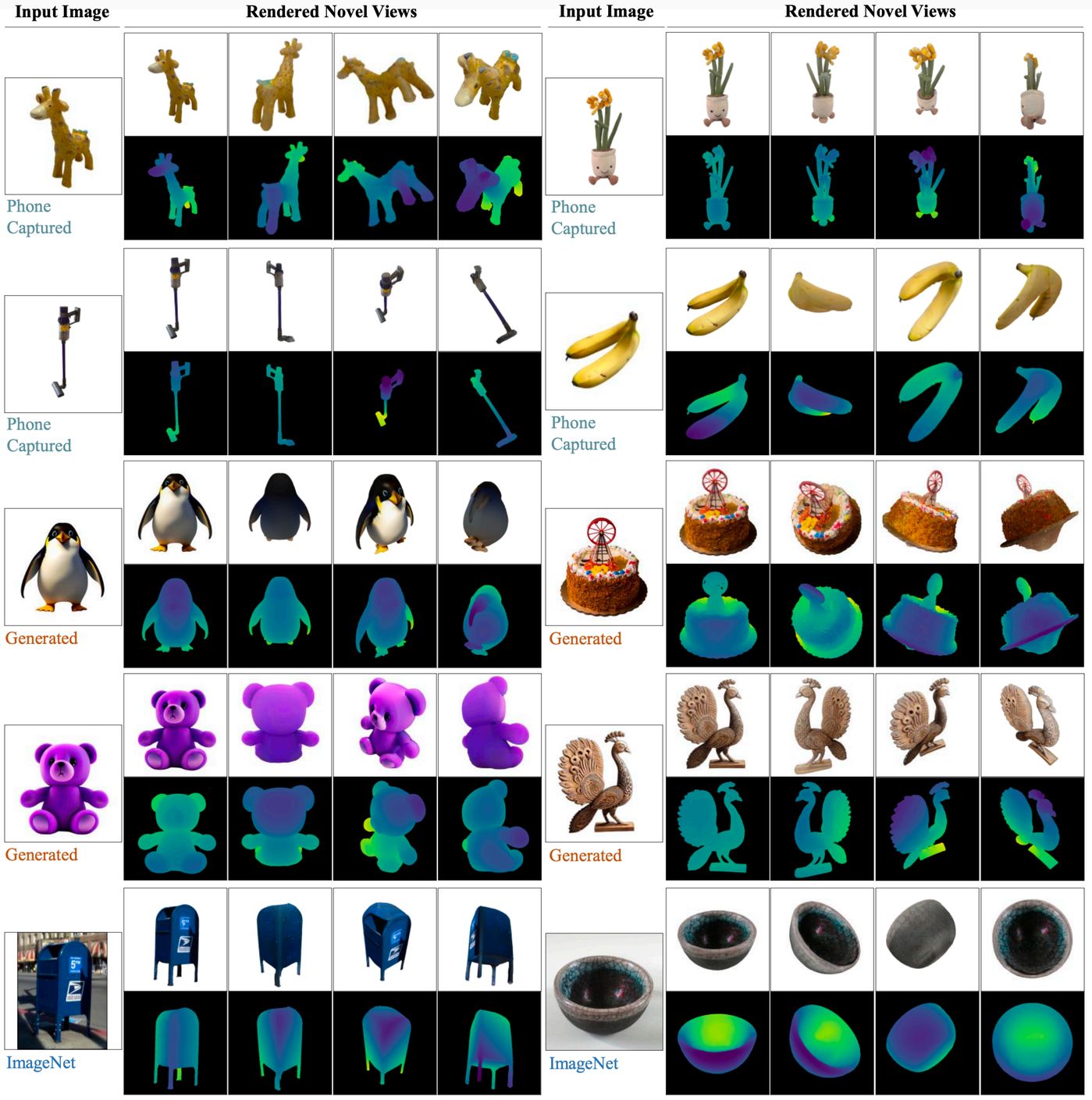


Figure 2: Rendered novel views (RGB and depth) of shapes reconstructed by our LRM from single images. None of the images are observed by the model during training. Generated images are created using Adobe Firefly. The last two rows compare our results to the rendered ground truth images of Objaverse objects (GT). Please zoom in for clearer visualization.

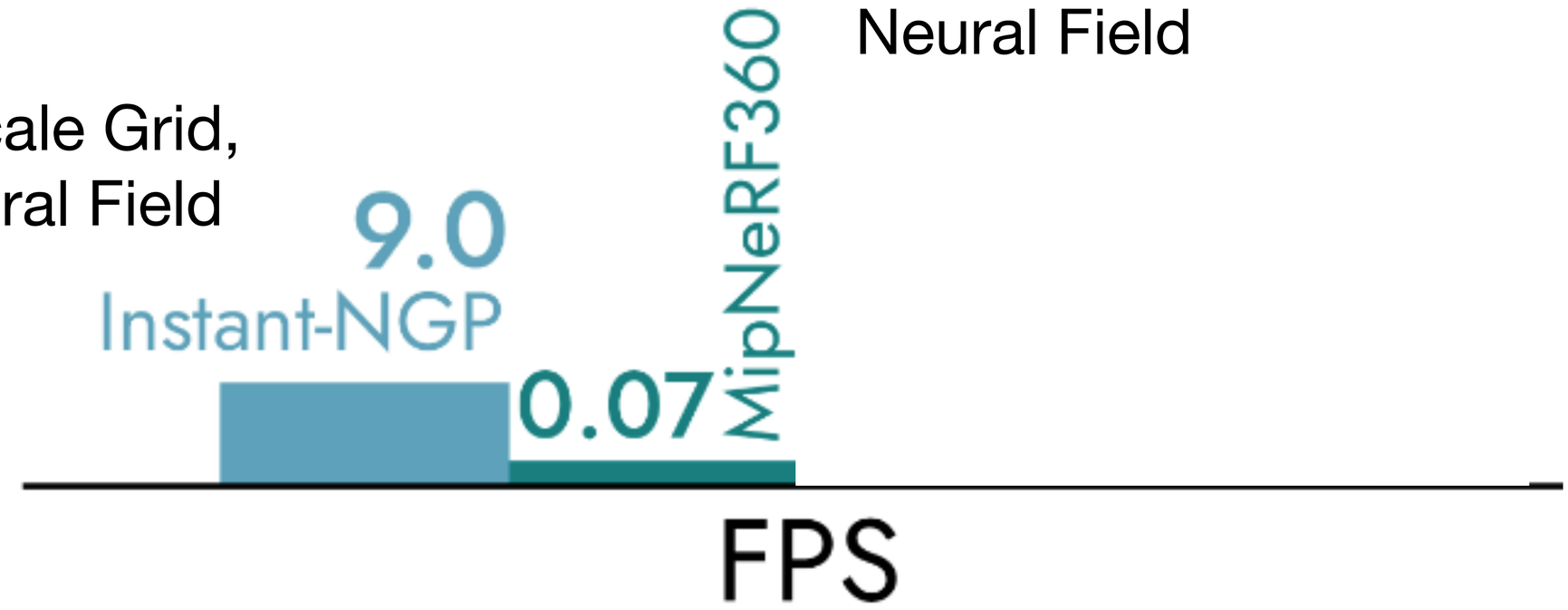

Input Image
Rendered Novel Views
Input Image
Rendered Novel Views
Phone Captured
Phone Captured
Phone Captured
Phone Captured
Generated
Generated
Generated
Generated
ImageNet
ImageNet

LRM open source version

- https://github.com/3DTopia/OpenLRM
-

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



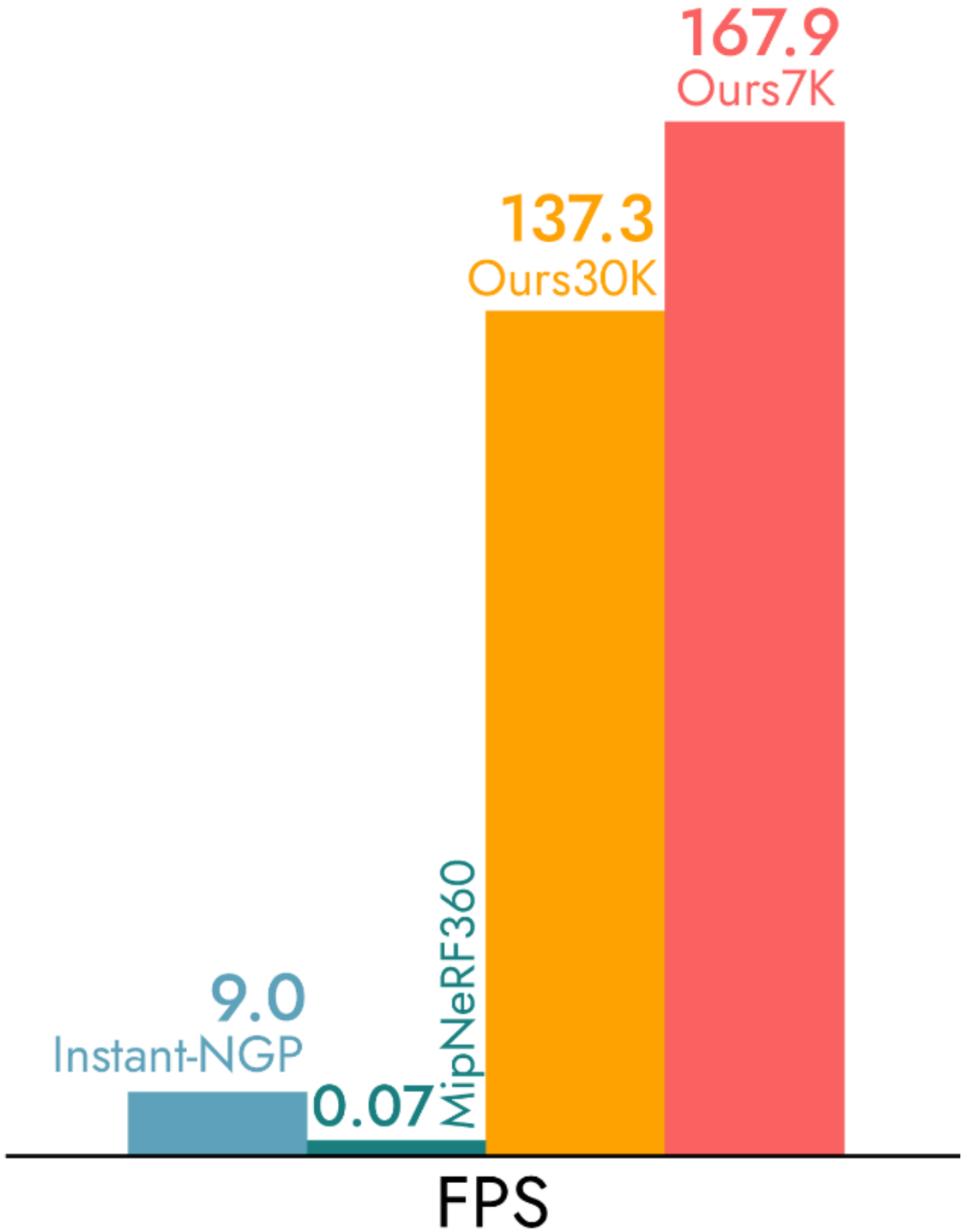
Fig. 1. Our method achieves real-time rendering of radiance fields while only requiring optimization times competitive with the faste performance is a novel 3D Gaussian scene representation coupled optimization and novel view synthesis. Note that for comparable trai this is the maximum quality they reach,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

EWA Splatting

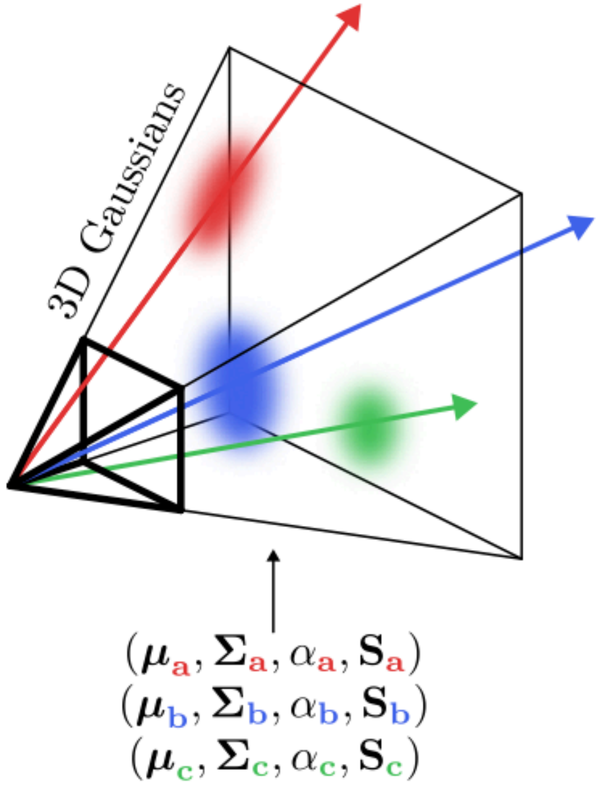
Matthias Zwicker, Hanspeter Pfister, *Member*, *IEEE*,
Jeroen van Baar, and Markus Gross, *Member*, *IEEE*

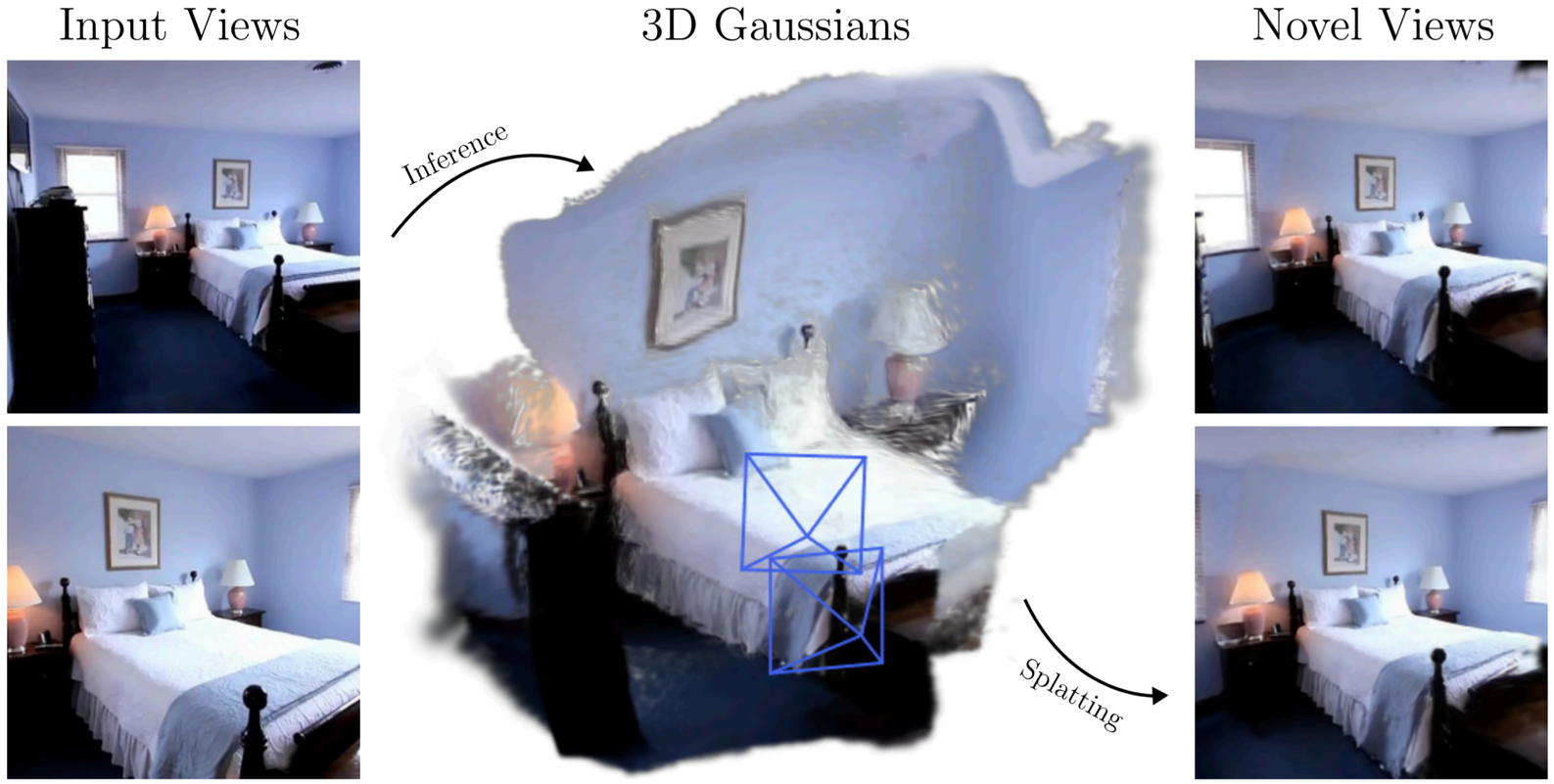
Abstract—In this paper, we present a framework for high quality splatting based on elliptical Gaussian kernels. To avoid aliasing artifacts, we introduce the concept of a resampling filter, combining a reconstruction kernel with a low-pass filter. Because of the similarity to Heckbert's EWA (elliptical 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.

