Implicit Scene Representations and Neural Rendering

Michael Niemeyer

Autonomous Vision Group
MPI for Intelligent Systems and University of Tübingen

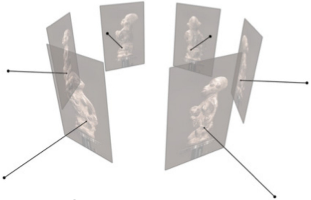
Implicit Scene Representations for 3D Reconstruction

Traditional 3D Reconstruction Pipeline

Input Images

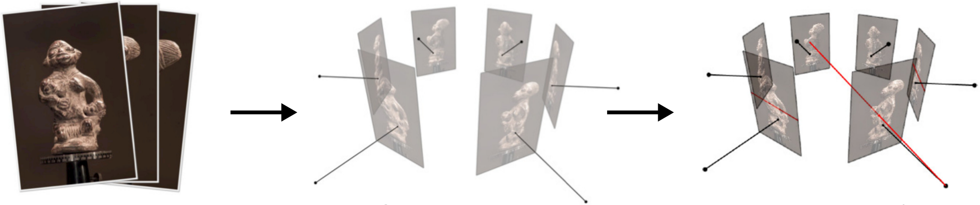
Traditional 3D Reconstruction Pipeline

Input Images



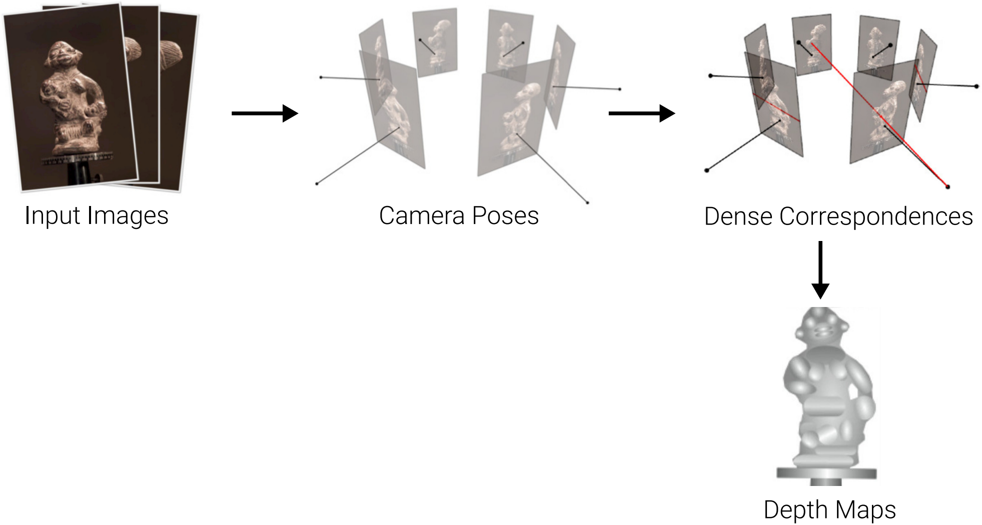
Camera Poses

Traditional 3D Reconstruction Pipeline



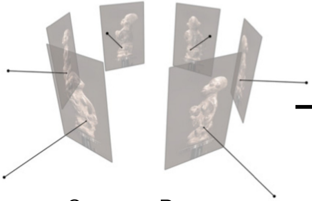
Input Images → Camera Poses → Dense Correspondences

Traditional 3D Reconstruction Pipeline

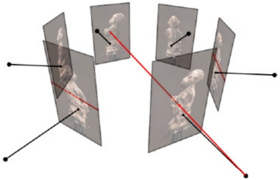


Traditional 3D Reconstruction Pipeline

Input Images

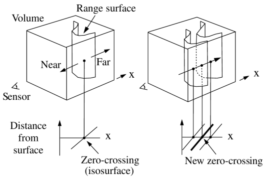


Camera Poses



Dense Correspondences

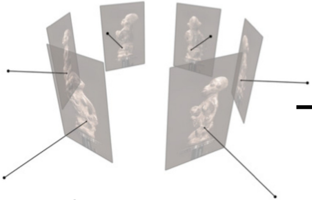
Depth Maps



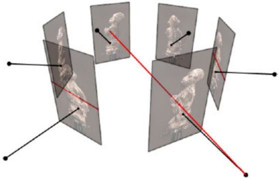
Depth Map Fusion

Traditional 3D Reconstruction Pipeline

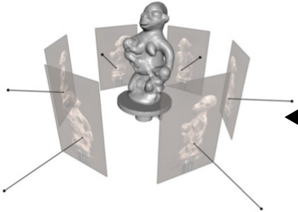
Input Images



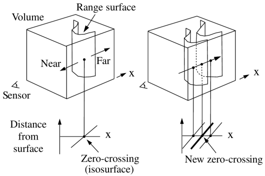
Camera Poses



Dense Correspondences



3D Reconstruction

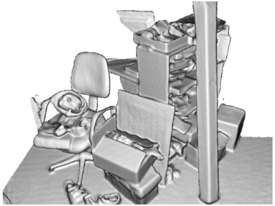


Depth Map Fusion

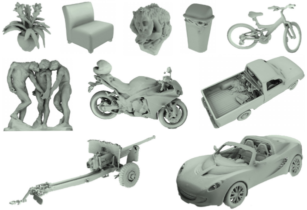
Depth Maps

Can we **learn** 3D reconstruction **from data**?

3D Datasets and Repositories



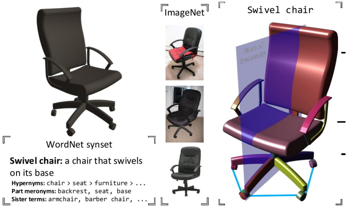
[Newcombe et al., 2011]



[Choi et al., 2011]

[Dai et al., 2017]

[Wu et al., 2015]

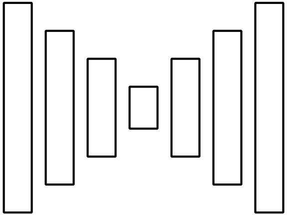


[Chang et al., 2015]

[Chang et al., 2017]

3D Reconstruction from a 2D Image

Input Images



Neural Network

3D Reconstruction

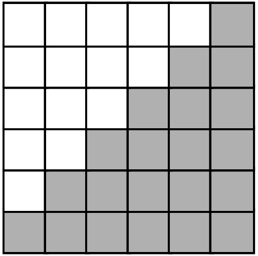
What is a good output representation?

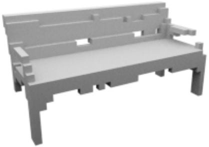
Learning-based 3D Reconstruction

Voxels:

- **Discretization** of 3D space into grid
- Easy to process with neural networks
- Cubic memory $O(n^3) \Rightarrow$ limited resolution
- Manhattan world bias

[Maturana et al., IROS 2015]

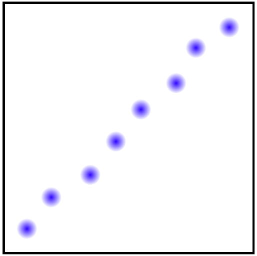




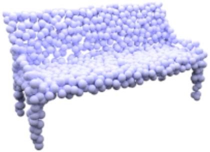
Learning-based 3D Reconstruction

Points:

- **Discretization** of surface into 3D points
- Does not model connectivity / topology
- Limited number of points
- Global shape description



[Fan et al., CVPR 2017]

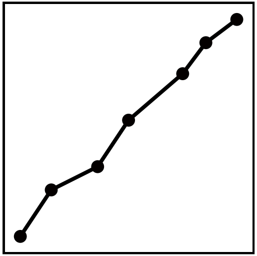


Learning-based 3D Reconstruction

Meshes:

- **Discretization** into vertices and faces
- Limited number of vertices / granularity
- Requires class-specific template – or –
- Leads to self-intersections

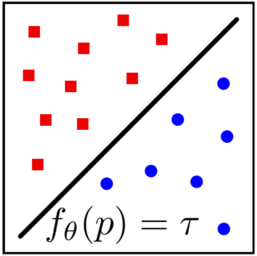
[Groueix et al., CVPR 2018]

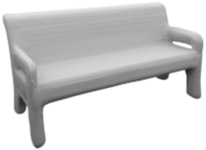


Learning-based 3D Reconstruction

This work:

- Implicit representation ⇒ **No discretization**
- Arbitrary topology & resolution
- Low memory footprint
- Not restricted to specific class





Occupancy Networks

Key Idea:

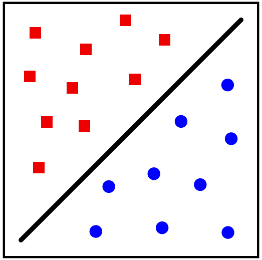
- Do not represent 3D shape explicitly

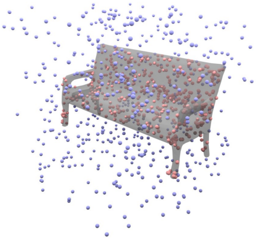
Occupancy Networks

Key Idea:

- Do not represent 3D shape explicitly
- Instead, consider surface **implicitly** as **decision boundary** of a non-linear classifier:

$$\underbrace{f_\theta : \mathbb{R}^3}_{\text{3D Location}} \times \underbrace{\mathcal{X}}_{\text{Condition (eg, Image)}} \rightarrow \underbrace{[0, 1]}_{\text{Occupancy Probability}}$$





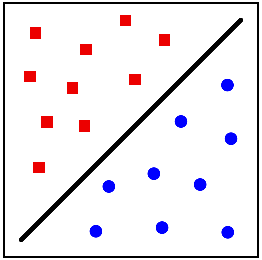
Occupancy Networks

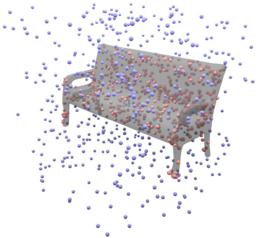
Key Idea:

- Do not represent 3D shape explicitly
- Instead, consider surface **implicitly** as **decision boundary** of a non-linear classifier:

$$f_\theta : \mathbb{R}^3 \times \mathcal{X} \to [0, 1]$$

3D Location — Condition (eg, Image) — Occupancy Probability





Remarks:

- The function f_θ models an **occupancy field**

Mescheder, Oechsle, Niemeyer, Nowozin and Geiger: Occupancy Networks: Learning 3D Reconstruction in Function Space. CVPR, 2019.

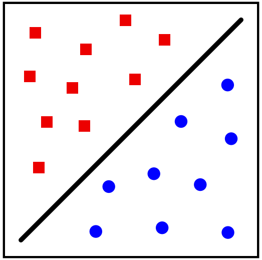
Occupancy Networks

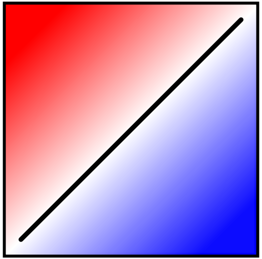
Key Idea:

- Do not represent 3D shape explicitly
- Instead, consider surface **implicitly** as **decision boundary** of a non-linear classifier:

$$f_\theta : \mathbb{R}^3 \times \mathcal{X} \to [0, 1]$$

3D Location | Condition (eg, Image) | Occupancy Probability

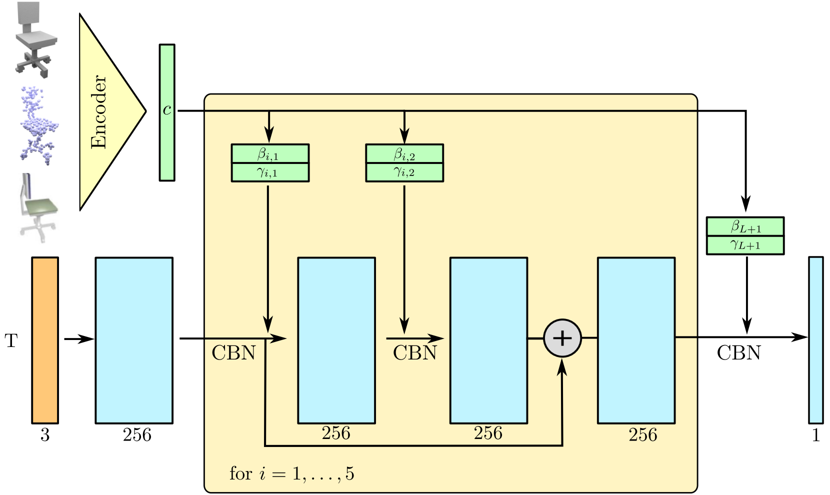




Remarks:

- The function f_θ models an **occupancy field**
- Also possible: **signed distance field** [Park et al., 2019]

Network Architecture



Mescheder, Oechsle, Niemeyer, Nowozin and Geiger: Occupancy Networks: Learning 3D Reconstruction in Function Space. CVPR, 2019.

Training Objective

Occupancy Network:

$$\mathcal{L}(\theta, \psi) = \sum_{j=1}^{K} \text{BCE}(f_\theta(p_{ij}, z_i), o_{ij})$$

- K: Randomly sampled 3D points ($K = 2048$)
- BCE: Cross-entropy loss

Mescheder, Oechsle, Niemeyer, Nowozin and Geiger: Occupancy Networks: Learning 3D Reconstruction in Function Space. CVPR, 2019.

Training Objective

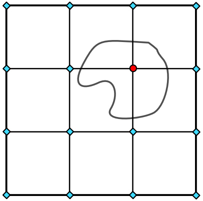
Variational Occupancy Encoder:

$$\mathcal{L}(\theta, \psi) = \sum_{j=1}^{K} \text{BCE}(f_\theta(p_{ij}, z_i), o_{ij}) + KL\left[q_\psi(z|(p_{ij}, o_{ij})_{j=1:K}) \,\|\, p_0(z)\right]$$

- K: Randomly sampled 3D points ($K = 2048$)
- BCE: Cross-entropy loss
- q_ψ: Encoder

Mescheder, Oechsle, Niemeyer, Nowozin and Geiger: Occupancy Networks: Learning 3D Reconstruction in Function Space. CVPR, 2019.

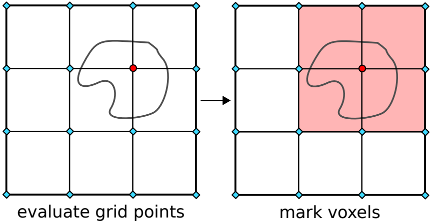
Occupancy Networks



Multiresolution IsoSurface Extraction (MISE):

- Build octree by incrementally querying the occupancy network

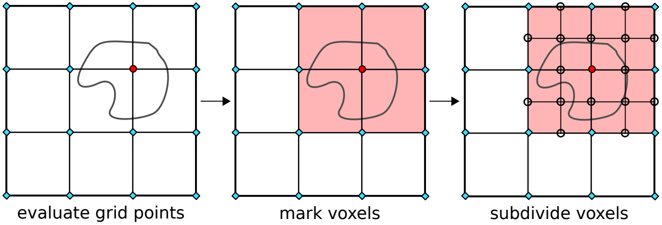
Occupancy Networks



Multiresolution IsoSurface Extraction (MISE):

- Build octree by incrementally querying the occupancy network

Occupancy Networks

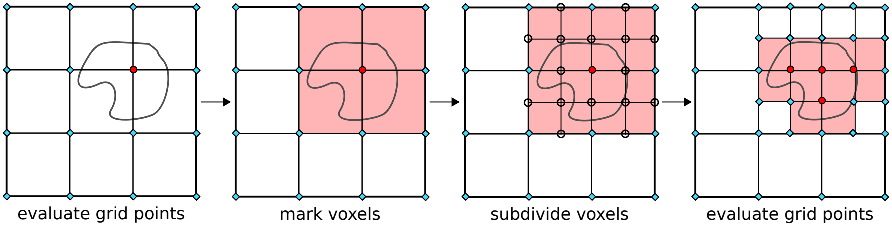


Multiresolution IsoSurface Extraction (MISE):

- Build octree by incrementally querying the occupancy network

Mescheder, Oechsle, Niemeyer, Nowozin and Geiger: Occupancy Networks: Learning 3D Reconstruction in Function Space. CVPR, 2019.

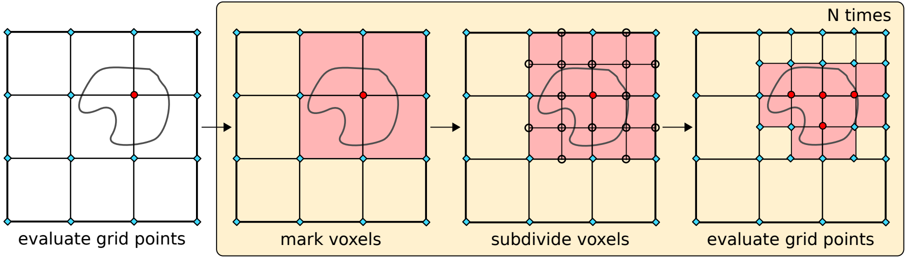
Occupancy Networks



Multiresolution IsoSurface Extraction (MISE):

- Build octree by incrementally querying the occupancy network

Occupancy Networks

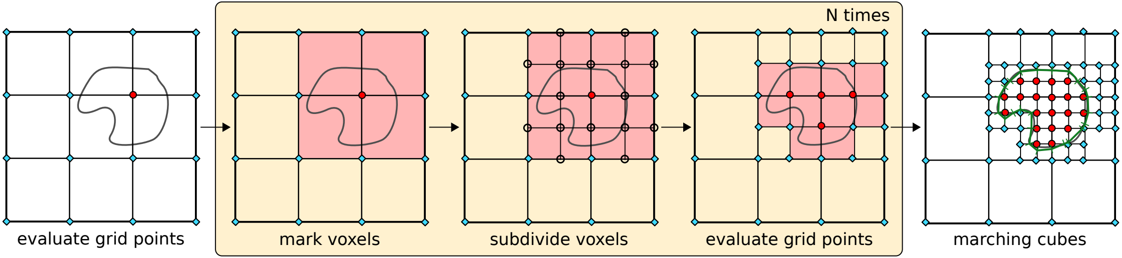


Multiresolution IsoSurface Extraction (MISE):

- Build octree by incrementally querying the occupancy network

Mescheder, Oechsle, Niemeyer, Nowozin and Geiger: Occupancy Networks: Learning 3D Reconstruction in Function Space. CVPR, 2019.

