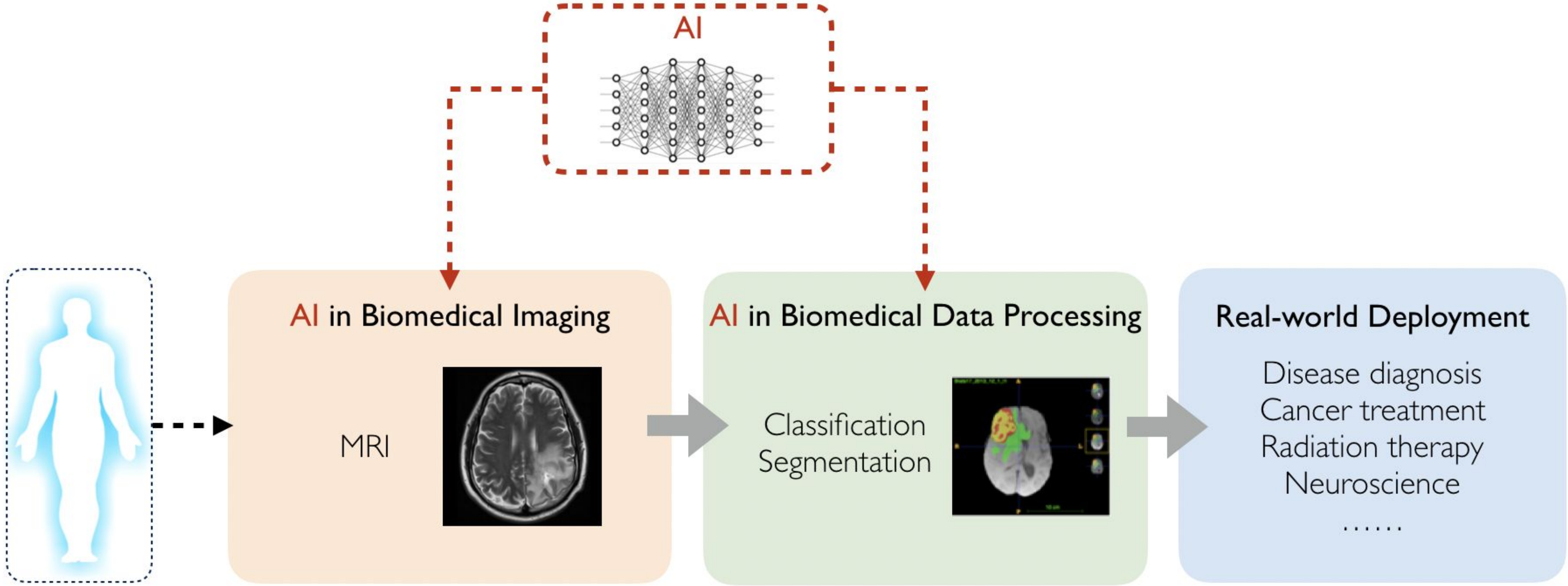
Lecture 3: Implicit Neural Representation Learning

Announcements

- Paper bidding:
  - Paper reading list
  - Complete google form by 11:59 pm EDT on September 8
  - 16 students have submitted response until this Monday
- Great lakes computing credits will be available for course project
  - More details coming soon!
- Lecture notes
  - Will be uploaded before the lecture
  - Will be updated after the lecture
- Lecture recording
  - Will be updated this week

In this class:

- Part I: AI in biomedical imaging
- Part II: AI in biomedical data processing



Last lecture

Biomedical Imaging with Deep Learning

- Motivation and physics background
- Conventional reconstruction method
- Deep learning-based reconstruction method
- Physics-informed learning
- Challenges

This lecture

- Implicit neural representation learning

Date	Lecture #	Topic	Papers	Instructor / Presenter
Tue 8/29	1	Introduction and course overview		Liyue Shen
Thu 8/31	2	Biomedical imaging with deep learning **[Fundamental]**		Liyue Shen
Tue 9/5	3	Implicit neural representation learning **[Advanced]**		Liyue Shen
Thu 9/7	4	Generative diffusion models **[Advanced]**		Liyue Shen
Tue 9/12	5	Medical image analysis **[Fundamental]**		Liyue Shen
Thu 9/14	6	Multimodal foundation models **[Advanced]**		Liyue Shen
Mon 9/18		Drop/add deadline for full term classes		
Tue 9/19	7	Implicit neural representation learning		
Thu 9/21	8	Implicit neural representation learning		
Tue 9/26	9	Implicit neural representation learning		
Thu 9/28	10	Implicit neural representation learning		
Tue 10/3	11	Generative diffusion models		
Thu 10/5	12	Generative diffusion models		
Tue 10/10	13	Generative diffusion models		
Thu 10/12	14	Generative diffusion models		
Tue 10/17		No class (fall study break)		
Thu 10/19	15	Self-supervised learning		
Tue 10/24	16	Self-supervised learning		
Thu 10/26	17	Multimodal learning		
Tue 10/31	18	Multimodal learning		
Thu 11/2	19	Transformer and LLM		
Tue 11/7	20	Transformer and LLM		

Today's agenda

- Neural scene representation
- Neural radiance fields (NeRF)
- Implicit neural representation learning (INR)
- INR in biomedical imaging
- Challenges

Today’s agenda

- Neural scene representation
- Neural radiance fields (NeRF)
- Implicit neural representation learning (INR)
- INR in biomedical imaging
- Challenges

Neural scene representation

- Motivation: How human understand a visual scene?
  - Our brains draw on prior knowledge to reason and to make inferences that go far beyond the patterns of light that hit our retinas

Eslami, et al., Neural scene representation and rendering, Science 2018.

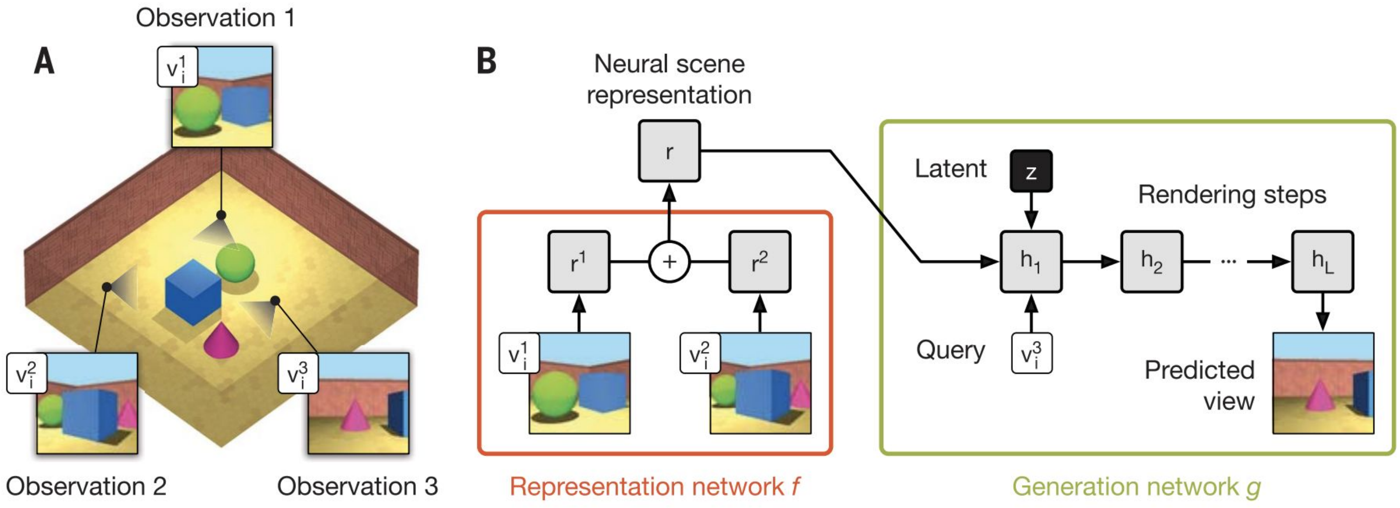
Neural scene representation

- Motivation: How human understand a visual scene?
  - Our brains draw on prior knowledge to reason and to make inferences that go far beyond the patterns of light that hit our retinas
  - Inference for unobserved viewpoints!

Eslami, et al., Neural scene representation and rendering, Science 2018.

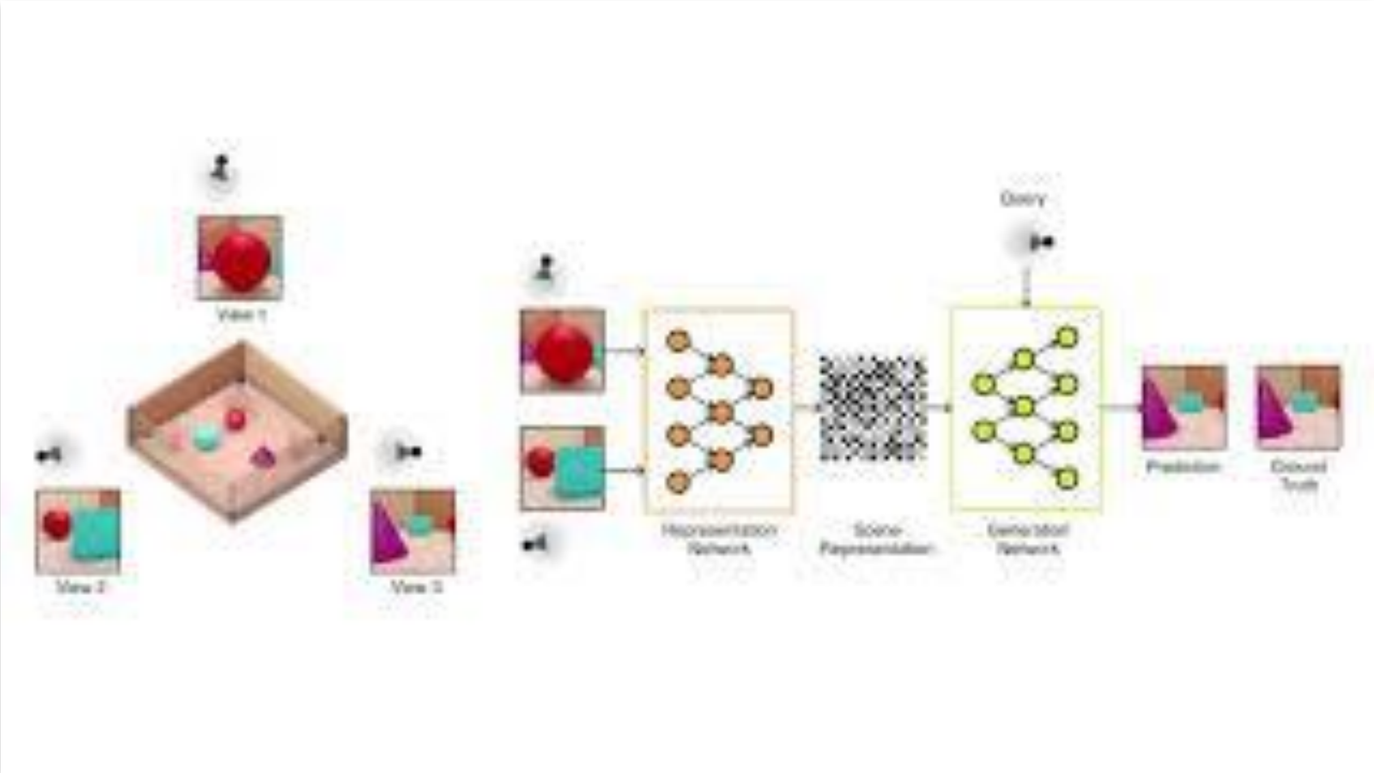
Neural scene representation

- Generative query network (GQN)
  - Machines learn to represent scenes using only their own sensors
  - Takes as input images of a scene taken from different viewpoints, constructs an internal representation, and uses this representation to predict the appearance of that scene from previously unobserved viewpoints



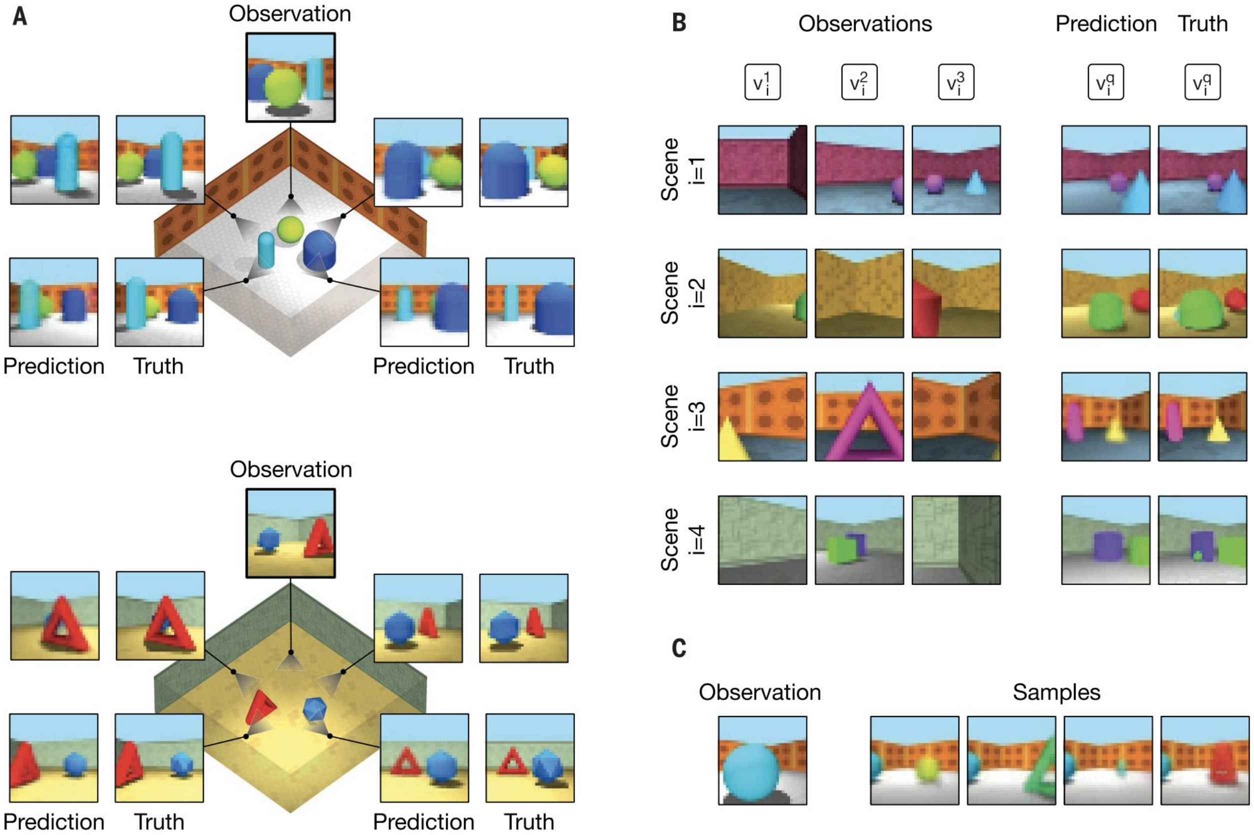
Neural scene representation

- Generative query network (GQN)



Neural scene representation

- Generative query network (GQN)
  - Single observation of a previously unseen test scene
  - Aggregate information from multiple views of different parts of the sce
  - Uncertainty owing to object occlusion



Neural scene representation

- Limitations:
  - Trained on synthetic scenes
  - Only low-resolution images

Eslami, et al., Neural scene representation and rendering, Science 2018.