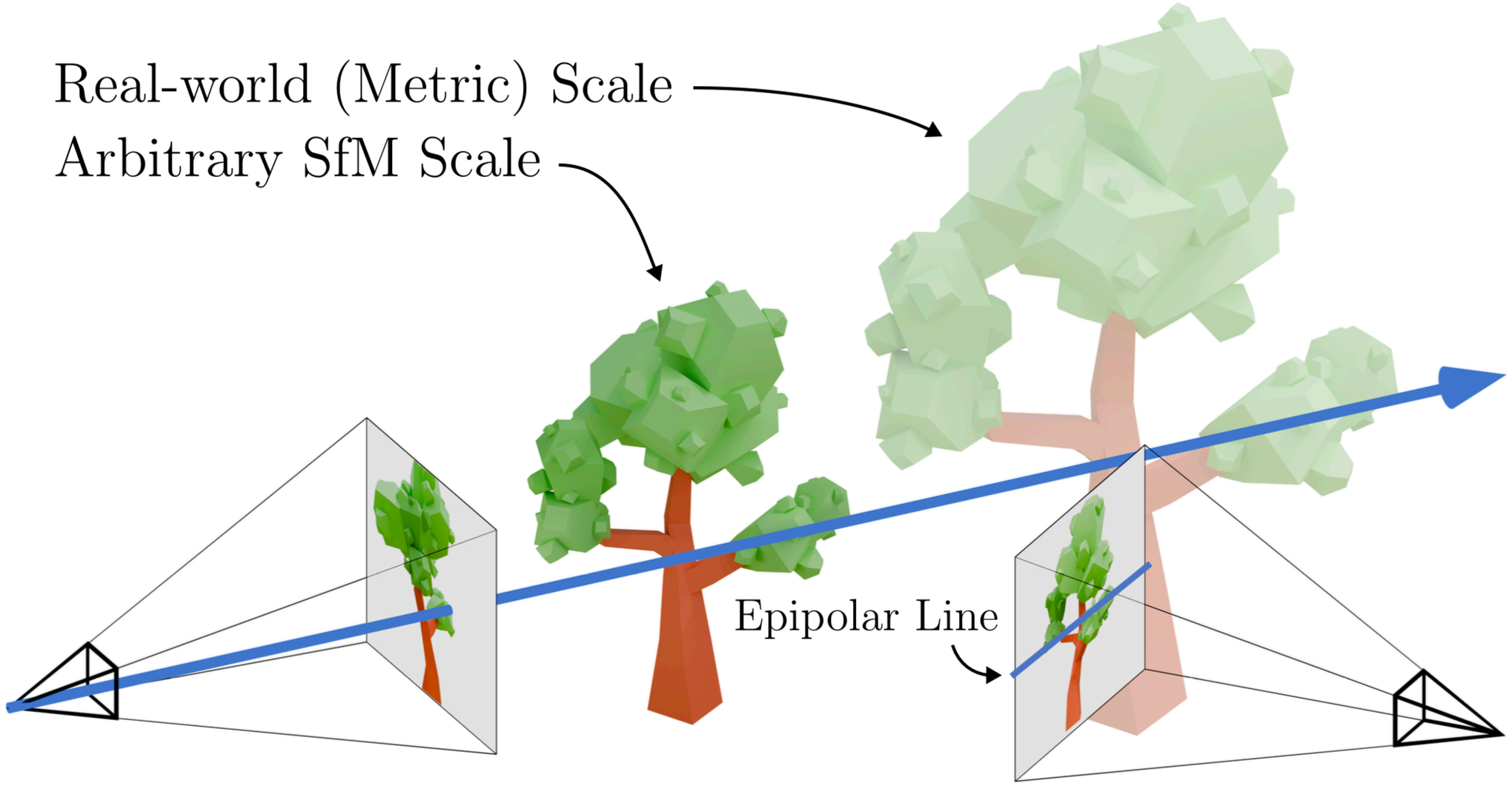

167.9
Ours7K
137.3
Ours30K
9.0
Instant-NGP
0.07
MipNeRF360
FPS

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

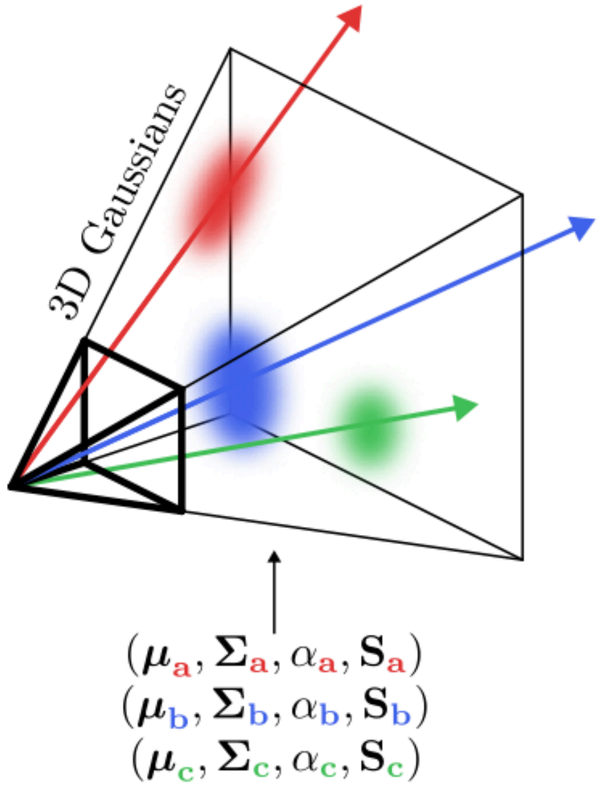




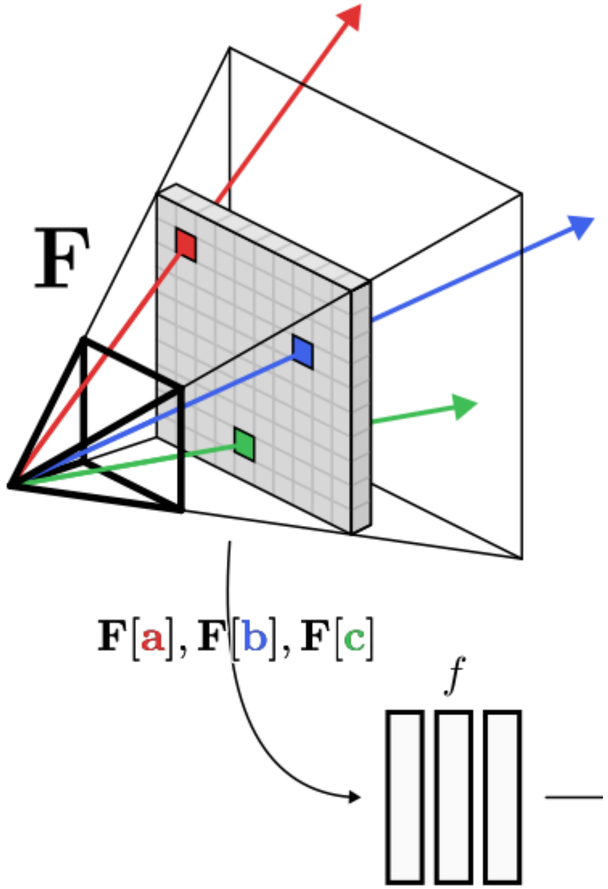
Core Challenge 1: Scale Ambiguity

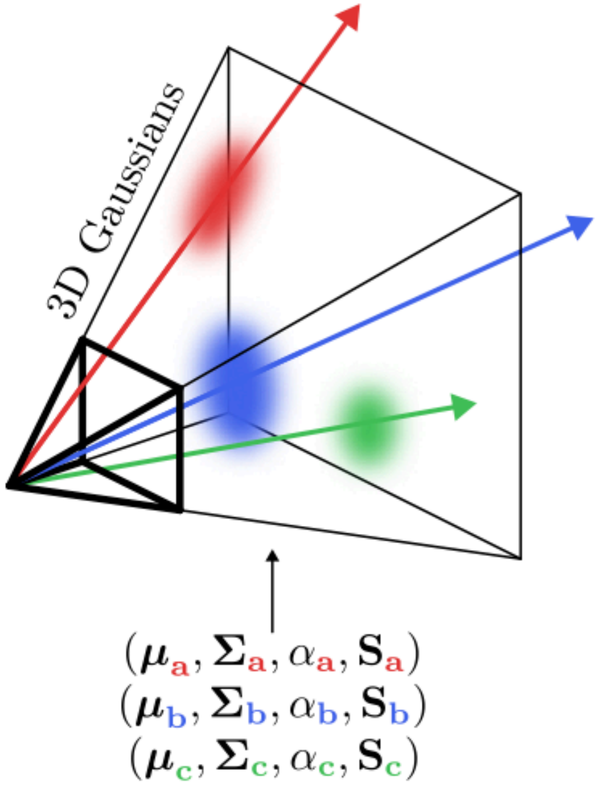


Core Challenge 2: Local Minima

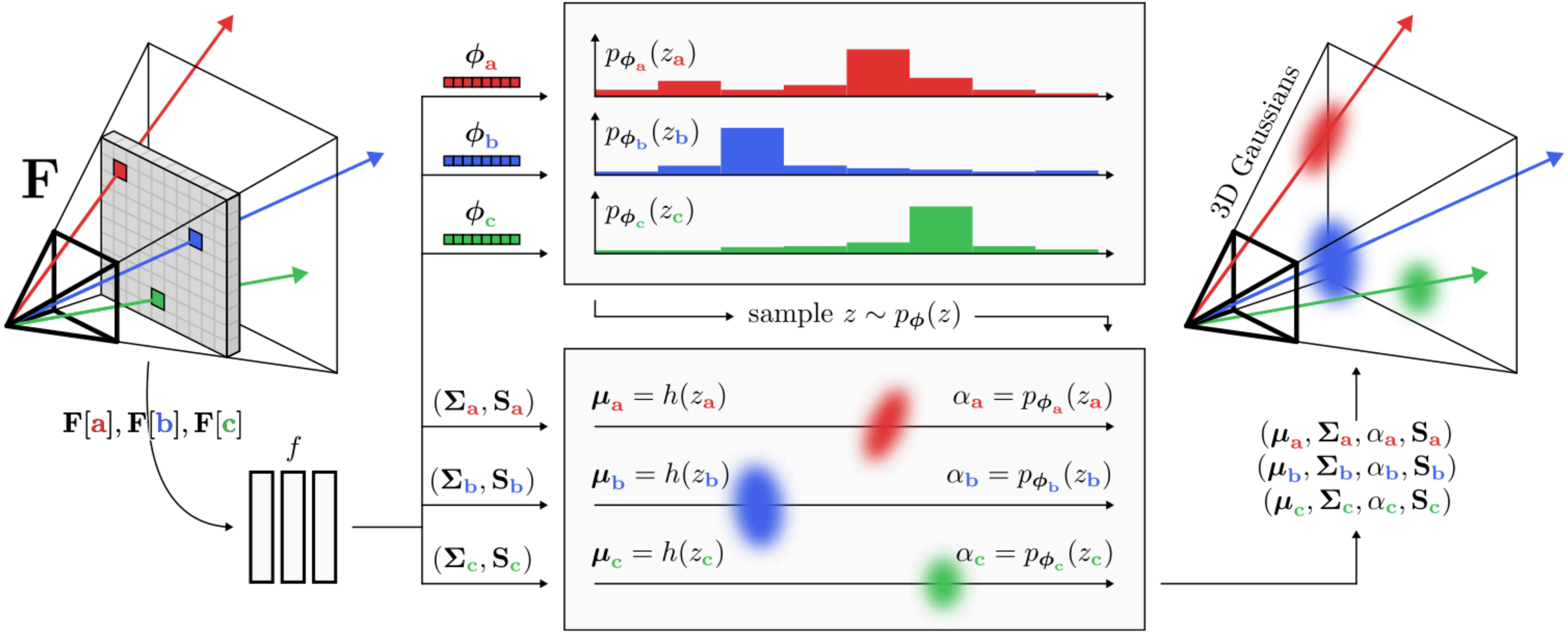


Core Challenge 2: Local Minima



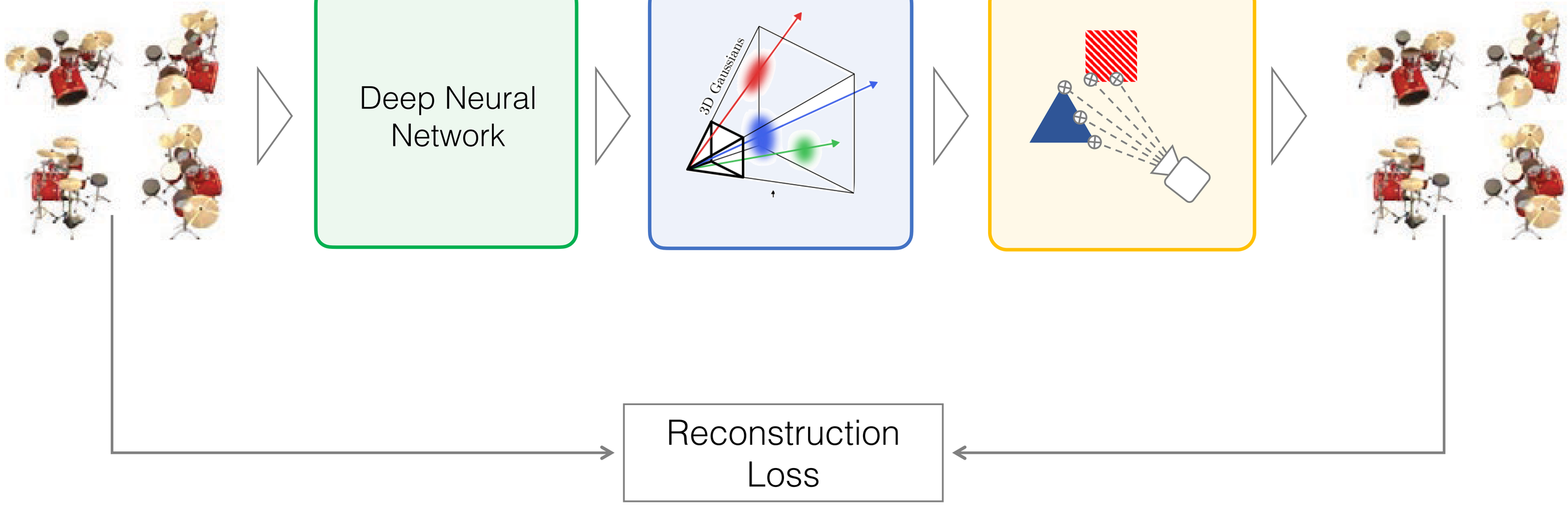


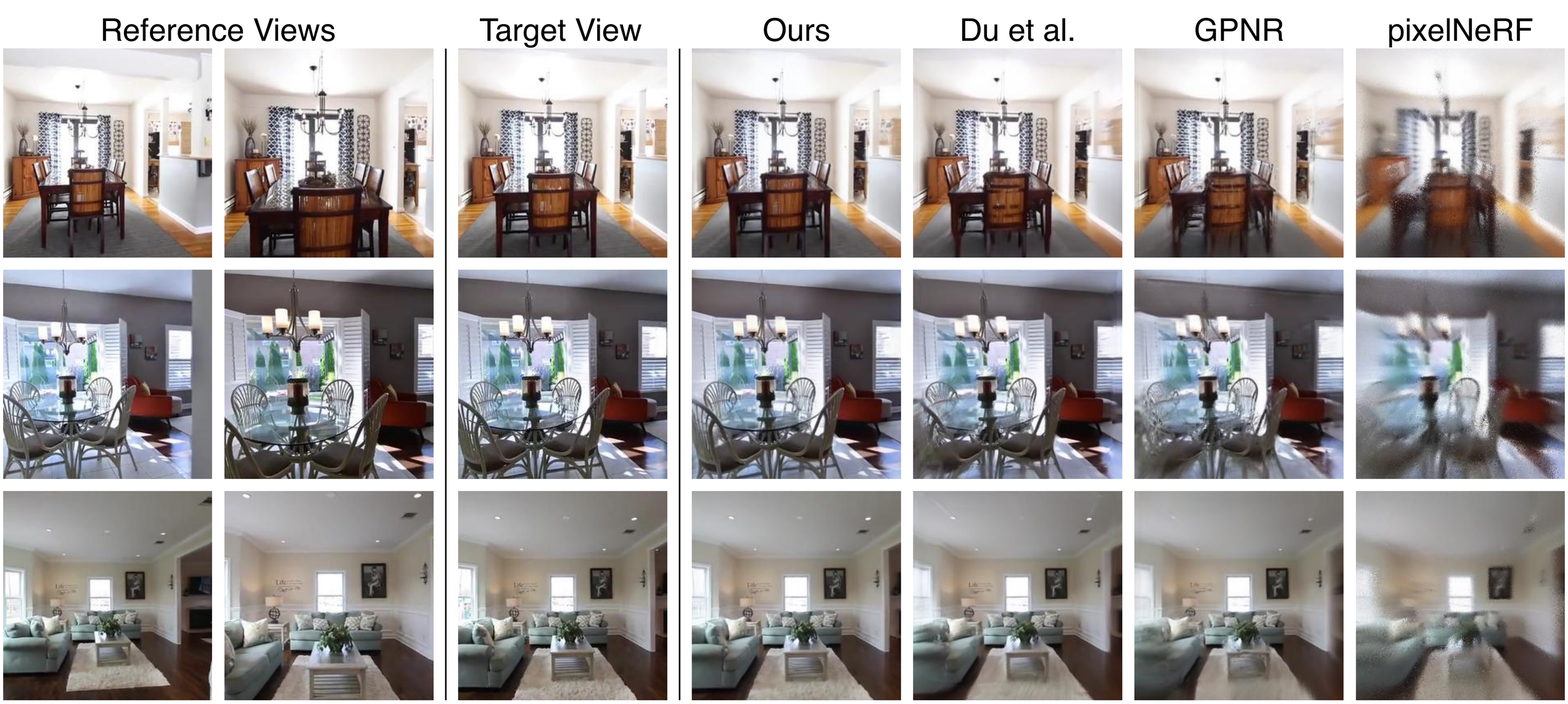
Core Challenge 2: Local Minima

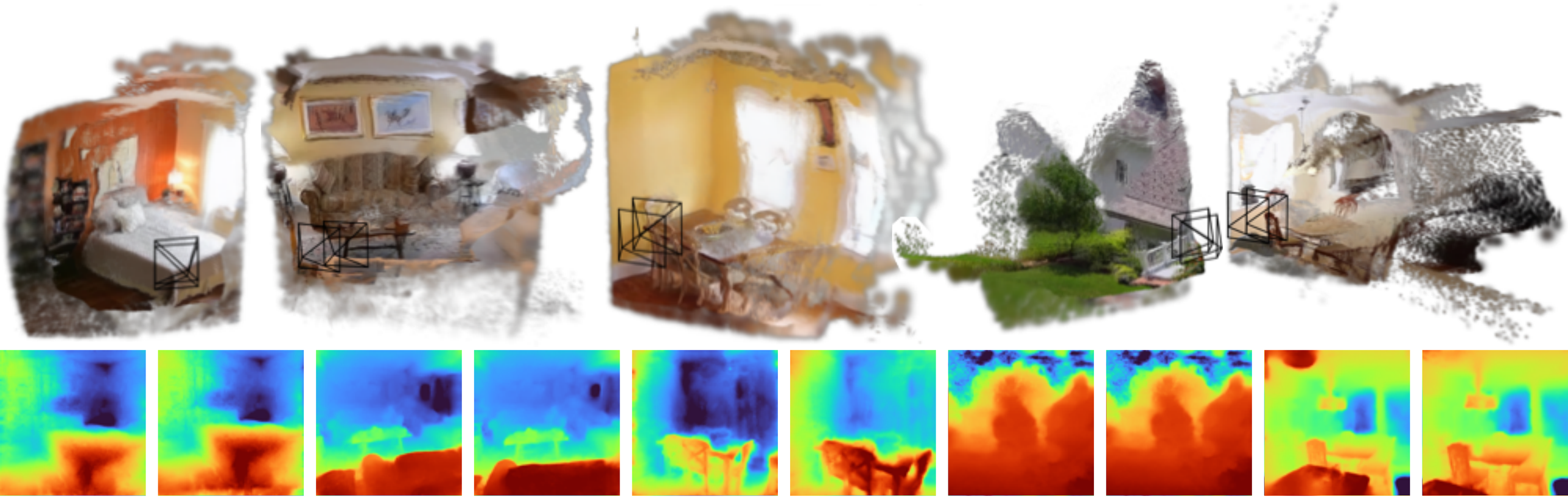


Amortized (=feedforward, generalizable) 3D Reconstruction

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)