Occupancy Networks

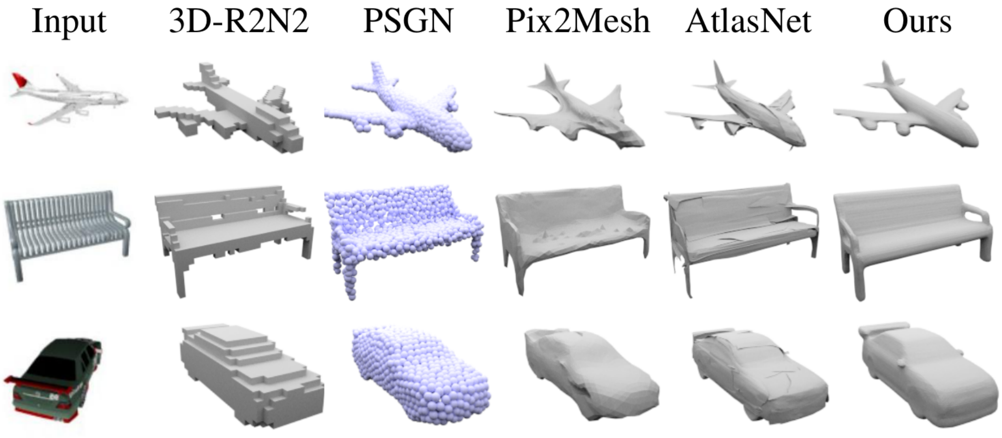


Multiresolution IsoSurface Extraction (MISE):

- Build octree by incrementally querying the occupancy network
- Extract triangular mesh using marching cubes algorithm (1-3 seconds in total)

Mescheder, Oechsle, Niemeyer, Nowozin and Geiger: Occupancy Networks: Learning 3D Reconstruction in Function Space. CVPR, 2019.

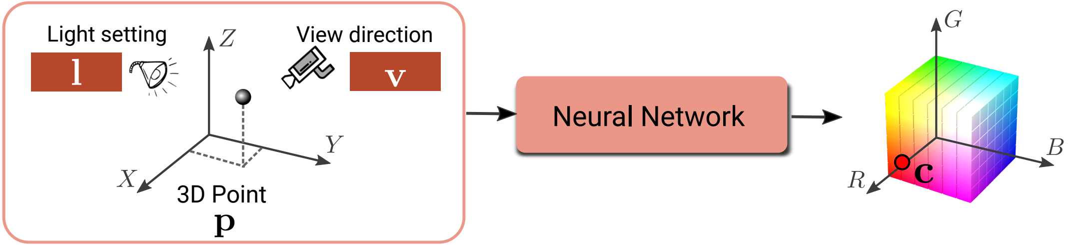
Results



Mescheder, Oechsle, Niemeyer, Nowozin and Geiger: Occupancy Networks: Learning 3D Reconstruction in Function Space. CVPR, 2019.

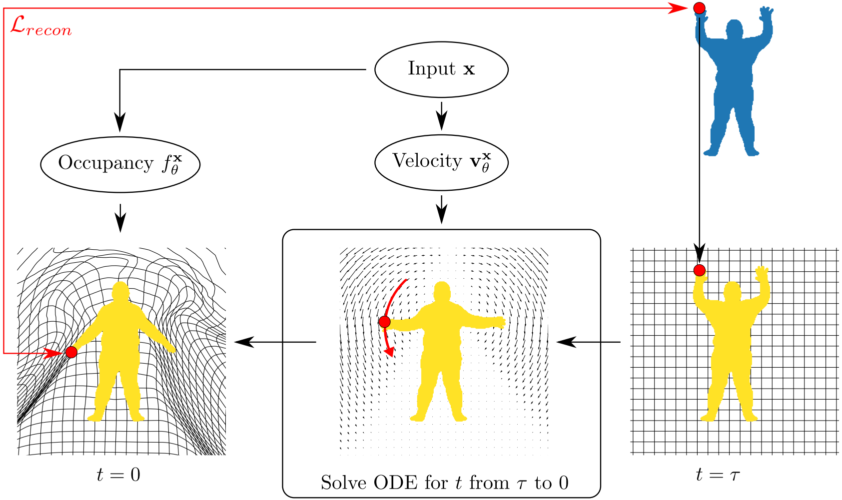
Applications

Appearance



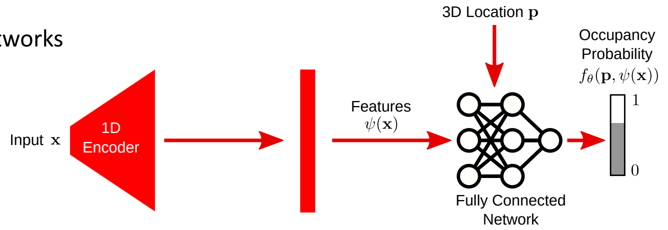
Light setting
l
Z
View direction
v
Y
X
3D Point
p
Neural Network
G
B
R
c

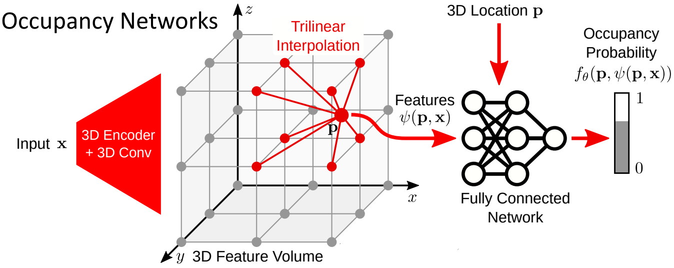
Motion



$\mathcal{L}_{recon}$
Input $\mathbf{x}$
Occupancy $f_\theta^{\mathbf{x}}$
Velocity $\mathbf{v}_\theta^{\mathbf{x}}$
$t = 0$
Solve ODE for t from τ to 0
$t = \tau$

3D Scenes

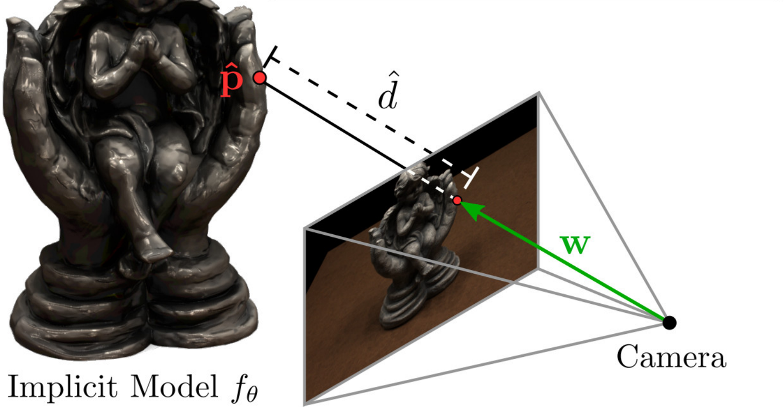




Peng, Niemeyer, Mescheder, Pollefeys and Geiger: Convolutional Occupancy Networks. ECCV, 2020.

Differentiable Rendering

$$\frac{\partial \hat{d}}{\partial \theta} = -\left(\frac{\partial f_\theta(\hat{\mathbf{p}})}{\partial \hat{\mathbf{p}}} \cdot \mathbf{w}\right)^{-1} \frac{\partial f_\theta(\hat{\mathbf{p}})}{\partial \theta}$$



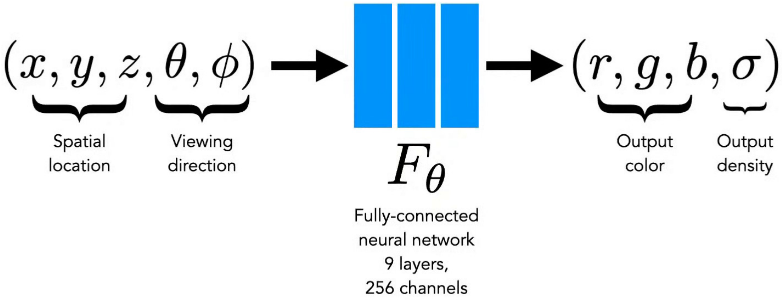
Neural Rendering: Neural Radiance Fields

Novel View Synthesis

- **Task:** Given a set of images of a scene (left), render image from novel viewpoint (right)

Mildenhall, Srinivasan, Tancik, Barron, Ramamoorthi and Ng: NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. ECCV, 2020.

