Neural Radiance Fields (NeRF)

Capturing Reality

The first photograph in Germany was taken in 1837 - Frauenkirche in Munich.

The photo is only 4 x 4cm large and was taken by Franz von Kobell, two years earlier than previously assumed.

Capturing Reality

First self-portrait Cornelius 1839

First Movie - Muybridge 1878

Capturing Reality – in 3D

Building Rome in a Day, Agarwal et al. ICCV 2009

Capturing Reality – in 3D

Google Earth 2016~

Capturing Reality – in 3D

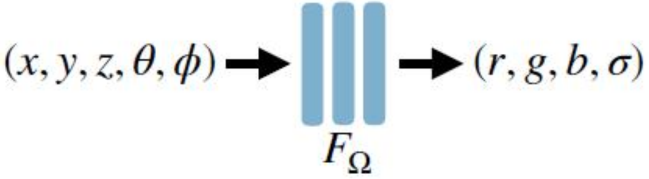
Neural Radiance Fields (NeRF), Mildenhall et al., ECCV 2020

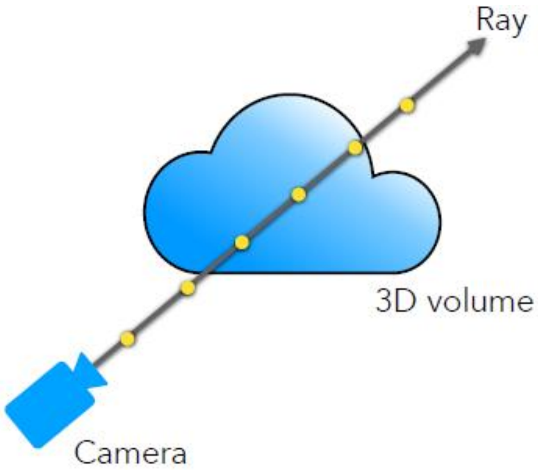
NeRF Problem Statement

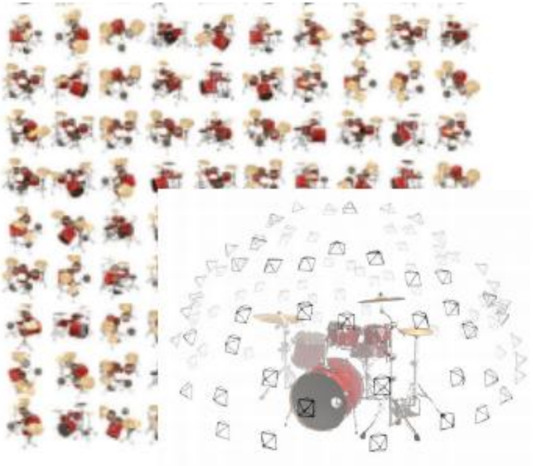
Input:
A set of calibrated RGB images

Output:
A 3D scene representation that renders novel views

Three Key Components







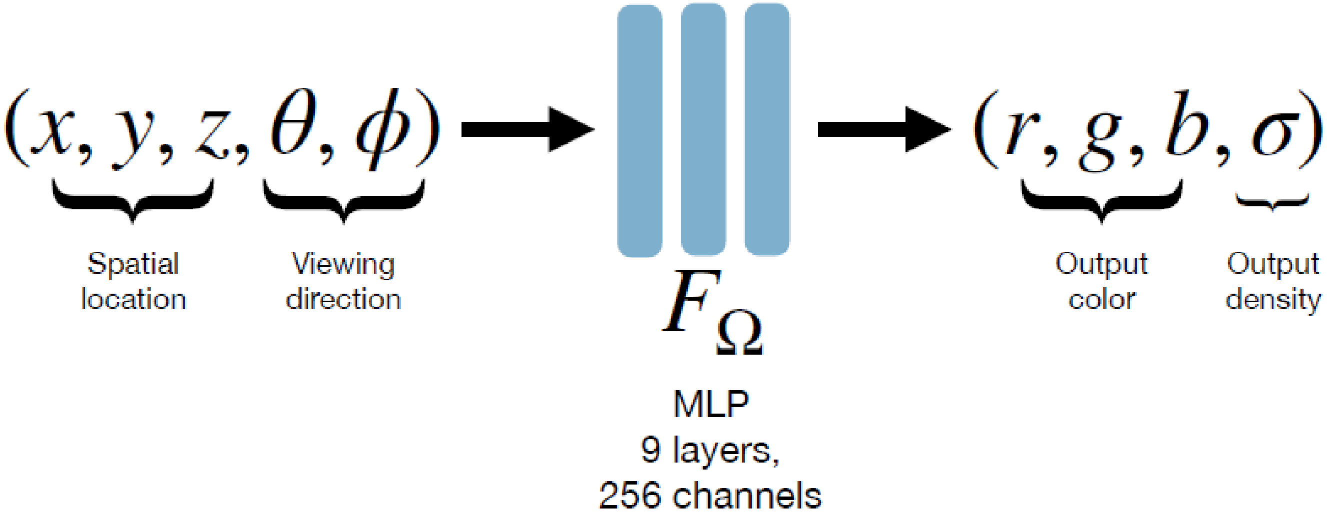
Neural Volumetric 3D Scene Representation

Differentiable Volumetric Rendering Function

Optimization via Analysis-by-Synthesis

Objective: Synthesize all input views

Representing a 3D Scene as a continuous 5D Function



→ The color emitted by every point is composited to render a pixel

Neural Volumetric Rendering

computing color
along rays through 3D
space

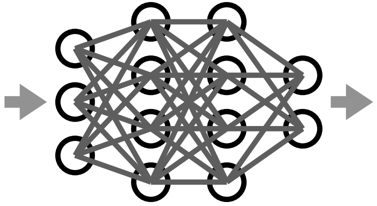
What color is this pixel?

Neural Volumetric Rendering

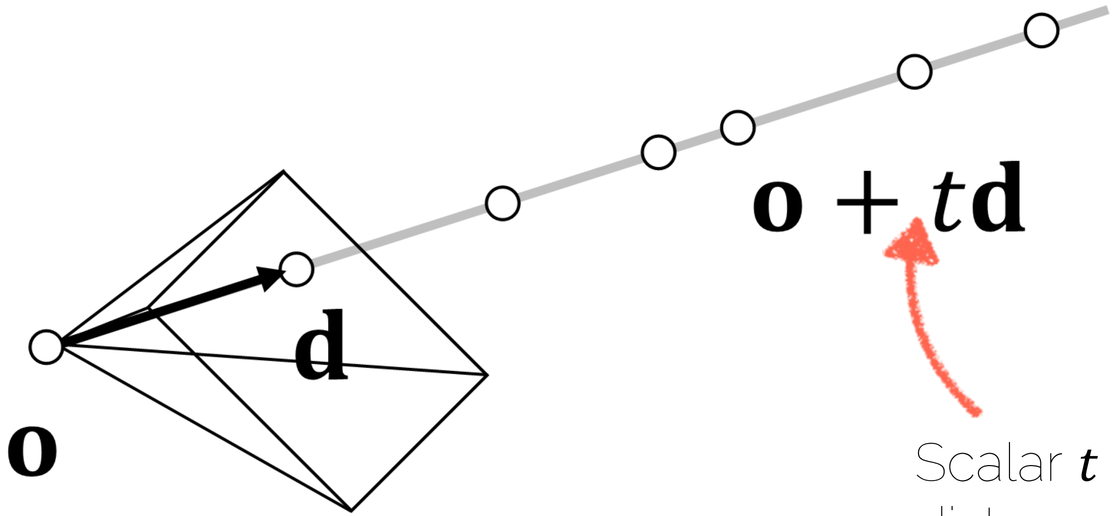
continuous, differentiable
rendering model without
concrete ray/surface
intersections

Neural Volumetric Rendering

using a neural network as a scene representation, rather than a voxel grid of data

(x, y, z)  Scene properties

Calculating Points along a Camera Ray



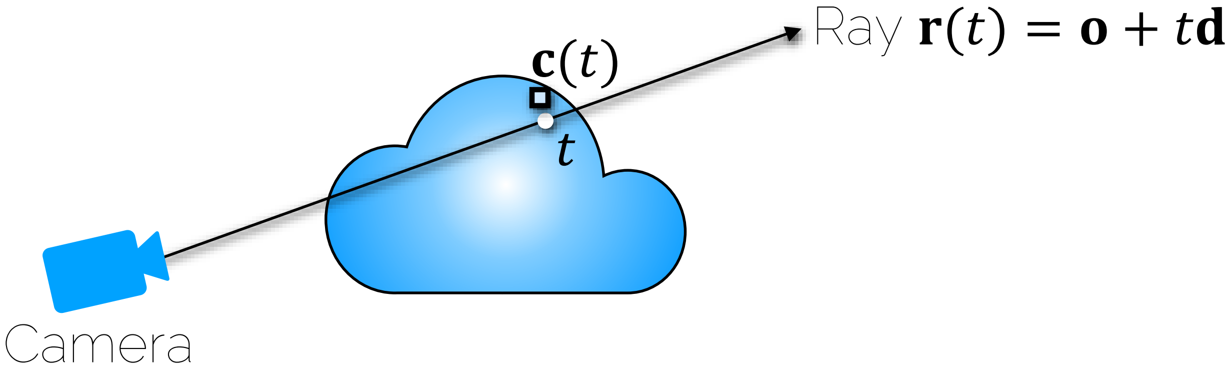
Scalar t controls distance along the ray

Camera with known intrinsics and extrinsics, e.g., from Structure from Motion (SfM)