arXiv:2312.13150v2 [cs.CV] 16 Apr 2024

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

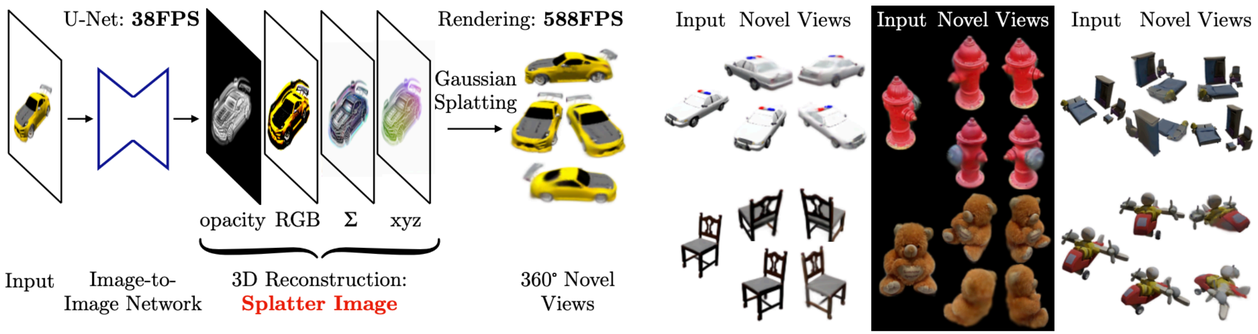


Figure 1. The **Splatter Image** is an ultra-efficient method for single- and few-view 3D reconstruction. It uses an image-to-image neural network to map the input image to another image that holds the parameters of one coloured 3D Gaussian per pixel. Splatter Image achieves excellent 3D reconstruction 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-

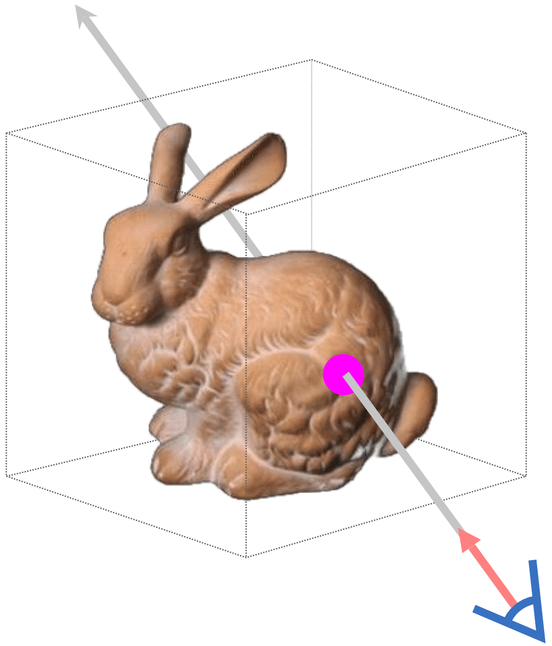
1. Introduction

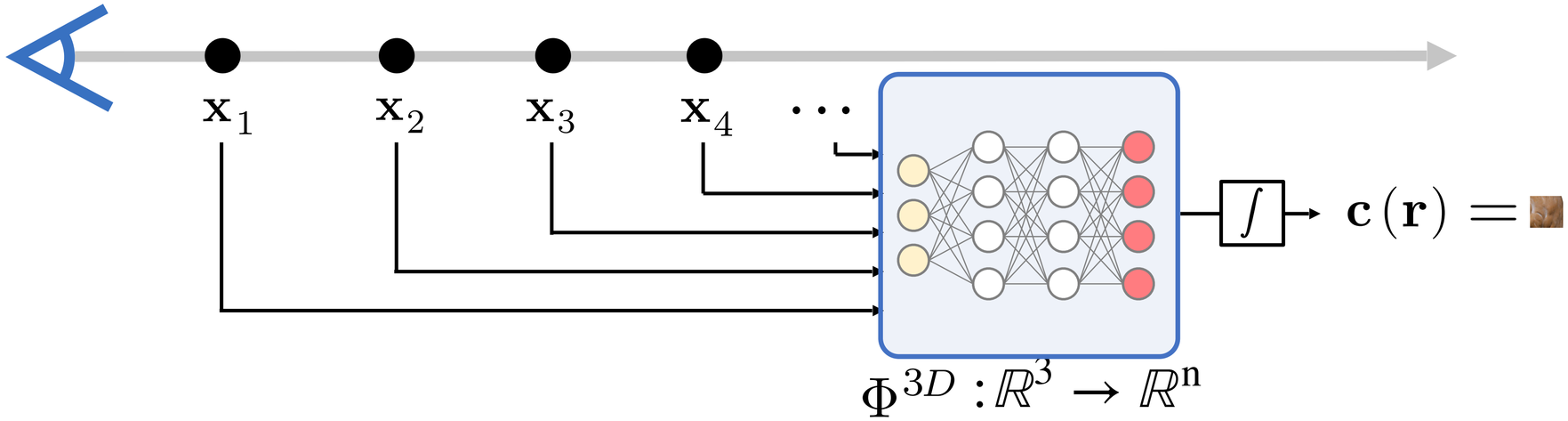
We contribute Splatter Image, a method that achieves ultra-fast single-view reconstruction of the 3D shape and appearance of objects. Splatter Image uses a set of 3D Gaussians as the 3D representation, taking advantage of the rendering quality and speed of Gaussian Splatting [22]. Splatter Image works 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-

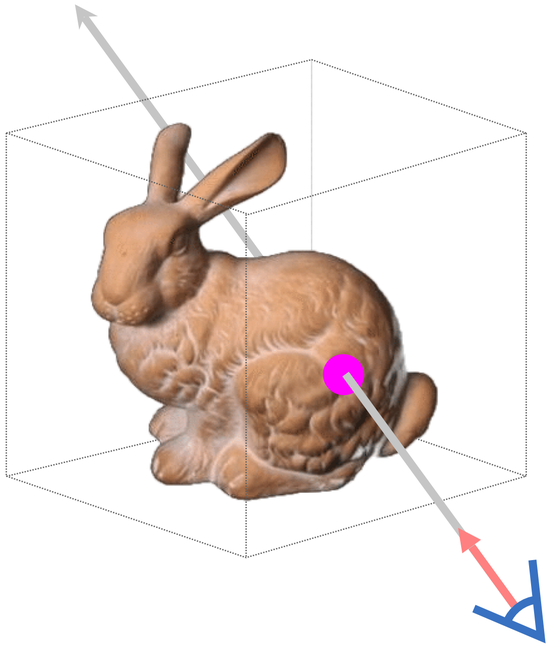
Concurrent work “SplatterImage” suggests that local minima not a huge deal

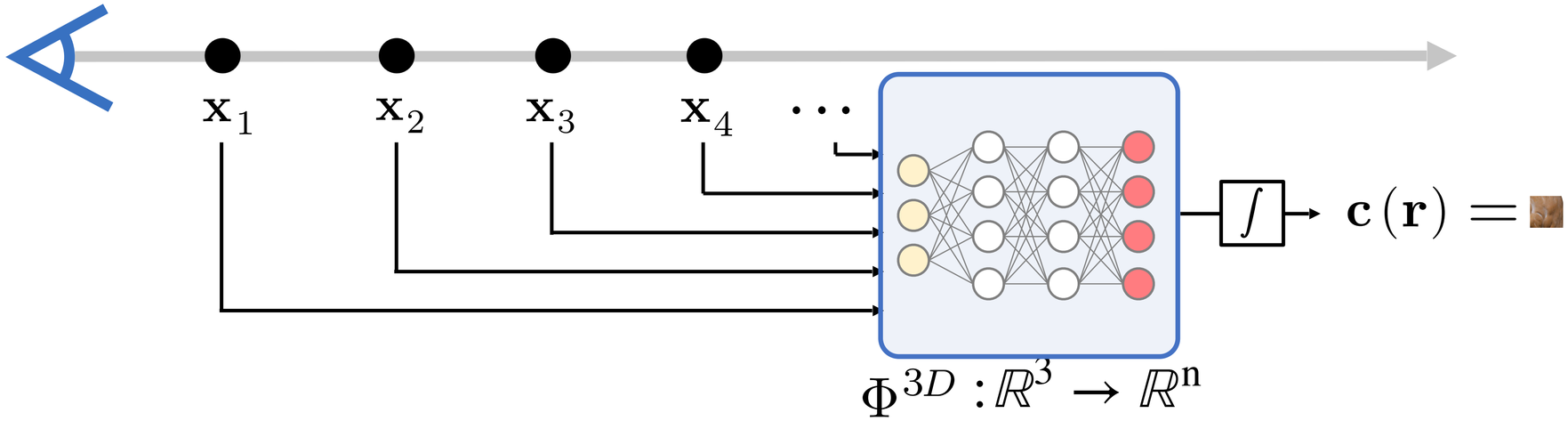
3D-structured Neural Scene Representations





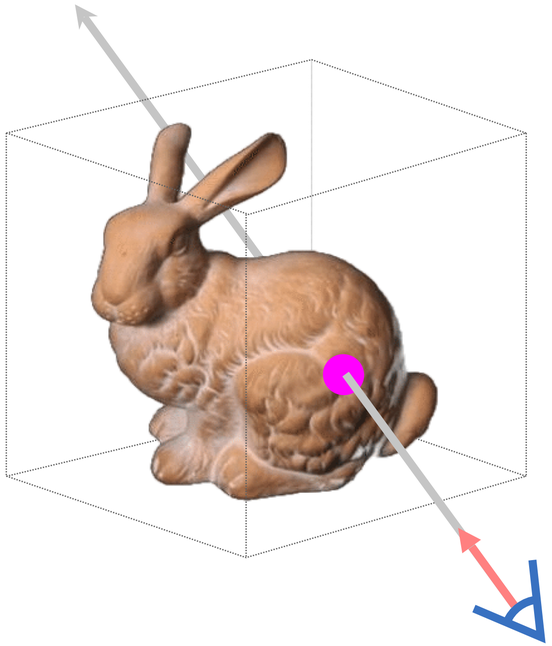
3D-structured Neural Scene Representations

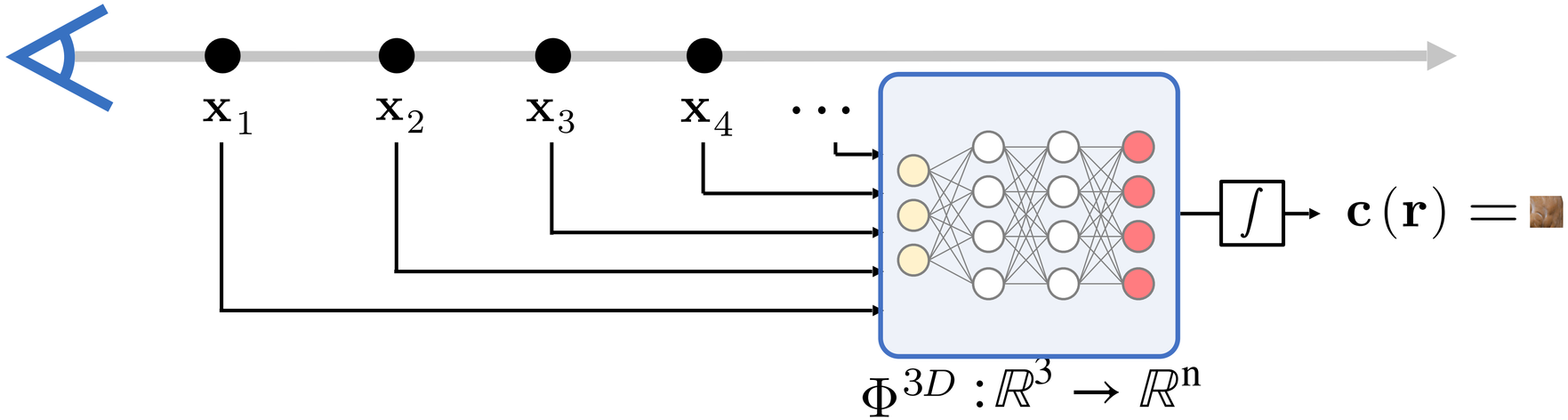




Hundreds of samples per ray.

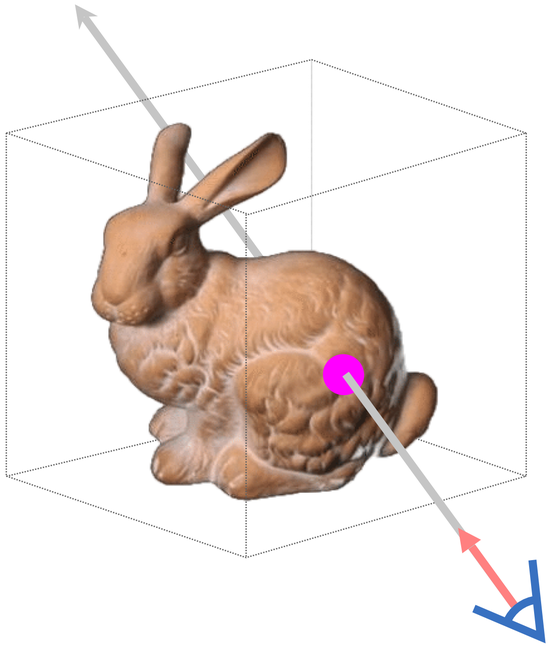
3D-structured Neural Scene Representations





Hundreds of samples per ray.

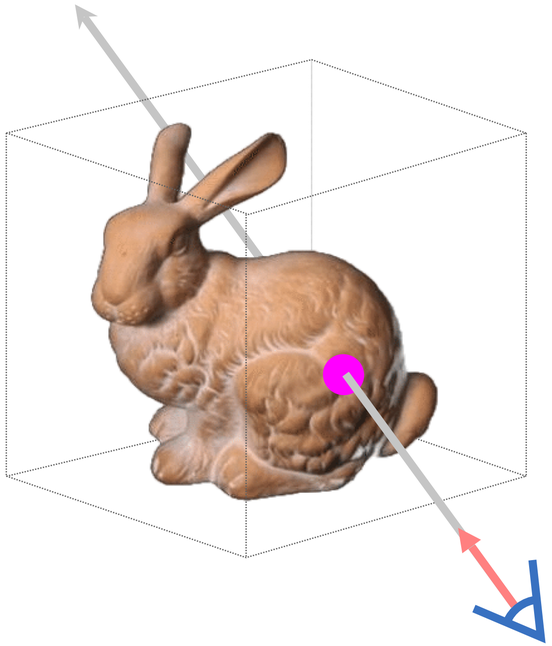
Time- and memory-intensive.