NeRF: Representing Scenes as Neural Radiance Fields



- Vanilla **ReLU MLP** that maps from **location/view direction to color/density**
- **Density** σ describes how solid/transparent a 3D point is (can model, e.g., fog)
- Conditioning on view direction allows for modeling **view-dependent effects**

Volume Rendering

Rendering model for ray r(t) = o + td:

$$C \approx \sum_{i=1}^{N} T_i \alpha_i c_i$$

weights

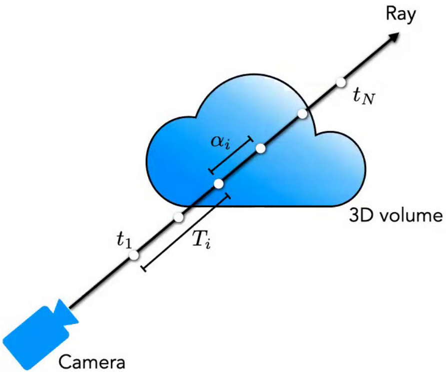
colors

How much light is blocked earlier along ray:

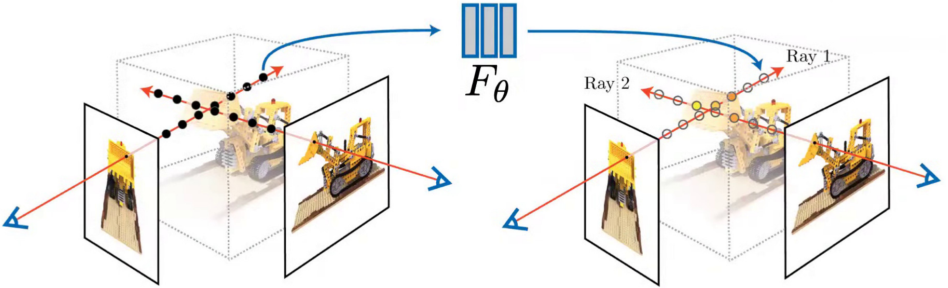
$$T_i = \prod_{j=1}^{i-1} (1 - \alpha_j)$$

How much light is contributed by ray segment *i*:

$$\alpha_i = 1 - e^{-\sigma_i \delta t_i}$$



NeRF Training



$$\min_\theta \sum_i ||\,\text{render}_i(F_\theta) - I_i||^2$$

- Shoot ray, render ray to pixel, minimize **reconstruction error** via backpropagation

Mildenhall, Srinivasan, Tancik, Barron, Ramamoorthi and Ng: NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. ECCV, 2020.

Fourier Features

NeRF (Naive)

NeRF (with positional encoding)

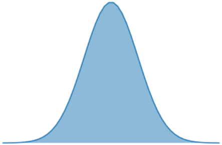
- Essential trick: Compute **positional encoding** for input point $\mathbf{x}$ and direction $\mathbf{d}$

Tancik et al.: Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains. NeurIPS, 2020.

Generative Neural Scene Representations

Generative Models

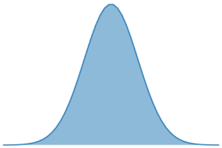
Sample a latent code from the prior distribution.



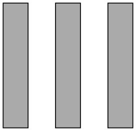
Latent Code

Generative Models

Pass latent code to trained generator G_θ.



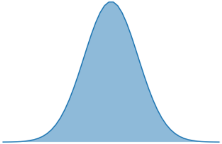
Latent Code



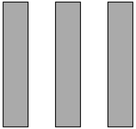
Generator G_θ

Generative Models

The generator outputs a synthesized image.



Latent Code



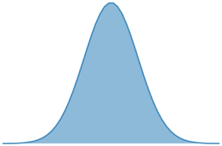
Generator G_θ

Generated Image*

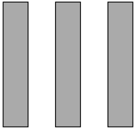
* The generated images are samples from StyleGAN2.

Generative Models

Sample more latent codes to get different generated images.



Latent Code



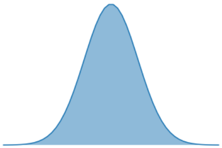
Generator G_θ

Generated Image*

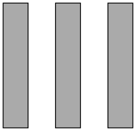
* The generated images are samples from StyleGAN2.

Generative Models

Sample more latent codes to get different generated images.



Latent Code



Generator G_θ

Generated Image*

* The generated images are samples from StyleGAN2.

Is the ability to sample photorealistic images
all we want?

Generative Models

For many applications, we require **control over the generation process**:

Generative Models

For many applications, we require **control over the generation process**:

Note: This and the following videos are only shown when opened with a supported PDF reader (e.g. Okular).

Animation Movies

Video Source: Disney's Toy Story 4 Trailer

Generative Models

For many applications, we require **control over the generation process**:

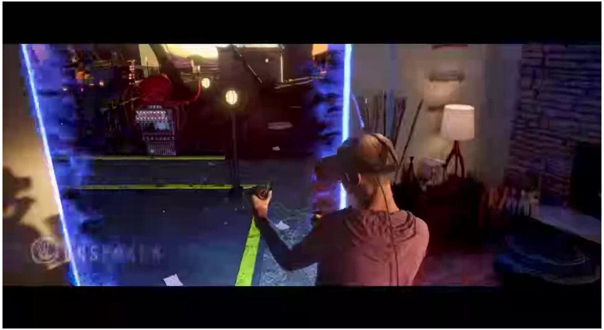
Video Games

Video Source: Gran Turismo 7 Trailer

Generative Models

For many applications, we require **control over the generation process**:

Virtual Reality



Video Source:Oculus Rift Trailer

Generative Models

Goal: A generative model for **3D-aware image synthesis** which allows us to:

Generative Models

Goal: A generative model for **3D-aware image synthesis** which allows us to:

- Generate photorealistic images

Generative Models

Goal: A generative model for **3D-aware image synthesis** which allows us to:

- ► Generate photorealistic images
- ► Control individual objects wrt. their pose, size, and position in 3D

Generative Models

Goal: A generative model for **3D-aware image synthesis** which allows us to:

- Generate photorealistic images
- Control individual objects wrt. their pose, size, and position in 3D
- Control camera viewpoint in 3D

Generative Models

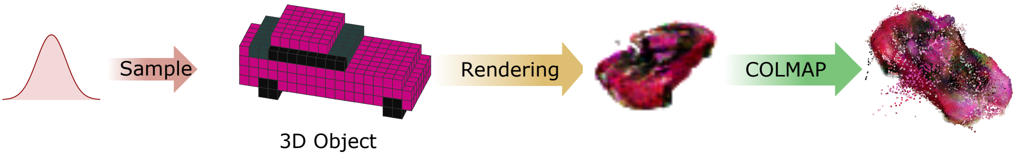
Goal: A generative model for **3D-aware image synthesis** which allows us to:

- Generate photorealistic images
- Control individual objects wrt. their pose, size, and position in 3D
- Control camera viewpoint in 3D
- Train from collections of unposed images

What representation should we use for 3D-aware image synthesis?

3D Representations

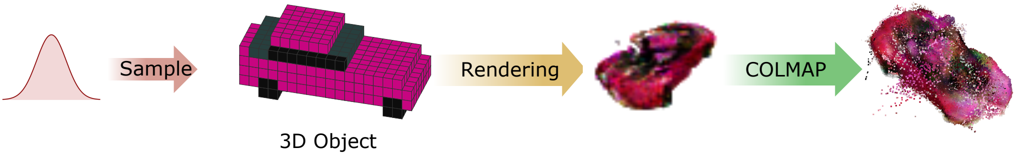
Voxel-based 3D Shape with Volumetric Rendering



PlatonicGAN [Henzler et al., ICCV 2019]

3D Representations

Voxel-based 3D Shape with Volumetric Rendering

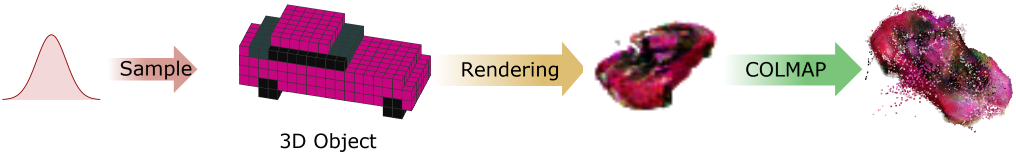


PlatonicGAN [Henzler et al., ICCV 2019]