Volumetric Formulation for NeRF

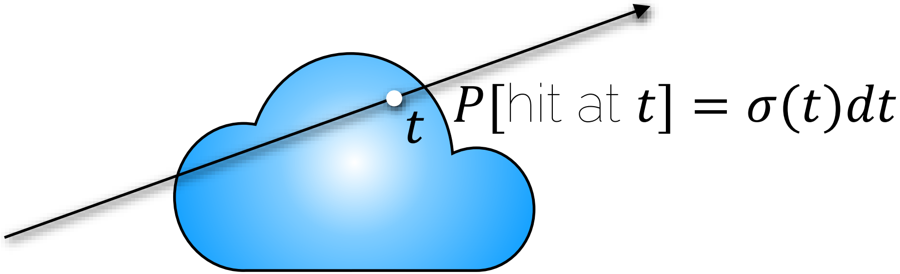
Scene is a cloud of tiny colored particles

Volumetric Formulation for NeRF



If a ray traveling through the scene hits a particle at distance t along the ray, we return its color $\mathbf{c}(t)$

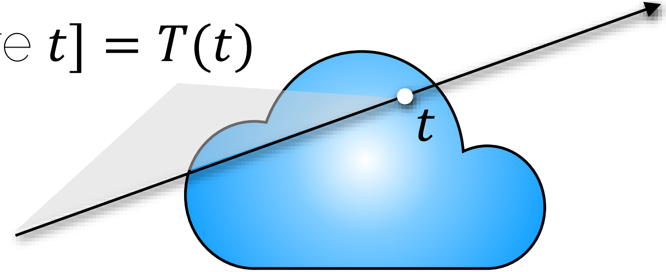
What does it mean for a ray to "hit" the volume?



This notion is *probabilistic*: chance that ray hits a particle in a small interval around t is $\sigma(t)dt$.
σ is called the "volume density"

Probabilistic Interpretation

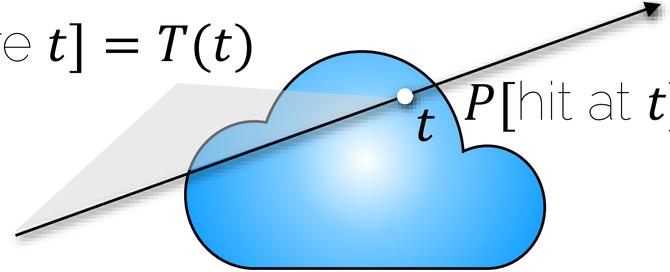
$P[\text{no hits before } t] = T(t)$



To determine if t is the first hit along the ray, need to know $T(t)$: the probability that the ray makes it through the volume up to t.
$T(t)$ is called "transmittance"

Probabilistic Interpretation

$P[\text{no hits before } t] = T(t)$



$P[\text{hit at } t] = \sigma(t)dt$

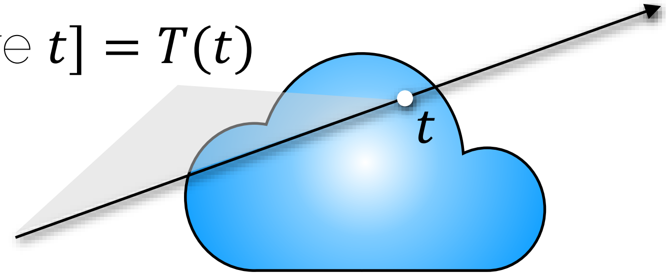
The product of these probabilities tells us how much you see the particles at t:

$P[\text{first hit at } t] = P[\textit{no hit before } t] \times P[\text{hit at } t]$

$= T(t)\sigma(t)dt$

Calculating Transmittance T given σ

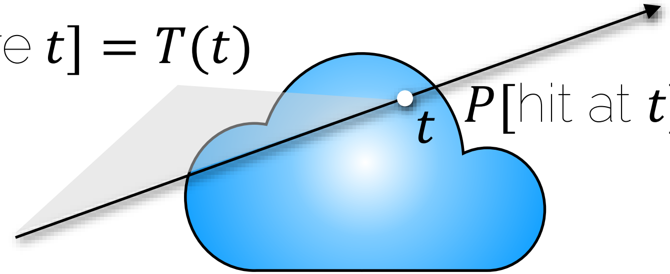
$P[\text{no hits before } t] = T(t)$



If σ is known, T can be computed… How?

Calculating Transmittance T given σ

$P[\text{no hits before } t] = T(t)$



$P[\text{hit at } t] = \sigma(t)dt$

σ and T are related by the probabilistic fact that

$$P[\text{no hit before } t + dt] = P[\text{no hit before } t] \times P[\text{no hit at } t]$$

$$T(t + dt) = T(t) \times (1 - \sigma(t)dt)$$

Solve for T

$$T(t + dt) = T(t)(1 - \sigma(t)dt)$$

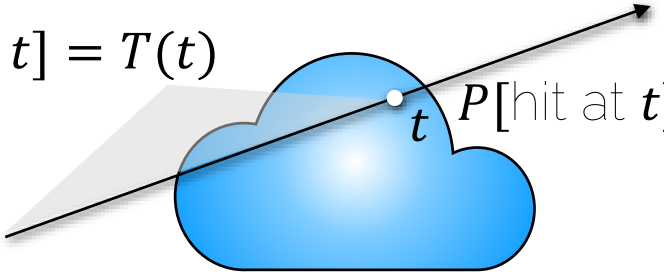
Taylor expansion for $T \Rightarrow T(t) + T'(t)dt = T(t) - T(t)\sigma(t)dt$

Rearrange $\Rightarrow \frac{T'(t)}{T(t)} dt = -\sigma(t)dt$

Integrate $\Rightarrow \log T(t) = -\int_{t_0}^{t} \sigma(s)ds$

Exponentiate $\Rightarrow T(t) = \exp\left(-\int_{t_0}^{t} \sigma(s)ds\right)$

PDF for Ray Termination



Finally, we can write the probability that a ray terminates at t as a function of only sigma:

$$\begin{aligned} P[\text{first hit at } t] &= P[\text{no hit before } t] \times P[\text{hit at } t] \\ &= T(t)\sigma(t)dt \\ &= \exp\left(-\int_{t_0}^{t} \sigma(s)ds\right)\sigma(t)dt \end{aligned}$$

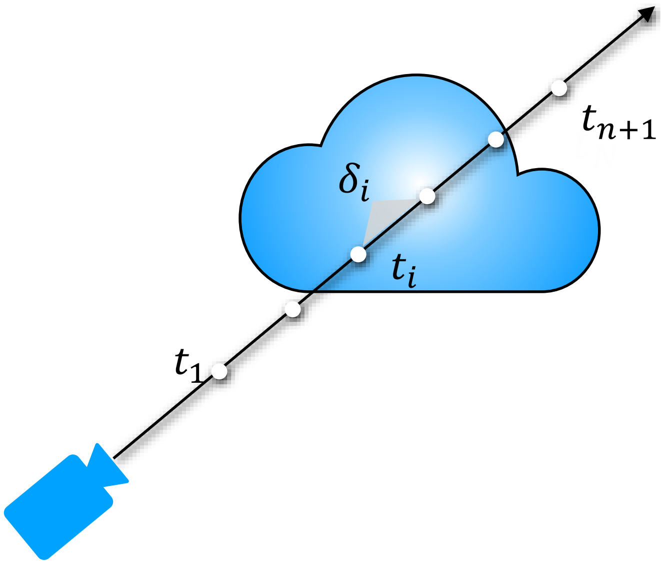
Expected Value of Color along Ray

This means the expected color returned by the ray will be

$$\int_{t_0}^{t_1} T(t)\sigma(t)\mathbf{c}(t)dt$$

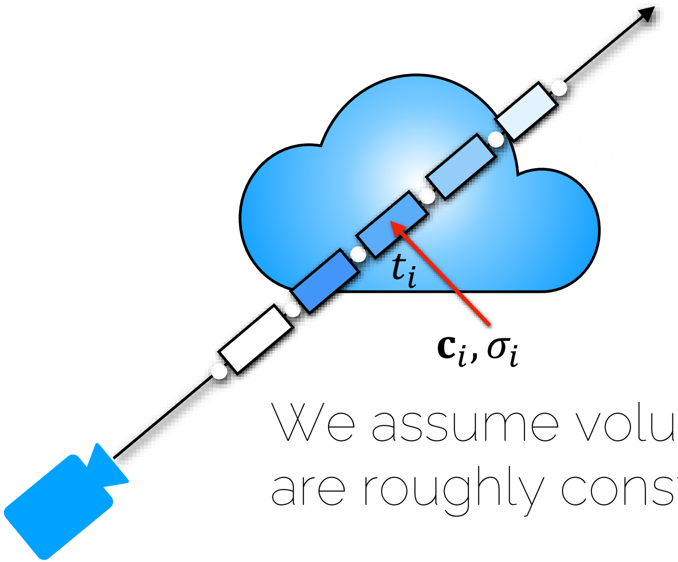
Note the nested integral!

Approximating the nested Integral



We use quadrature to approximate the nested integral, splitting the ray up into n segments with endpoints $\{t_1, t_2, \dots, t_{n+1}\}$ with lengths $\delta_i = t_{i+1} - t_i$

Approximating the nested Integral



We assume volume density and color are roughly constant within each interval.