Light Field



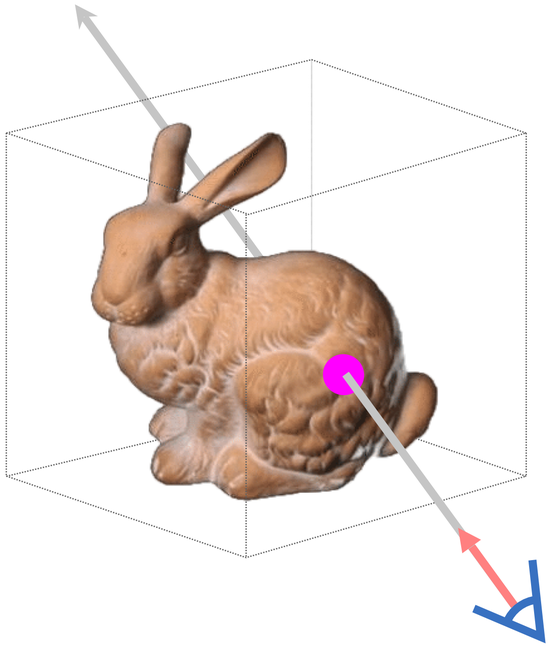
[Adelson et al. 1991, Levoy et al. 1996, Gortler et al. 1996]