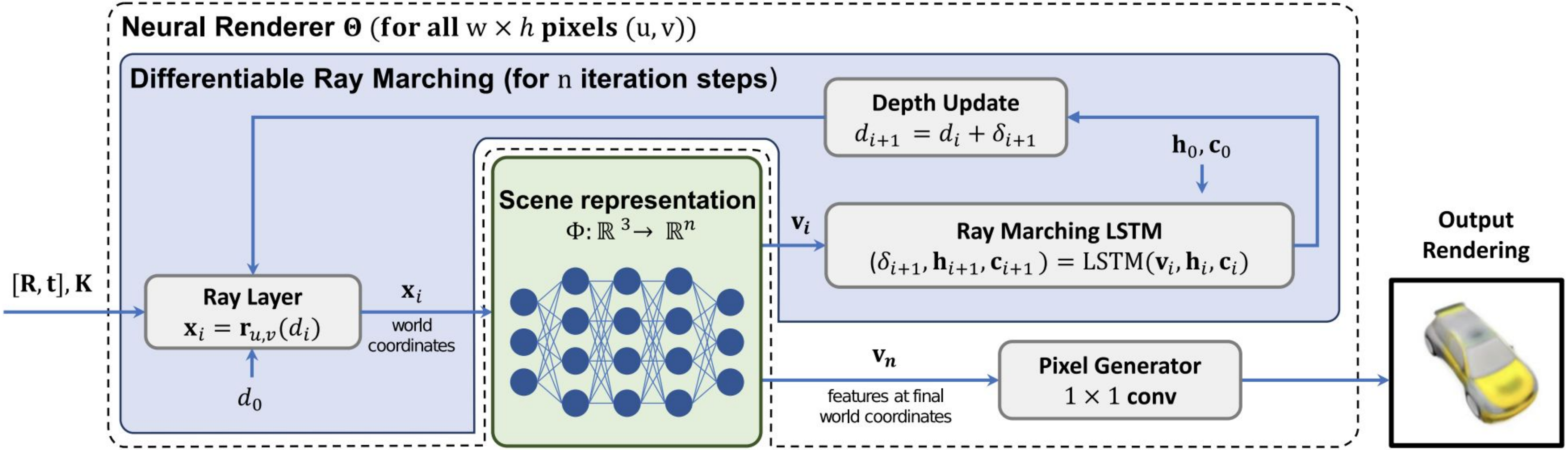
Neural scene representation

- Scene representation network (SRN)

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

Neural scene representation

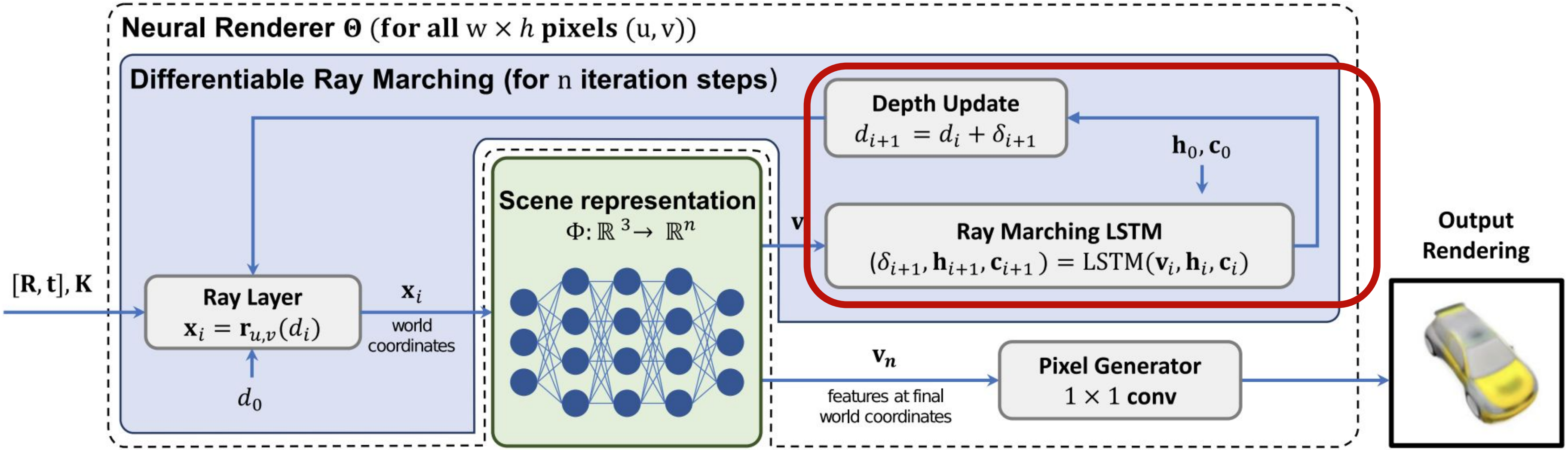
- Scene representation network (SRN)
  - A continuous, 3D structure-aware scene representation that encodes both geometry and appearance
  - Represent scenes as continuous functions that map world coordinates to a feature representation of local scene properties



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

Neural scene representation

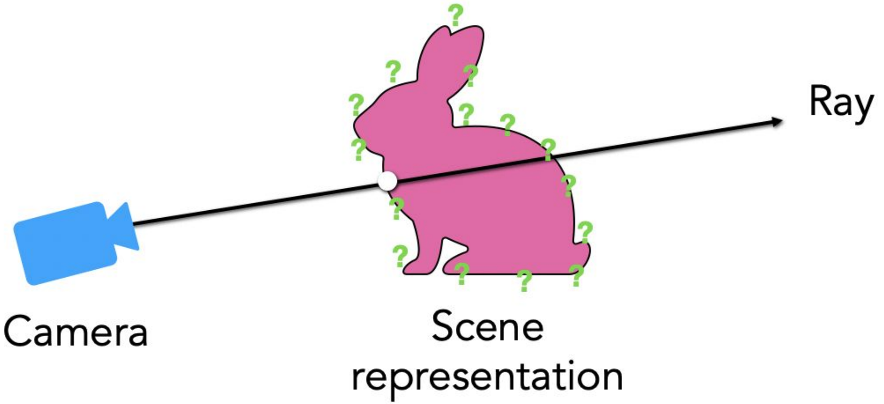
- Ray marching LSTM
  - Find the world coordinates of the intersections of the respective camera rays with scene geometry
  - Predict step-length along ray direction in each iteration



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

Neural scene representation

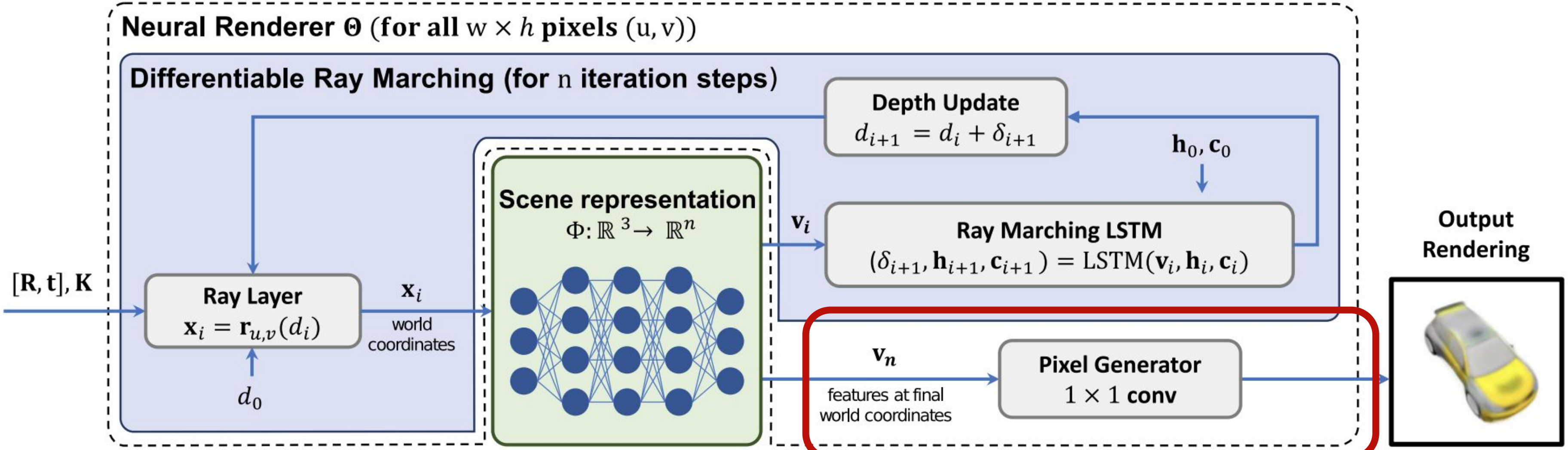
- Ray marching LSTM
  - Find the world coordinates of the intersections of the respective camera rays with scene geometry
  - Predict step-length along ray direction in each iteration



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

Neural scene representation

- Pixel generator
  - Take as input the 2D feature at world coordinates of ray-surface intersections and map it to an estimate of the observed image



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

Neural scene representation

- Generalize across scenes
  - Hypernetwork: a neural network that regresses the parameters of another neural network
  - Shared across scenes:
    - Neural rendering function: $\boldsymbol{\Theta}$
    - Hypernetwork: $\boldsymbol{\Psi}$
  - Scene-specific variables:
    - Latent code: z_j
    - Scene representation network: $\boldsymbol{\phi}_j$
  - Joint optimization during training

$$\Psi : \mathbb{R}^k \to \mathbb{R}^l, \quad \mathbf{z}_j \mapsto \Psi(\mathbf{z}_j) = \phi_j$$

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

Neural scene representation

- SRN outperforms GQN in view synthesis
  - Training set: 50 observations
  - Testing set: 1 or 2 observations

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

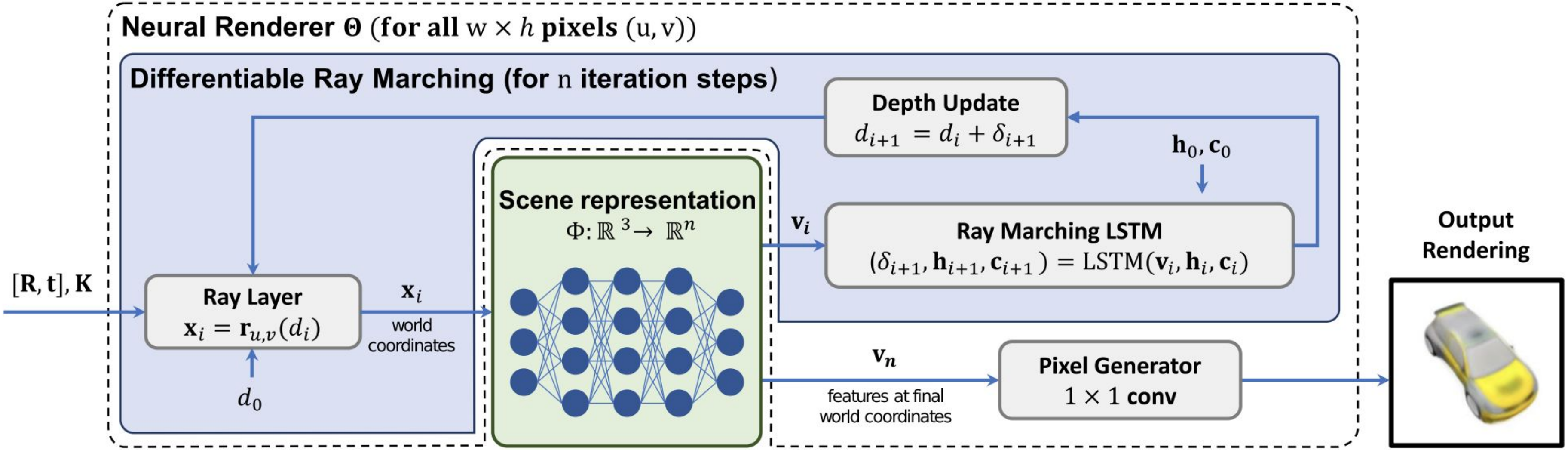
Neural scene representation

- Few-shot reconstruction
  - Fix Neural rendering function: $\boldsymbol{\Theta}$ and Hypernetwork: $\boldsymbol{\Psi}$
  - Minimize latent code: z

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

Neural scene representation

- Limitations for both GQN and SRN:
  - Small dataset or objects
  - Synthetic objects or scenes
  - Low-resolution images

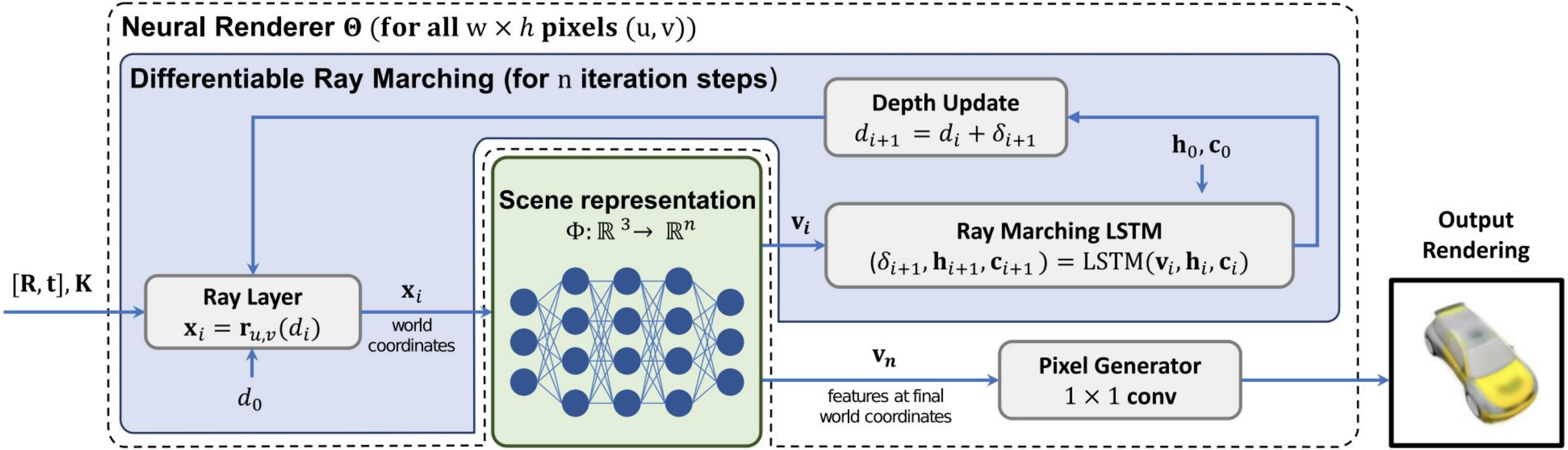


Sitzmann, et al., Scene Representation Networks: Continuous 3D-Structure-Aware Neural Scene Representations, NeurIPS 2019.

Neural scene representation

- Limitations
  - Small dataset or objects
  - Synthetic objects or scenes
  - Low-resolution images

How to apply to high-resolution images of real scenes?



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

Today's agenda

- Neural scene representation
  - Generative query network (GQN)
  - Scene representation network (SRN)
- Neural radiance fields (NeRF)
- Implicit neural representation learning (INR)
- INR in biomedical imaging
- Challenges

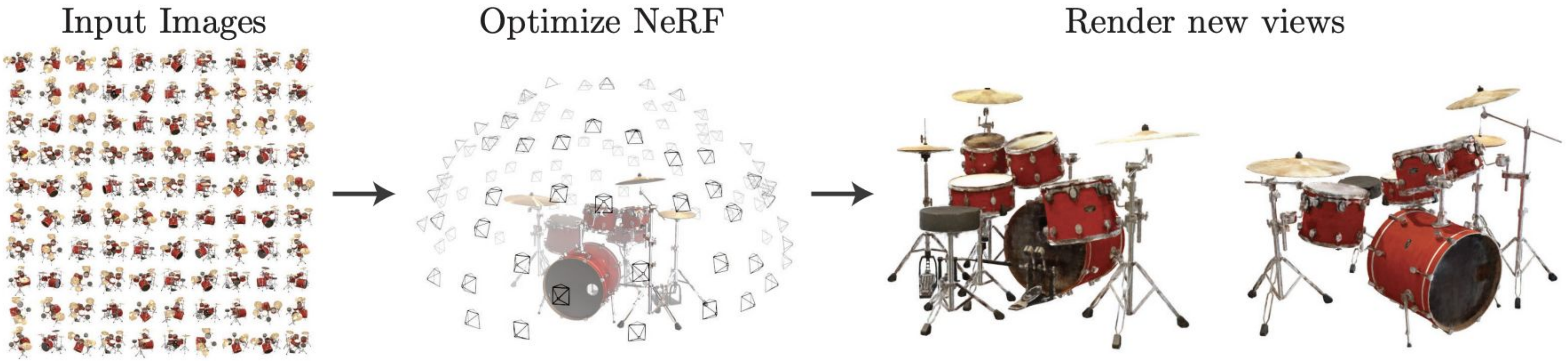
Today’s agenda

- Neural scene representation
- Neural radiance fields (NeRF)
- Implicit neural representation learning (INR)
- INR in biomedical imaging
- Challenges

**Some slides in this section are adapted from: [Deep Dive into the Volumetric Rendering Function, Ben Mildenhall, ECCV 2022 Tutorial]*

Neural radiance fields (NeRF)

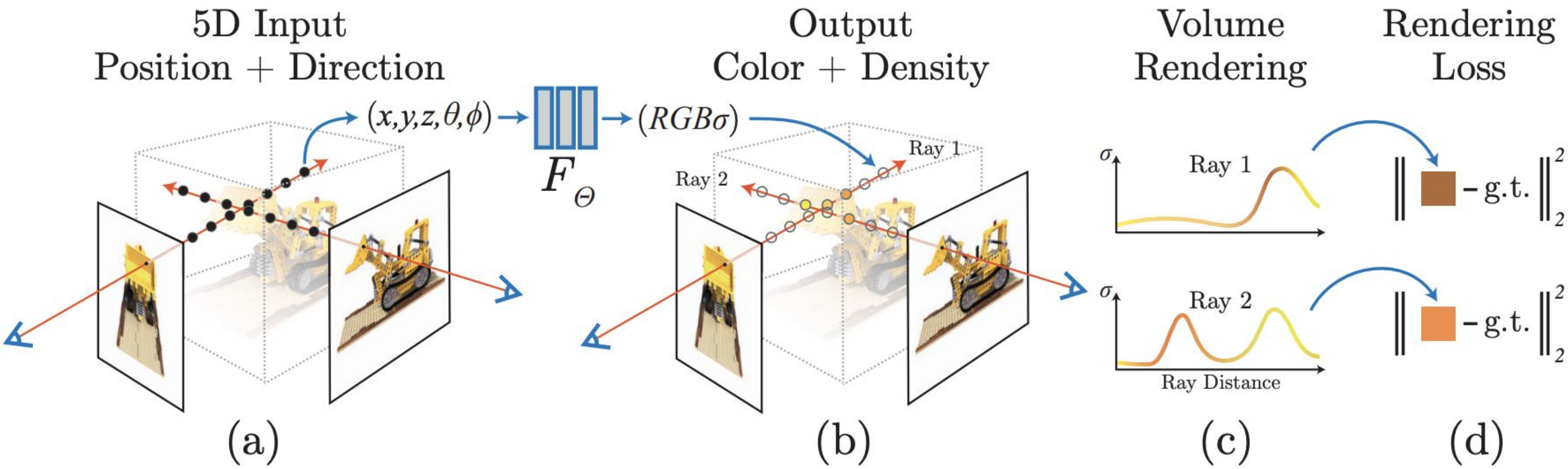
- Novel view synthesis
  - Observe a set of input images
  - Optimizes a continuous 5D neural radiance field representation
    - Volume density and view-dependent color at any continuous location
  - Render the scene from any novel (unobserved) viewpoint



Mildenhall, et al., NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2019.