Approximation with quadrature rule is described in [Max, N 1995] Optical models for direct volume rendering

Resulting Estimate of the Volume Rendering Integral

Rendering model for ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$:

$$\boxed{\mathbf{c} \approx \sum_{i=1}^{n} T_i \alpha_i \mathbf{c}_i}$$

differentiable w.r.t. $\mathbf{c}, \sigma$

colors

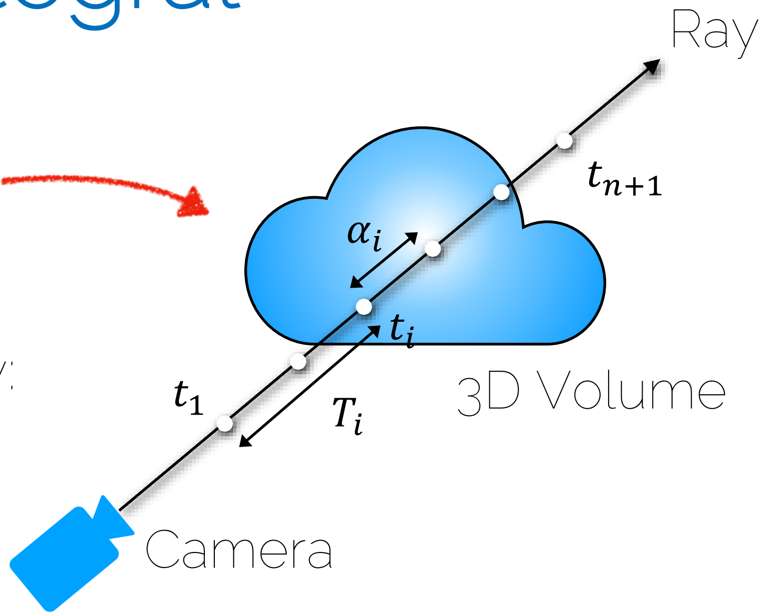
weights

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*:

$$\alpha_i = 1 - \exp(-\sigma_i \delta_i)$$



Rendering weight PDF is important

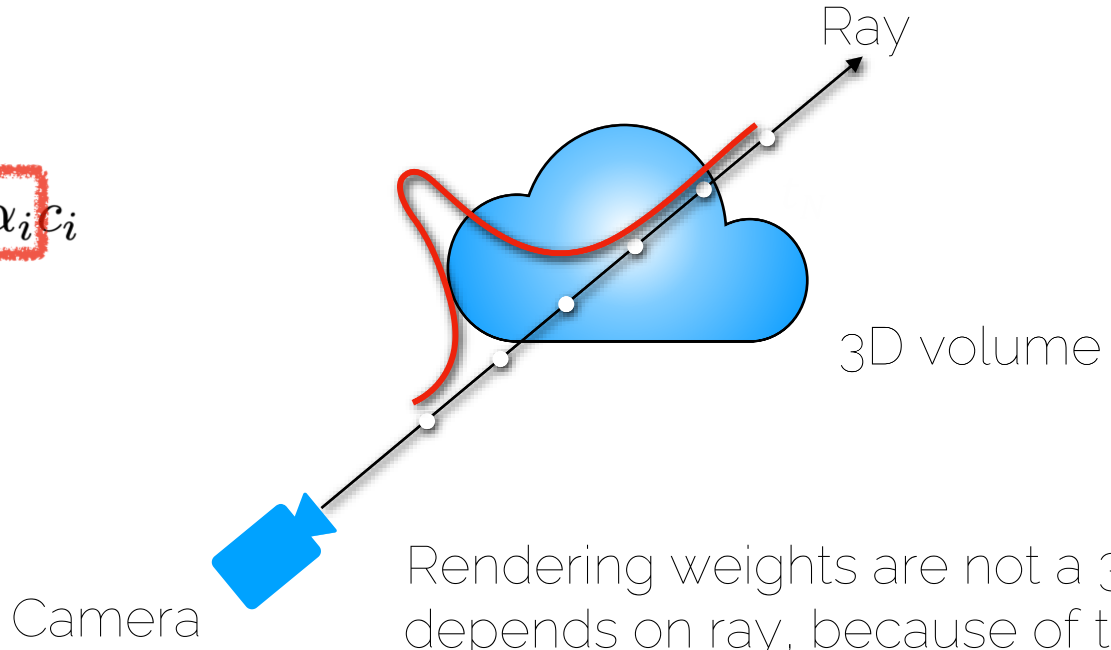
Remember, expected color is equal to

$$\int T(t)\sigma(t)\mathbf{c}(t)dt \approx \sum_i T_i \alpha_i \mathbf{c}_i$$

$T(t)\sigma(t)$ and $T_i\alpha_i$ are "rendering weights" — probability distribution along the ray (continuous and discrete, respectively)

Rendering weight is not just 3D function

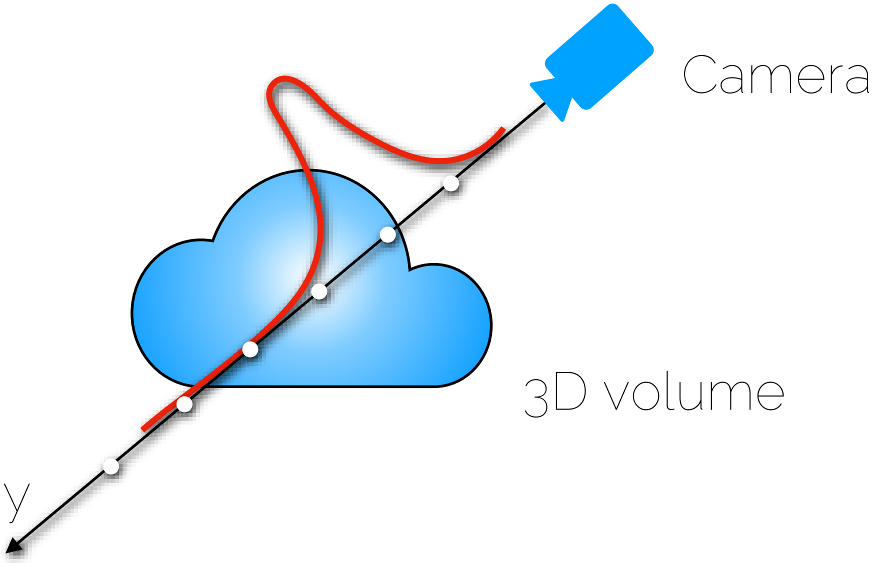
$$C \approx \sum_{i=1}^{N} \boxed{T_i \alpha_i} c_i$$



Rendering weights are not a 3D function —
depends on ray, because of transmittance!

Rendering weight is not just 3D function

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



Rendering weights are not a 3D function — depends on ray, because of transmittance!

Rendering weight PDF is important – Depth

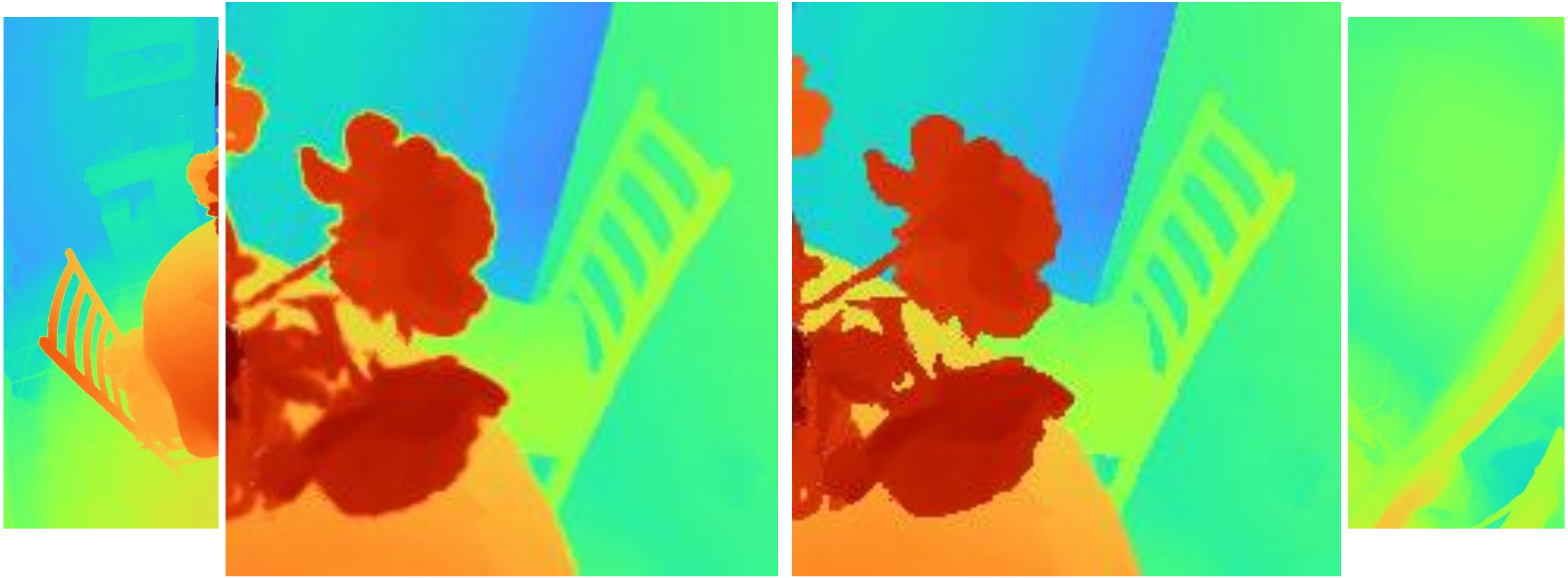
We can use this distribution to compute expectations for other quantities, e.g. "expected depth":

$$\bar{t} = \sum_i T_i \alpha_i t_i$$

This is often how people visualise NeRF depth maps.

Alternatively, other statistics like median can be used.