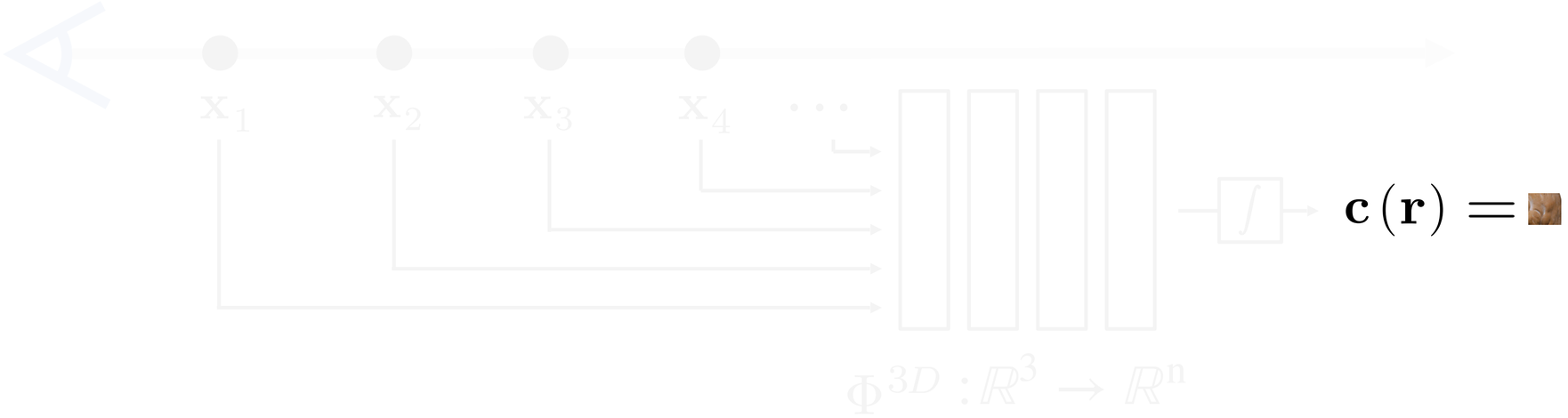


Light Field Networks

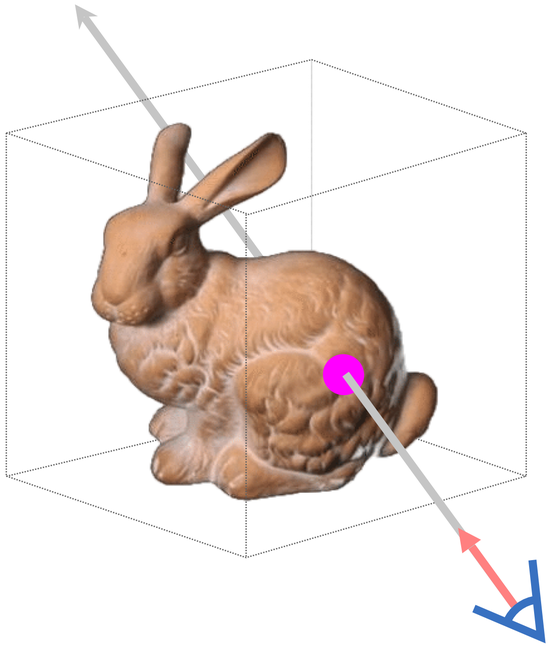
$\mathbf{c}(\mathbf{r}) =$

Light Field Networks





Light Field Networks

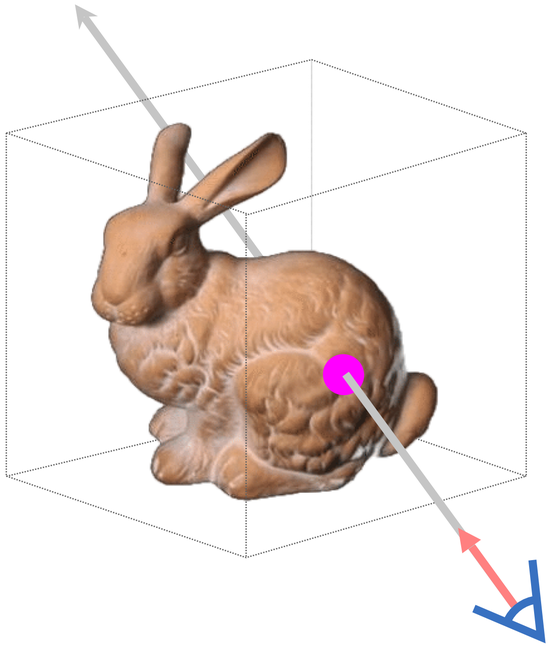


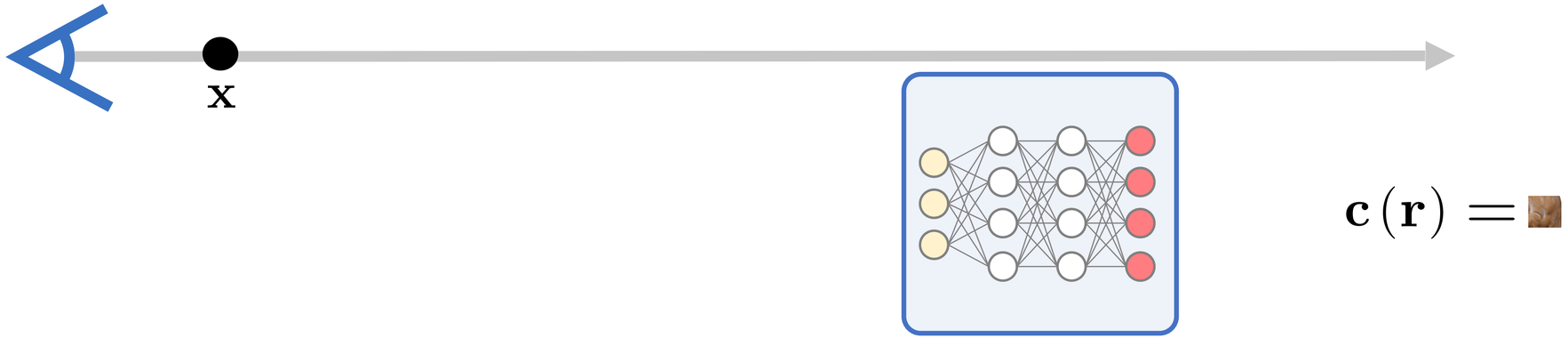


$\mathbf{c}(\mathbf{r}) =$

An Alternative Scene Representation

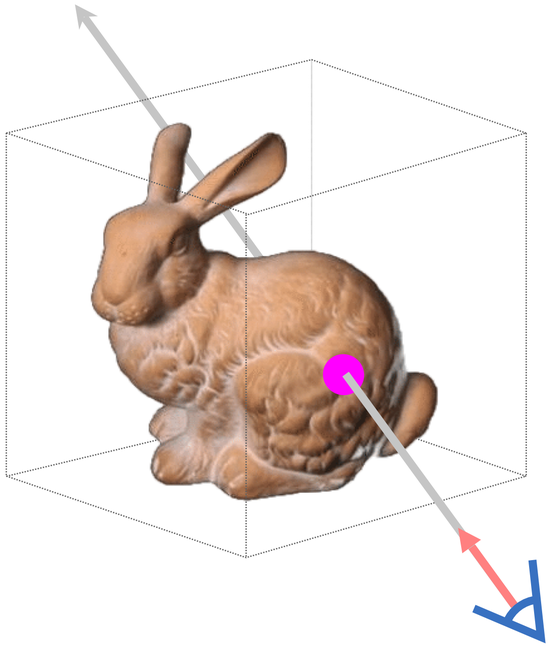
Light Field Networks





An Alternative Scene Representation

Light Field Networks

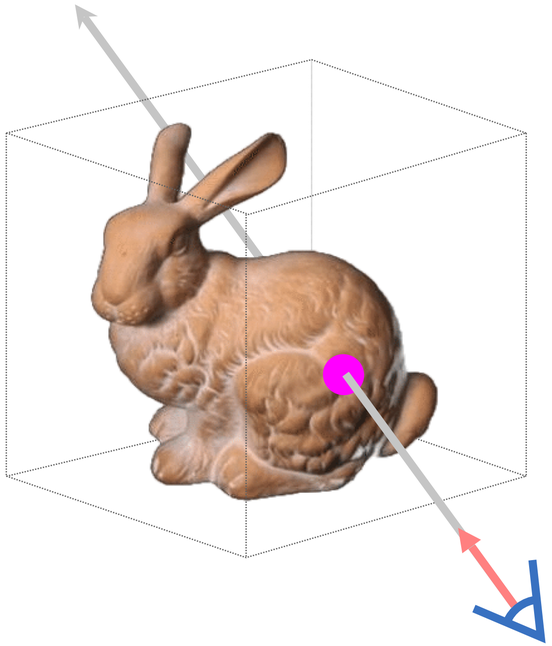


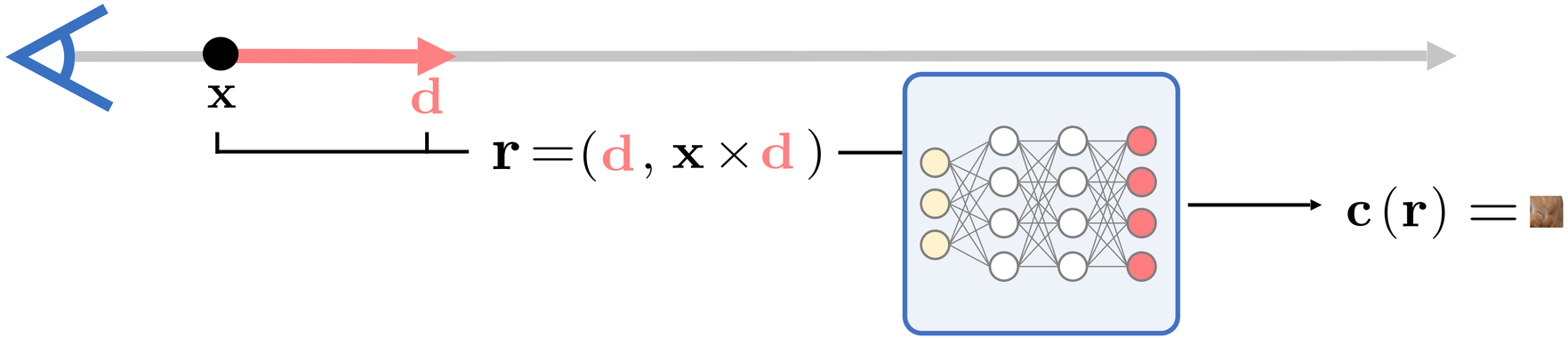


$$\mathbf{c}(\mathbf{r}) =$$

An Alternative Scene Representation

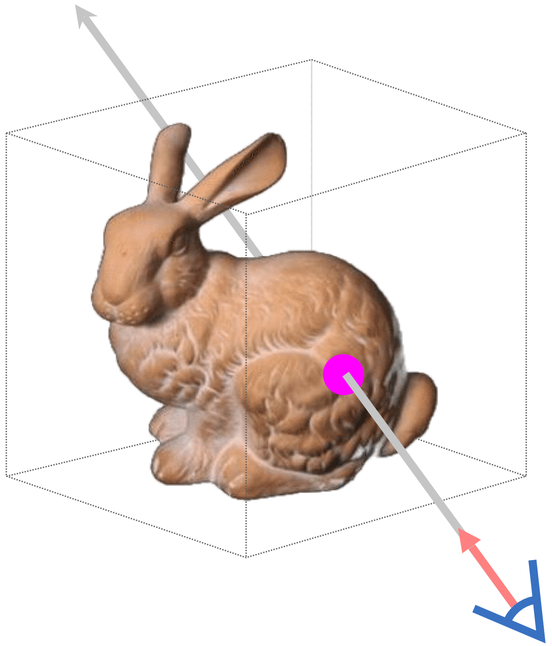
Light Field Networks

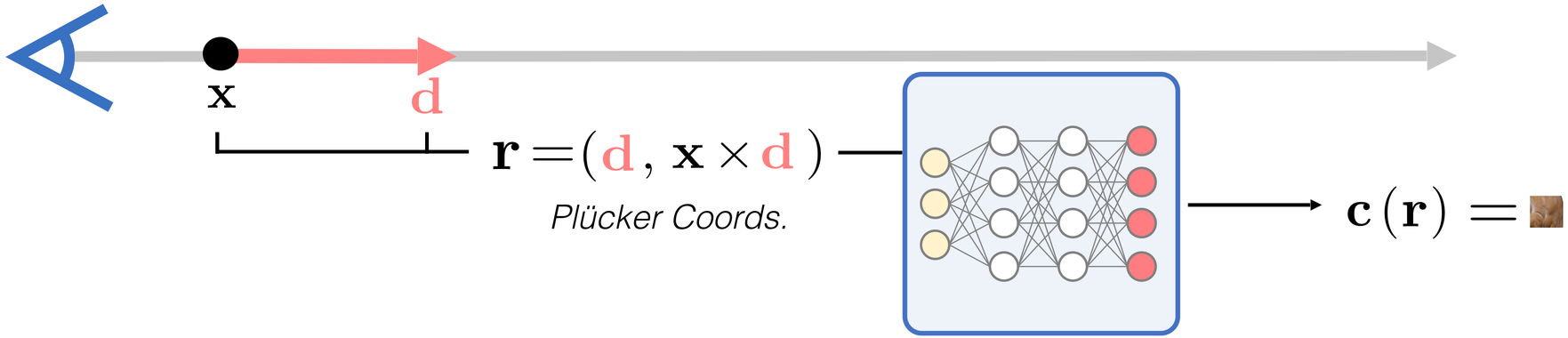




An Alternative Scene Representation

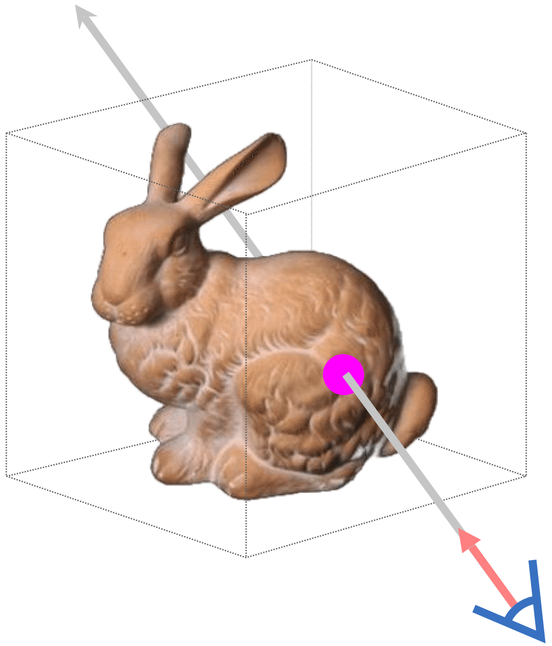
Light Field Networks

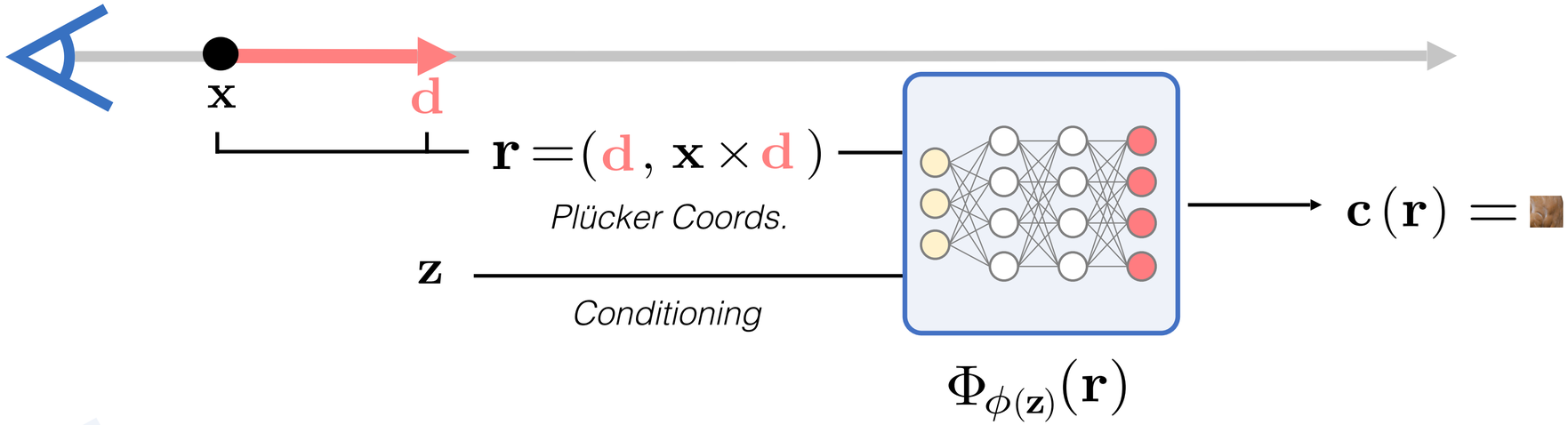




An Alternative Scene Representation

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



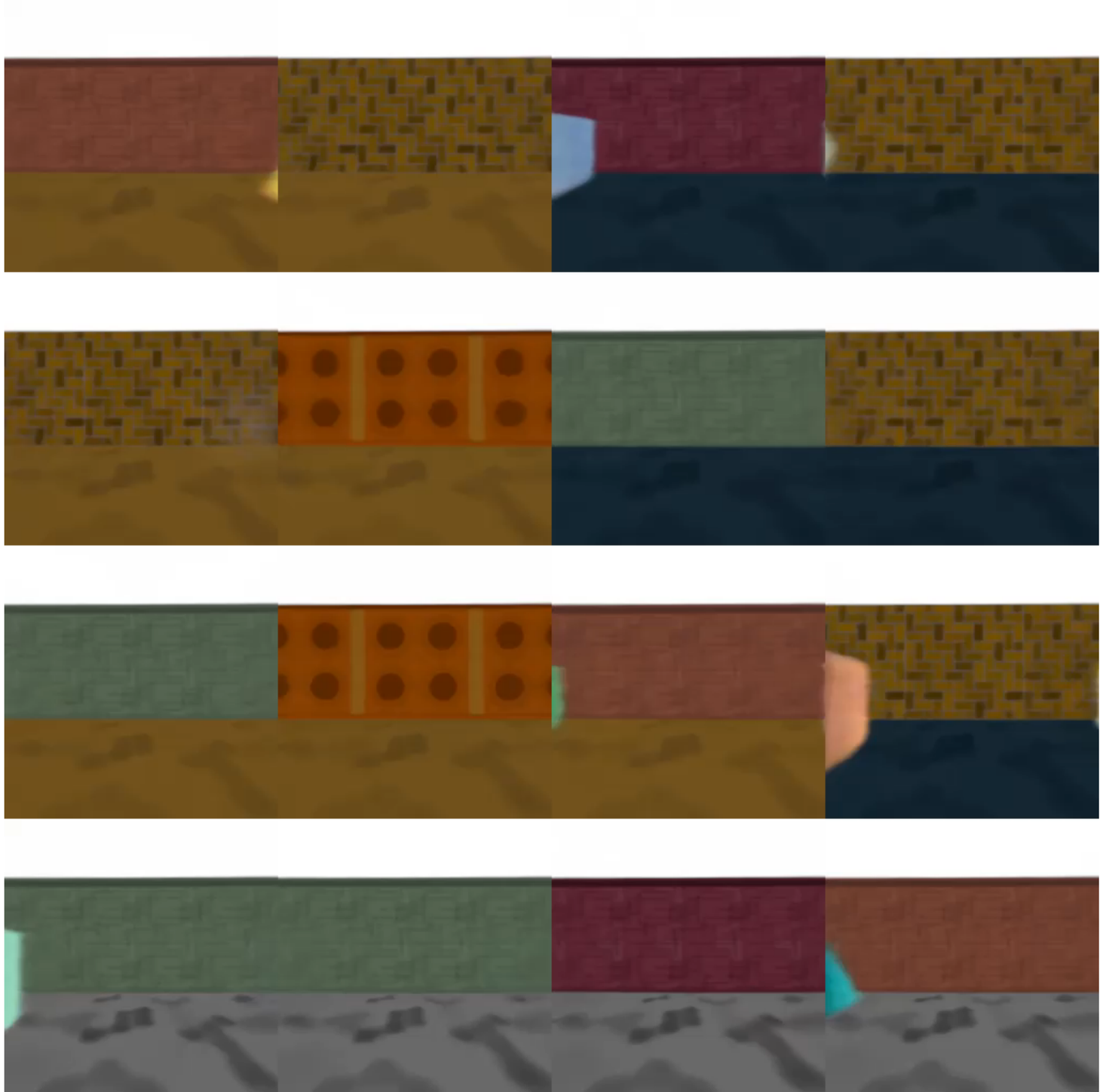
100x speed



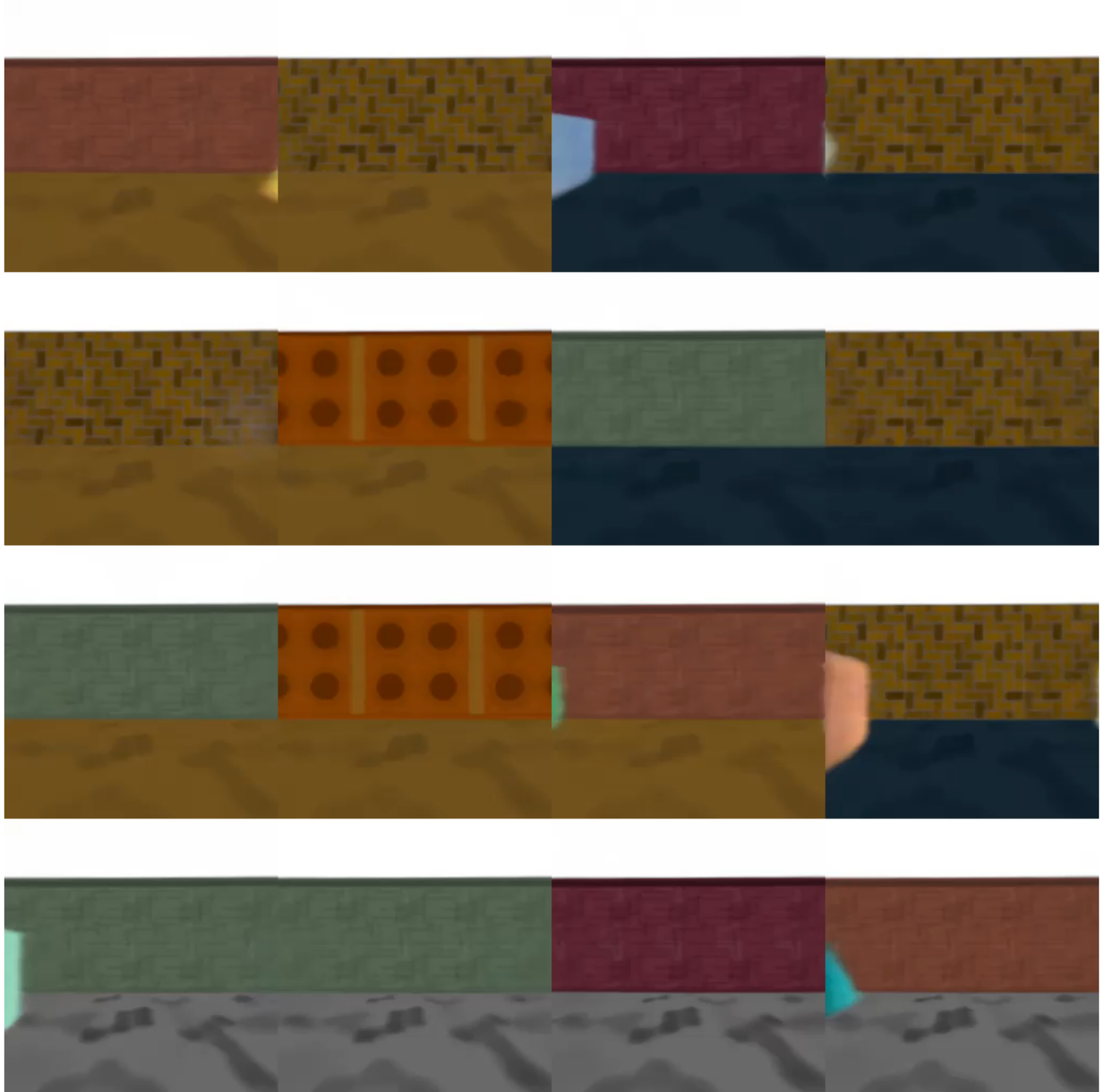


Real-time.
>100x reduction in memory: Can be trained on small GPUs!

Light Field Networks
500 FPS
1 evaluation per ray



Light Field Networks
500 FPS
1 evaluation per ray



arXiv:2410.17242v1 [cs.CV] 22 Oct 2024

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/.

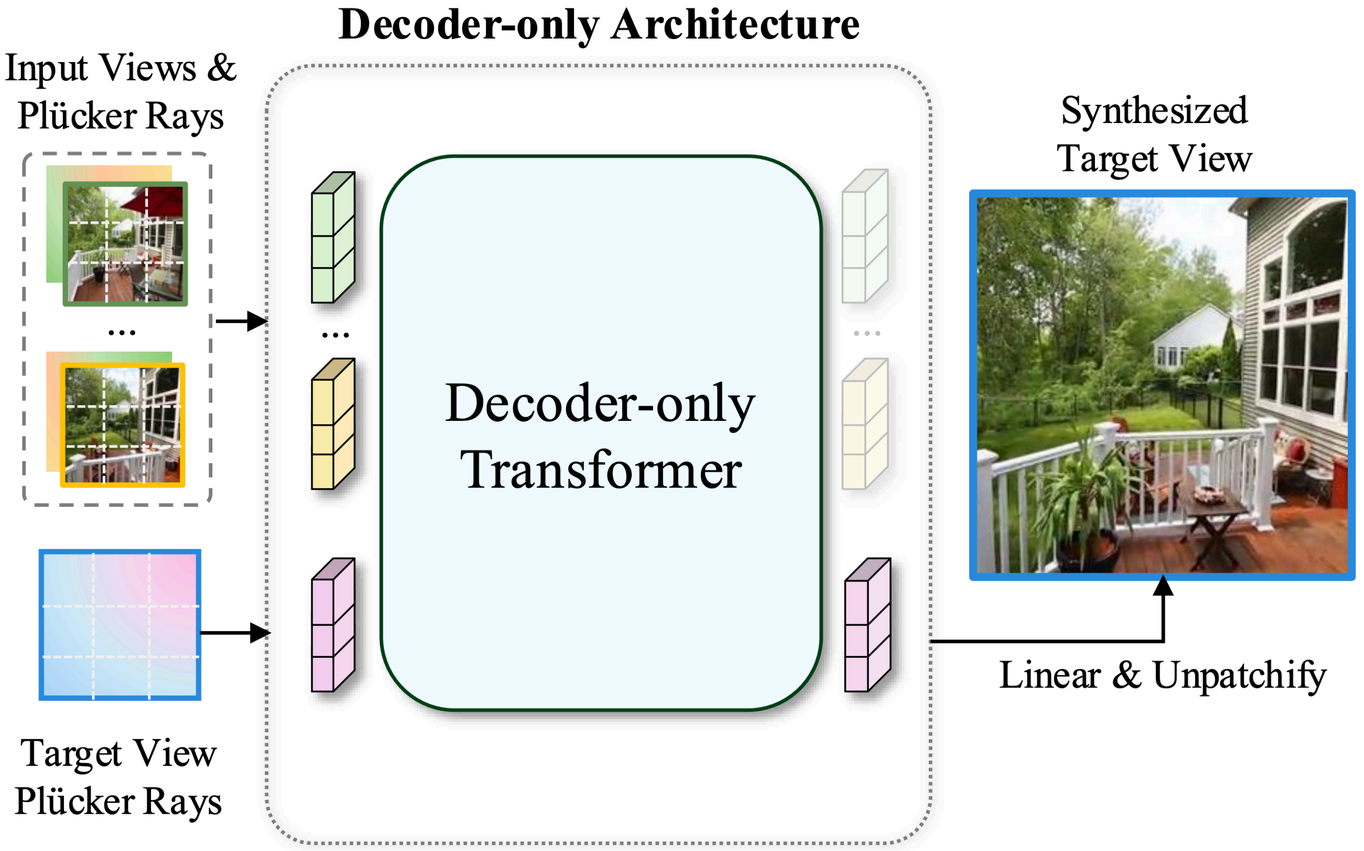
1 Introduction

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



arXiv:2410.17242v1 [cs.CV] 22 Oct 2024

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

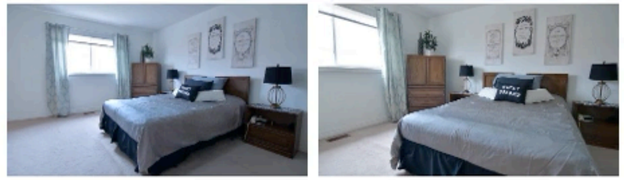
Novel view synthesis is a long-standing challenge in vision and graphics. For decades, the community has generally relied on various 3D inductive biases, incorporating 3D priors and handcrafted structures to simplify the task and improve synthesis quality. Recently, NeRF, 3D Gaussian Splatting (3DGS), 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

Only inductive bias & supervision:
Plucker ray coordinates & camera poses.
No representation or forward rendering model

arXiv:2410.17242v1 [cs.CV] 22 Oct 2024

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

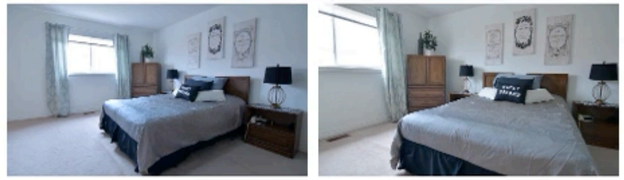
Novel view synthesis is a long-standing challenge in vision and graphics. For decades, the community has generally relied on various 3D inductive biases, incorporating 3D priors and handcrafted structures to simplify the task and improve synthesis quality. Recently, NeRF, 3D Gaussian Splatting (3DGS), 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

Only inductive bias & supervision:
Plucker ray coordinates & camera poses.
No representation or forward rendering model

3D Inverse Graphics Taxonomy

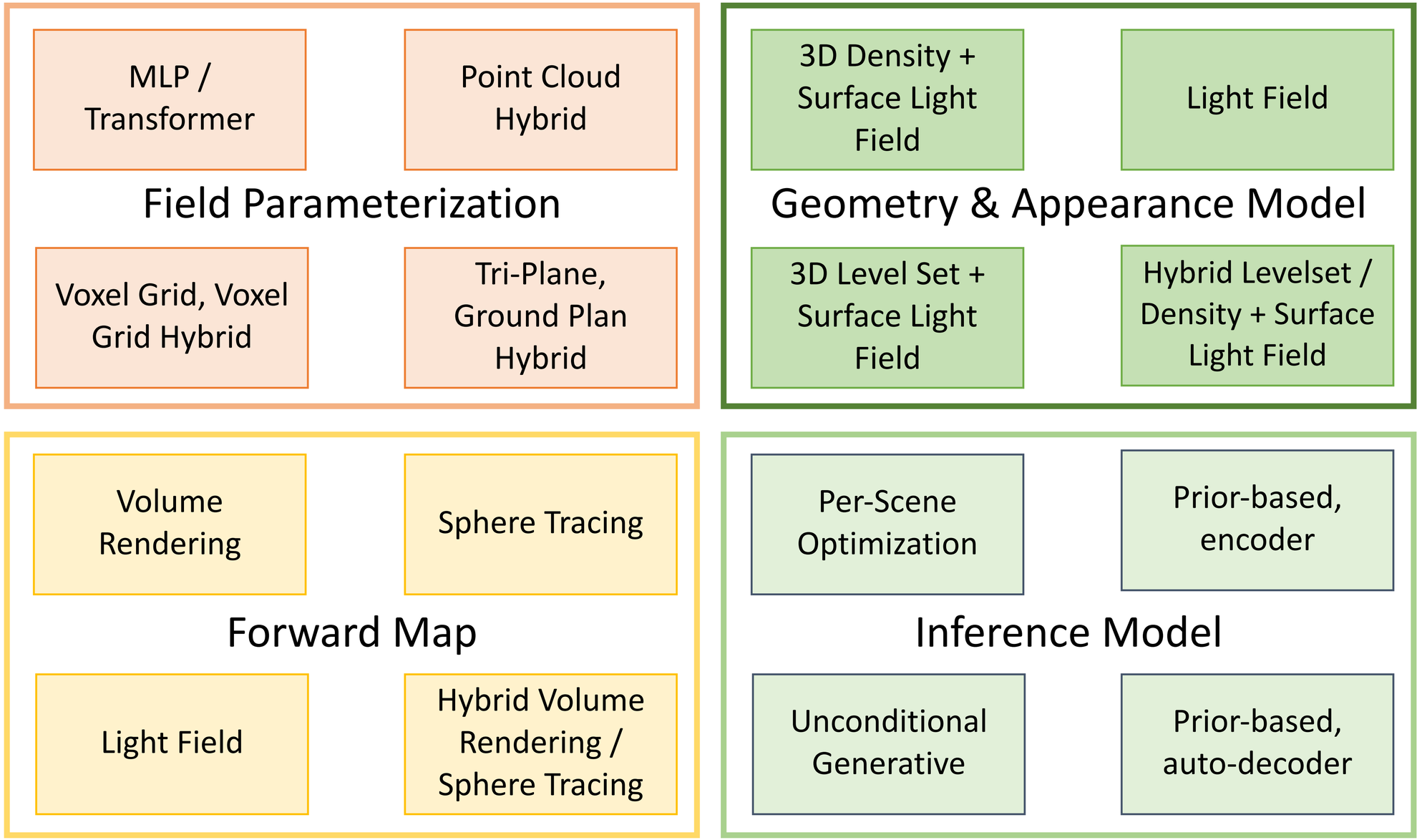
Field Parameterization

Geometry & Appearance Model

Forward Map

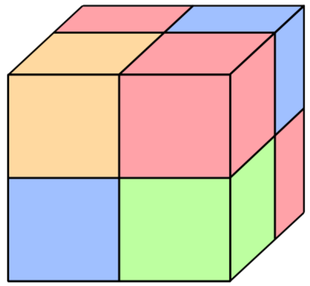
Inference Method

3D Inverse Graphics Taxonomy



DeepVoxels (CVPR 2018)

Field Parameterization



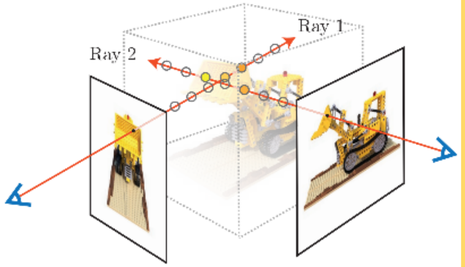
Geometry & Appearance Model

$$\Phi(\mathbf{x}) = \mathbf{f}$$

$$\mathbf{f} \in \mathbb{R}^n$$

Feature-based Representation

Forward Map



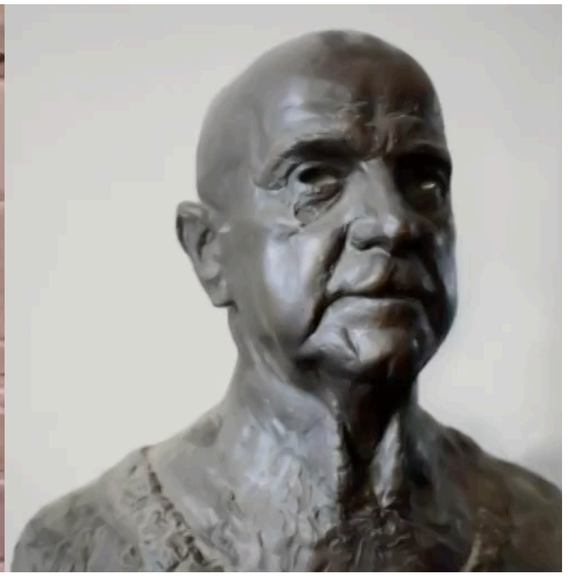
Volumetric Sampling, Softmax Attention

Inference Model



Per-Scene Optimization

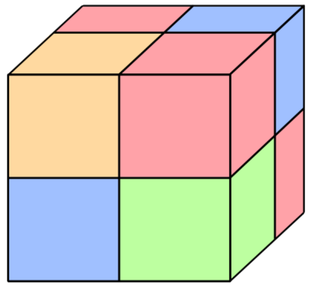




- Reasonable quality overfitting
- Fast rendering
- Noisy geometry
- Requires Many Images

DeepVoxels (CVPR 2018)

