6.1820/MAS.453: Mobile and Sensor Computing
aka IoT Systems

https://6mobile.github.io/

Lecture 14: Wireless NeRFs

Some slides adapted from Amballa et. al

Course Staff	Announcements
Lecturers: Fadel Adib (fadel@mit.edu)	1- Lab 4 due April 7
TAs: Maisy Lam(mllam@mit.edu)	2- PSet 2 due April 9
Laura Dodds (lddods@mit.edu)	3- Midterm April 16

Today in IoT

- Did anyone travel for spring break?
- What were some issues you faced while traveling / in the airport?
  - Unknown wait times?

Pro:
- Already deployed
- High visual detail
- AI

Con:
- Occlusions
- Invade privacy

BOSTON'S MORNING NEWSLETTER

Logan Airport to launch new wait time tracker for security lines

By Hanna Ali
March 23, 2026

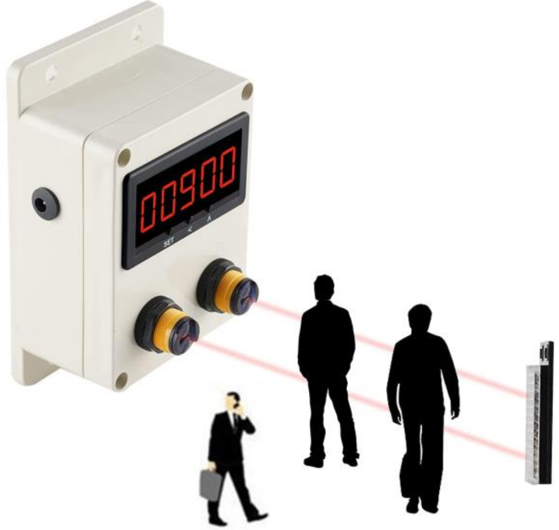
Share

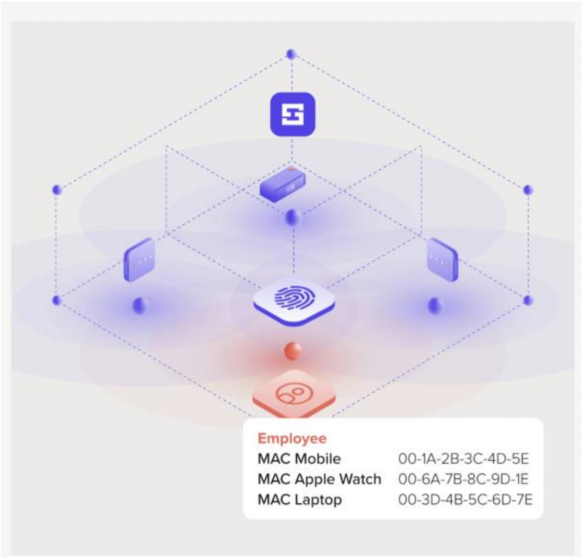
An overhead shot of passenger traffic through the JetBlue check-in area at Logan Airport in 2020. (Robin Lubbock/WBUR)

We have covered lots of sensing modalities!

- WiFi
- mmWave
- RFID
- Bluetooth
- IMUs
- Ultrasound
- Acoustics
- GPS
- LiDAR

How else can we build a queue detector?





This Lecture

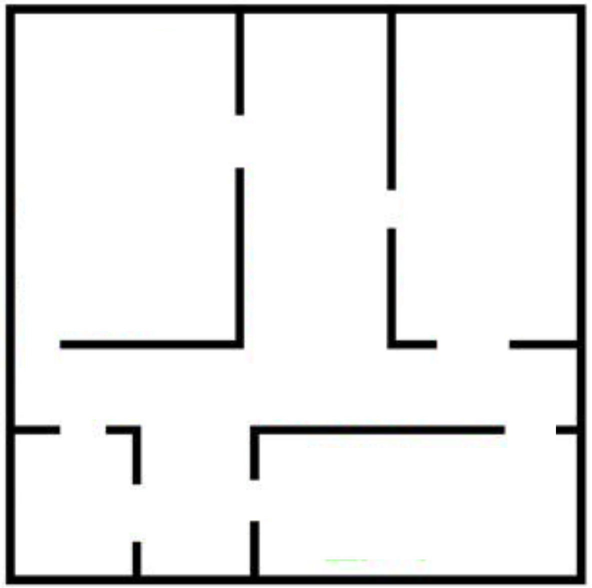
How can **NERFS** & **wireless signals** be used for **scene reconstruction**?