+ Multi-view consistent

3D Representations

Voxel-based 3D Shape with Volumetric Rendering



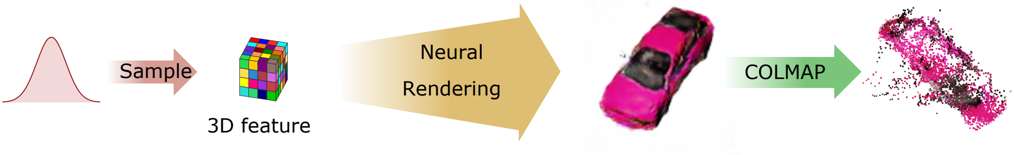
PlatonicGAN [Henzler et al., ICCV 2019]

+ Multi-view consistent

− Low image fidelity, high memory consumption

3D Representations

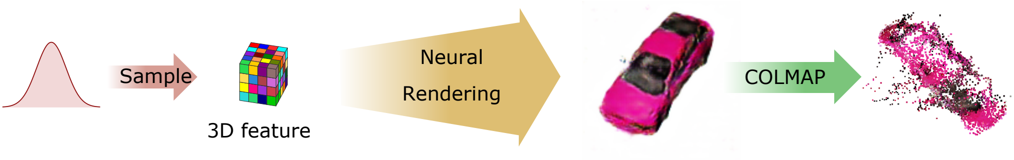
Voxel-based 3D Latent Feature with Learnable Projection



HoloGAN [Nguyen-Phuoc et al., ICCV 2019]

3D Representations

Voxel-based 3D Latent Feature with Learnable Projection



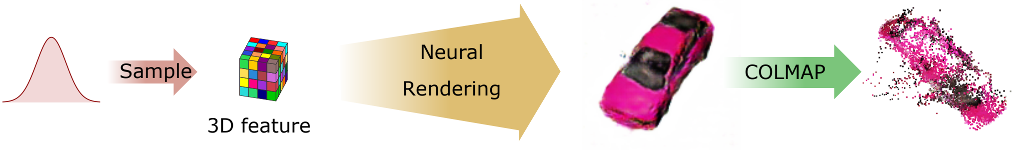
HoloGAN [Nguyen-Phuoc et al., ICCV 2019]

+ High image fidelity

Schwarz, Liao, Niemeyer, Geiger: GRAF: Generative Radiance Fields for 3D-Aware Image Synthesis. NeurIPS, 2020

3D Representations

Voxel-based 3D Latent Feature with Learnable Projection



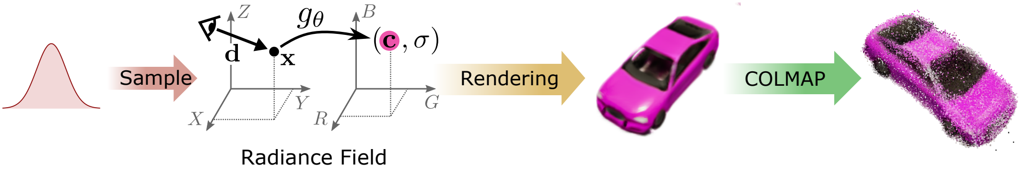
HoloGAN [Nguyen-Phuoc et al., ICCV 2019]

+ High image fidelity

− Object identity may vary with viewpoint due to learnable projection

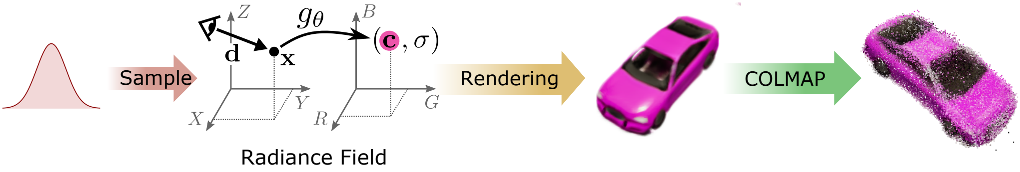
3D Representations

Generative Radiance Fields



3D Representations

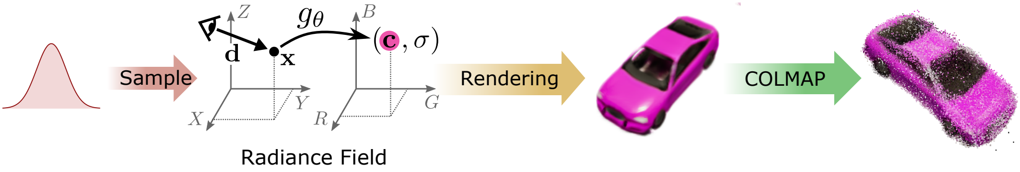
Generative Radiance Fields



+ Continuous representation, multi-view consistent

3D Representations

Generative Radiance Fields



- Continuous representation, multi-view consistent
- High image fidelity, low memory consumption

Schwarz, Liao, Niemeyer, Geiger: GRAF: Generative Radiance Fields for 3D-Aware Image Synthesis. NeurIPS, 2020

Generative Radiance Fields

Generative Radiance Fields

Sample camera matrix $\mathbf{K}$, camera pose $\boldsymbol{\xi} \sim p_\xi$, and patch sampling pattern $\boldsymbol{\nu} \sim p_\nu$.

$\mathbf{K}$

$\boldsymbol{\xi} \sim p_\xi$

$\boldsymbol{\nu} \sim p_\nu$

Schwarz, Liao, Niemeyer, Geiger: GRAF: Generative Radiance Fields for 3D-Aware Image Synthesis. NeurIPS, 2020

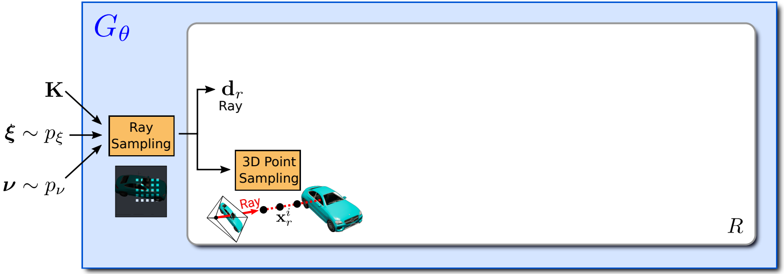
Generative Radiance Fields

Pass $\mathbf{K}$, $\boldsymbol{\xi}$, and $\boldsymbol{\nu}$ to generator G_θ and sample pixels / rays on image plane.

Schwarz, Liao, Niemeyer, Geiger: GRAF: Generative Radiance Fields for 3D-Aware Image Synthesis. NeurIPS, 2020

Generative Radiance Fields

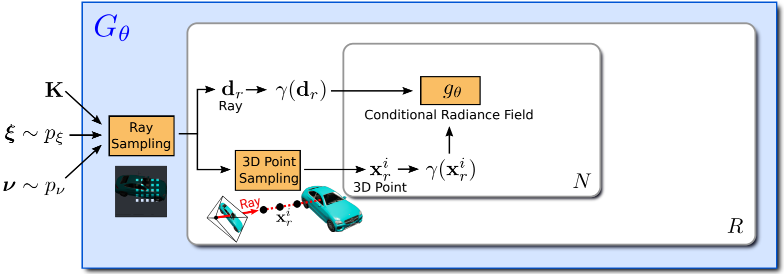
For each ray, get viewing direction $\mathbf{d}_r$ and sample 3D points $\mathbf{x}_r^i$ along ray.



Patch Generator

Generative Radiance Fields

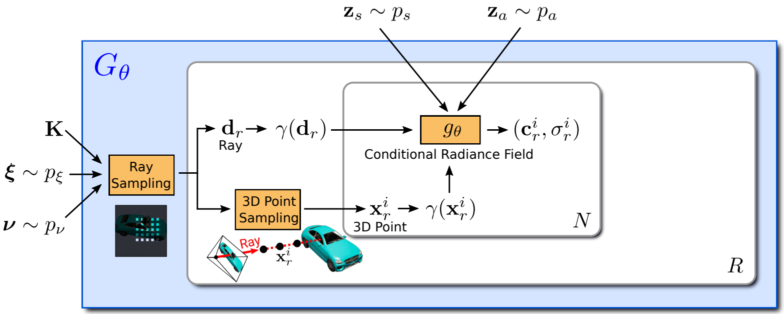
Pass $\mathbf{d}_r$ and $\mathbf{x}_r^i$ to positional encoding γ and then to the conditional radiance field g_θ.



Patch Generator

Generative Radiance Fields

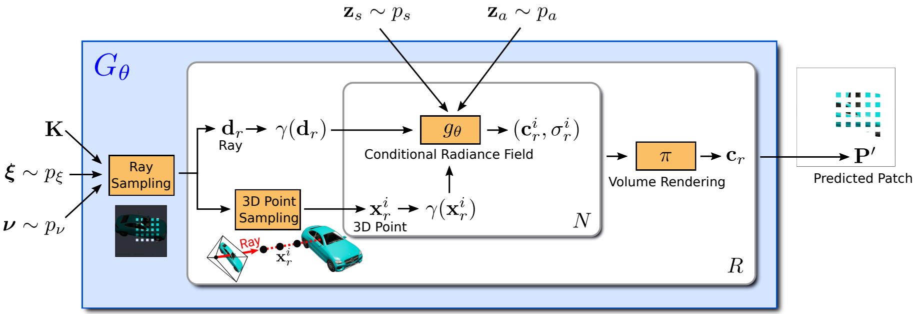
Sample latent shape and appearance codes $\mathbf{z}_s, \mathbf{z}_a$ and pass them to g_θ.