Field Parameterization



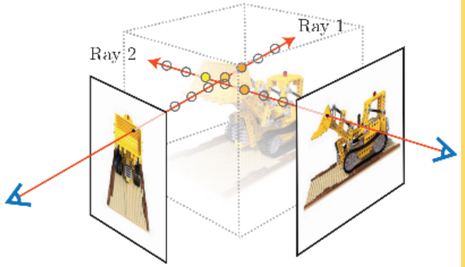
Geometry & Appearance Model

$$\Phi(\mathbf{x}) = \mathbf{f}$$

$$\mathbf{f} \in \mathbb{R}^n$$

Feature-based Representation

Forward Map



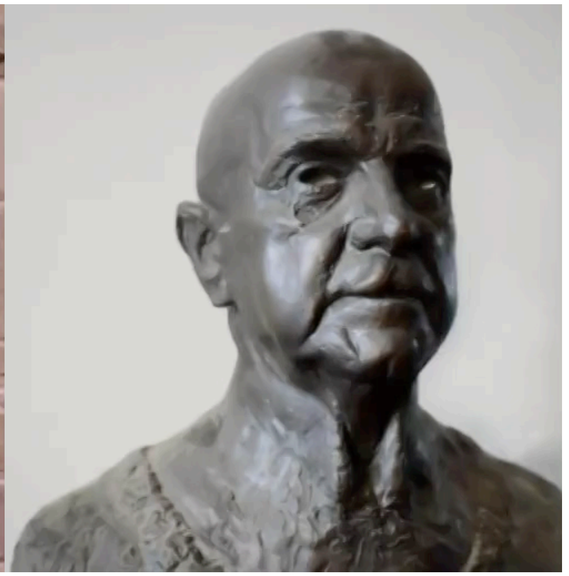
Volumetric Sampling, Softmax Attention

Inference Model



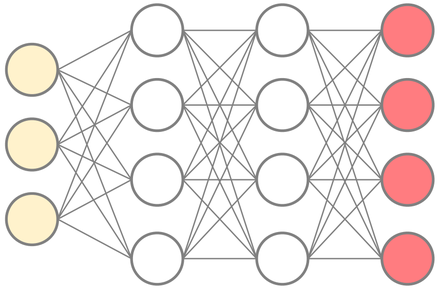
Per-Scene Optimization



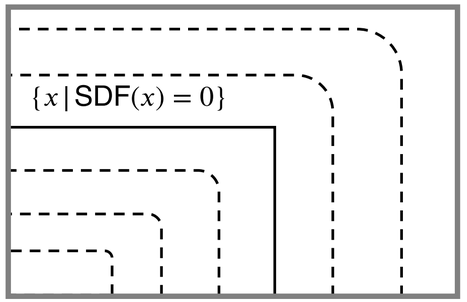


- Reasonable quality overfitting
- Fast rendering
- Noisy geometry
- Requires Many Images

Scene Representation Networks (NeurIPS 2019)



Field Parameterization



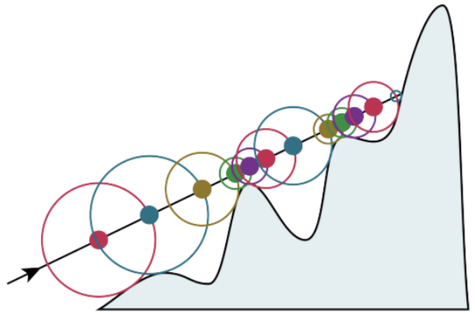
SDF

$$x \mapsto C$$

View-independent Radiance

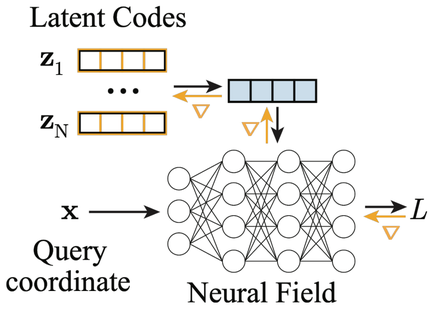
Geometry & Appearance Model

LSTM-based sphere tracing



Forward Map

Auto-Decoding or Per-Scene Optimization

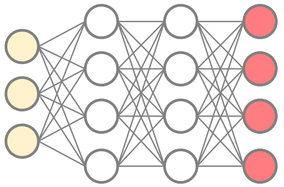




Inference Model

Scene Representation Networks (NeurIPS 2019)

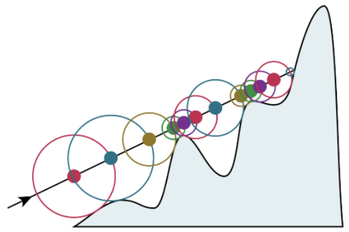
Field Parameterization



Geometry & Appearance Model

SDF
+
View-independent Radiance

Forward Map



Volume Rendering

Inference Model

Auto-Decoding
OR
Per-Scene Optimization

Overfitting



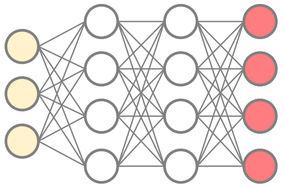
Single Image



- When overfitting: Not photorealistic
- Non-compositional scenes: Single-image reconstruction
- Well-defined geometry
- Low storage cost (weights)

Scene Representation Networks (NeurIPS 2019)

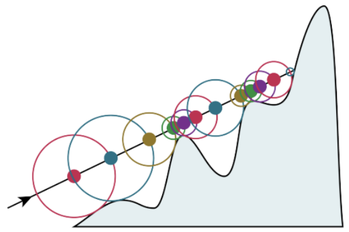
Field Parameterization



Geometry & Appearance Model

SDF
+
View-independent Radiance

Forward Map



Volume Rendering

Inference Model

Auto-Decoding
OR
Per-Scene Optimization

Overfitting

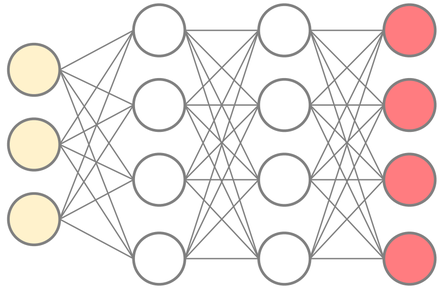


Single Image



- When overfitting: Not photorealistic
- Non-compositional scenes: Single-image reconstruction
- Well-defined geometry
- Low storage cost (weights)

NeRF (Mildenhall et al. ECCV 2020)



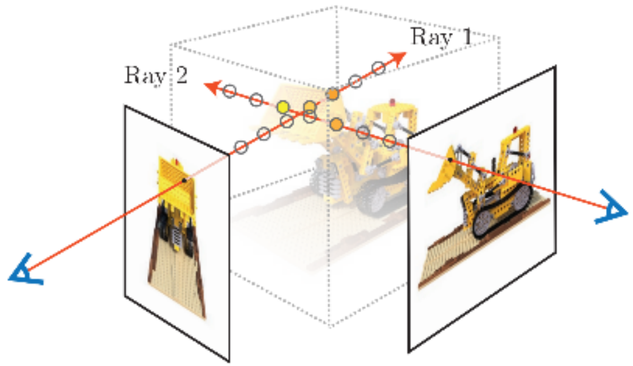
Field Parameterization

$$(x, d) \mapsto (\sigma, C)$$

View-Dependent Density + Radiance

Geometry & Appearance Model

Volume Rendering



Forward Map

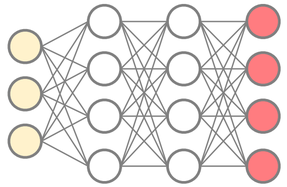
Per-Scene Optimization



Inference Model

NeRF (Mildenhall et al. ECCV 2020)

Field Parameterization

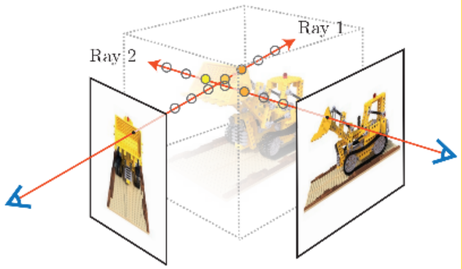


Geometry & Appearance Model

$$(x, d) \mapsto (\sigma, C)$$

View-Dependent Density + Radiance

Forward Map



Volume Rendering

Inference Model



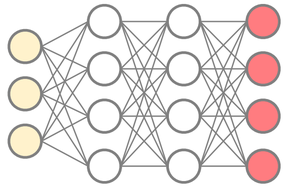
Per-Scene Optimization



- When overfitting: Almost photorealistic!
- Slow Rendering
- High memory cost in forward pass
- Low Storage Cost (weights)
- Noisy or Cloudy Geometry
- Requires Many Images

NeRF (Mildenhall et al. ECCV 2020)

Field Parameterization

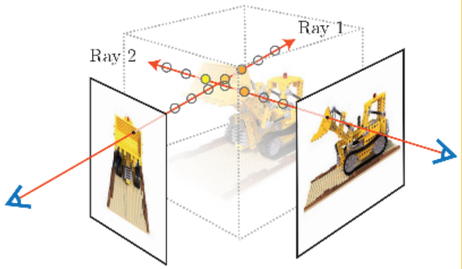


Geometry & Appearance Model

$$(x, d) \mapsto (\sigma, C)$$

View-Dependent Density + Radiance

Forward Map



Volume Rendering

Inference Model



Per-Scene Optimization

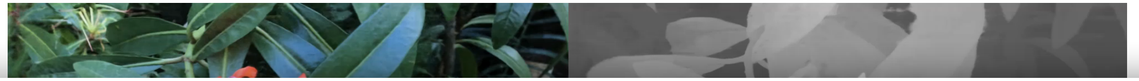


- When overfitting: Almost photorealistic!
- Slow Rendering
- High memory cost in forward pass
- Low Storage Cost (weights)
- Noisy or Cloudy Geometry
- Requires Many Images

NeRF (Mildenhall et al. ECCV 2020)

Field

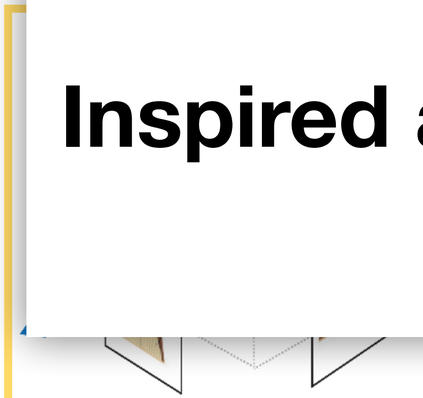
Geometry &



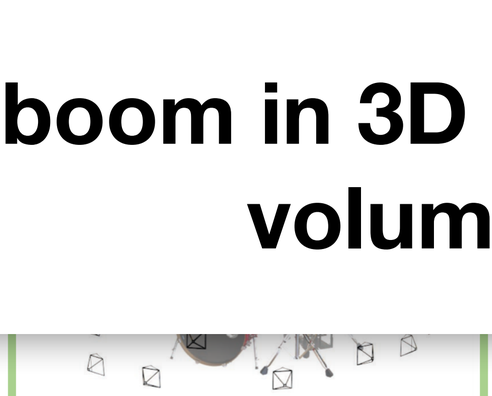
Great out-of-the-box performance
Great looking results for single-scene reconstruction
Ben donated slides ;)

Inspired a boom in 3D reconstruction via differentiable volumetric rendering!

ic!



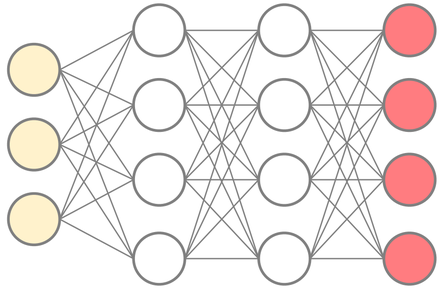
Volume Rendering



Per-Scene Optimization

- Noisy or Cloudy Geometry
- Requires Many Images

NeRF (Mildenhall et al. ECCV 2020)



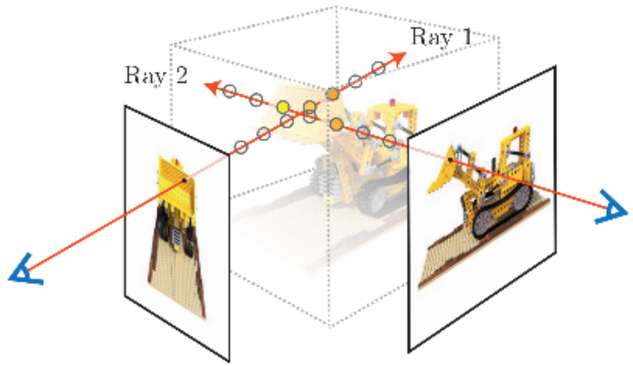
Field Parameterization

$$(x, d) \mapsto (\sigma, C)$$

View-Dependent Density + Radiance

Geometry & Appearance Model

Volume Rendering



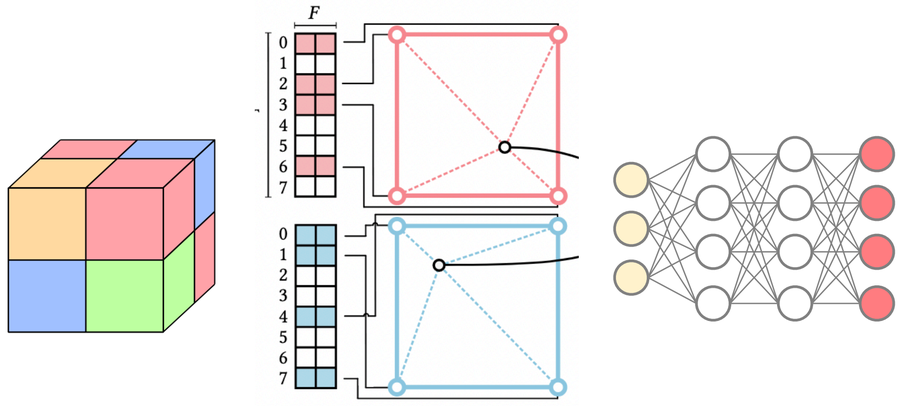
Forward Map

Per-Scene Optimization



Inference Model

Faster Radiance Fields w/ Hybrid Data Structures



Field Parameterization

$$(x, d) \mapsto (\sigma, C)$$

View-Dependent Density + Radiance

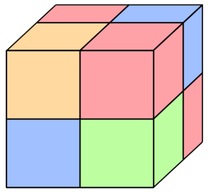
Geometry & Appearance Model

Volume Rendering

Forward Map

Per-Scene Optimization

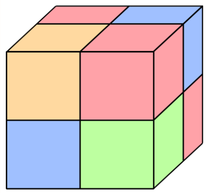
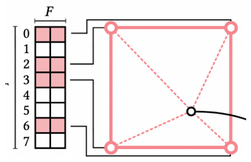
Inference Model

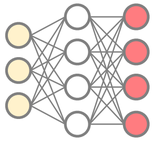


Voxel Grid

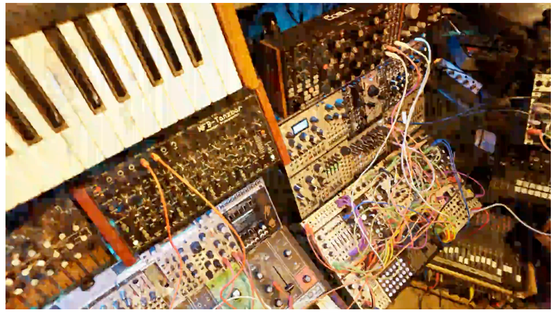


Plenoxels: Radiance Fields …
[Yu et al. 2022]
Direct Voxel Grid Optimization
[Sun et al. 2021]

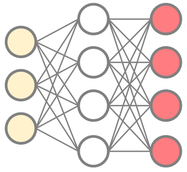





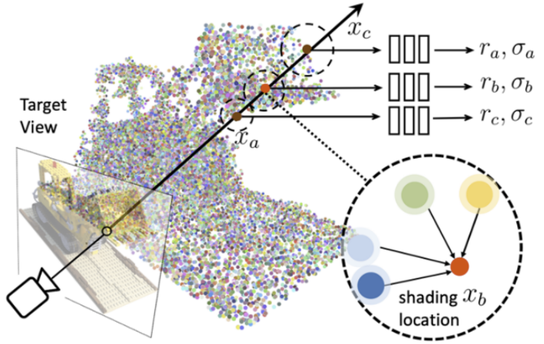
Voxel Grid + Hashmap + MLP



InstantNGP: Instant Neural …
[Müller et al. 2022]

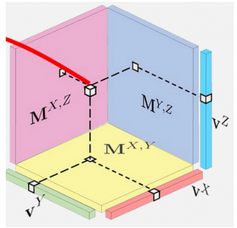



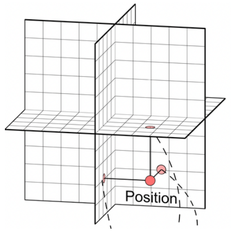
Point Cloud + MLP

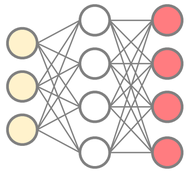


(b) Point-NeRF Representation with Volum

PointNeRF: Point-based Neural …
[Xu et al. 2022]



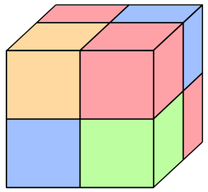




Tensor Factorization / Triplane + MLP



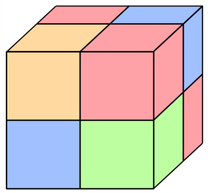
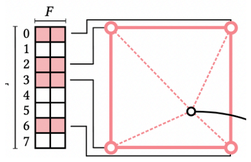
Efficient Geometry-aware 3D…
[Chan et al. 2022]
TensorRF: Tensor Radiance Fields
[Chen & Xu et al. 2022]

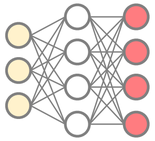


Voxel Grid

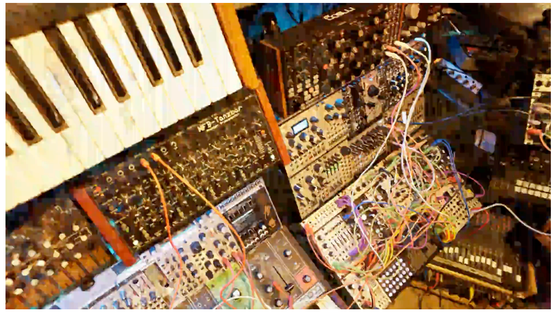


Plenoxels: Radiance Fields …
[Yu et al. 2022]
Direct Voxel Grid Optimization
[Sun et al. 2021]

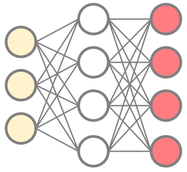





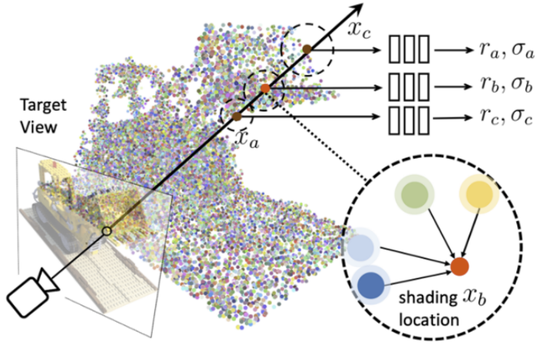
Voxel Grid + Hashmap + MLP



InstantNGP: Instant Neural …
[Müller et al. 2022]