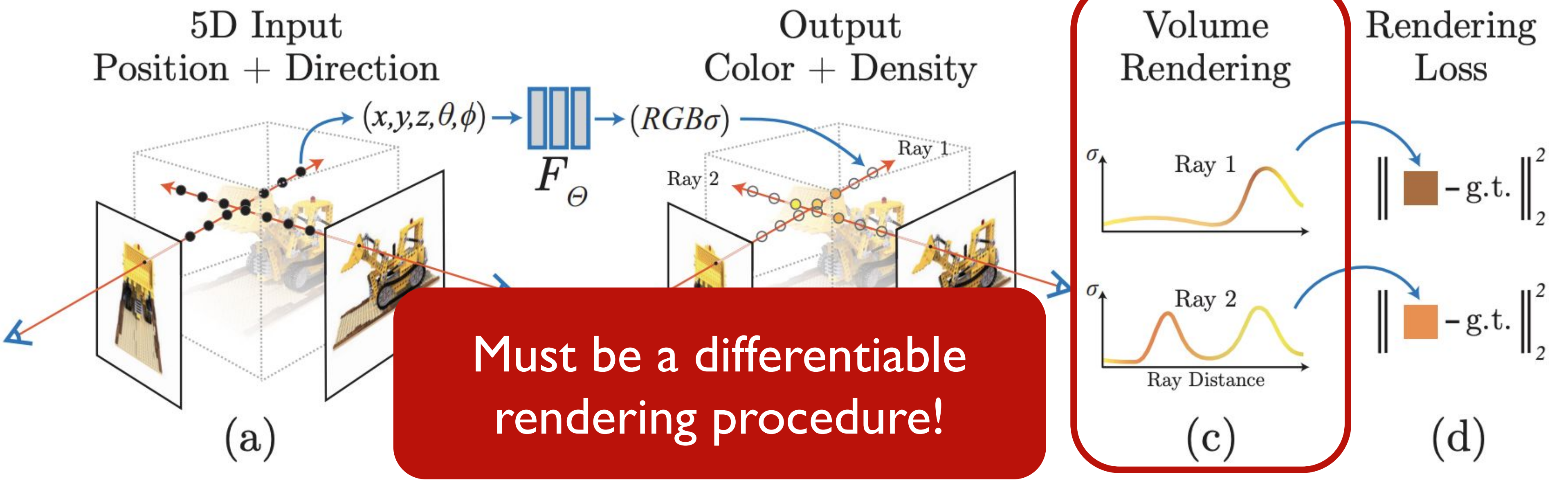
Neural radiance fields (NeRF)

- Continuous volumetric scene function
    - Represent a scene using a fully-connected (non-convolutional) deep network
    - Input: a single continuous 5D coordinate (spatial location (x, y, z) and viewing direction (θ, ϕ))
    - Output: volume density and view-dependent emitted radiance at that spatial location



Neural radiance fields (NeRF)

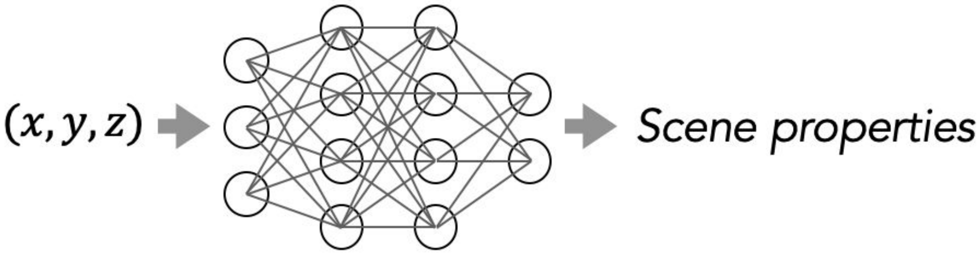
- Continuous volumetric scene function
  - Represent a scene using a fully-connected (non-convolutional) deep network
  - Input: a single continuous 5D coordinate (spatial location (x, y, z) and viewing direction (θ, ϕ))
  - Output: volume density and view-dependent emitted radiance at that spatial location



Must be a differentiable rendering procedure!

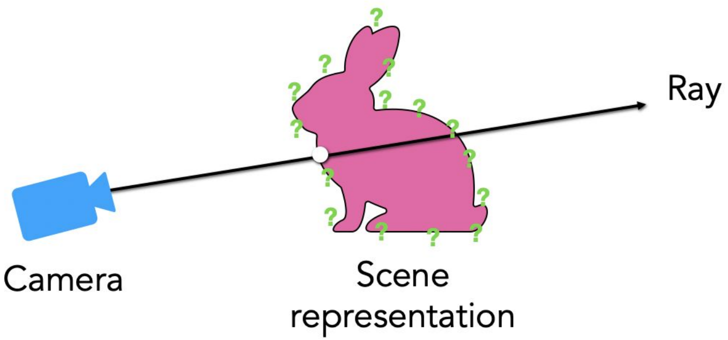
Neural radiance fields (NeRF)

- Neural volumetric rendering
  - Neural: using a neural network as a scene representation, rather than a voxel grid of data

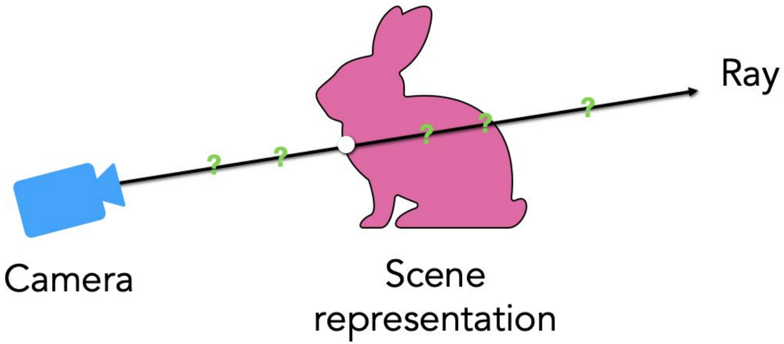


Neural radiance fields (NeRF)

- Neural volumetric rendering
  - Neural: using a neural network as a scene representation, rather than a voxel grid of data
  - Volumetric: continuous, differentiable rendering model without concrete ray/surface intersections



Surface rendering

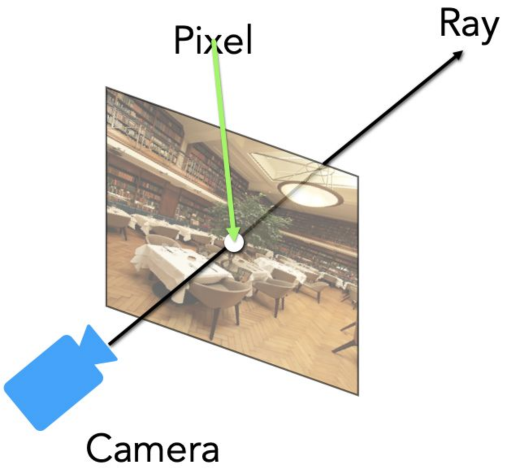


Volume rendering

Neural radiance fields (NeRF)

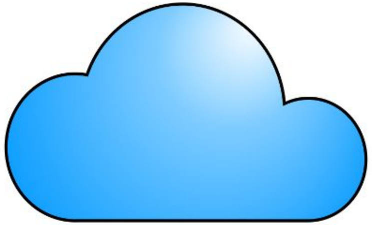
- Neural volumetric rendering
  - Neural: using a neural network as a scene representation, rather than a voxel grid of data
  - Volumetric: continuous, differentiable rendering model without concrete ray/surface intersections
  - Rendering: computing color along rays through 3D space

What color is this pixel?
- Learning function ray → color



NeRF: Neural volumetric rendering

- Volumetric formulation

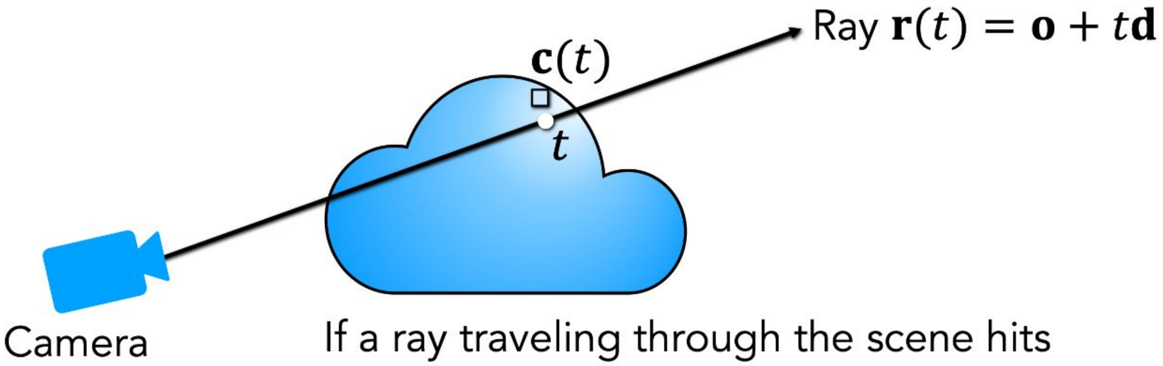


Scene is a cloud of tiny colored particles

Max and Chen, Local and Global Illumination in the Volume Rendering Integral, 2010.

NeRF: Neural volumetric rendering

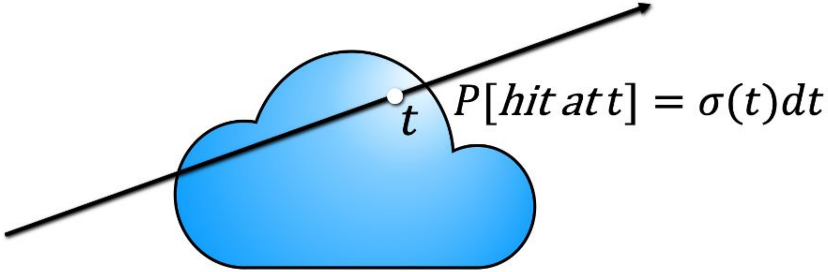
- Volumetric formulation



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

NeRF: Neural volumetric rendering

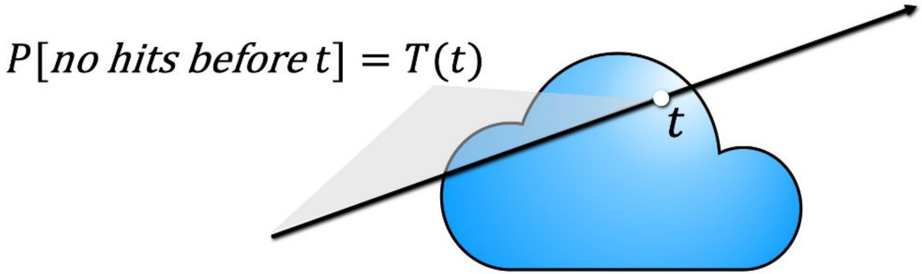
- What does it mean for a ray to "hit" the volume?



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"

NeRF: Neural volumetric rendering

- Probabilistic interpretation

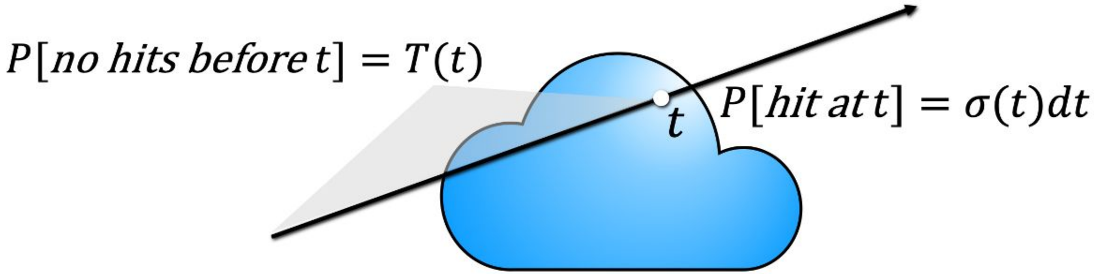


To determine if t is the *first* hit along the ray, need to know $T(t)$: the probability that the ray makes it through the volume up to t.

$T(t)$ is called "transmittance"

NeRF: Neural volumetric rendering

- Probabilistic interpretation

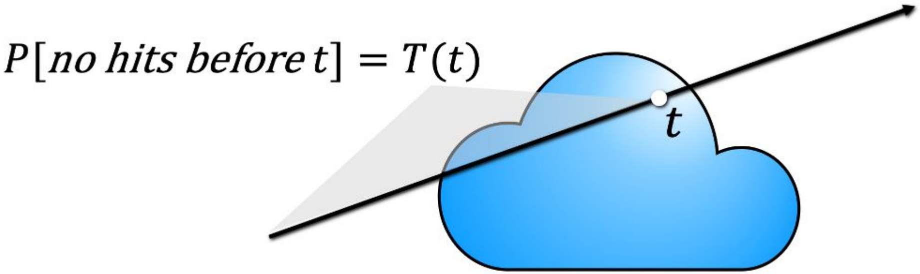


The product of these probabilities tells us how much you see the particles at t:

$P[first\ hit\ at\ t] = P[no\ hit\ before\ t] \times P[hit\ at\ t]$
$= T(t)\sigma(t)dt$

NeRF: Neural volumetric rendering

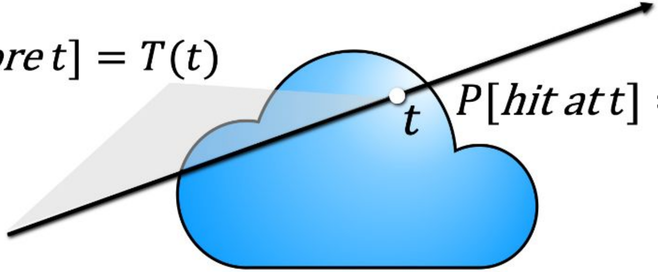
- Calculating T given σ



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

NeRF: Neural volumetric rendering

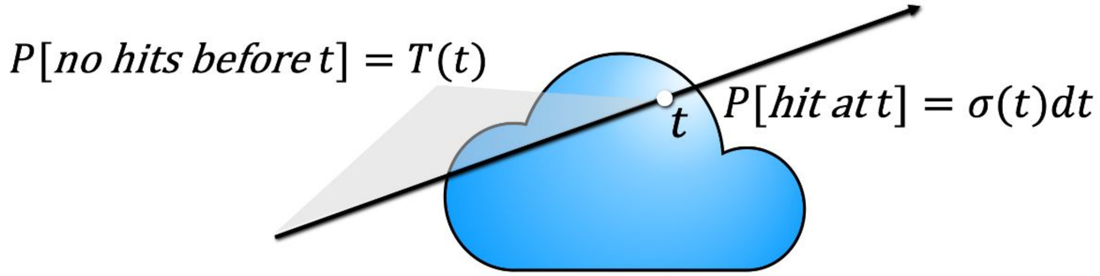
- Calculating T given σ



σ and T are related by the probabilistic fact that
$P[no\ hit\ before\ t + dt] = P[no\ hit\ before\ t] \times P[no\ hit\ at\ t]$

NeRF: Neural volumetric rendering

- Calculating transmittance T



σ and T are related by the probabilistic fact that

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

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

NeRF: Neural volumetric rendering

- Calculating transmittance T

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

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

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

NeRF: Neural volumetric rendering

- Solve for T

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

NeRF: Neural volumetric rendering

- Solve for T

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

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

NeRF: Neural volumetric rendering

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

NeRF: Neural volumetric rendering

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

NeRF: Neural volumetric rendering

- Solve for T

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

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

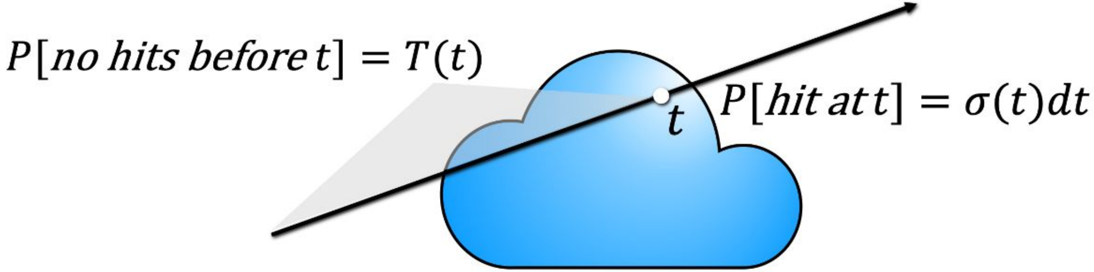
Rearrange$\Rightarrow \frac{T'(t)}{T(t)}dt = -\sigma(t)dt$

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

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

NeRF: Neural volumetric rendering

- Probabilistic interpretation

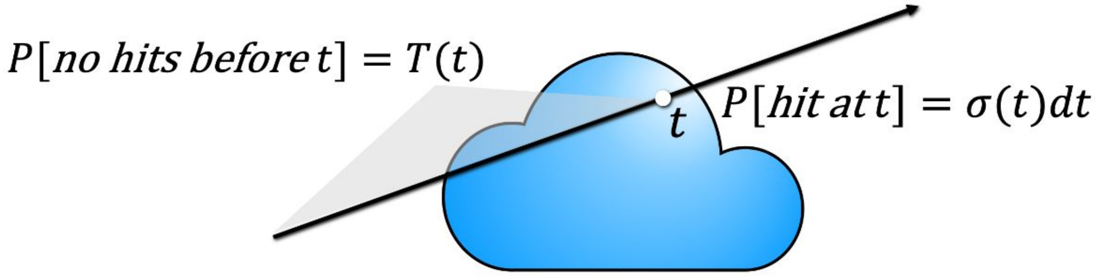


The product of these probabilities tells us how much you see the particles at t:

$P[\textit{first hit at}\ t] = P[\textit{no hit before}\ t] \times P[\textit{hit at}\ t]$
$= T(t)\sigma(t)dt$

NeRF: Neural volumetric rendering

- Probability density function (PDF) for ray termination



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

$$P[first\ hit\ at\ t] = P[no\ hit\ before\ t] \times P[hit\ at\ t]$$

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

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

NeRF: Neural volumetric rendering

- Expected value of color along ray

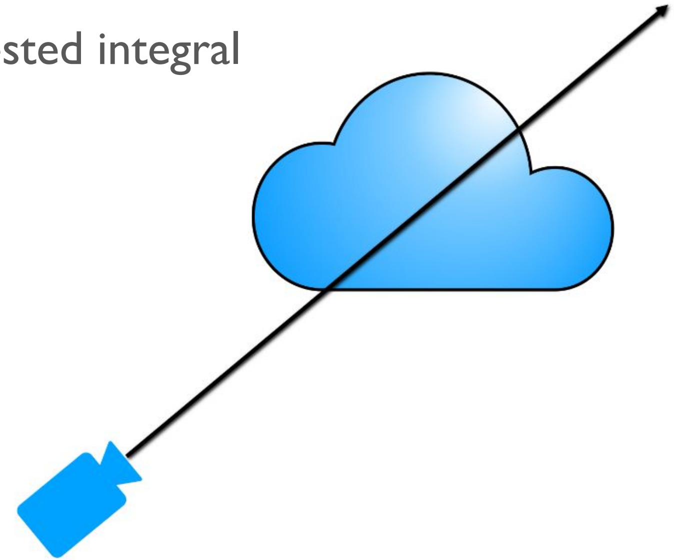
This means the expected color returned by the ray will be

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

Note the nested integral!