Rendering weight PDF is important – Depth



Mean depth

Median depth

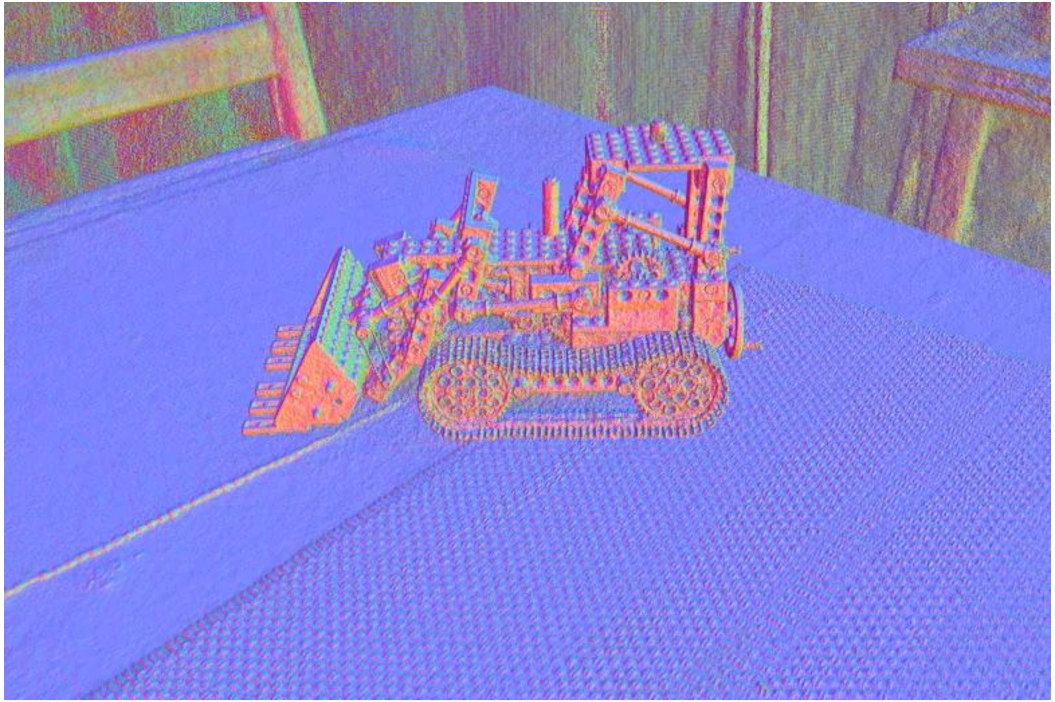
Volume Rendering other Quantities

This idea can be used for any quantity we want to "volume render" into a 2D image. If $\mathbf{v}$ lives in 3D space (semantic features, normal vectors, etc.)

$$\sum_i T_i \alpha_i \mathbf{v}_i$$

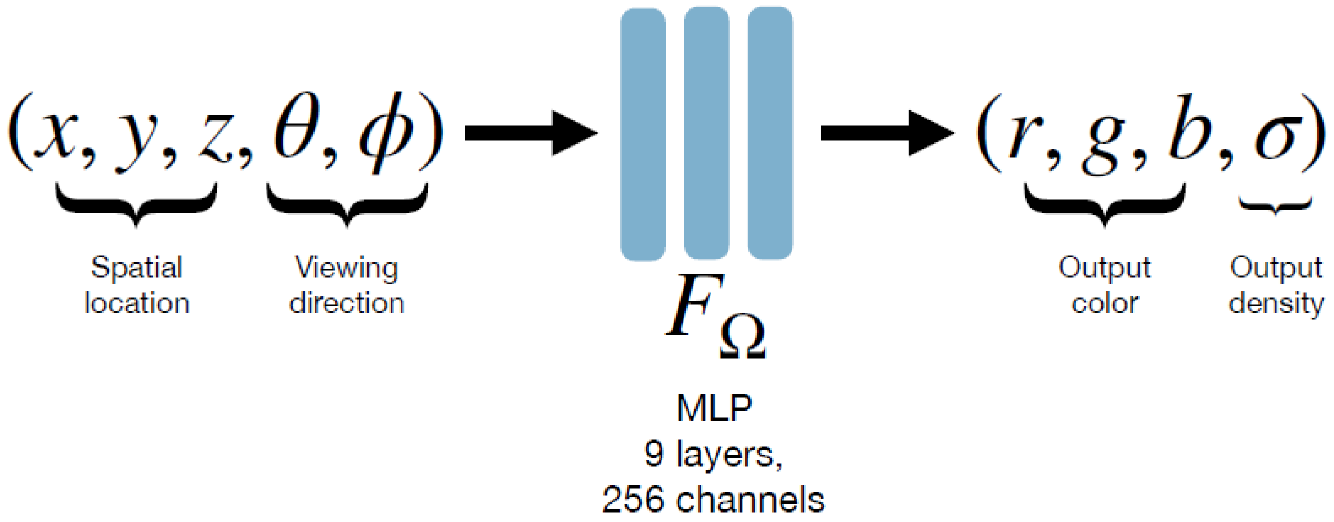
can be taken per-ray to produce 2D output images.

Density as Geometry



Normal vectors (from analytic gradient of density)

Representing a 3D Scene as a continuous 5D Function



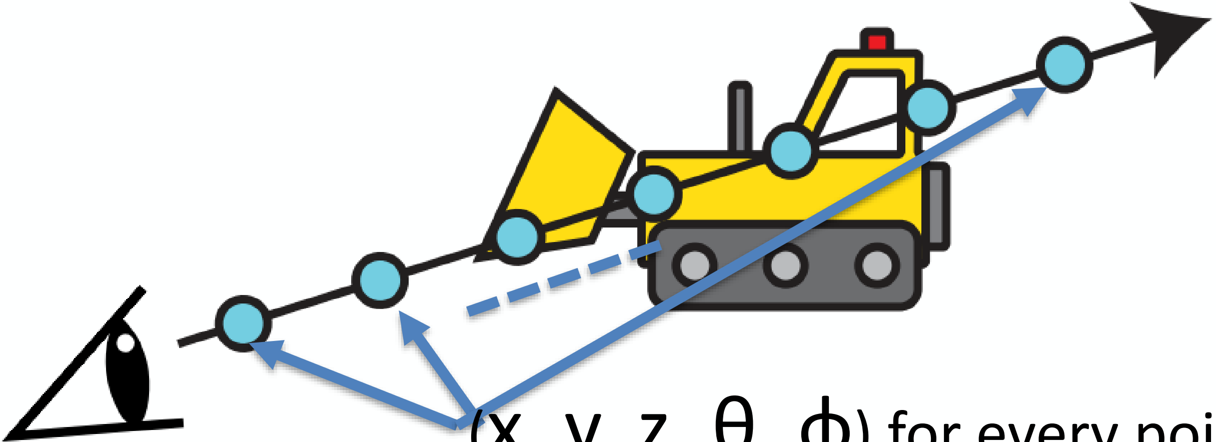
→ The color emitted by every point is composited to render a pixel

NeRF Results

Supervision of Neural Fields

Representation in NeRFs

- Input Domain [See detailed intro in the following course]

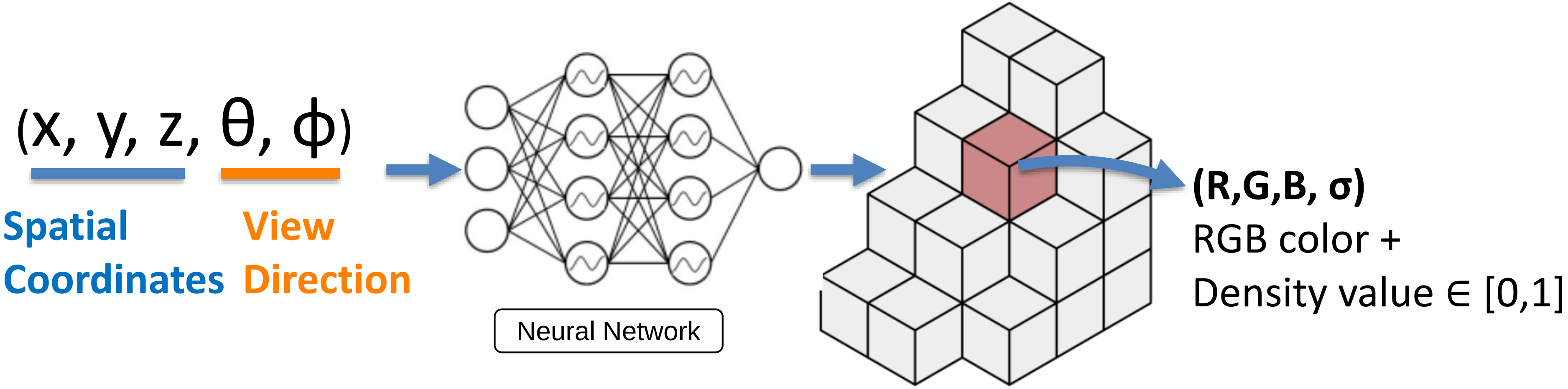


(x, y, z, θ, ϕ) for every point sampled in a ray

Spatial Coordinates **View Direction**

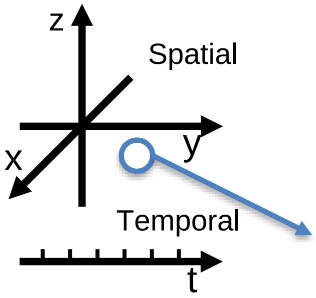
Representation in NeRFs

- Output Domain

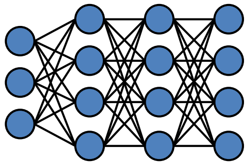


General Architectures

A forward pass

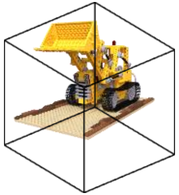


x, y, z,
[t], …



What we want to reconstruct:

Radiance Field



Or Signed Distance / Occupancy Field / …

Coordinate Sampling

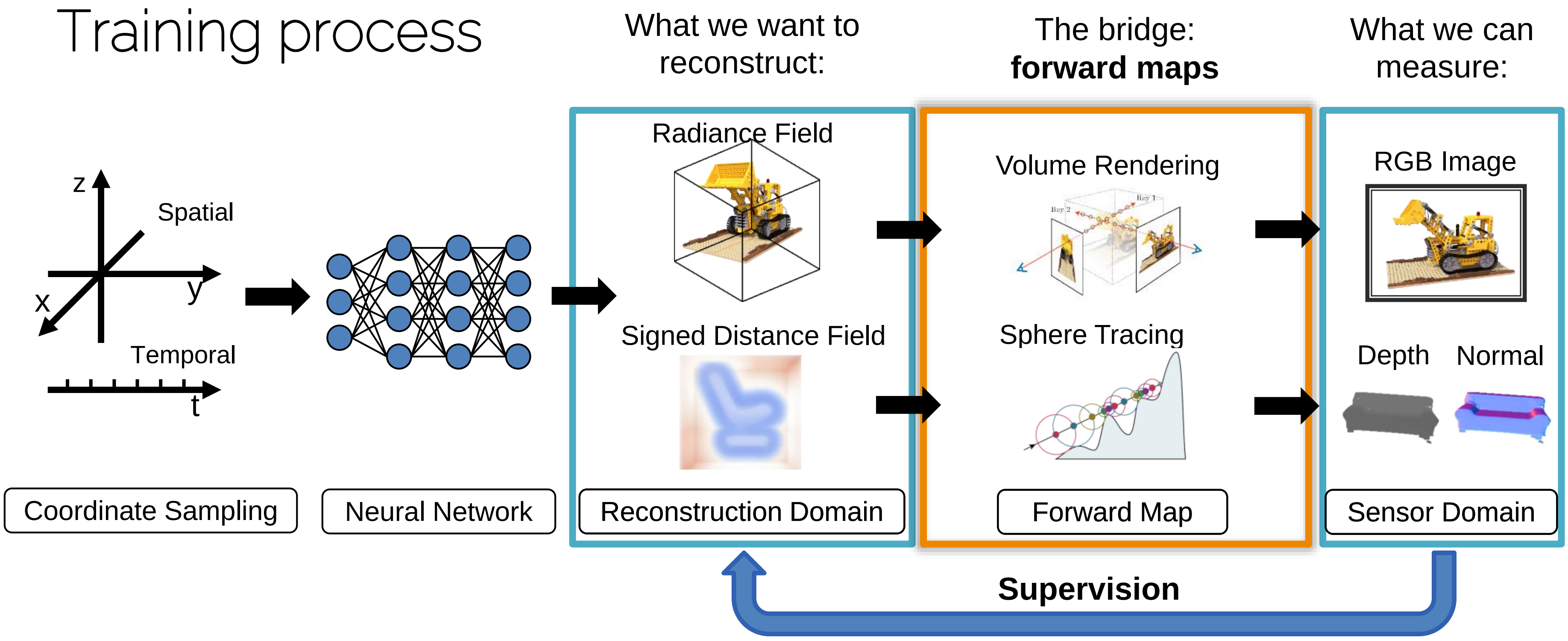
Neural Network

Reconstruction Domain

(R,G,B, σ) for all points
*Density value **σ** ∈ [0,1]

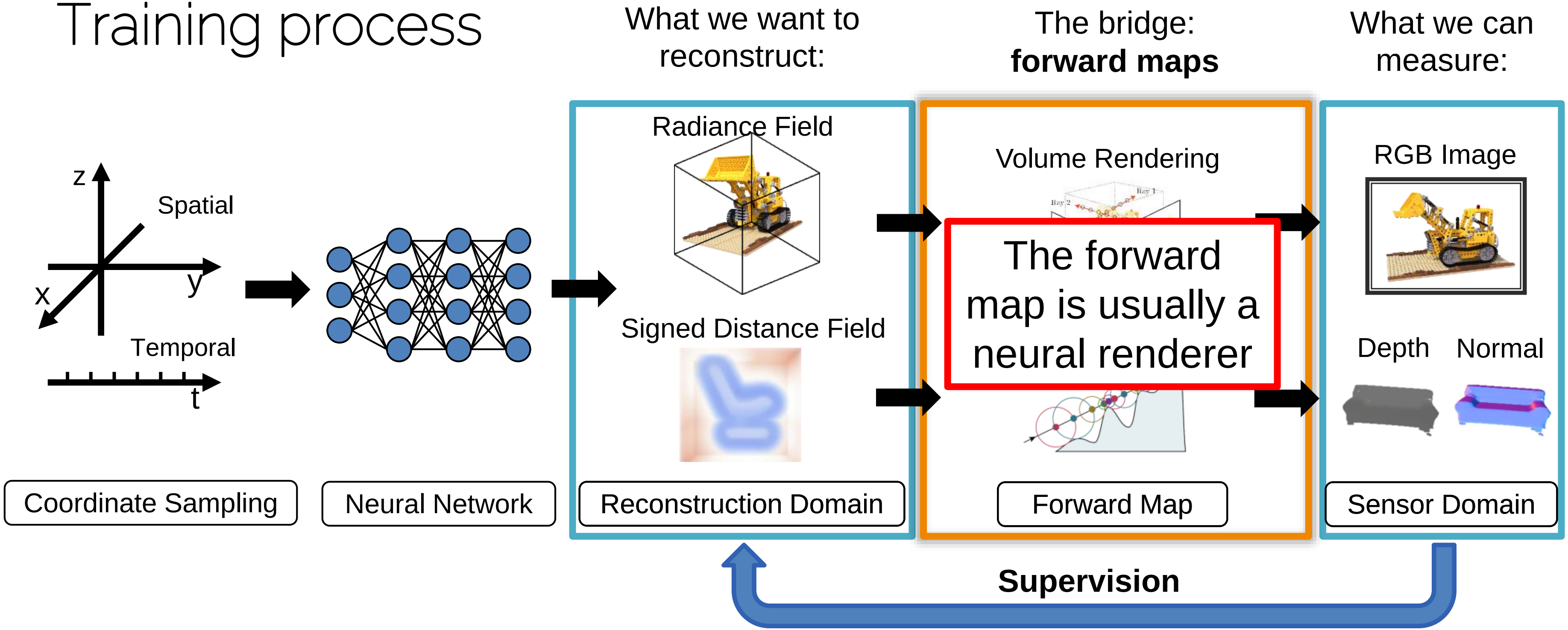
Signed distance /Occupancy /…
for all points