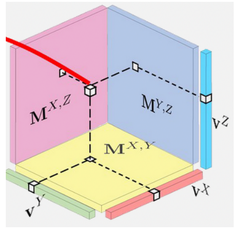



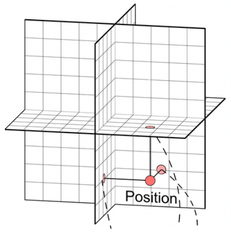
Point Cloud + MLP

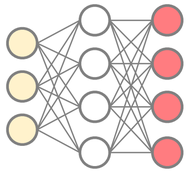


(b) Point-NeRF Representation with Volum

PointNeRF: Point-based Neural …
[Xu et al. 2022]



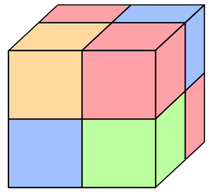




Tensor Factorization / Triplane + MLP

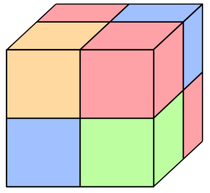


Efficient Geometry-aware 3D…
[Chan et al. 2022]
TensorRF: Tensor Radiance Fields
[Chen & Xu et al. 2022]





Plenoxels: Radiance Fields …
[Yu et al. 2022]
Direct Voxel Grid Optimization
[Sun et al. 2021]

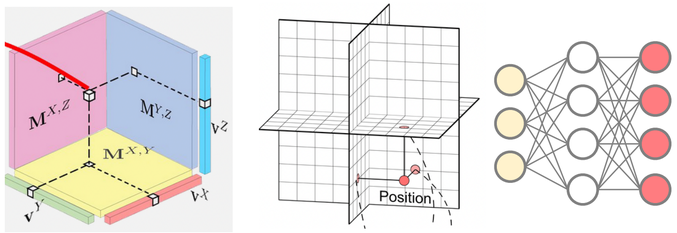


Voxel Grid

- Faster fitting
- Faster rendering
- Lower forward pass memory footprint
- May achieve higher quality than vanilla NeRF
- Voxels: Memory intense



Point Cloud + MLP

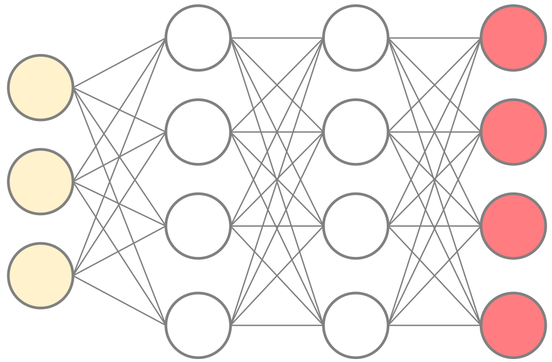


Tensor Factorization / Triplane + MLP

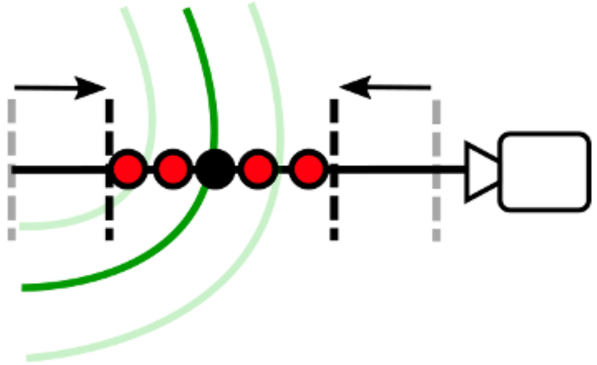


Efficient Geometry-aware 3D…
[Chan et al. 2022]
TensorRF: Tensor Radiance Fields
[Chen & Xu et al. 2022]

Hybrid Volumetric / Level Set Representations

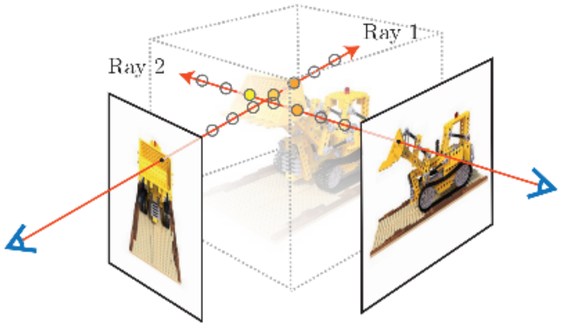


Field Parameterization



Geometry & Appearance Model

Volume Rendering



Forward Map

Per-Scene Optimization

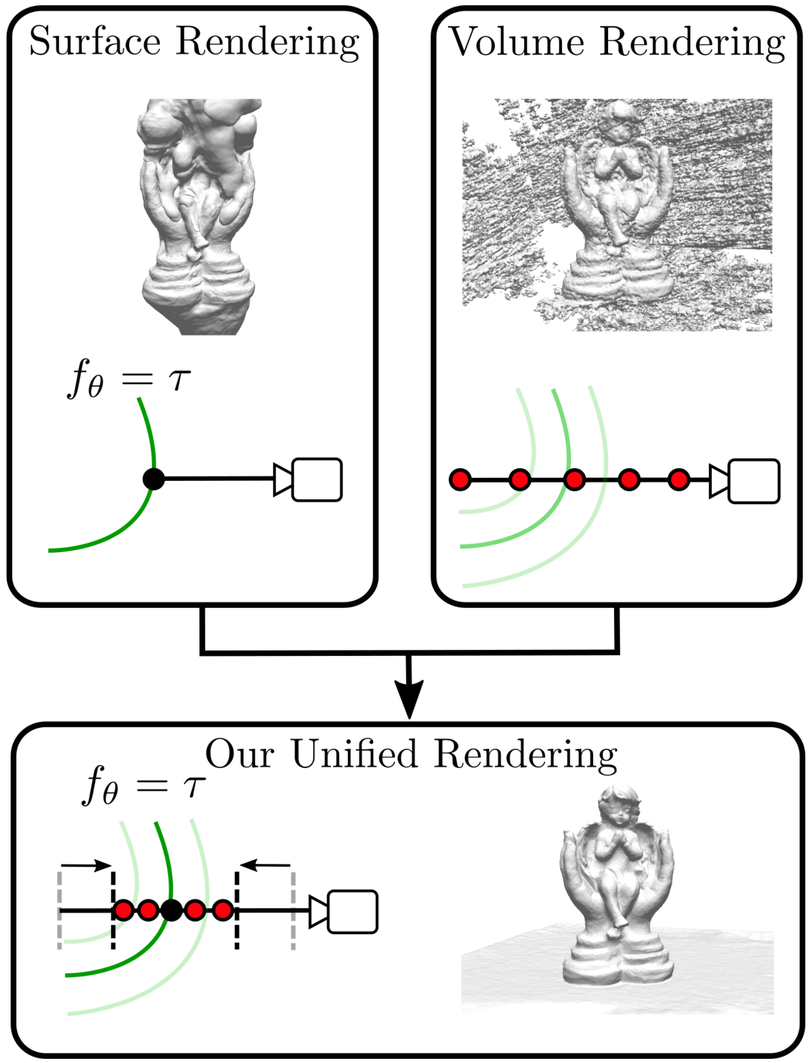


Inference Model

Hybrid Volumetric / Level Set Representations

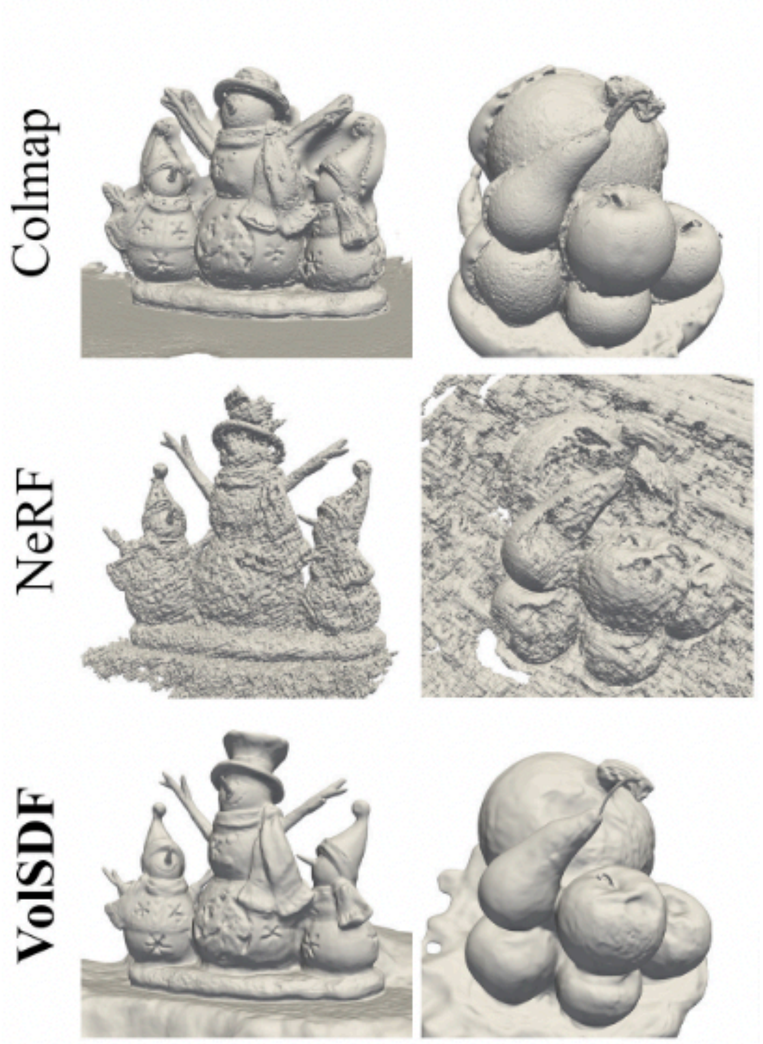
UNISURF

[Oechsle et al., ICCV 2021]



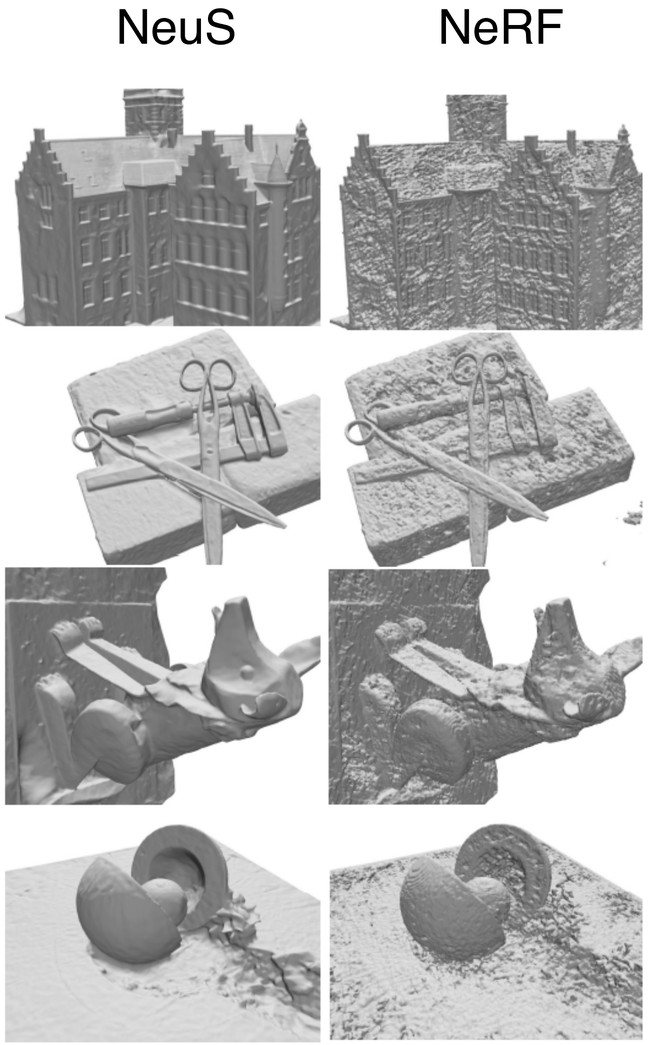
VolSDF

[Yariv et al., NeurIPS 2021]



NeuS

[Wang et al., NeurIPS 2021]

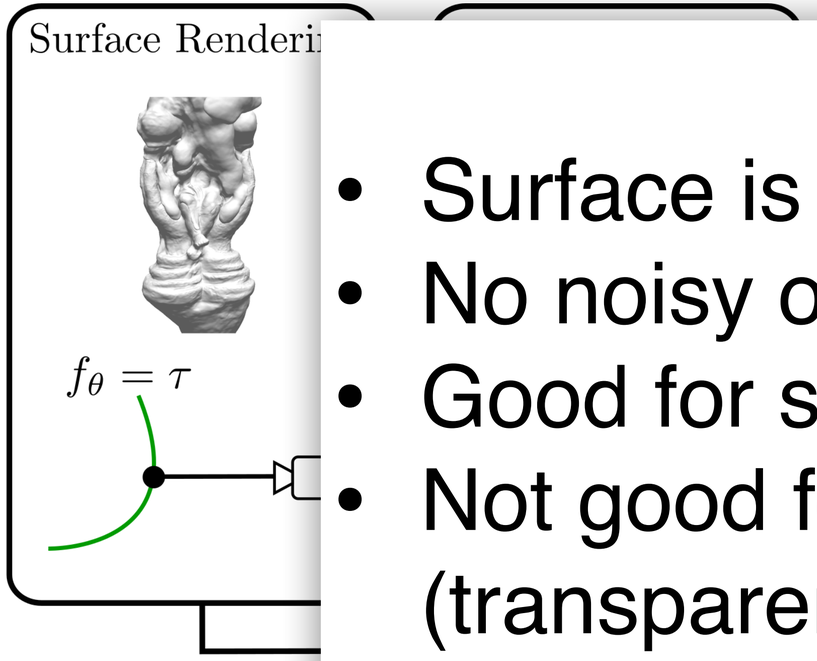


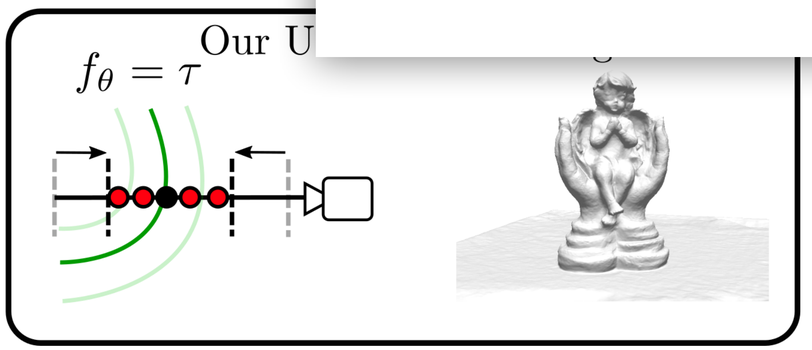
Hybrid Volumetric / Level Set Representations

UNISURF
[Oechsle et al., ICCV 2021]

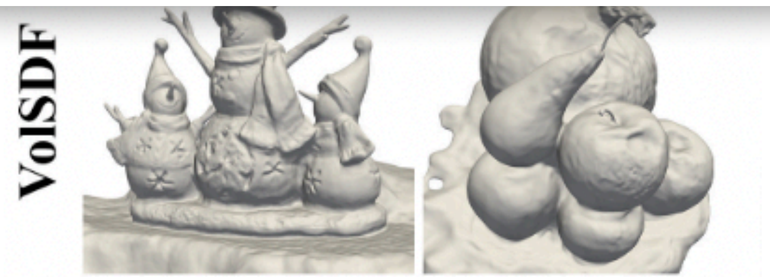
VolSDF
[Yariv et al., NeurIPS 2021]

NeuS
[Wang et al., NeurIPS 2021]