NeRF: Neural volumetric rendering

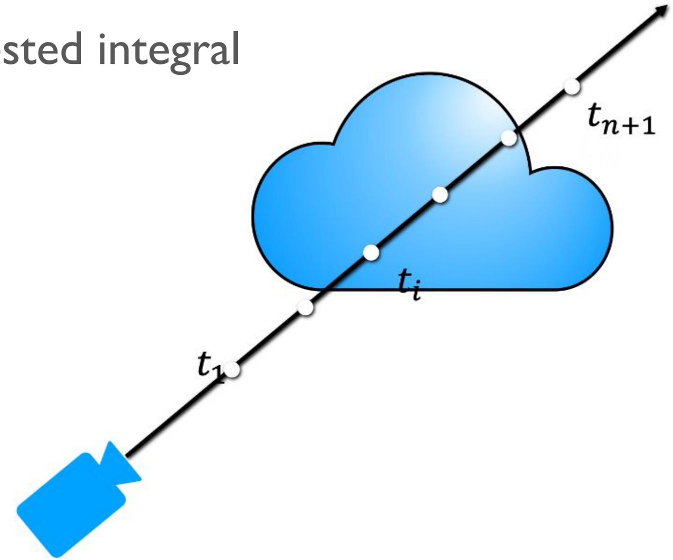
- Approximating the nested integral



We use quadrature to approximate the nested integral,

NeRF: Neural volumetric rendering

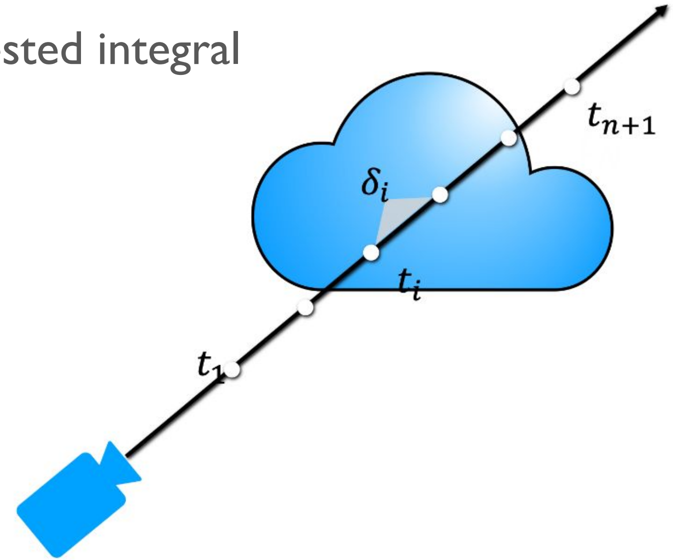
- Approximating the nested integral



We use quadrature to approximate the nested integral, splitting the ray up into n segments with endpoints $\{t_1, t_2, \ldots, t_{n+1}\}$

NeRF: Neural volumetric rendering

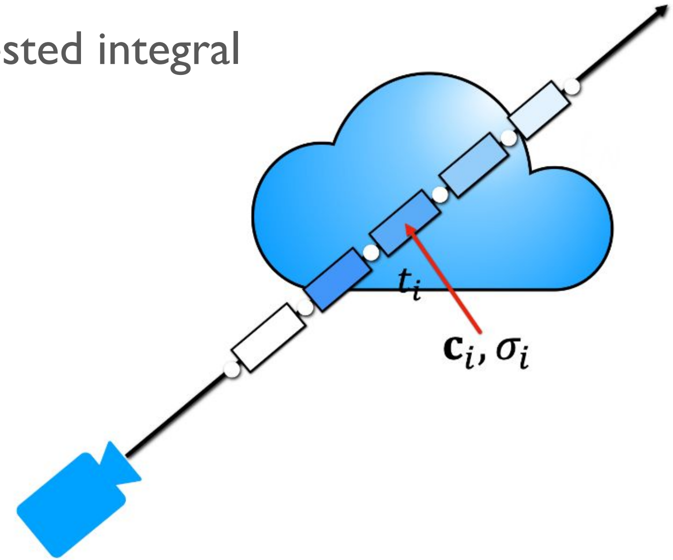
- Approximating the nested integral



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 $\delta_i = t_{i+1} - t_i$

NeRF: Neural volumetric rendering

- Approximating the nested integral



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

NeRF: Neural volumetric rendering

- Deriving quadrature estimate

$$\int T(t)\sigma(t)\mathbf{c}(t)dt \approx$$

This allows us to break the outer integral

NeRF: Neural volumetric rendering

- Deriving quadrature estimate

$$\int T(t)\sigma(t)\mathbf{c}(t)dt \approx \sum_{i=1}^{n} \int_{t_i}^{t_{i+1}} T(t)\sigma_i \mathbf{c}_i dt$$

This allows us to break the outer integral
into a sum of analytically tractable integrals

NeRF: Neural volumetric rendering

- Deriving quadrature estimate

$$\int T(t)\sigma(t)\mathbf{c}(t)dt \approx \sum_{i=1}^{n}\int_{t_i}^{t_{i+1}} \boxed{T(t)}\,\sigma_i\mathbf{c}_i dt$$

Caveat: piecewise constant density and color
do not imply constant transmittance!

NeRF: Neural volumetric rendering

- Deriving quadrature estimate

$$\int T(t)\sigma(t)\mathbf{c}(t)dt \approx \sum_{i=1}^{n}\int_{t_i}^{t_{i+1}} \boxed{T(t)}\sigma_i\mathbf{c}_i dt$$

Caveat: piecewise constant density and color **do not** imply constant transmittance!

Important to account for how early part of a segment blocks later part when σ_i is high

NeRF: Neural volumetric rendering

- Evaluating T for piecewise constant density

$$\text{For } t \in [t_i, t_{i+1}],\ T(t) = \exp\left(-\int_{t_1}^{t_i} \sigma_i ds\right) \exp\left(-\int_{t_i}^{t} \sigma_i ds\right)$$

We need to evaluate at continuous t values
that can lie *partway through* an interval

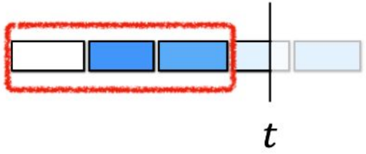
NeRF: Neural volumetric rendering

- Evaluating T for piecewise constant density

For $t \in [t_i, t_{i+1}]$, $T(t) = \exp\left(-\int_{t_1}^{t_i} \sigma_i ds\right) \exp\left(-\int_{t_i}^{t} \sigma_i ds\right)$

$$\exp\left(-\sum_{j=1}^{i-1} \sigma_j \delta_j\right) = T_i$$

"How much light is blocked by all previous segments?"



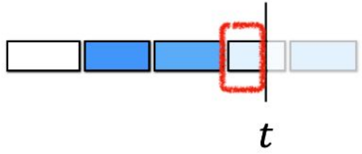
NeRF: Neural volumetric rendering

- Evaluating T for piecewise constant density

$$\text{For } t \in [t_i, t_{i+1}],\ T(t) = \exp\left(-\int_{t_1}^{t_i} \sigma_i ds\right) \exp\left(-\int_{t_i}^{t} \sigma_i ds\right)$$

"How much light is blocked partway through the current segment?"

$$\exp(-\sigma_i(t - t_i))$$



NeRF: Neural volumetric rendering

- Deriving quadrature estimate

$$\int T(t)\sigma(t)\mathbf{c}(t)dt \approx \sum_{i=1}^{n} \int_{t_i}^{t_{i+1}} T(t)\sigma_i \mathbf{c}_i dt$$

NeRF: Neural volumetric rendering

- Deriving quadrature estimate

$$\int T(t)\sigma(t)\mathbf{c}(t)dt \approx \sum_{i=1}^{n} \int_{t_i}^{t_{i+1}} T(t)\sigma_i \mathbf{c}_i dt$$

$$\text{Substitute} = \sum_{i=1}^{n} T_i \sigma_i \mathbf{c}_i \int_{t_i}^{t_{i+1}} \exp(-\sigma_i (t - t_i)) dt$$

NeRF: Neural volumetric rendering

- Deriving quadrature estimate

$$\int T(t)\sigma(t)\mathbf{c}(t)dt \approx \sum_{i=1}^{n} \int_{t_i}^{t_{i+1}} T(t)\sigma_i \mathbf{c}_i dt$$

$$= \sum_{i=1}^{n} T_i \sigma_i \mathbf{c}_i \int_{t_i}^{t_{i+1}} \exp(-\sigma_i(t - t_i))dt$$

Integrate $= \sum_{i=1}^{n} T_i \sigma_i \mathbf{c}_i \frac{\exp(-\sigma_i(t_{i+1} - t_i)) - 1}{-\sigma_i}$

NeRF: Neural volumetric rendering

- Deriving quadrature estimate

$$\int T(t)\sigma(t)\mathbf{c}(t)dt \approx \sum_{i=1}^{n} \int_{t_i}^{t_{i+1}} T(t)\sigma_i \mathbf{c}_i dt$$

$$= \sum_{i=1}^{n} T_i \sigma_i \mathbf{c}_i \int_{t_i}^{t_{i+1}} \exp(-\sigma_i(t - t_i))dt$$

$$= \sum_{i=1}^{n} T_i \sigma_i \mathbf{c}_i \frac{\exp(-\sigma_i(t_{i+1} - t_i)) - 1}{-\sigma_i}$$

Cancel σ_i $= \sum_{i=1}^{n} T_i \mathbf{c}_i (1 - \exp(-\sigma_i \delta_i))$

NeRF: Neural volumetric rendering

- Deriving quadrature estimate

$$= \sum_{i=1}^{n} T_i \mathbf{c}_i (1 - \underbrace{\exp(-\sigma_i \delta_i))}_{\text{segment opacity } \alpha_i}$$

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

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

NeRF: Neural volumetric rendering

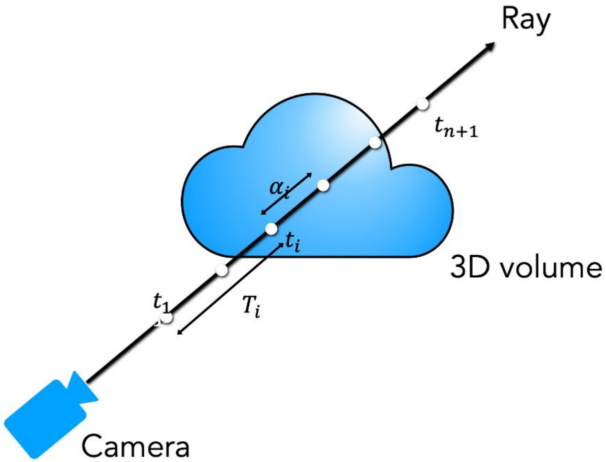
- Volume rendering integral estimate:
  - How to estimate the color of a ray traversing a volume parameterized by color and volume density

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

$$\mathbf{c} \approx \sum_{i=1}^{n} \underbrace{T_i \alpha_i}_{\text{weights}} \underbrace{\mathbf{c}_i}_{\text{colors}}$$

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

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



NeRF: Neural volumetric rendering

- Volume rendering is differentiable

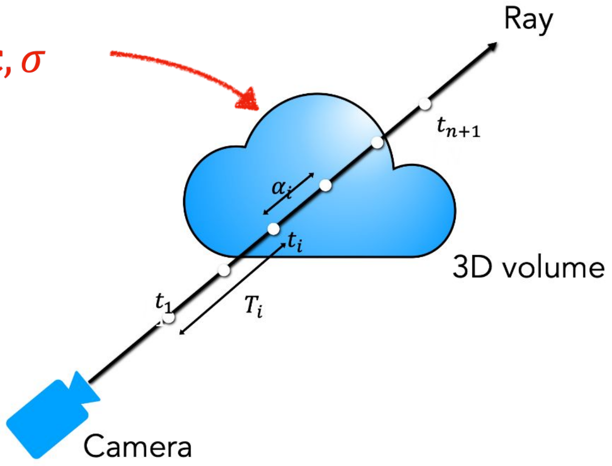
Rendering model for ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$:

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

colors

weights

differentiable w.r.t. c, σ



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

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

NeRF: Neural volumetric rendering with radiance fields

- The expected color C(r) of camera ray r(t) = o + td:

$$C(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t), \mathbf{d})dt\,, \text{ where } T(t) = \exp\left(-\int_{t_n}^{t} \sigma(\mathbf{r}(s))ds\right)$$

- Estimate C(r) with the quadrature rule:

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} T_i(1 - \exp(-\sigma_i\delta_i))\mathbf{c}_i\,, \text{ where } T_i = \exp\left(-\sum_{j=1}^{i-1} \sigma_j\delta_j\right)$$

Mildenhall, et al., NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2019.

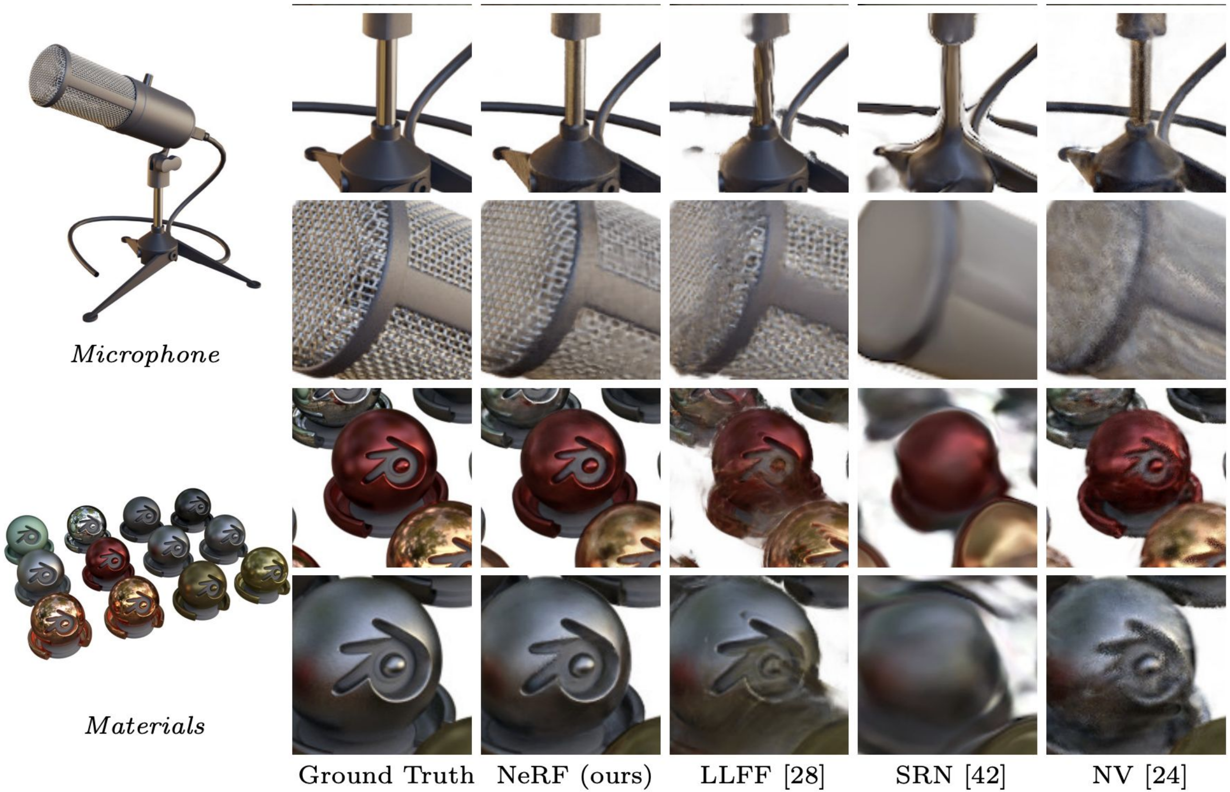
NeRF

- More details in the paper:
  - Positional encoding
  - Hierarchical volume sampling
  - ……

Mildenhall, et al., NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2019.