- EchoNeRF
- NERFs for vision
- Bonus: Alternative methods for indoor scene reconstruction using wireless signals

What is the goal of EchoNeRF?

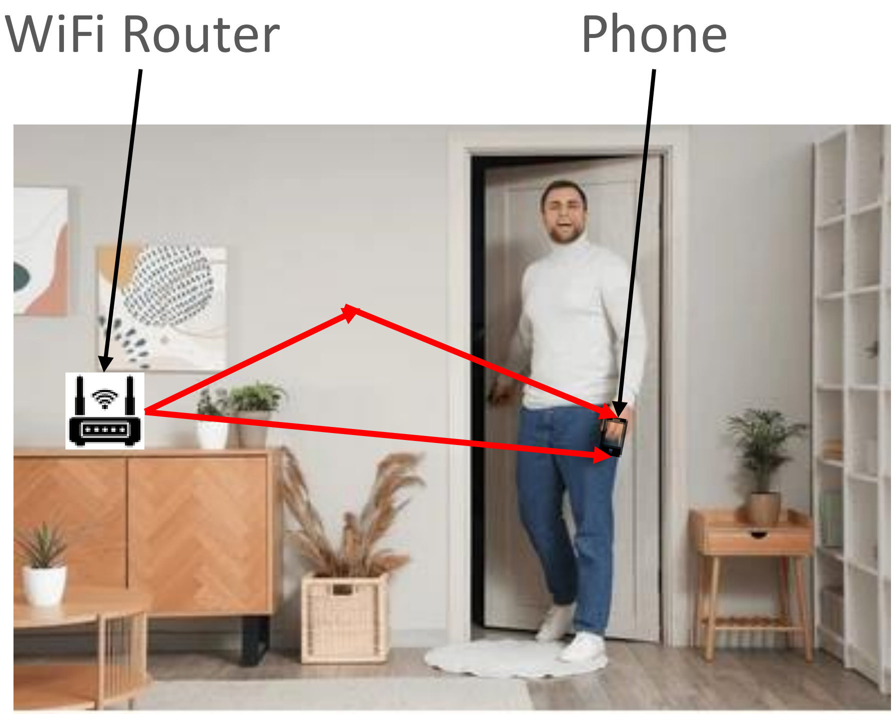
Learn a floor plan from a user walking with a phone