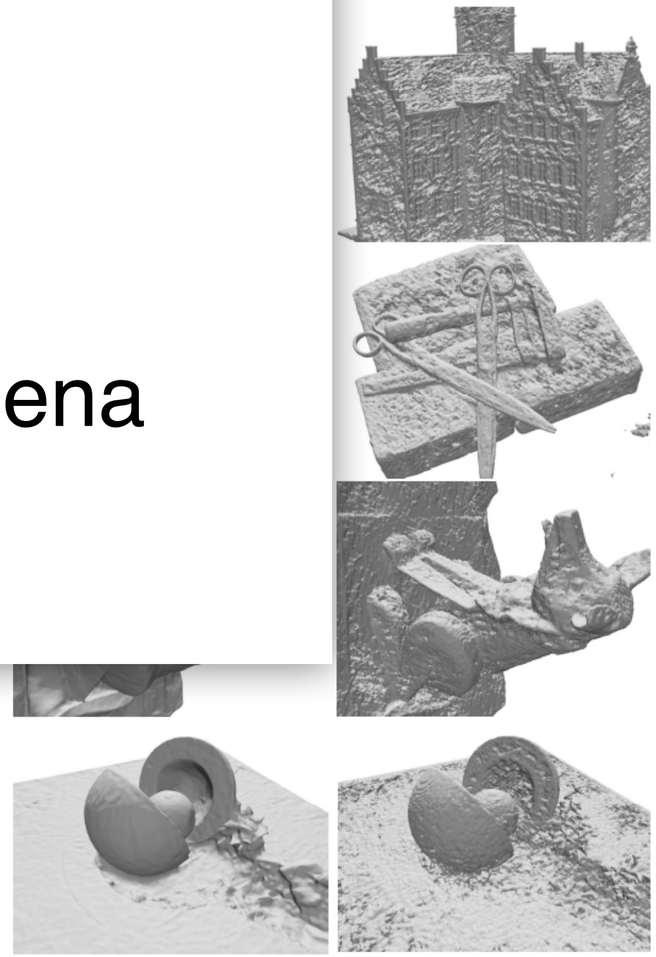




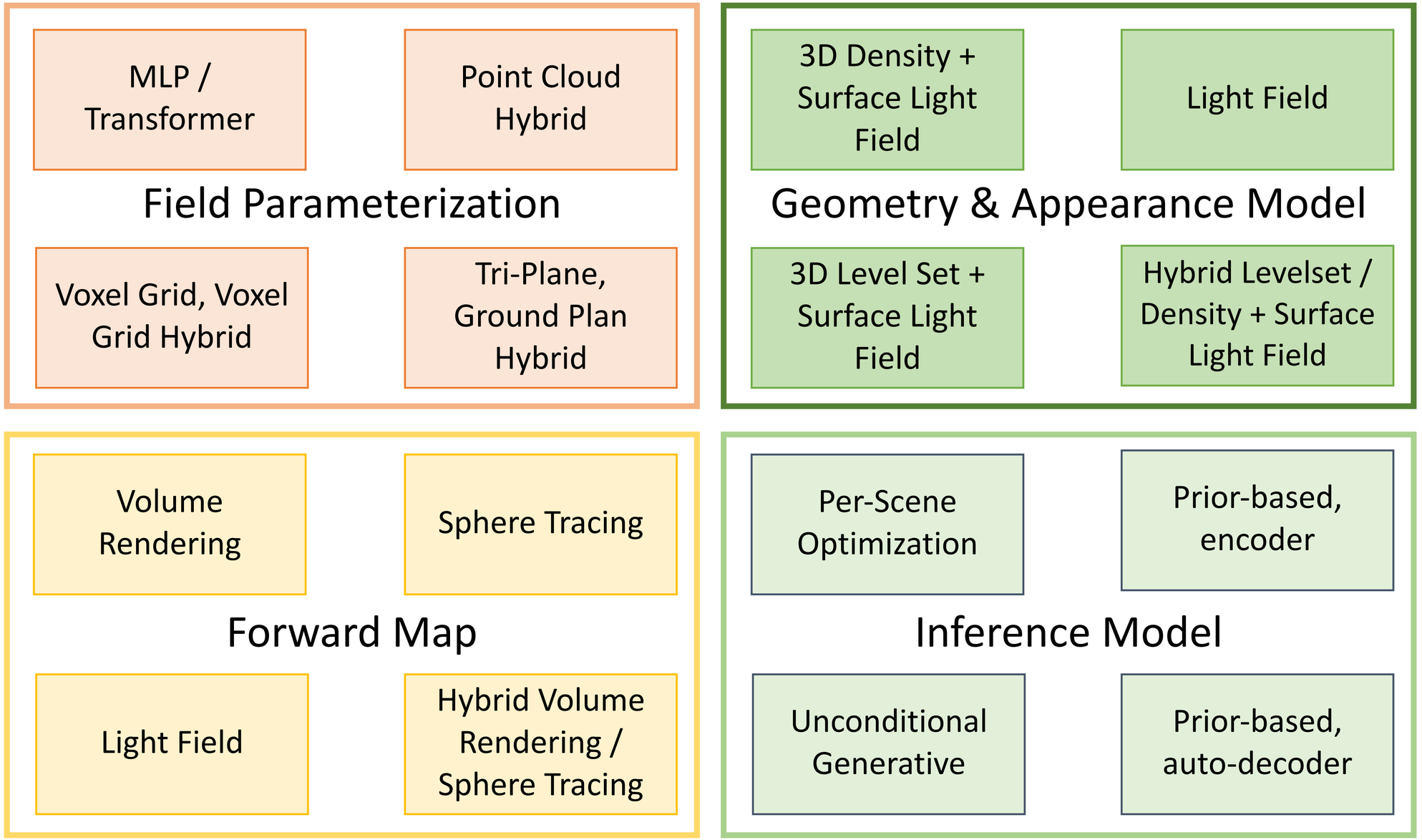
- Surface is well defined
- No noisy or cloudy geometry
- Good for solid objects
- Not good for complex scene phenomena (transparency)



NeRF



3D Inverse Graphics Taxonomy



arXiv:2111.11426v4 [cs.CV] 5 Apr 2022

EUROGRAPHICS 2022
D. Meneveaux and G. Patanè
(Guest Editors)

Volume 41 (2022), Number 2
STAR – State of The Art Report

Neural Fields in Visual Computing and Beyond

Yiheng Xie[1,2] Towaki Takikawa[3,4] Shunsuke Saito[5] Or Litany[4] Shiqin Yan[1] Numair Khan[1] Federico Tombari[6,7]
James Tompkin[1] Vincent Sitzmann[8†] Srinath Sridhar[1†]

[1]Brown University [2]Unity Technologies [3]University of Toronto [4]NVIDIA [5]Meta Reality Labs Research [6]Google [7]Technical University of Munich
[8]Massachusetts Institute of Technology [†]*Equal advising*

https://neuralfields.cs.brown.edu/

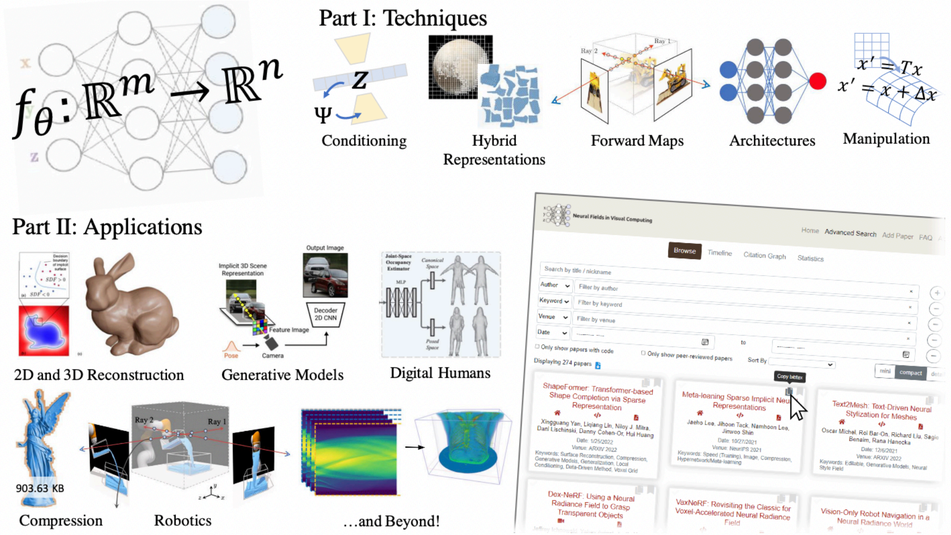
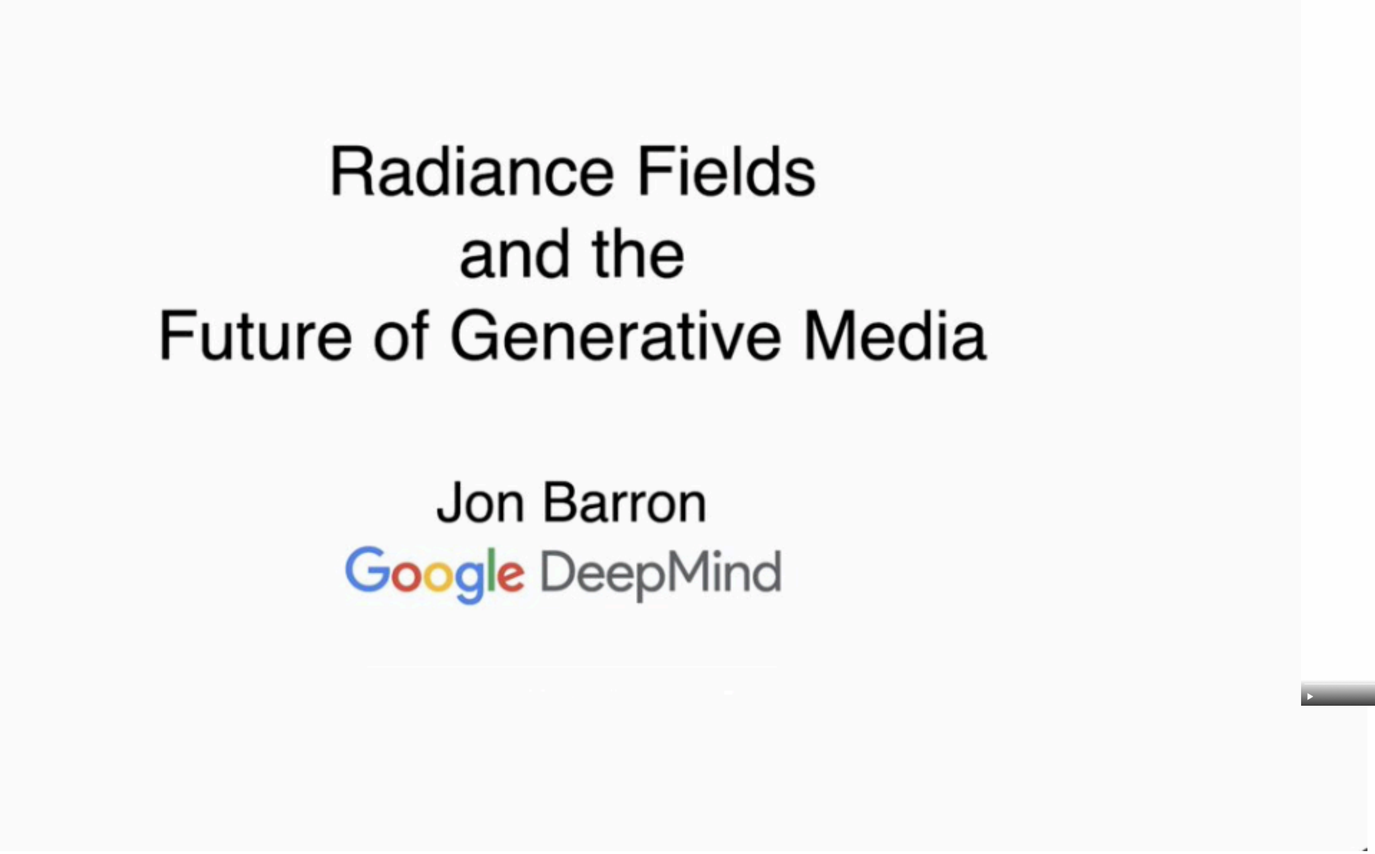
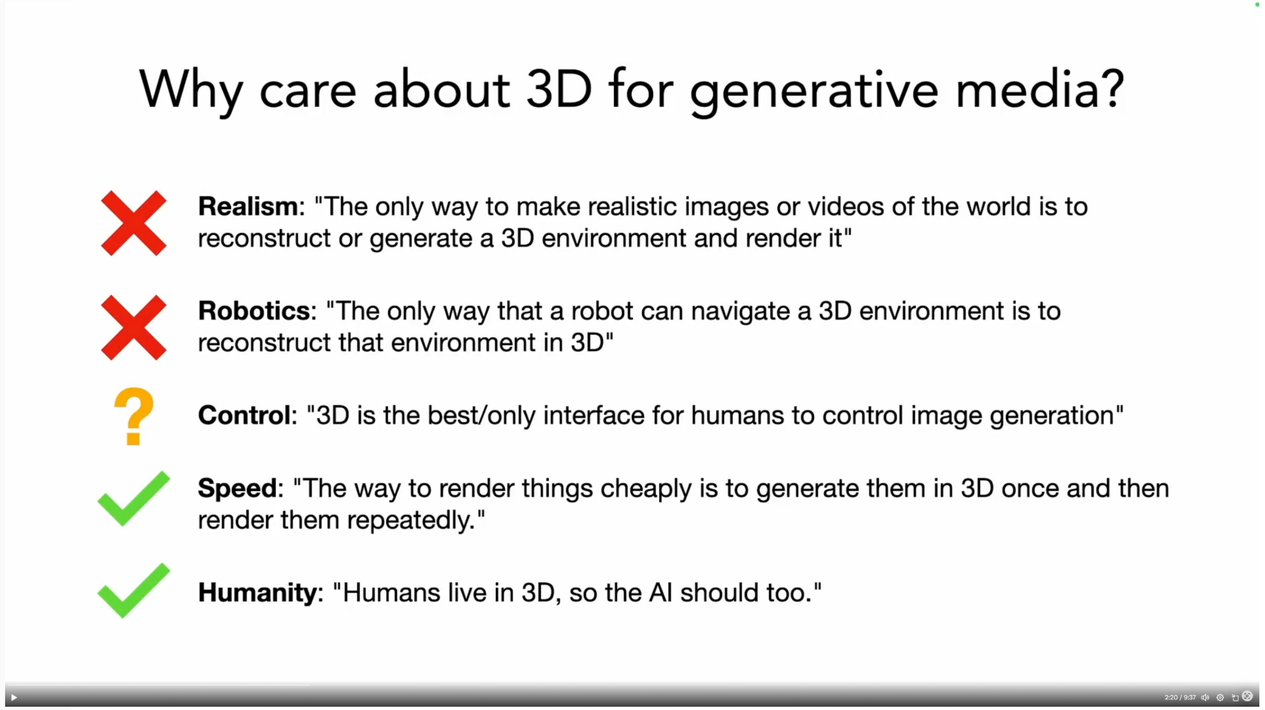


Figure 1: **Contribution of this report.** Following a survey of over 250 papers, we provide a review of (**Part I**) techniques in neural fields such as prior learning and conditioning, representations, forward maps, architectures, and manipulation, and of (**Part II**) applications in visual computing including 2D image processing, 3D scene reconstruction, generative modeling, digital humans, compression, robotics, and beyond. This report is complemented by a community-driven website with search, filtering, bibliographic, and visualization features.

Great Jon Barron talk

https://x.com/jon_barron/status/1916918872177054171





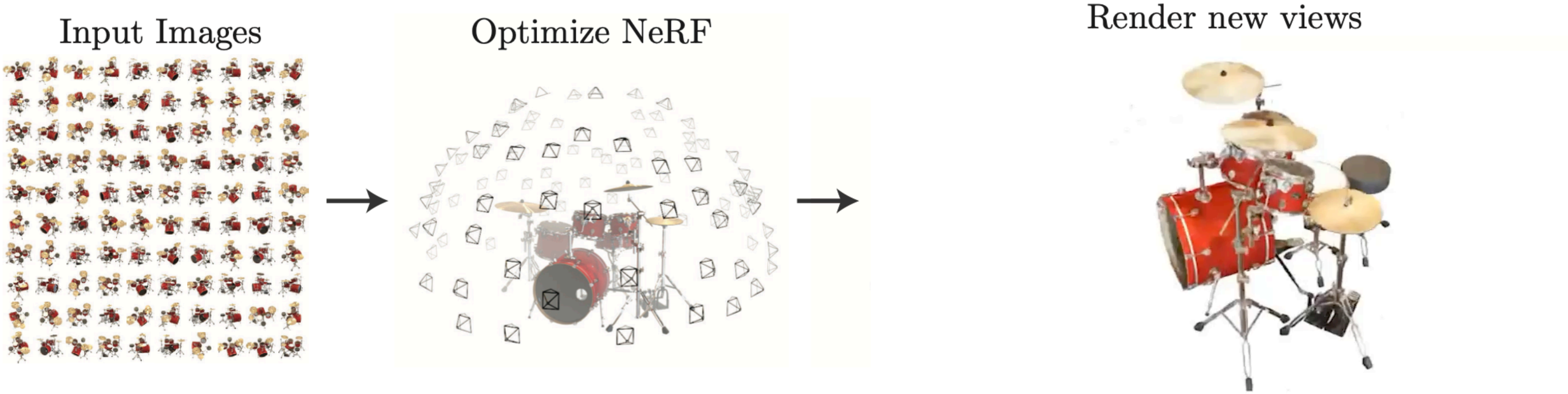
Recap

New representations and rendering methods enable differentiable/inverse rendering and novel view synthesis

- Fuzziness, implicit/SDF, neural, hybrid, pure primitives, 4D light field

Single scene overfitting vs training & inference

Less and less inductive bias is needed as training data grows



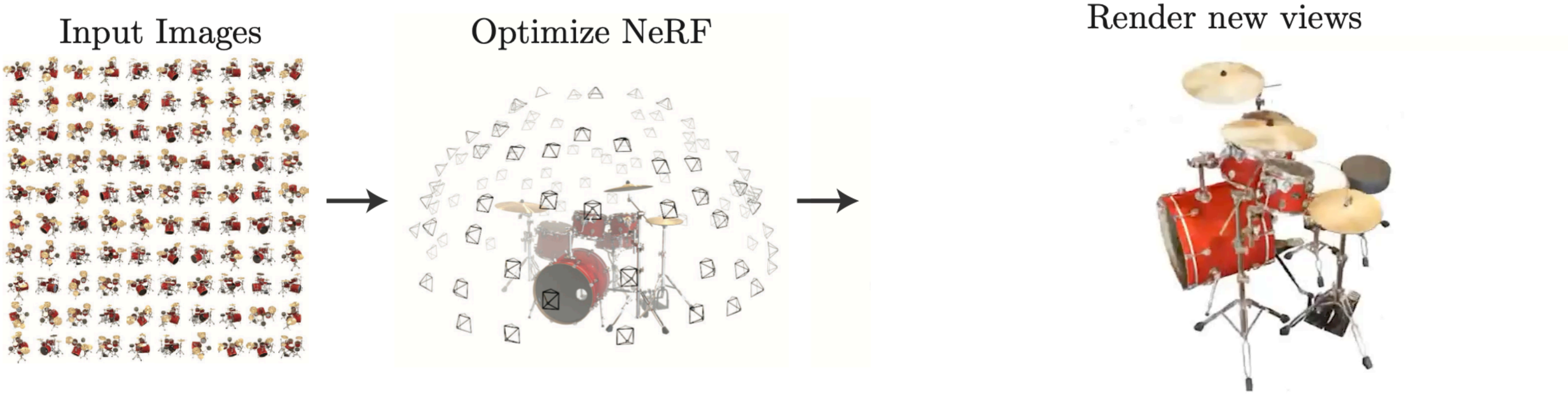
Recap

New representations and rendering methods enable differentiable/inverse rendering and novel view synthesis

- Fuzziness, implicit/SDF, neural, hybrid, pure primitives, 4D light field

Single scene overfitting vs training & inference

Less and less inductive bias is needed as training data grows



Quick Philosophy

Bitter Lesson

The problem **only needs learning**, no inductive bias/intelligent manually-coded solution **if you have enough compute & data**

http://www.incompleteideas.net/IncIdeas/BitterLesson.html

https://www.vincentsitzmann.com/blog/bitter_lesson_of_cv/

Parable of the Parser

The problem is not even relevant (often because bitter lesson allows us to skip it.) It was just an **intermediate fake tasks** for a manually coded solution

https://drive.google.com/file/d/1VodGljuEhBKwZIXQwN-ApH6g2wBAVAdK/view

https://x.com/jon_barron/status/1916918872177054171

Is 3D still needed ? Is it a fake task?