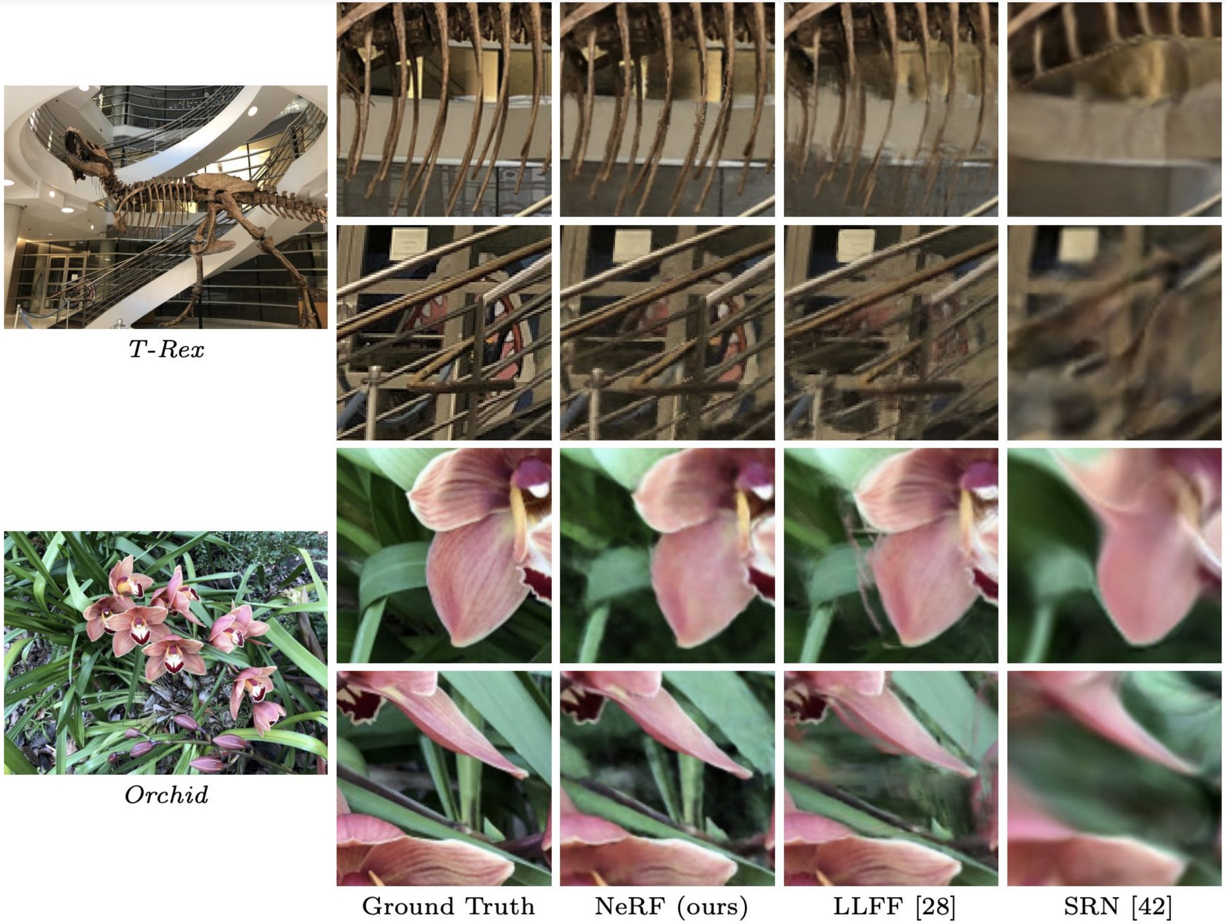
NeRF

- Results:
  - NeRF outperforms SRN
  - Synthesize high-resolution images of real scenes / objects!



NeRF

- Results:
  - NeRF outperforms SRN
  - Synthesize high-resolution images of real scenes / objects!



Today's agenda

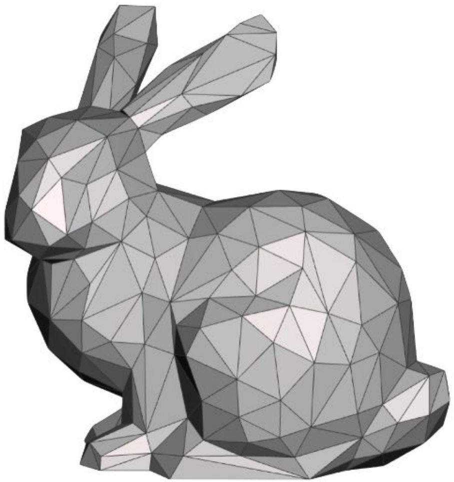
- Neural scene representation
- Neural radiance fields (NeRF)
- Implicit neural representation learning (INR)
- INR in biomedical imaging
- Challenges

Today's agenda

- Neural scene representation
- Neural radiance fields (NeRF)
- Implicit neural representation learning (INR)
- INR in biomedical imaging
- Challenges

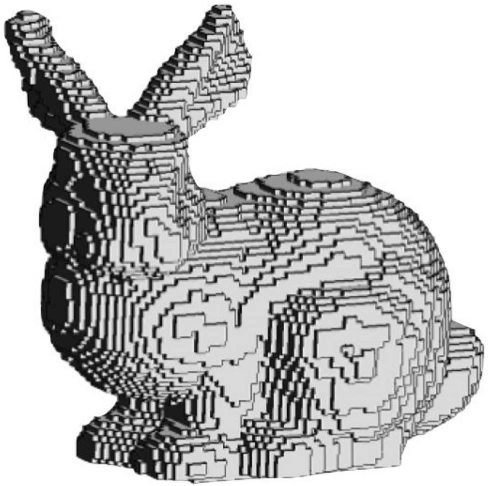
**Some slides in this section are adapted from: [Encoding and Representing 3D Volumes, Matt Tancik, ECCV 2022 Tutorial]*

Geometry representations



Mesh Representation

Small memory footprint
Hard to optimize

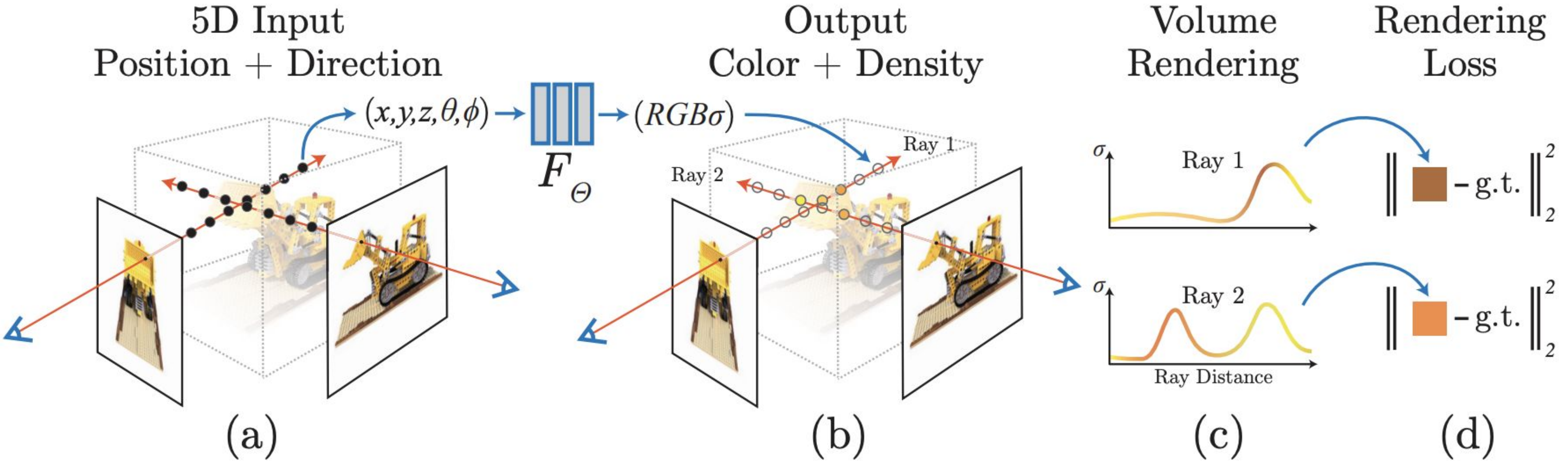


Voxel Representation

Easy to optimize
Large memory footprint

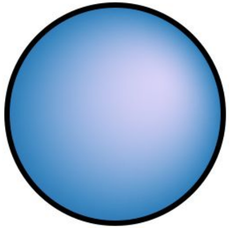
NeRF: Network-parameterized function representation

- Continuous volumetric scene function
  - Represent a scene using a fully-connected (non-convolutional) deep network
  - Small memory footprint
  - Easy to optimize

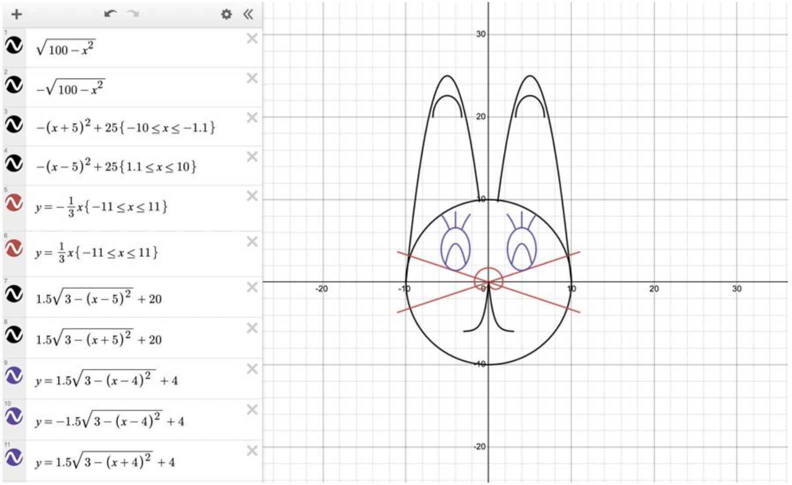


Generalize to implicit function representations

- Implicit functions

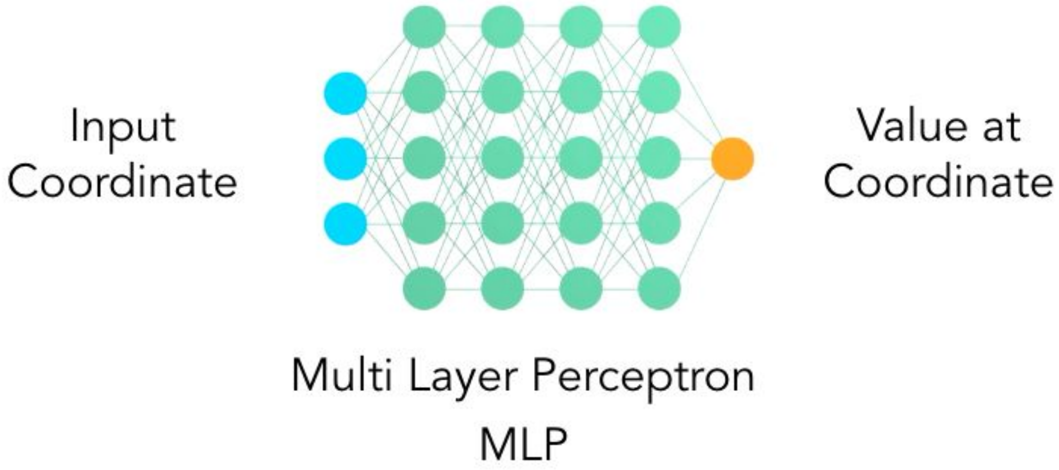


$x^2 + y^2 + z^2 = 1$



Implicit neural representations (INR)

- Coordinate-based neural networks



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

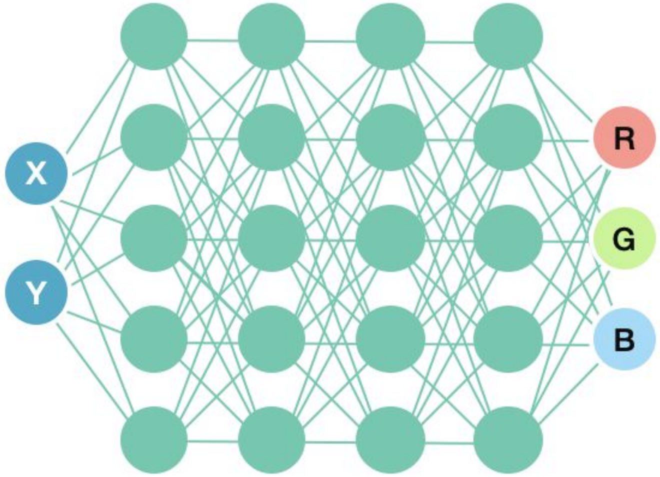
Implicit neural representations (INR)

- Coordinate-based neural networks

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

Implicit neural representations (INR)

- INR for RGB image representations



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

Implicit neural representations (INR)

- Challenges:
  - How to get MLPs to represent higher frequency functions?

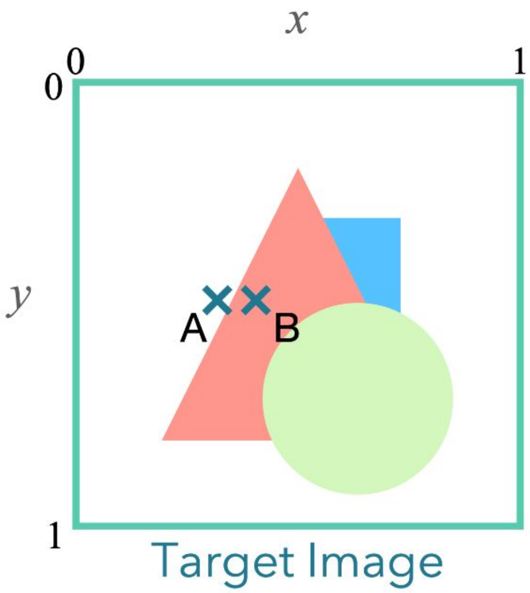
MLP output

Supervision image

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

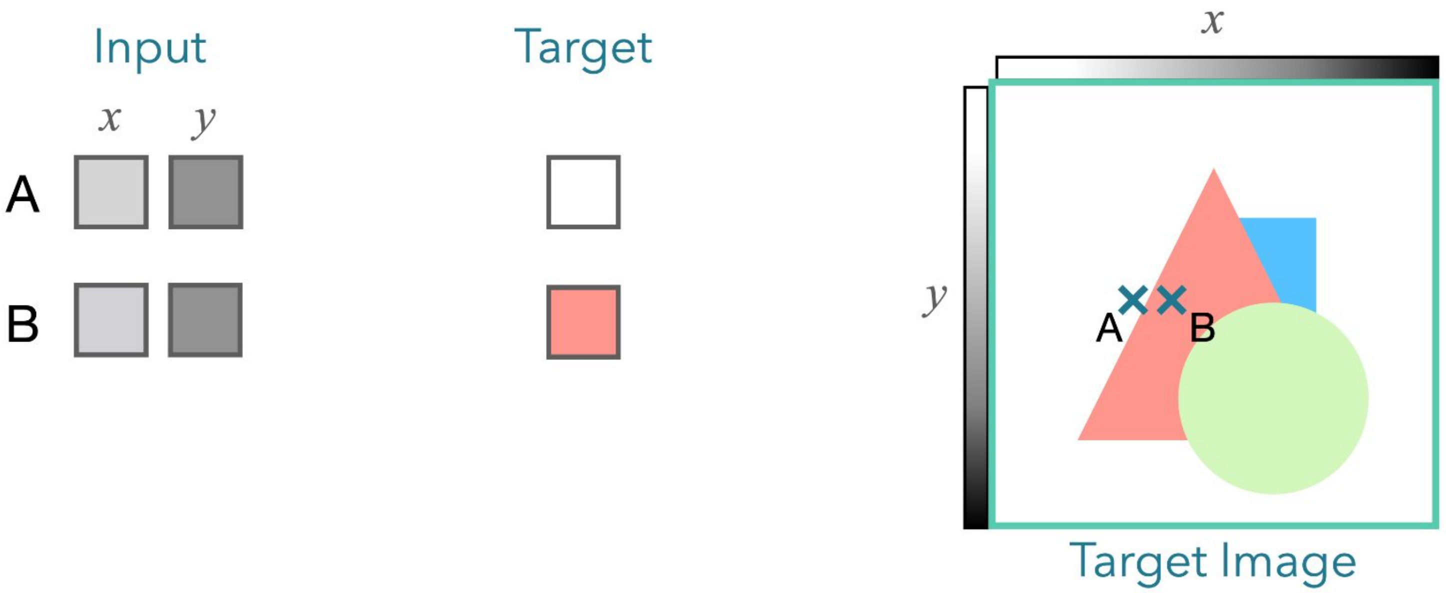
Implicit neural representations (INR)

	Input x	Input y	Target
A	.36	.5	
B	.38	.5	



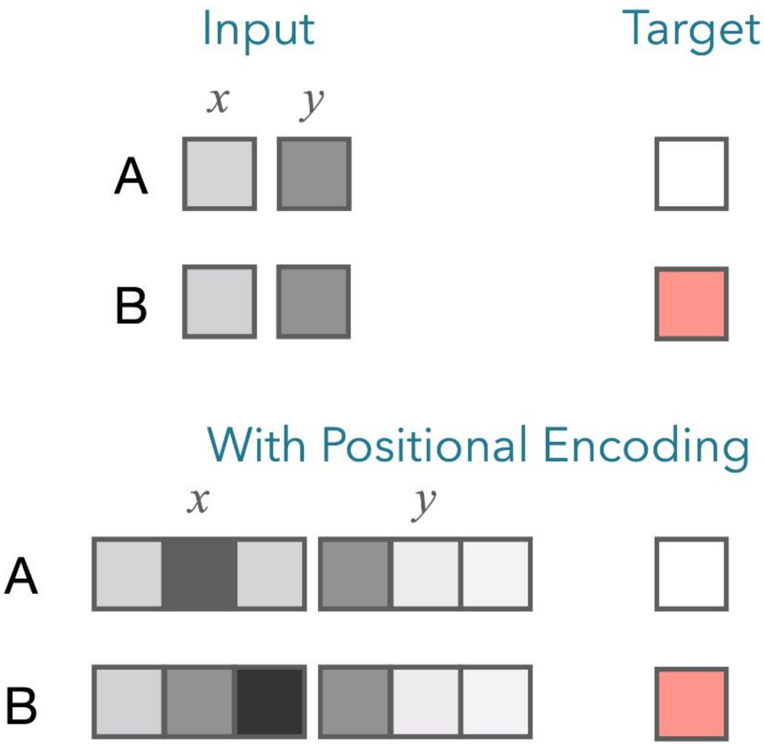
Tancik, et al., *Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains*, NeurIPS 2020.