If have 2D supervision

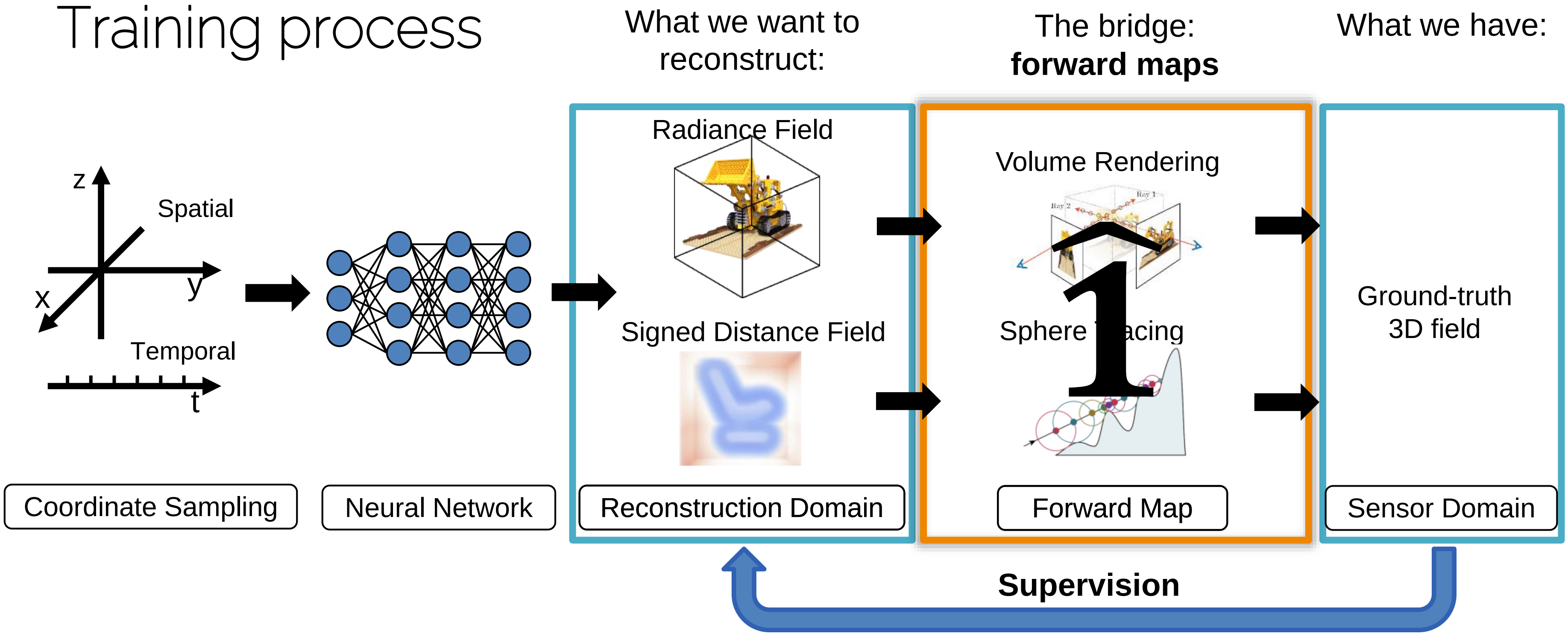


If have 2D supervision

Training process

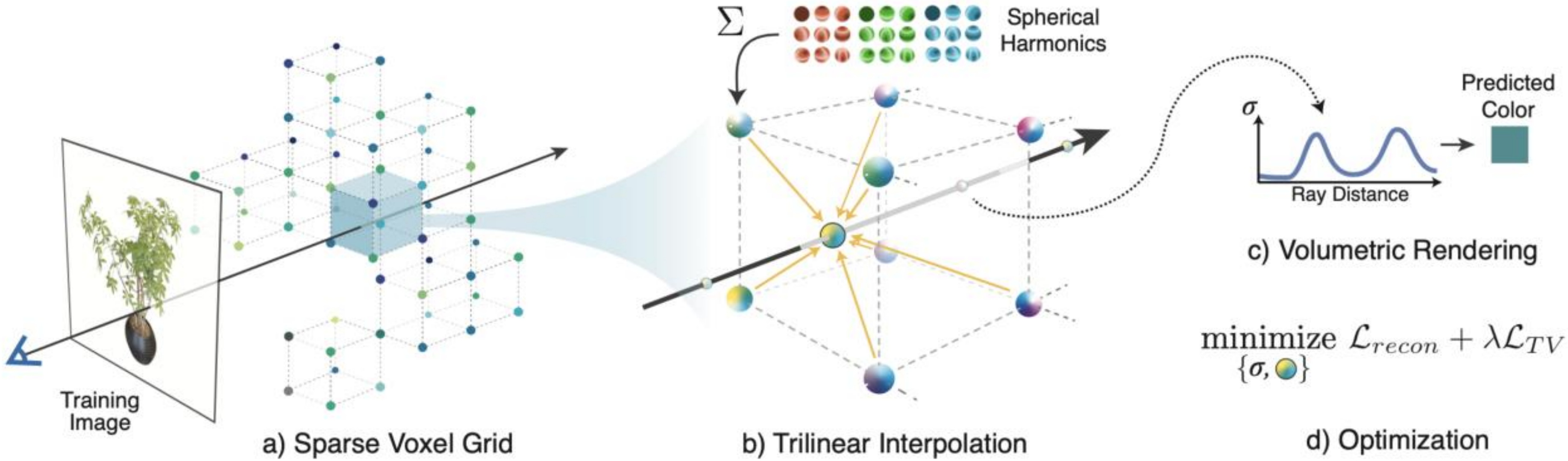


If have 3D supervision



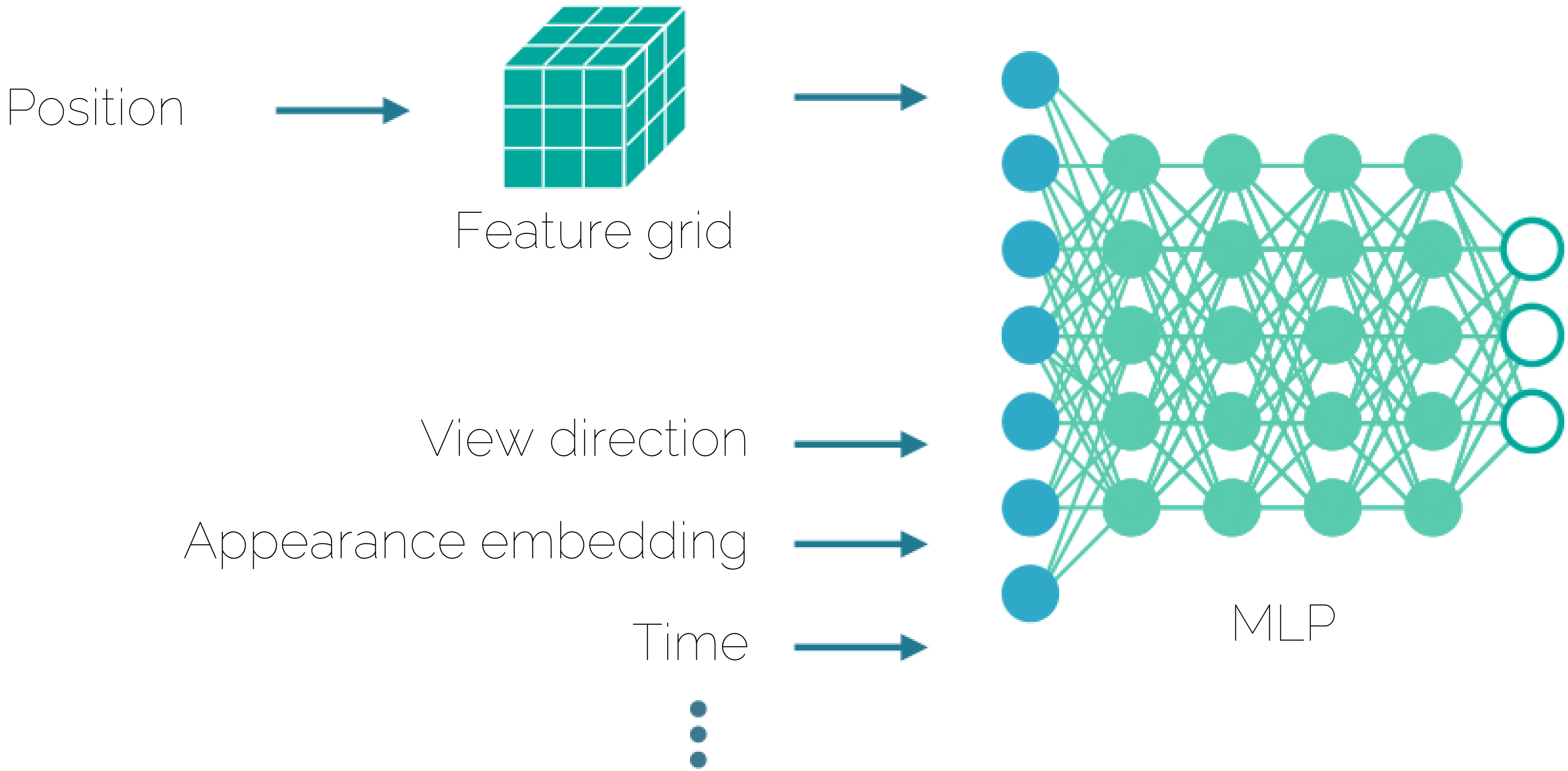
Back to NeRFs

MLPs are not required…



Yu et al. "Plenoxels: Radiance Fields without Neural Networks" CVPR 2022

But MLPs are convenient



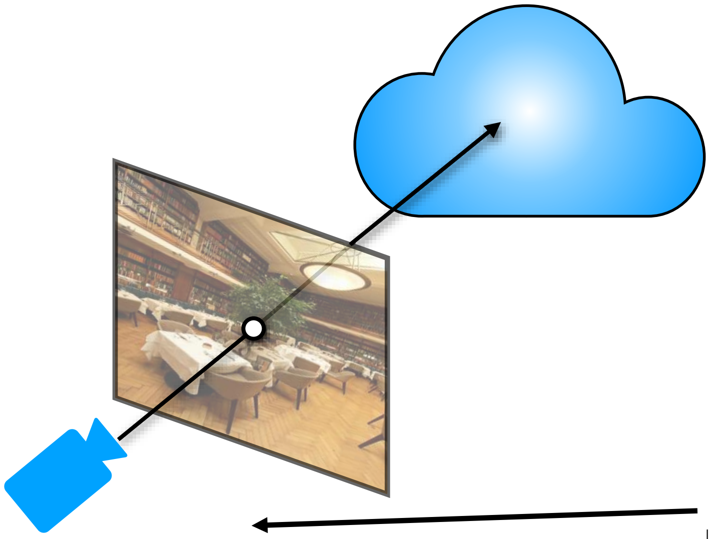
NeRF Challenges

Relaxing the Assumptions

- $\boxed{\text{Unknown or inaccurate camera poses}}$
- Dynamic scene
- Dynamic lighting
- Generalization

Camera Pose Optimization

Small noise in the camera can be made robust by also optimizing the camera



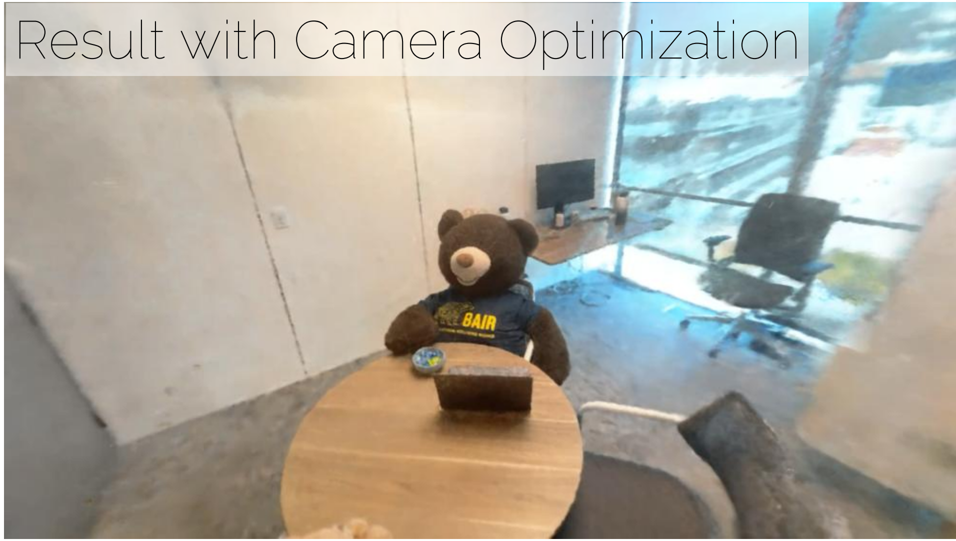
So far we've been optimizing this

Also do backdrop on the camera parameters

Camera

Camera Pose Optimization

Small noise in the results can be improved:



Optimizing poses from scratch is still challenging.