Patch Generator

Generative Radiance Fields

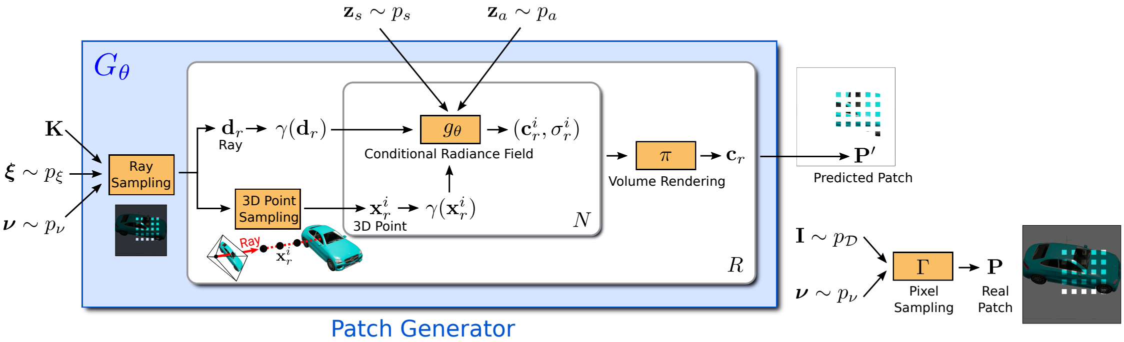
Perform volume-rendering for each ray and get predicted patch $\mathbf{P}'$.



Patch Generator

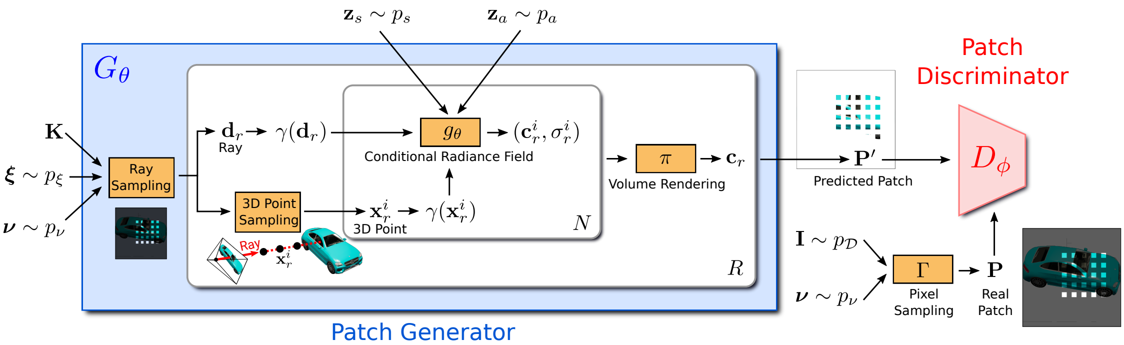
Generative Radiance Fields

Sample patch $\mathbf{P}$ from real image $\mathbf{I}$ drawn from the data distribution $p_{\mathcal{D}}$.



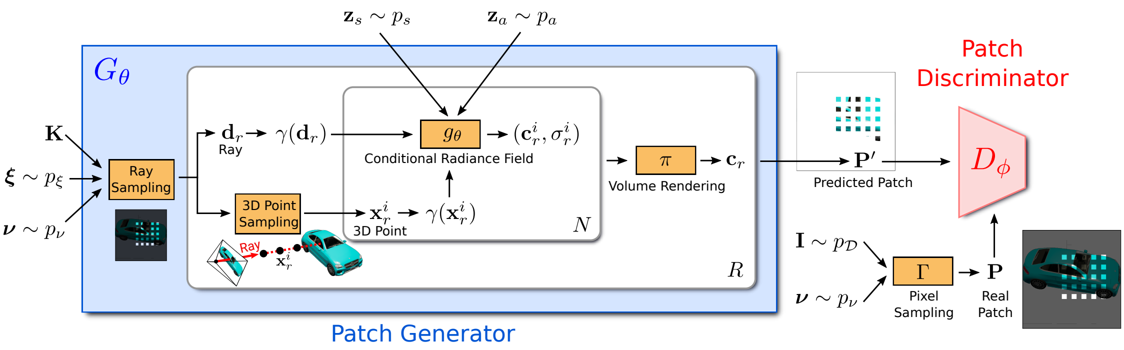
Generative Radiance Fields

Pass fake and real patch $\mathbf{P}', \mathbf{P}$ to discriminator D_ϕ and train with adversarial loss.



Schwarz, Liao, Niemeyer, Geiger: GRAF: Generative Radiance Fields for 3D-Aware Image Synthesis. NeurIPS, 2020

Generative Radiance Fields



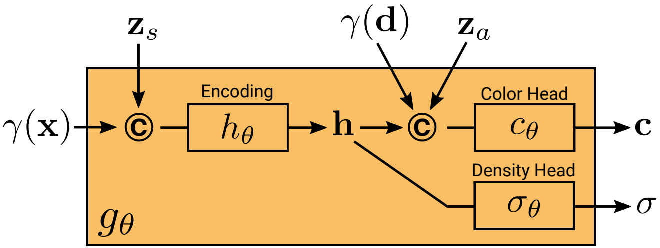
- Generator/discriminator for **image patches** of size 32×32 pixels
- Patches sampled at **random scale** using dilation

Schwarz, Liao, Niemeyer, Geiger: GRAF: Generative Radiance Fields for 3D-Aware Image Synthesis. NeurIPS, 2020

How do we parametrize

Conditional Radiance Fields?

Conditional Radiance Fields

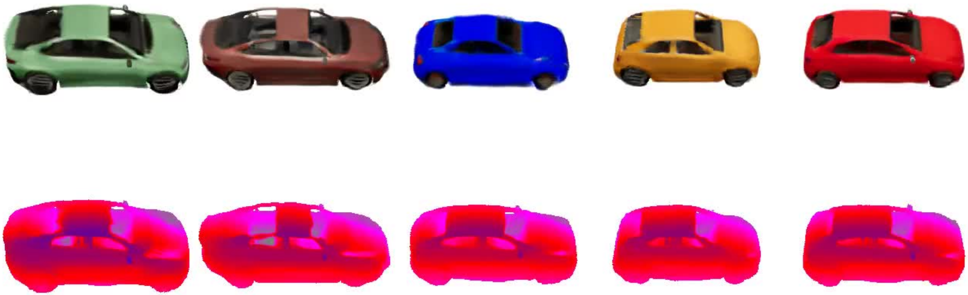


- Conditional radiance fields as fully-connected MLPs with ReLU activation
- Shape code $\mathbf{z}_s$ concatenated with encoded 3D location $\gamma(\mathbf{x})$
- Appearance code $\mathbf{z}_a$ concatenated with encoded viewing direction $\gamma(\mathbf{d})$

How well does it work?

Generative Radiance Fields

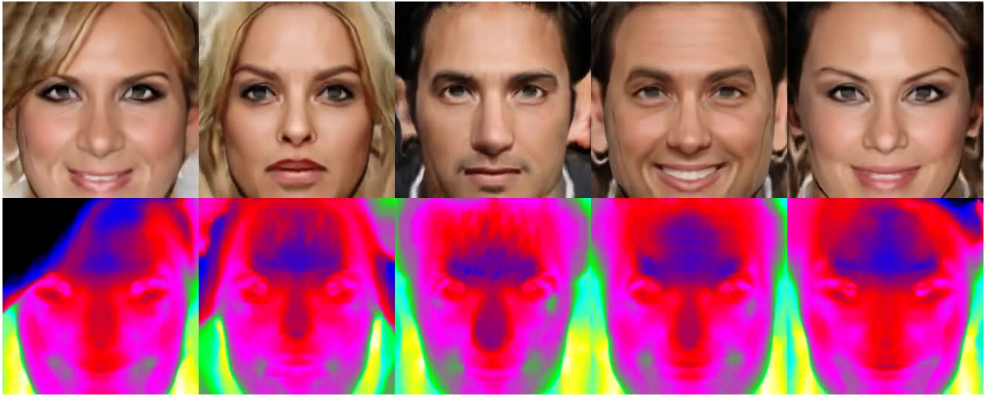
Results on synthetic Carla dataset at 256^2 pixels:



Schwarz, Liao, Niemeyer, Geiger: GRAF: Generative Radiance Fields for 3D-Aware Image Synthesis. NeurIPS, 2020

Generative Radiance Fields

Results on real CelebA-HQ dataset at 256^2 pixels:



Schwarz, Liao, Niemeyer, Geiger: GRAF: Generative Radiance Fields for 3D-Aware Image Synthesis. NeurIPS, 2020

How can we scale to
more complex, multi-object scenes?