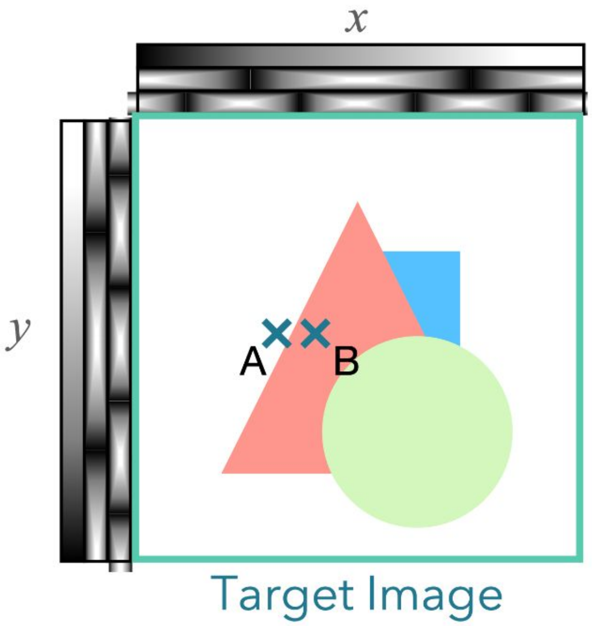
Implicit neural representations (INR)



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

Implicit neural representations (INR)





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

Implicit neural representations (INR)

- Positional encoding:
  - Map input coordinates into a higher dimensional feature space before passing them through the network
- Fourier feature mapping:
  - v: input points (x, y)
  - B: random Gaussian matrix, where each entry is drawn independently from a normal distribution N(0, σ^2)

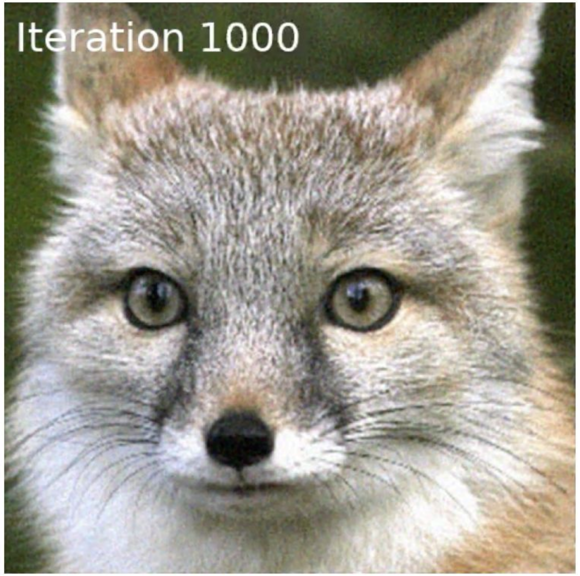
$$\gamma(\mathbf{v}) = [\cos(2\pi\mathbf{B}\mathbf{v}), \sin(2\pi\mathbf{B}\mathbf{v})]^{\mathrm{T}}$$

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

Implicit neural representations (INR)

- Fourier features let networks learn high frequency functions in low dimensional domains

Standard MLP

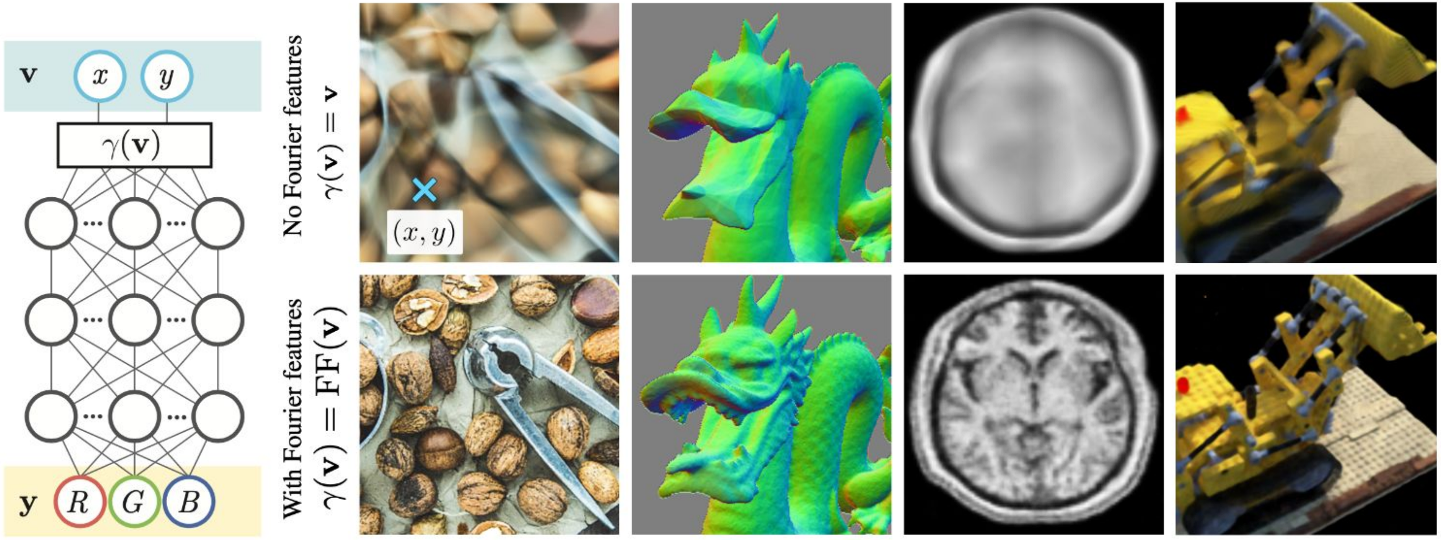


MLP with Fourier features

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

Implicit neural representations (INR)

- Fourier features let networks learn high frequency functions in low dimensional domains



(a) Coordinate-based MLP

(b) Image regression $(x,y) \to$ RGB

(c) 3D shape regression $(x,y,z) \to$ occupancy

(d) MRI reconstruction $(x,y,z) \to$ density

(e) Inverse rendering $(x,y,z) \to$ RGB, density

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

Implicit neural representations (INR)

- Fourier feature mapping:
  - Different Gaussian scales σ

$\sigma = 1$ $\sigma = 2$ $\sigma = 10$ $\sigma = 32$ $\sigma = 64$

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

Implicit neural representations (INR)

- Other positional encoding methods:
  - Sparsity
  - Low-rank approximation
  - Dictionary methods
  - ……

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

Today’s agenda

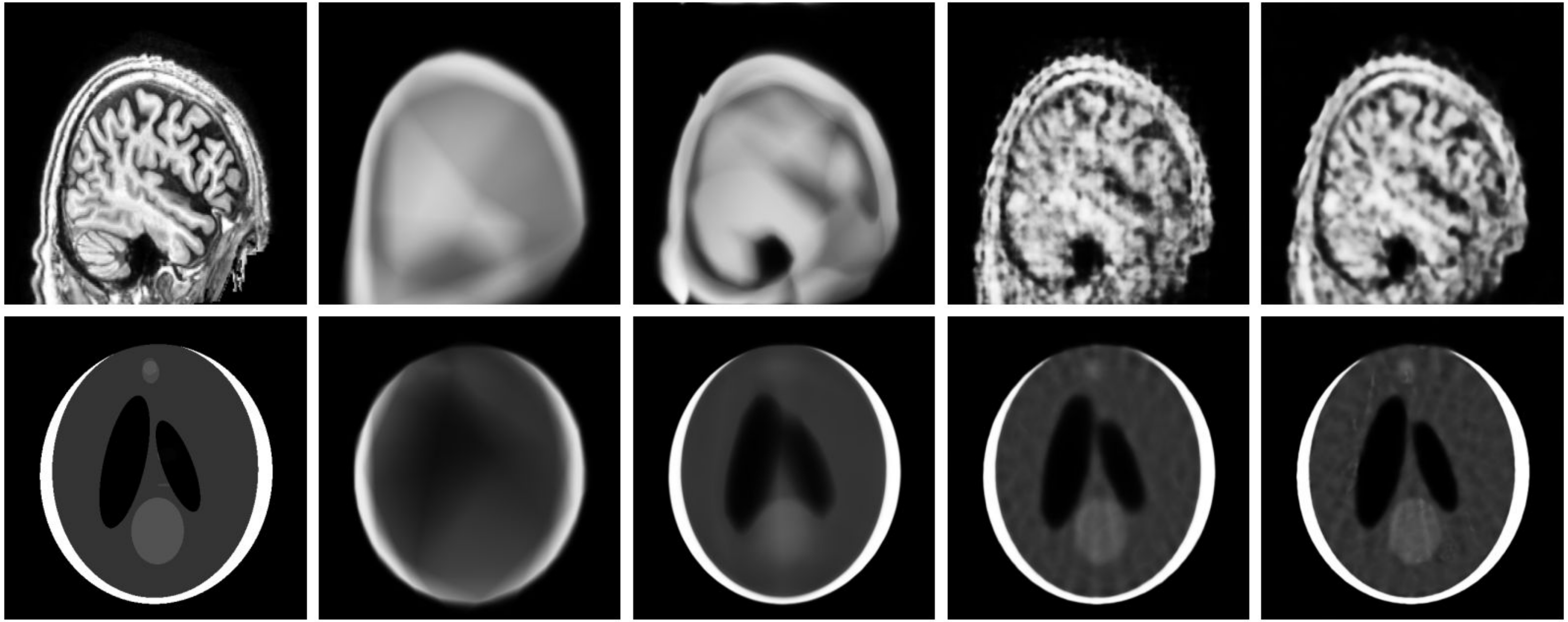
- Neural scene representation
- Neural radiance fields (NeRF)
- Implicit neural representation learning (INR)
- INR in biomedical imaging
- Challenges

Today's agenda

- Neural scene representation
- Neural radiance fields (NeRF)
- Implicit neural representation learning (INR)
- INR in biomedical imaging
- Challenges

Implicit neural representations (INR)

- Unsatisfying quality for biomedical imaging reconstruction

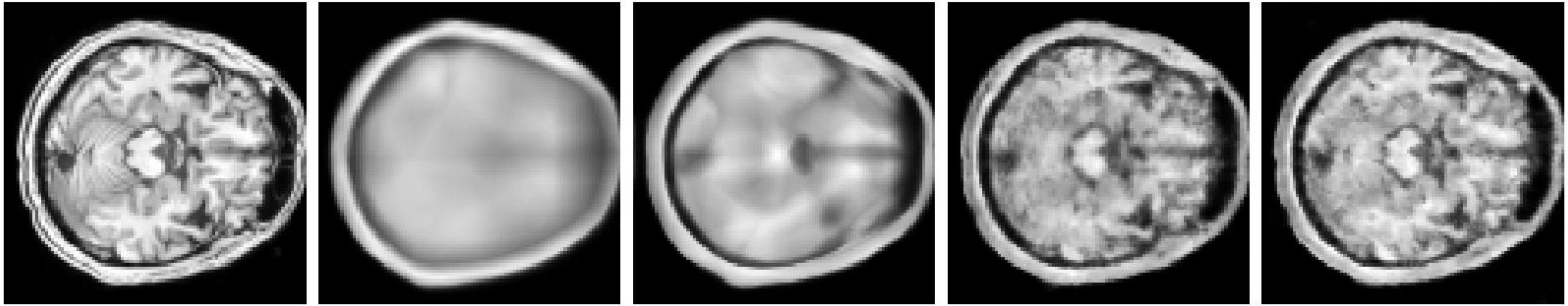


(a) Ground Truth (b) No mapping (c) Basic (d) Positional enc. (e) Gaussian

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

Implicit neural representations (INR)

- Unsatisfying quality for biomedical imaging reconstruction



(a) Ground Truth (b) No mapping (c) Basic (d) Positional enc. (e) Gaussian

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

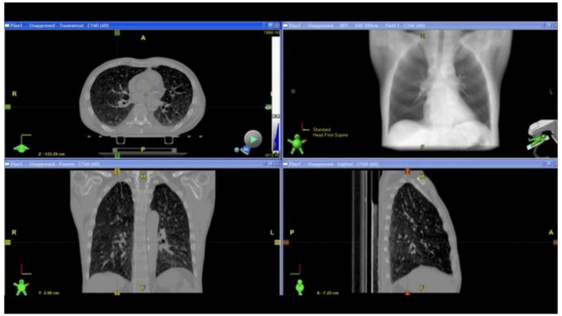
INR for biomedical imaging

- Limitations:
  - Unsatisfying quality for biomedical imaging reconstruction
- Advantages:
  - Small memory footprint
  - Easy to optimize
  - ?

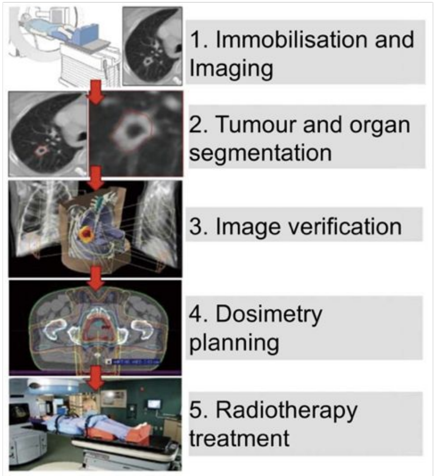
INR for biomedical imaging

- Motivation:
  - Time series data is common and important in biomedical tasks
  - Longitudinal data contain personalized prior knowledge of patient's anatomy

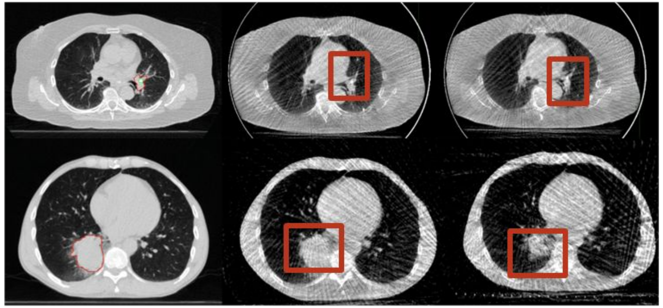
4D CT / 4D MRI characterize motion



Radiation therapy with multiple fractions



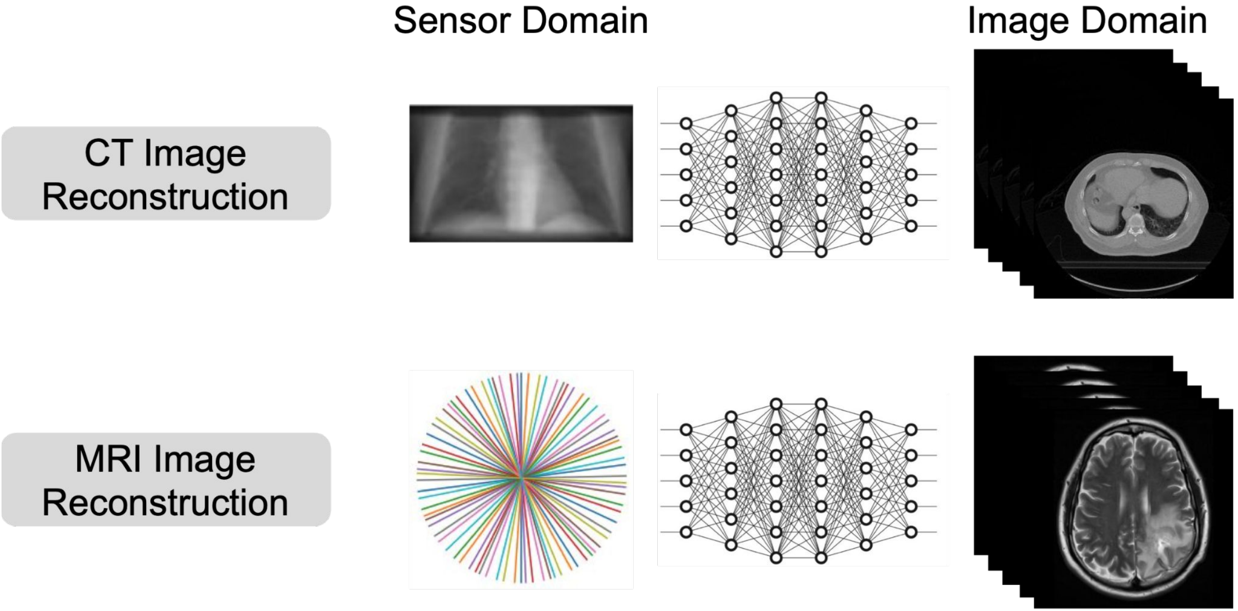
Evaluate tumor response to surgery and therapy



Shen, et al., NeRP: implicit neural representation learning with prior embedding for sparsely sampled image reconstruction, TNNLS 2022.