User walks around a building with phone, measuring WiFi signals

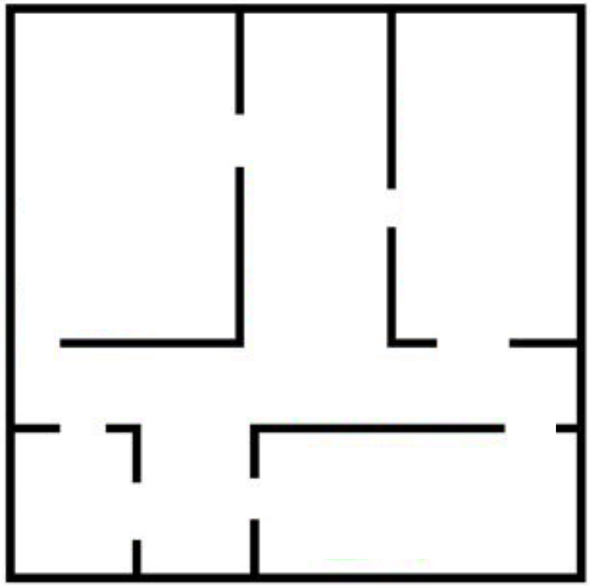


Floor plan of the room

EchoNeRF: Learn a floor plan from a user walking with a phone



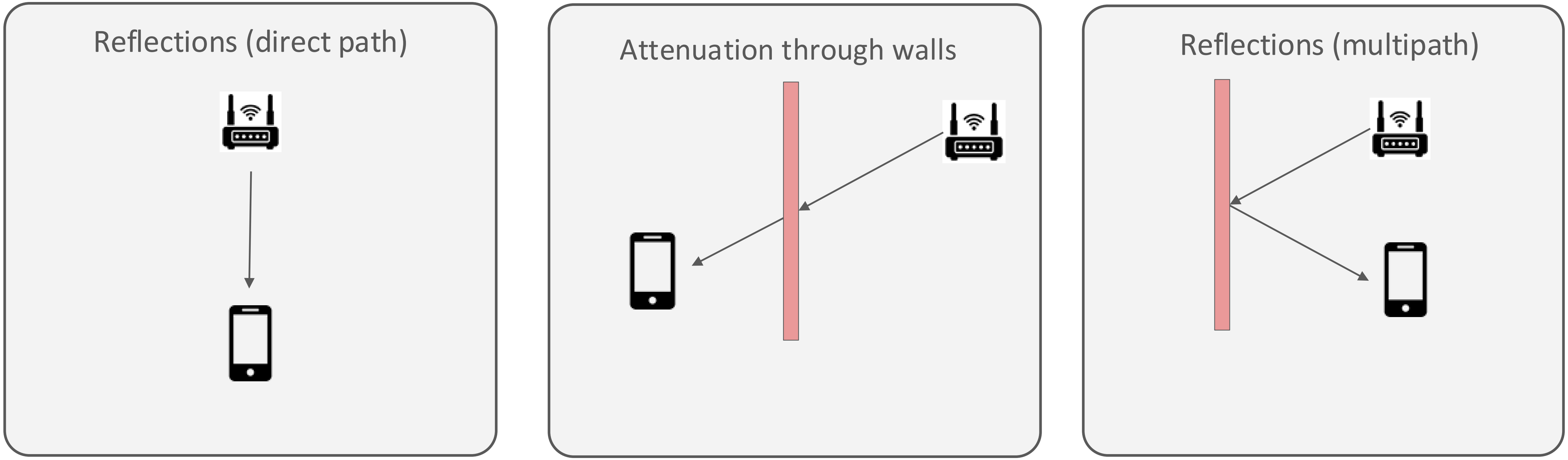
User walks around in a building with phone, measuring WiFi signals



Floor plan of the room

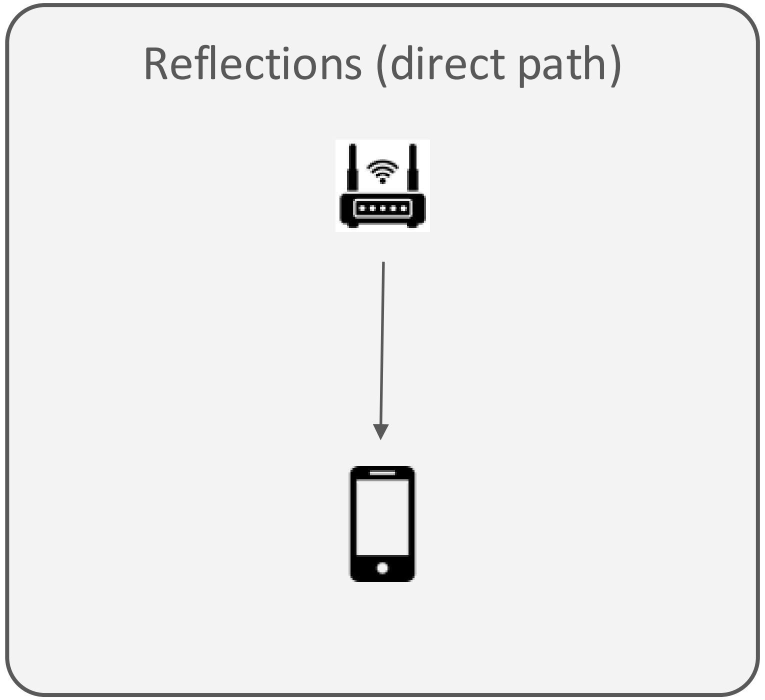
How do wireless signals travel indoors?