GIRAFFE: Compositional Generative Neural Feature Fields

GRAF:

- Incorporate a **3D representation** into the generative model

GIRAFFE: Compositional Generative Neural Feature Fields

GRAF:

- Incorporate a **3D representation** into the generative model

GIRAFFE:

- Incorporate a **compositional 3D scene representation** into the generative model

GIRAFFE: Compositional Generative Neural Feature Fields

GRAF:

- Incorporate a **3D representation** into the generative model

GIRAFFE:

- Incorporate a **compositional 3D scene representation** into the generative model
- Incorporate a **neural renderer** to yield fast and high-quality inference

Niemeyer, Geiger: GIRAFFE: Representing Scenes as Compositional Generative Neural Feature Fields. CVPR, 2021

GIRAFFE

GIRAFFE

Sample N shape and appearance codes.

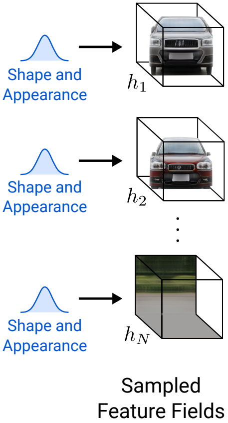
Shape and
Appearance

Shape and
Appearance

Shape and
Appearance

GIRAFFE

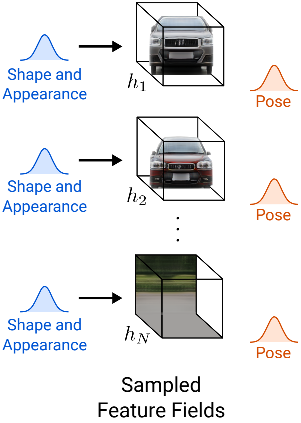
Get N feature fields. Note: We show features in RGB color for clarity.



Niemeyer, Geiger: GIRAFFE: Representing Scenes as Compositional Generative Neural Feature Fields. CVPR, 2021

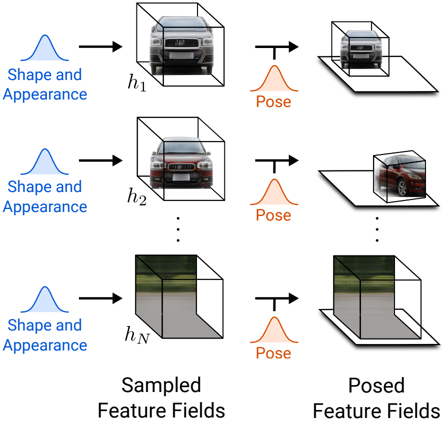
GIRAFFE

Sample size and pose for each feature field.



GIRAFFE

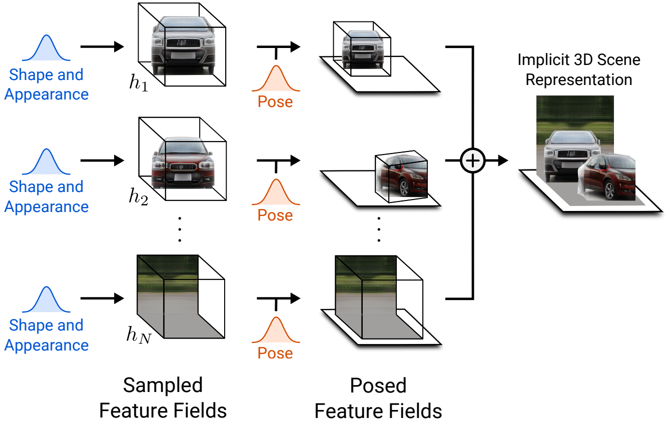
Get posed feature fields.



Niemeyer, Geiger: GIRAFFE: Representing Scenes as Compositional Generative Neural Feature Fields. CVPR, 2021

GIRAFFE

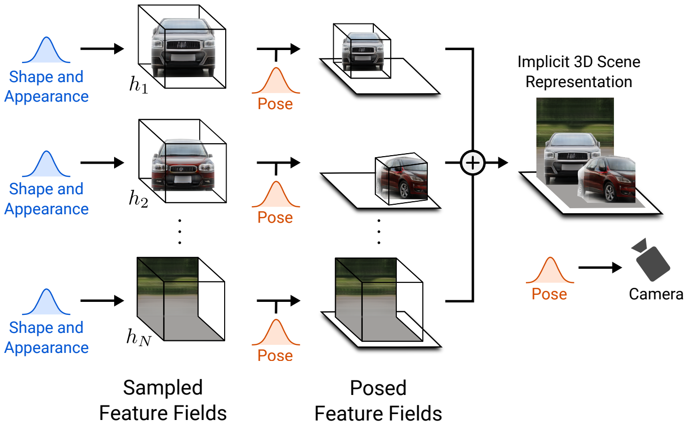
Composite all feature feature fields to one 3D scene representation.



Niemeyer, Geiger: GIRAFFE: Representing Scenes as Compositional Generative Neural Feature Fields. CVPR, 2021

GIRAFFE

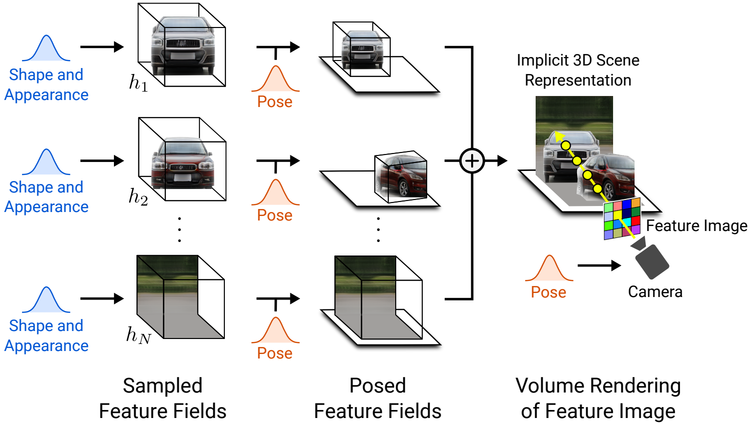
Sample a camera pose.



Niemeyer, Geiger: GIRAFFE: Representing Scenes as Compositional Generative Neural Feature Fields. CVPR, 2021

GIRAFFE

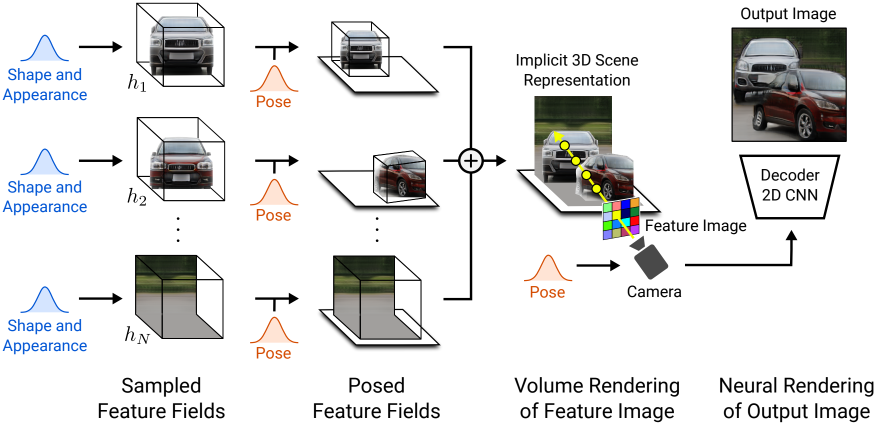
Perform volume rendering and get feature image.



Niemeyer, Geiger: GIRAFFE: Representing Scenes as Compositional Generative Neural Feature Fields. CVPR, 2021

GIRAFFE

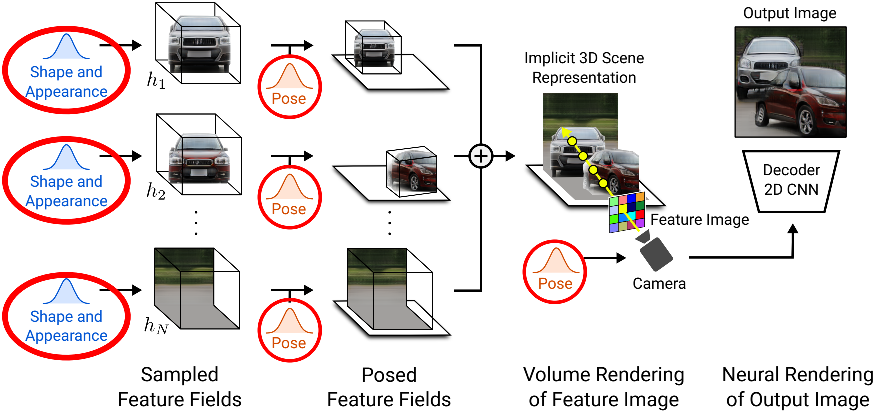
Pass feature image to neural renderer to obtain final output.