Previous approach for biomedical imaging

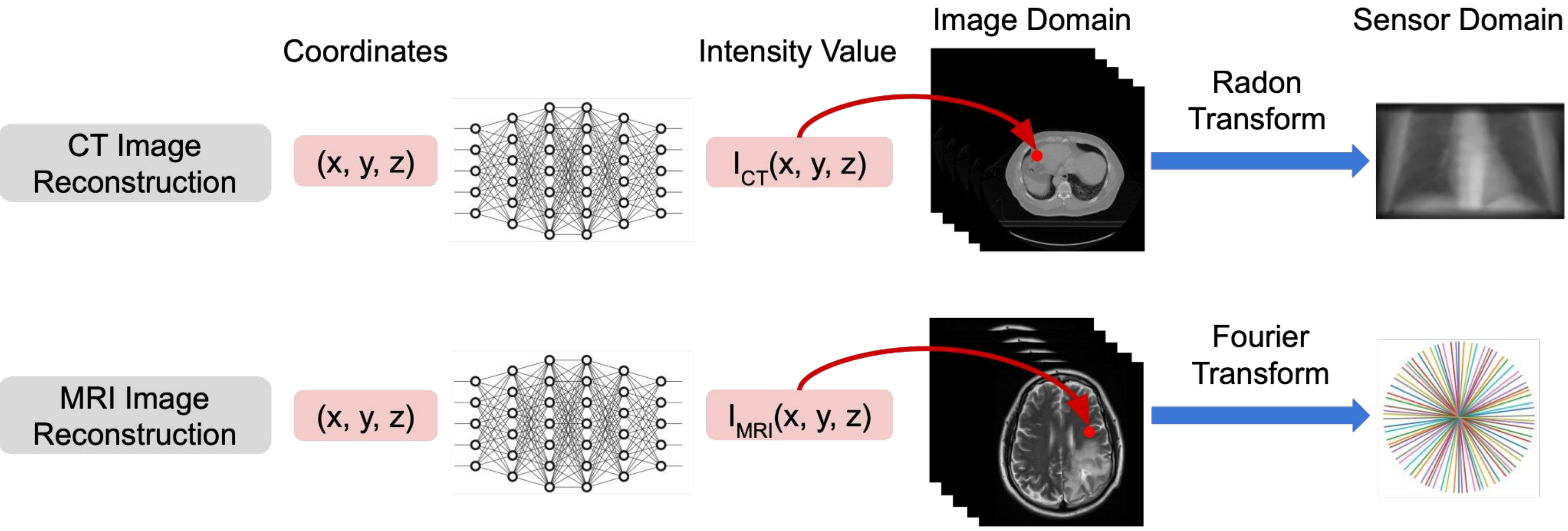
- Network learns mapping from sensor measurements to the reconstructed image



Shen, et al., NeRP: implicit neural representation learning with prior embedding for sparsely sampled image reconstruction, TNNLS 2022.

INR for biomedical imaging

- Network learns implicit neural representation of the reconstructed image



Shen, et al., NeRP: implicit neural representation learning with prior embedding for sparsely sampled image reconstruction, TNNLS 2022.

INR for biomedical imaging

- Reformulate as a continuous function optimization problem

Optimization objective:
- x: image
- y: observed measurements
- A: forward model
- data fidelity + regularizer

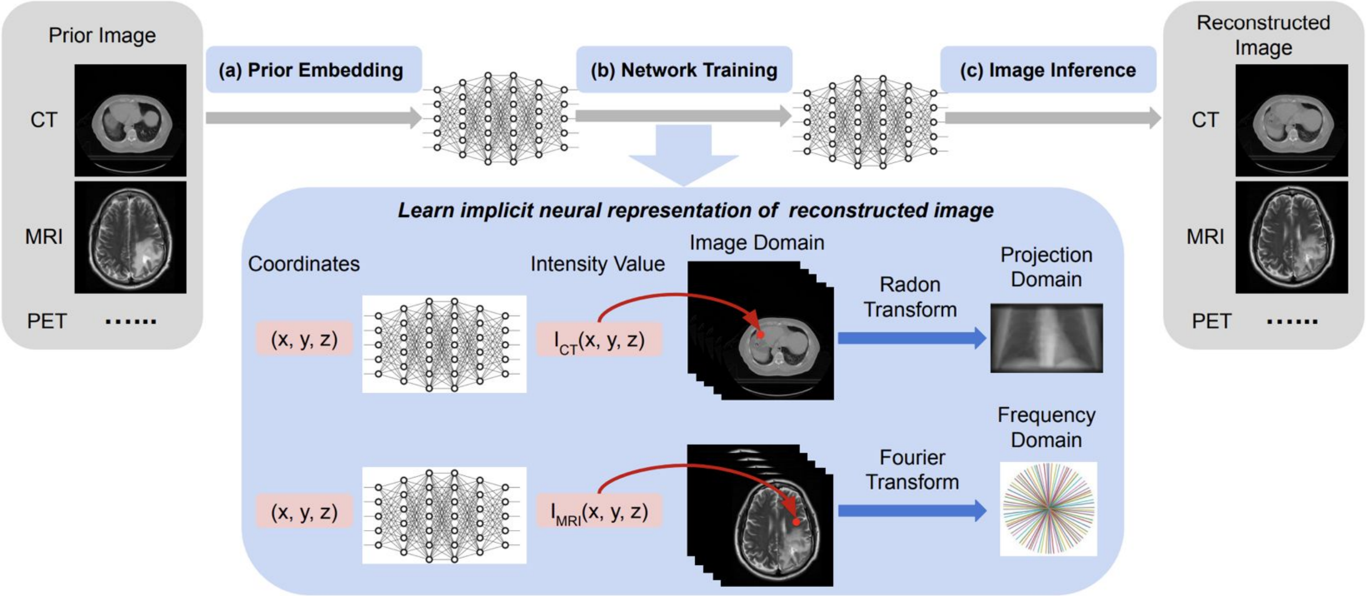
$$x^* = \underset{x}{\operatorname{argmin}} \boxed{\mathcal{E}(Ax, y)} + \boxed{\rho(x)}$$

Implicit regularization captured by the network parametrization

Shen, et al., NeRP: implicit neural representation learning with prior embedding for sparsely sampled image reconstruction, TNNLS 2022.

NeRP: Neural Representation learning with Prior embedding

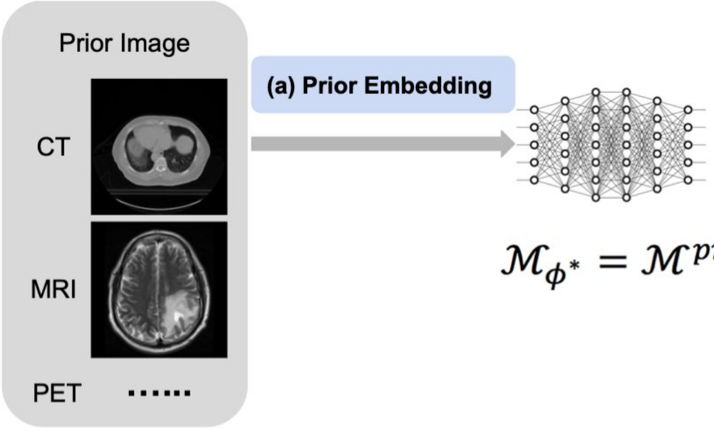
- Exploit internal information from personalized prior and physics of sparsely sampled measurements to learn the representation of unknown subject



Shen, et al., NeRP: implicit neural representation learning with prior embedding for sparsely sampled image reconstruction, TNNLS 2022.

NeRP: Neural Representation learning with Prior embedding

1. Prior embedding: fit the prior image into the neural network weights



x^{pr}

$\mathcal{M}_{\phi^*} = \mathcal{M}^{pr}$

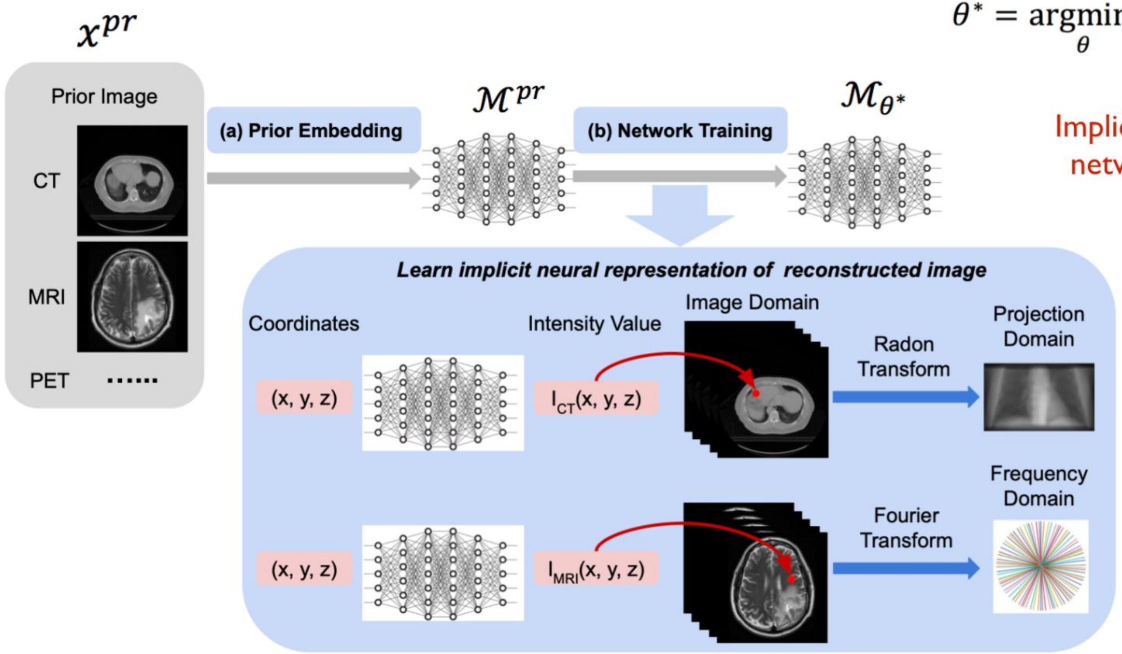
$$\phi^* = \underset{\phi}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} \left\| \mathcal{M}_\phi(c_i) - x_i^{pr} \right\|_2^2$$

x: image
i: spatial grid index
c: spatial coordinates
M: network
Φ: network weights

NeRP: Neural Representation learning with Prior embedding

2. Network training: fine-tune the network to match the constraints of sparsely sampled measurements

$$\theta^* = \underset{\theta}{\operatorname{argmin}}\ \mathcal{E}(A\boxed{\mathcal{M}_\theta}, y; \boxed{\mathcal{M}^{pr}}), \quad x^* = \mathcal{M}_{\theta^*}$$

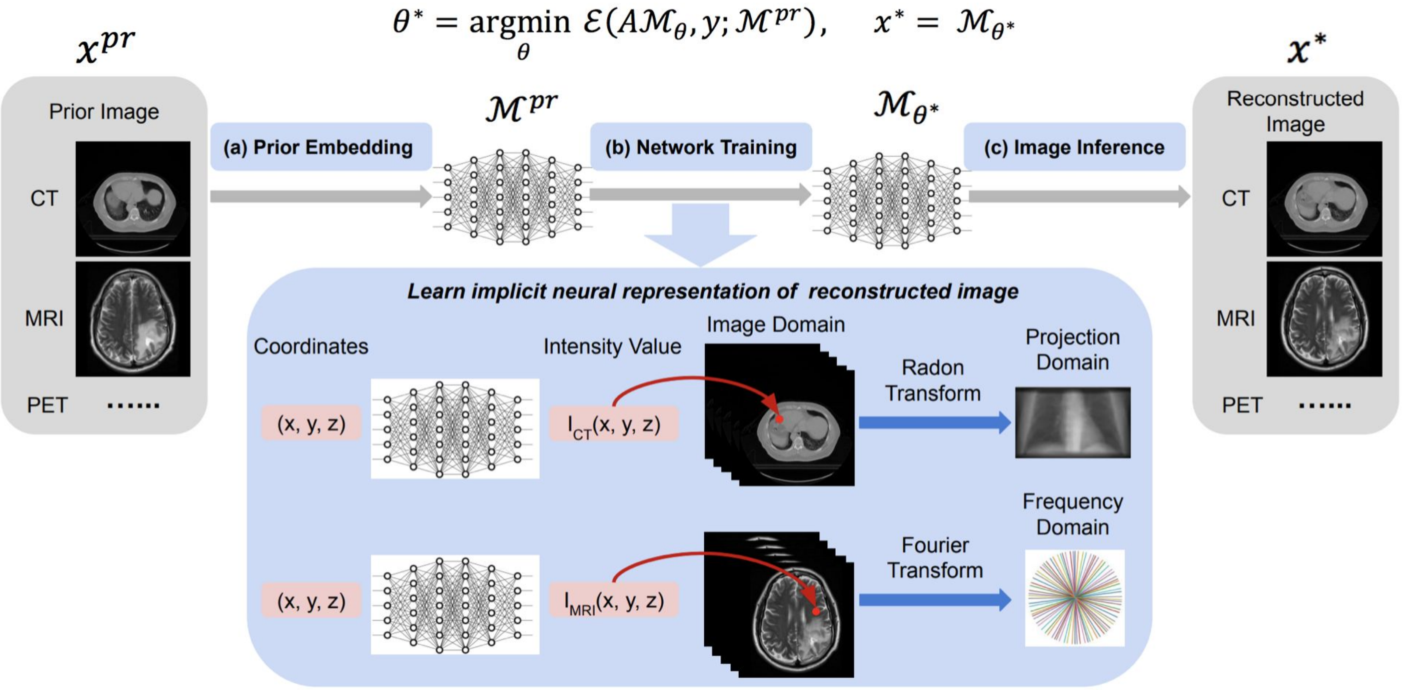


Implicit regularization captured by the network parametrization and prior-embedded initialization

x: image
y: observed measurements
M: network
θ: network weights
A: forward model

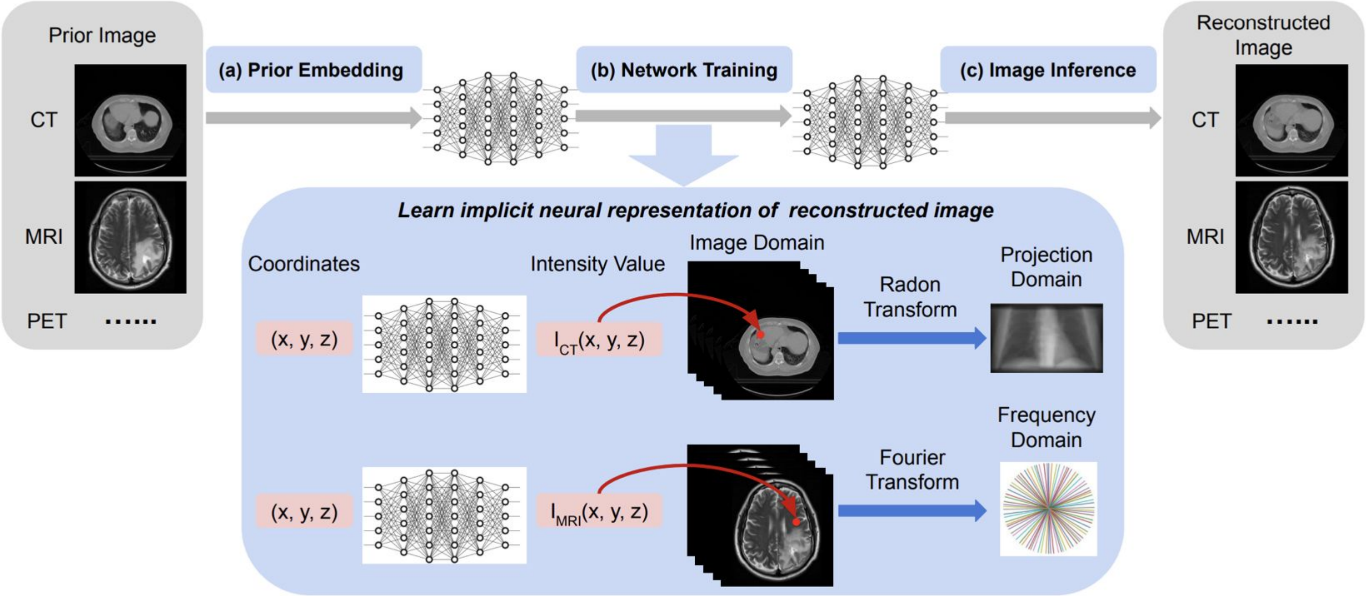
NeRP: Neural Representation learning with Prior embedding