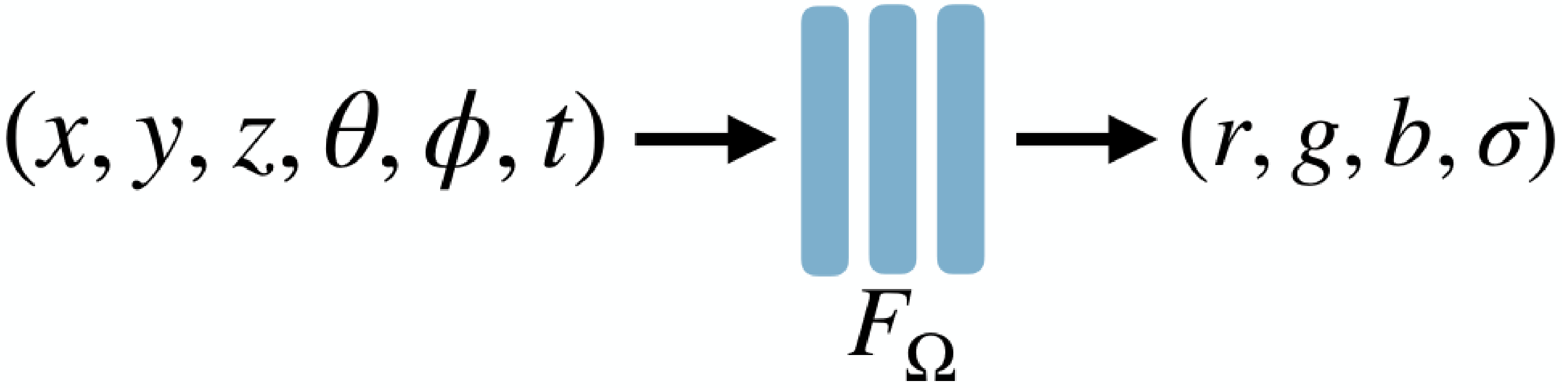
Lin et al. "BARF: Bundle-Adjusting Neural Radiance Fields" ICCV 2021
Wang et al. "NeRF--: Neural Radiance Fields Without Known Camera Parameters" 2021

Relaxing the Assumptions

- Unknown or inaccurate camera poses
- Dynamic scene
- Dynamic lighting
- Generalization

Dynamic Novel View Synthesis

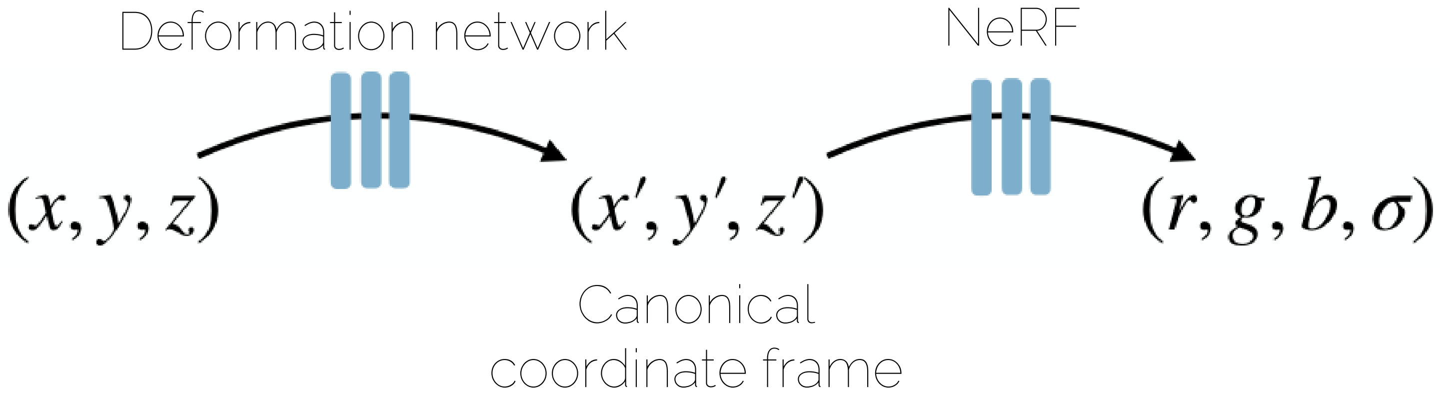
Simple baseline for adding time



Hard without simultaneous multiple view!

Dynamic Novel View Synthesis

Through a deformation network



Still very under constrained

Pumarola et al. "D-NeRF: Neural Radiance Fields for Dynamic Scenes" CVPR 2021
Park et al. "Nerfies: Deformable Neural Radiance Fields" ICCV 2021
Tretschk et al. "Non-Rigid Neural Radiance Fields: Reconstruction and Novel View Synthesis of a Dynamic Scene From Monocular Video" ICCV 2021
Park et al. "A Higher-Dimensional Representation for Topologically Varying Neural Radiance Fields" ACM Trans. Graph. 2021

Dynamic Novel View Synthesis

In-the-wild monocular capture still hard:

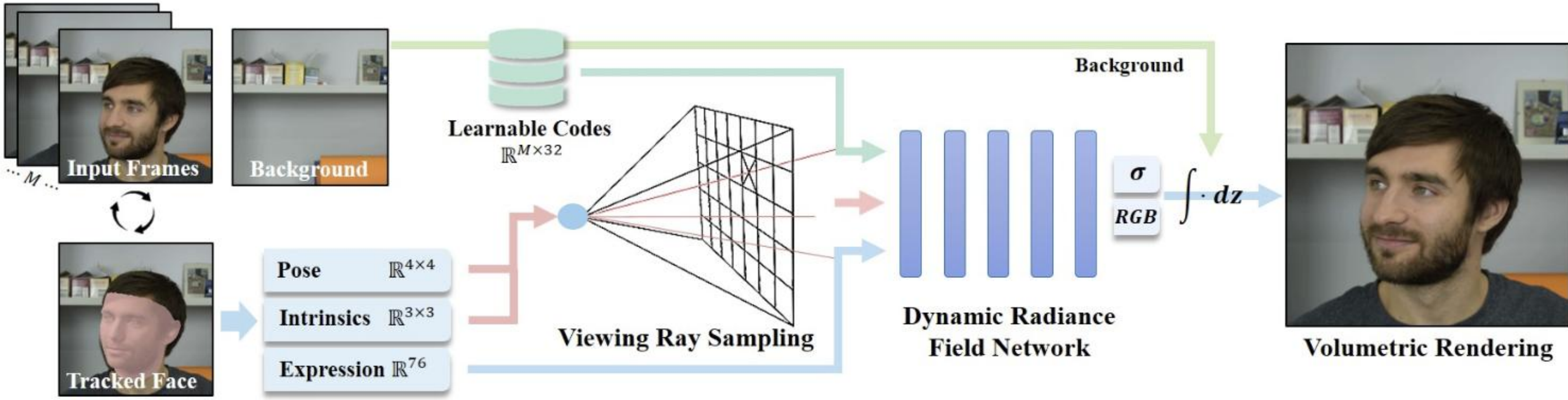
Pumarola et al. "D-NeRF: Neural Radiance Fields for Dynamic Scenes" CVPR 2021
Gao et al. "Monocular Dynamic View Synthesis: A Reality Check" NeurIPS 2022
Park et al. "Nerfies: Deformable Neural Radiance Fields" ICCV 2021

Dynamic Novel View Synthesis

Using prior knowledge about the deformations (e.g., human body model) helps to better constrain the problem:

Weng et al. "HumanNeRF: Free-viewpoint Rendering of Moving People from Monocular Video" CVPR 2021

Dynamic Novel View Synthesis



Gafni et al. "NeRFace: Dynamic Neural Radiance Fields for Monocular 4D Facial Avatar Reconstruction

Relaxing the Assumptions

- Unknown or inaccurate camera poses
- Dynamic scene
- Dynamic lighting
- Generalization