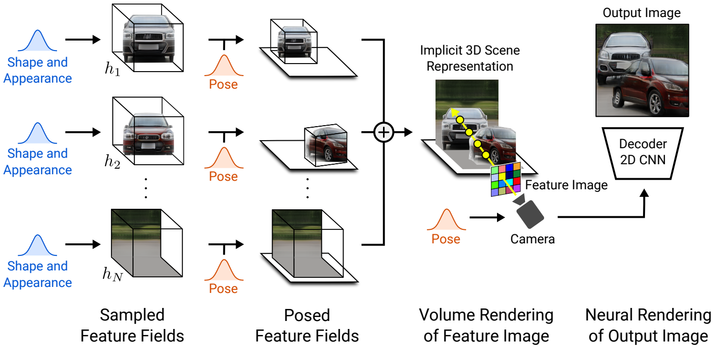
Niemeyer, Geiger: GIRAFFE: Representing Scenes as Compositional Generative Neural Feature Fields. CVPR, 2021

GIRAFFE

At test time, we can sample individual codes and **control the poses**.



GIRAFFE



- We train with adversarial loss **on full image**
- We volume-render the feature image at 16×16 pixels

How well does it work?

GIRAFFE

We compare object translation for a 2D-based GAN (left) and our method (right):

Niemeyer, Geiger: GIRAFFE: Representing Scenes as Compositional Generative Neural Feature Fields. CVPR, 2021

GIRAFFE

We can perform more complex operations like circular translations

Niemeyer, Geiger: GIRAFFE: Representing Scenes as Compositional Generative Neural Feature Fields. CVPR, 2021

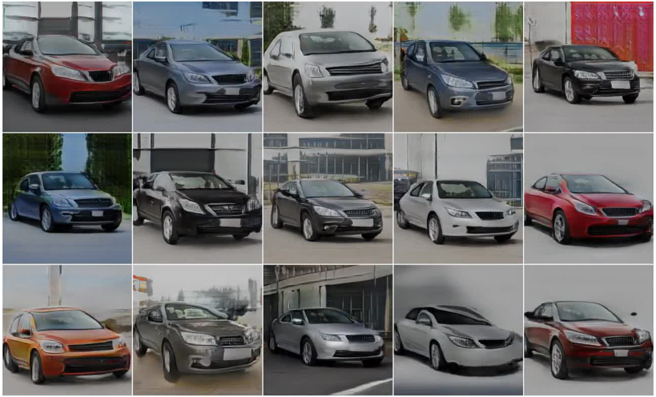
GIRAFFE

We can add more objects at test time (trained on two-object)

Niemeyer, Geiger: GIRAFFE: Representing Scenes as Compositional Generative Neural Feature Fields. CVPR, 2021

GIRAFFE

We can rotate the object



Niemeyer, Geiger: GIRAFFE: Representing Scenes as Compositional Generative Neural Feature Fields. CVPR, 2021

GIRAFFE

We can translate the object

Niemeyer, Geiger: GIRAFFE: Representing Scenes as Compositional Generative Neural Feature Fields. CVPR, 2021

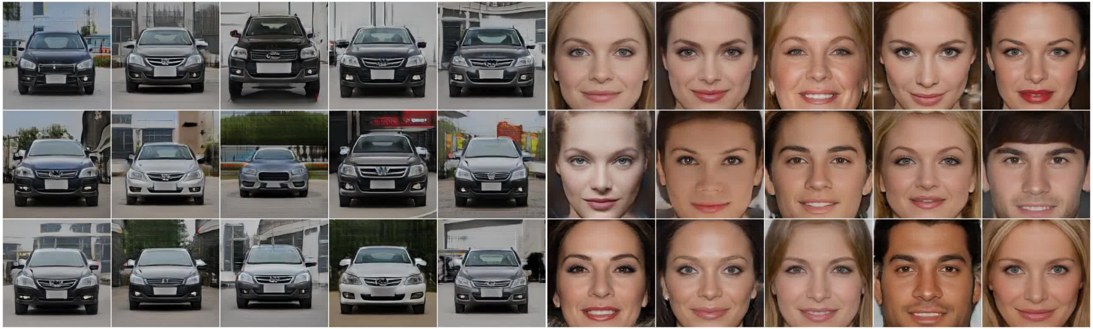
GIRAFFE

We can change the object shape

Niemeyer, Geiger: GIRAFFE: Representing Scenes as Compositional Generative Neural Feature Fields. CVPR, 2021

GIRAFFE

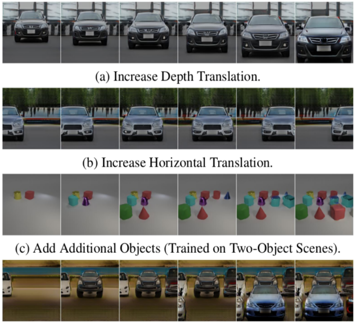
We can change the object appearance



Niemeyer, Geiger: GIRAFFE: Representing Scenes as Compositional Generative Neural Feature Fields. CVPR, 2021

GIRAFFE

We can generate out-of-distribution samples



(a) Increase Depth Translation.

(b) Increase Horizontal Translation.

(c) Add Additional Objects (Trained on Two-Object Scenes).

(d) Add Additional Objects (Trained on Single-Object Scenes).

Summary

Summary

- Occupancy Networks: Represent Surfaces **implicitly** as the decision boundary of a neural network

Summary

Summary

- Occupancy Networks: Represent Surfaces **implicitly** as the decision boundary of a neural network
- This **coordinate-based neural representation** can also be used for other applications ranging from light fields to motion and more

Summary

Summary

- Occupancy Networks: Represent Surfaces **implicitly** as the decision boundary of a neural network
- This **coordinate-based neural representation** can also be used for other applications ranging from light fields to motion and more
- Neural Radiance Fields: Represent scenes as volumes which can be used for view synthesis

Niemeyer, Geiger: GIRAFFE: Representing Scenes as Compositional Generative Neural Feature Fields. CVPR, 2021

Summary

Summary

- Occupancy Networks: Represent Surfaces **implicitly** as the decision boundary of a neural network
- This **coordinate-based neural representation** can also be used for other applications ranging from light fields to motion and more
- Neural Radiance Fields: Represent scenes as volumes which can be used for view synthesis
- Generative Radiance Fields: **3D-aware generative model** of single-object scenes

Niemeyer, Geiger: GIRAFFE: Representing Scenes as Compositional Generative Neural Feature Fields. CVPR, 2021

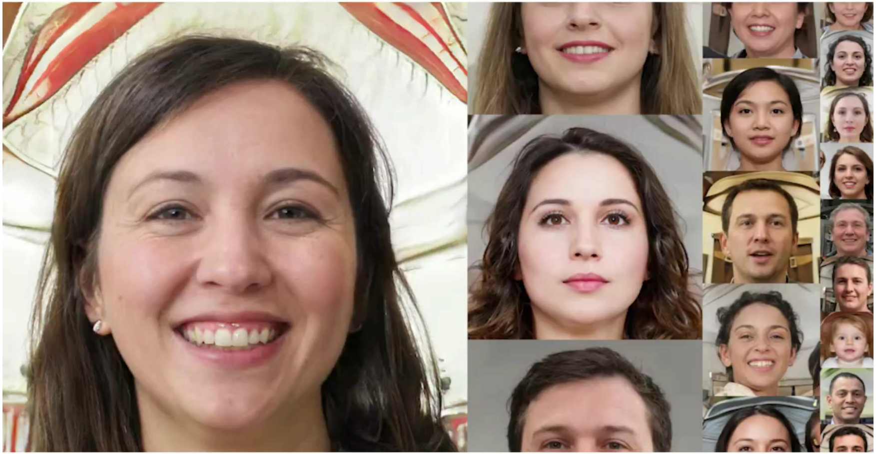
Summary

Summary

- Occupancy Networks: Represent Surfaces **implicitly** as the decision boundary of a neural network
- This **coordinate-based neural representation** can also be used for other applications ranging from light fields to motion and more
- Neural Radiance Fields: Represent scenes as volumes which can be used for view synthesis
- Generative Radiance Fields: **3D-aware generative model** of single-object scenes
- GIRAFFE: Generative Model of **complex, multi-object scenes**

Summary

This research is very activate and leads to state-of-the-art results:



Gu, Jiatao, et al. "StyleNeRF: A Style-based 3D-Aware Generator for High-resolution Image Synthesis." arXiv preprint arXiv:2110.08985 (2021).

Thank you!