3. Image inference: infer the trained network across all the spatial coordinates in the image grid



NeRP: Neural Representation learning with Prior embedding

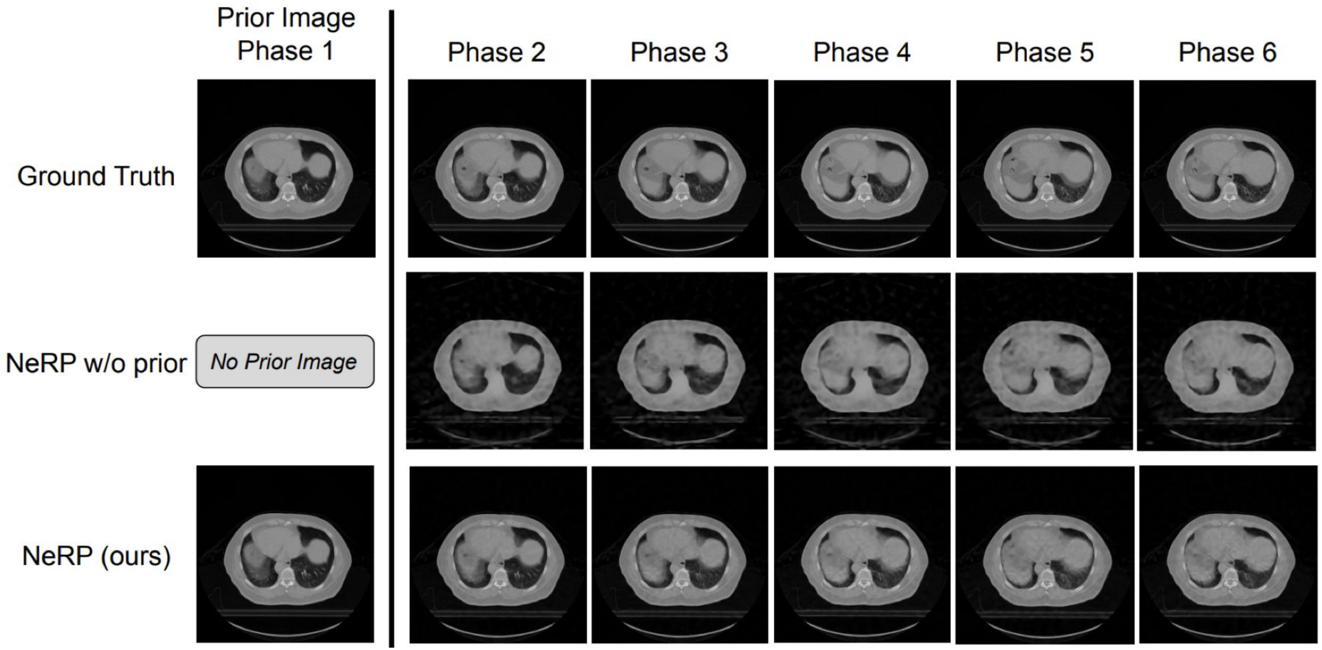
- Exploit internal information from personalized prior and physics of sparsely sampled measurements to learn the representation of unknown subject



Shen, et al., NeRP: implicit neural representation learning with prior embedding for sparsely sampled image reconstruction, TNNLS 2022.

NeRP: Neural Representation learning with Prior embedding

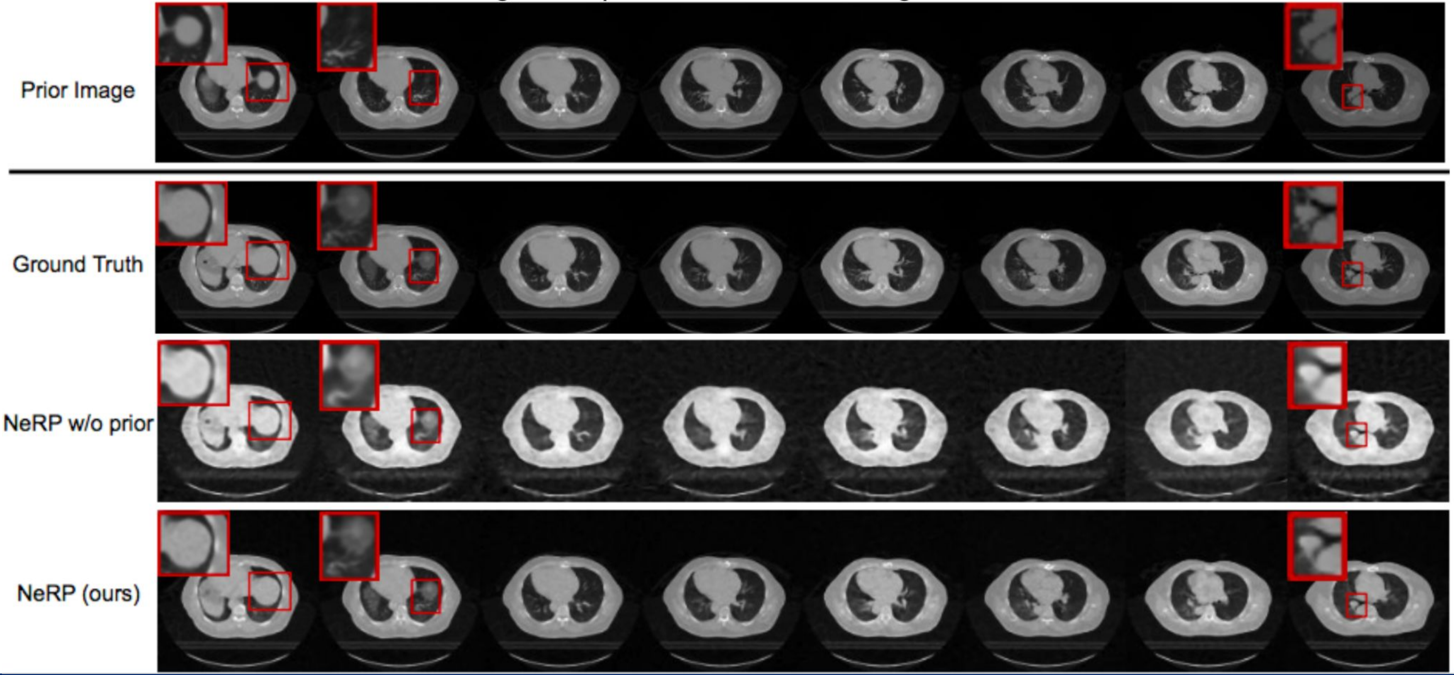
- Data-efficiency:
  - No large-scale data from external subjects is required to train model except for a longitudinal prior image and sparsely sampled measurements



NeRP: Neural Representation learning with Prior embedding

- Reliability:
  - Robustly capture subtle but significant structural changes required for assessing tumor shrink or progression

Clinical lung cancer patient case with two longitudinal 3D CT



NeRP: Neural Representation learning with Prior embedding

- Reliability:
  - Robustly capture subtle but significant structural changes required for assessing tumor shrink or progression

TABLE I
RESULTS OF 3D CT IMAGE RECONSTRUCTION
USING 5 / 10 / 20 PROJECTIONS ON DIFFERENT ANATOMICAL SITES

Methods	Pancreas CT	HeadNeck CT	Lung CT
Projections = 10			
FBP	17.95 / 0.461	23.05 / 0.653	21.49 / 0.597
GRFF [20]	28.07 / 0.855	29.38 / 0.864	27.80 / 0.835
NeRP w/o prior	28.88 / 0.850	30.40 / 0.858	30.98 / 0.880
NeRP (ours)	**37.66 / 0.981**	**36.92 / 0.976**	**32.73 / 0.941**
Projections = 20			
FBP	18.23 / 0.610	23.42 / 0.750	21.74 / 0.717
GRFF [20]	29.27 / 0.893	32.56 / 0.931	32.75 / 0.935
NeRP w/o prior	32.41 / 0.927	32.59 / 0.920	32.86 / 0.929
NeRP (ours)	**39.06 / 0.986**	**38.81 / 0.985**	**36.52 / 0.972**
Projections = 30			
FBP	18.31 / 0.650	23.54 / 0.773	21.83 / 0.7443
GRFF [20]	31.53 / 0.932	32.34 / 0.927	33.13 / 0.942
NeRP w/o prior	33.88 / 0.953	33.53 / 0.942	33.97 / 0.951
NeRP (ours)	**39.65 / 0.987**	**39.50 / 0.987**	**37.66 / 0.980**

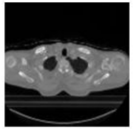
Evaluation metric: PSNR / SSIM values are reported.
PSNR (dB), peak signal noise ratio; SSIM, structural similarity.

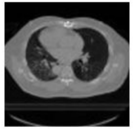
Our method with very sparse sampling:
17%~34% improvement in PSNR
13%~25% improvement in SSIM

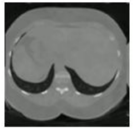
NeRP: Neural Representation learning with Prior embedding

- Generalization:
  - Easily generalize to different anatomic sites, imaging modalities, and sampling processes

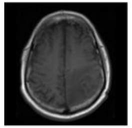
Different anatomic sites

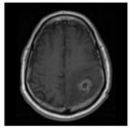

HeadNeck CT

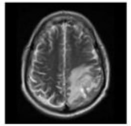

Lung CT

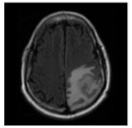

Pancreas CT

Different imaging modalities


T1 MRI


T1-contrast MRI


T2 MRI


FLAIR MRI

Different sampling processes


Different projections in CT

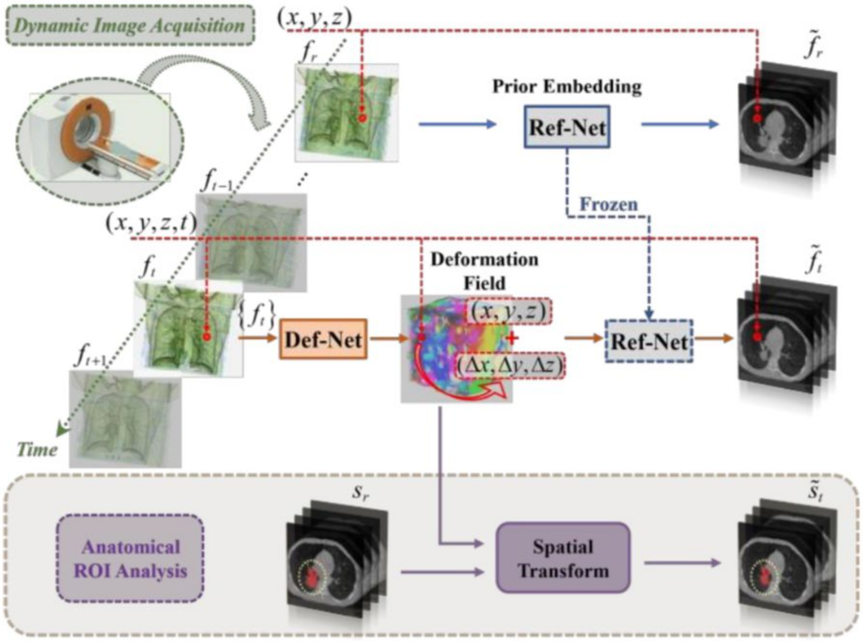
Different sampling masks in MRI

INR for biomedical imaging

- ~~Limitations:~~
  - ~~Unsatisfying quality for biomedical imaging reconstruction~~
- Advantages:
  - Small memory footprint
  - Easy to optimize
  - Leverage personalized prior knowledge
  - ?

INR for biomedical imaging

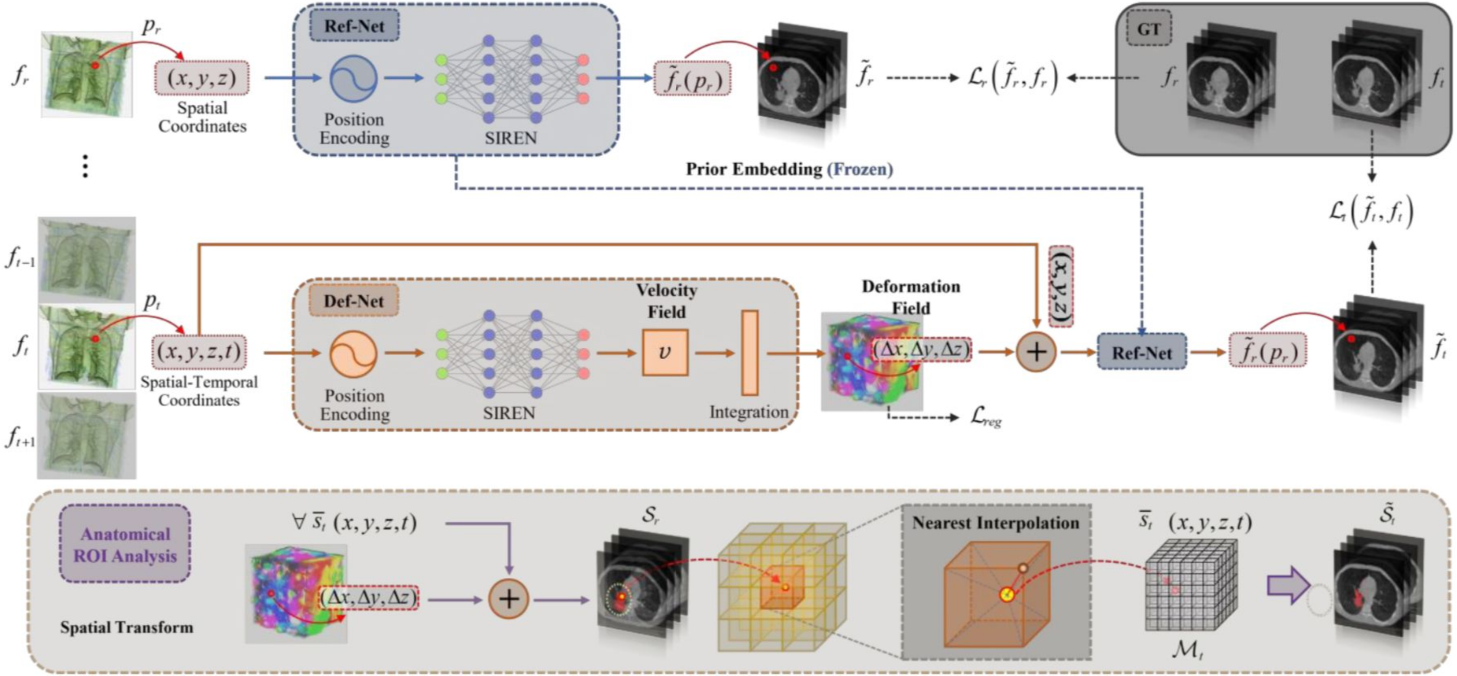
- D-NeRP: Dynamic Neural Representation Learning with Prior Embedding for Patient-Specific Longitudinal Imaging Study



Qiu, et al., under submission, 2023.

INR for biomedical imaging

- D-NeRP: Dynamic Neural Representation Learning with Prior Embedding for Patient-Specific Longitudinal Imaging Study



INR for biomedical imaging

- Advantages:
  - Small memory footprint
  - Easy to optimize
  - Leverage personalized prior knowledge
  - Longitudinal imaging study
  - ?

Today's agenda

- Neural scene representation
- Neural radiance fields (NeRF)
- Implicit neural representation learning (INR)
- INR in biomedical imaging
- Challenges

Today's agenda

- Neural scene representation
- Neural radiance fields (NeRF)
- Implicit neural representation learning (INR)
- INR in biomedical imaging
- Challenges

INR for biomedical imaging

- Challenges:
  - Good patient priors?
  - Model structure choice for MLP network?
  - Parameter choice for Gaussian embedding?
  - More efficient positional encoding methods?
  - High-resolution high-dimensional medical images?
  - Time-efficiency and memory-efficiency?
  - Incorporate population priors?
  - ……

Next lecture

- Generative diffusion models

Date	Lecture #	Topic	Papers	Instructor / Presenter
Tue 8/29	1	Introduction and course overview		Liyue Shen
Thu 8/31	2	Biomedical imaging with deep learning **[Fundamental]**		Liyue Shen
Tue 9/5	3	Implicit neural representation learning **[Advanced]**		Liyue Shen
Thu 9/7	4	Generative diffusion models **[Advanced]**		Liyue Shen
Tue 9/12	5	Medical image analysis **[Fundamental]**		Liyue Shen
Thu 9/14	6	Multimodal foundation models **[Advanced]**		Liyue Shen
Mon 9/18		Drop/add deadline for full term classes		
Tue 9/19	7	Implicit neural representation learning		
Thu 9/21	8	Implicit neural representation learning		
Tue 9/26	9	Implicit neural representation learning		
Thu 9/28	10	Implicit neural representation learning		
Tue 10/3	11	Generative diffusion models		
Thu 10/5	12	Generative diffusion models		
Tue 10/10	13	Generative diffusion models		
Thu 10/12	14	Generative diffusion models		
Tue 10/17		No class (fall study break)		
Thu 10/19	15	Self-supervised learning		
Tue 10/24	16	Self-supervised learning		
Thu 10/26	17	Multimodal learning		
Tue 10/31	18	Multimodal learning		
Thu 11/2	19	Transformer and LLM		
Tue 11/7	20	Transformer and LLM		