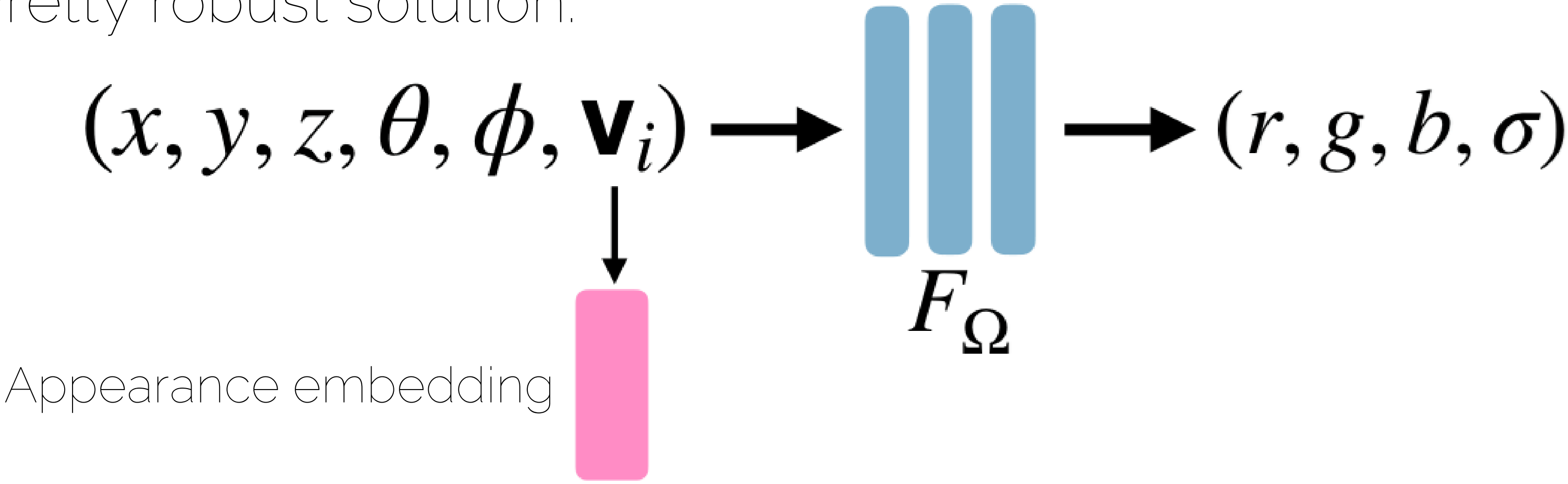
Appearance Changes

- Exposure differences
- Lighting changes (day, night)
- Clouds passing by

Martin-Brualla et al. "NeRF in the Wild Neural Radiance Fields for Unconstrained Photo Collections" CVPR 2021

Appearance Embedding

Pretty robust solution:



N-dim vector optimized *per* image: "Auto-Decoding"
i.e. GLO: Generative Latent Optimization [Bojanowski et al. ICML 2018]

Martin-Brualla et al. "NeRF in the Wild Neural Radiance Fields for Unconstrained Photo Collections" CVPR 2021

Appearance Embedding Interpolation

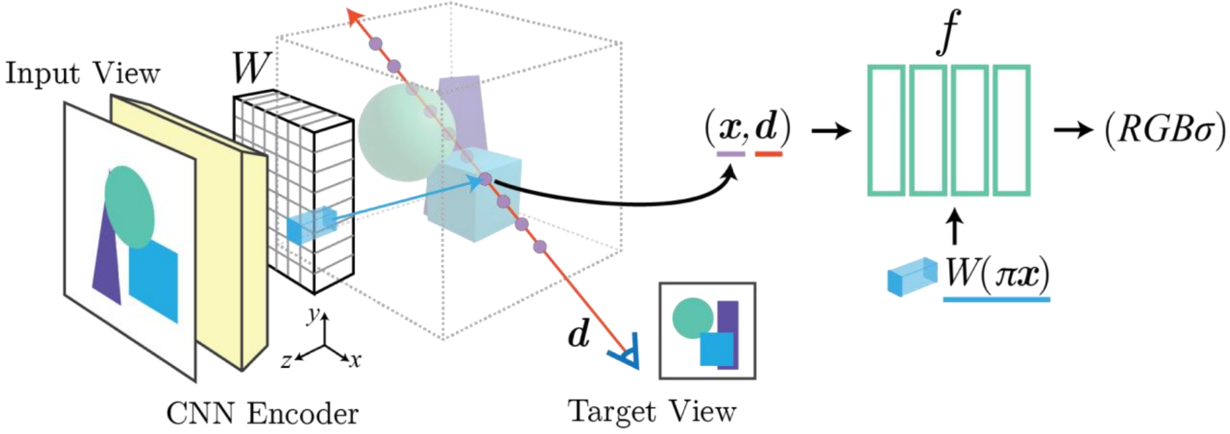
Martin-Brualla et al. "NeRF in the Wild Neural Radiance Fields for Unconstrained Photo Collections" CVPR 2021

Relaxing the Assumptions

- Unknown or inaccurate camera poses
- Dynamic scene
- Dynamic lighting
- $\boxed{\text{Generalization}}$

Generalization - Few-shot NeRF

- One-shot (single-view): pixelNeRF



3 input views:

- Few-shot (3~10 views): pixelNeRF, MVSNet, …

- Challenging for predicting completely unseen real scenes

Yu et al. "pixelNeRF: Neural Radiance Fields from One or Few Images" CVPR 2021
Chen et al. "MVSNeRF: Fast Generalizable Radiance Field Reconstruction from Multi-View Stereo" ICCV 2021

NeRF vs 3D Meshes

[Neural Radiance Fields] vs [3D Meshes]

Reconstruction:
- Optimization-based
- Need good gradients
- Not just surface rep!

Rendering:
- Need only surface
- Fast access / efficient
- No gradients needed

What's beyond NeRF?

3D Gaussian Splatting

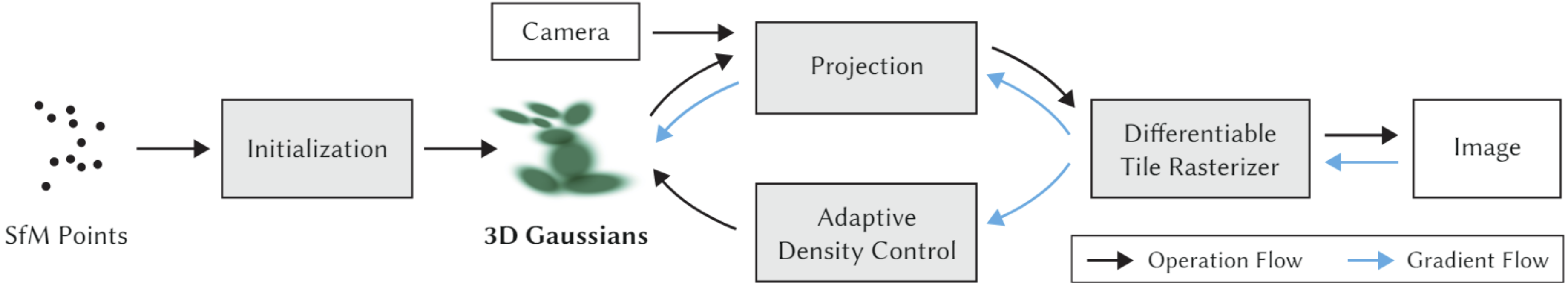


Fig. 2. Optimization starts with the sparse SfM point cloud and creates a set of 3D Gaussians. We then optimize and adaptively control the density of this set of Gaussians. During optimization we use our fast tile-based renderer, allowing competitive training times compared to SOTA fast radiance field methods. Once trained, our renderer allows real-time navigation for a wide variety of scenes.

[Kerbl et al 2023, Siggraph] 3D Gaussian Splatting for Real-time Radiance Field Rendering

3D Gaussian Splatting

[Kerbl et al 2023, Siggraph] 3D Gaussian Splatting for Real-time Radiance Field Rendering

Reading Homework

- [Mildenhall et al. 2020] NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis
  - https://arxiv.org/pdf/2003.08934.pdf

- [Müller at al. 2022] Instant Neural Graphics Primitives with a Multiresolution Hash Encoding
  - https://arxiv.org/pdf/2201.05989.pdf

- [Kerbl et al 2023, Siggraph] 3D Gaussian Splatting for Real-time Radiance Field Rendering
  - https://arxiv.org/pdf/2308.04079.pdf

Literature

[Mildenhall et al. 2020] NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis

- Volume rendering
  - [Max, N 1995] Optical models for direct volume rendering
- Positional encoding
  - [Tancik at al. 2020] Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains

Literature

- NeRF compression techniques, fast NeRFs
  - [Lui et al. 2020] Neural Sparse Voxel Fields
  - [Yu et al. 2022] Plenoxels: Radiance Fields without Neural Networks
  - [Sun et al. 2022] Direct Voxel Grid Optimization: Super-fast Convergence for Radiance Fields Reconstruction
  - [Chen et al. 2022] TensoRF Tensorial Radiance Fields
  - [Müller et al. 2022] Instant Neural Graphics Primitives with a Multiresolution Hash Encoding

Literature

- NeRF for unbounded scenes
  - [Barron et al. 2022] Mip-NeRF 360: Unbounded Anti-Aliased Neural Radiance Fields
- NeRF with unknown camera poses
  - [Lin et al. 2021] BARF: Bundle-Adjusting Neural Radiance Fields
  - [Wang et al. 2021] NeRF--: Neural Radiance Fields Without Known Camera Parameters

Literature

- Dynamic NeRFs
  - [Pumarola et al. 2021] D-NeRF: Neural Radiance Fields for Dynamic Scenes
  - [Park et al. 2021] Nerfies: Deformable Neural Radiance Fields
  - [Tretschk et al. 2021] Non-Rigid Neural Radiance Fields: Reconstruction and Novel View Synthesis of a Dynamic Scene From Monocular Video
  - [Park et al. 2021] A Higher-Dimensional Representation for Topologically Varying Neural Radiance Fields
  - [Gao et al. 2022] Monocular Dynamic View Synthesis: A Reality Check

Literature

- NeRF with appearance embedding
  - [Martin-Brualla et al. 2021] NeRF in the Wild Neural Radiance Fields for Unconstrained Photo Collections
- Generalizable NeRFs, few-shot NeRFs
  - [Yu et al. 2021] pixelNeRF: Neural Radiance Fields from One or Few Images
  - [Chen et al. 2021] MVSNeRF: Fast Generalizable Radiance Field Reconstruction from Multi-View Stereo

Thanks for watching!

Some Slides adapted from…

- NeRF Tutorial ECCV 2022, Matt Tancik, Ben Mildenhall, Pratul Srinivasan, Jon Barron, Angjoo Kanazawa
  - https://sites.google.com/berkeley.edu/nerf-tutorial/home