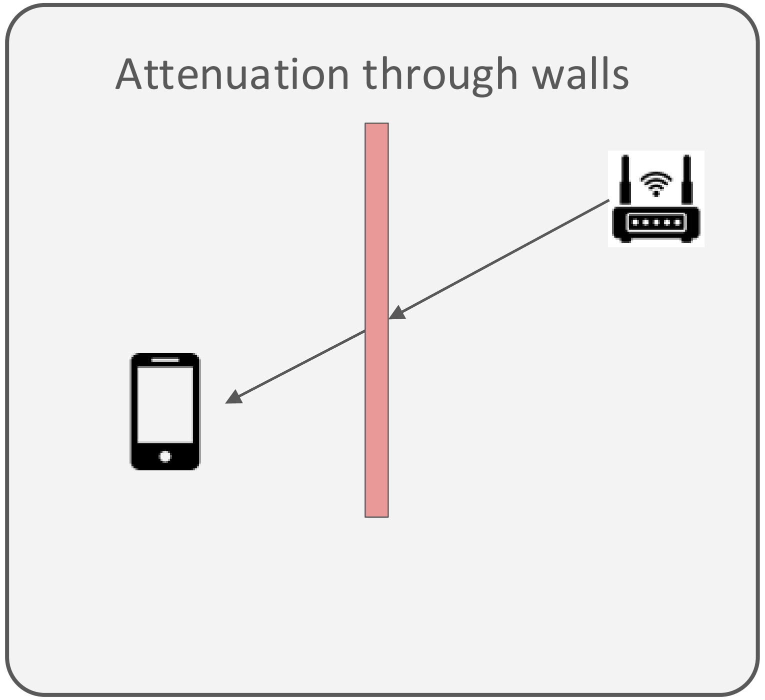
Indoor Signal Propagation

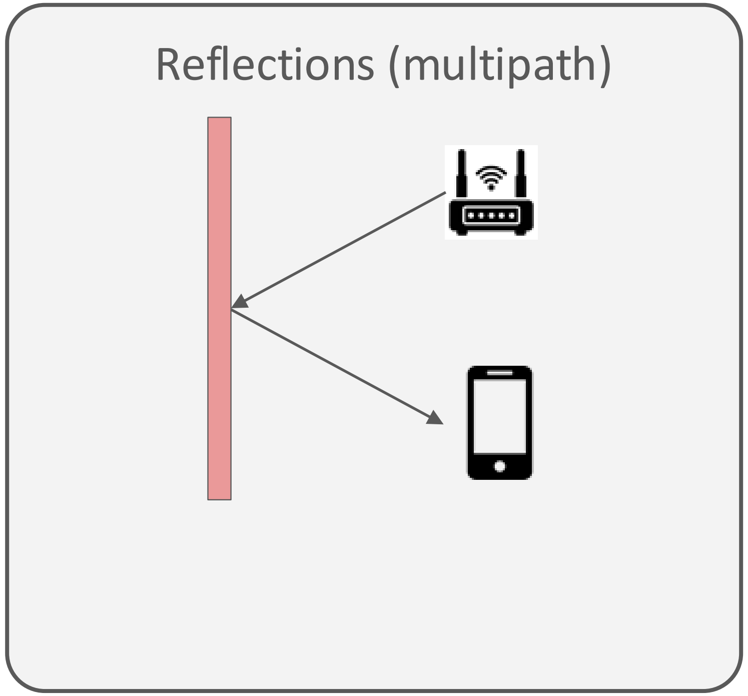


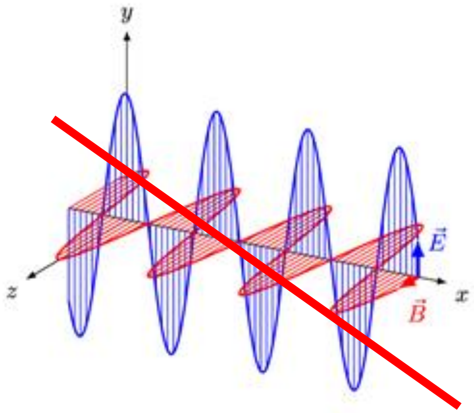
Different types of attenuation indoors

Indoor Signal Propagation

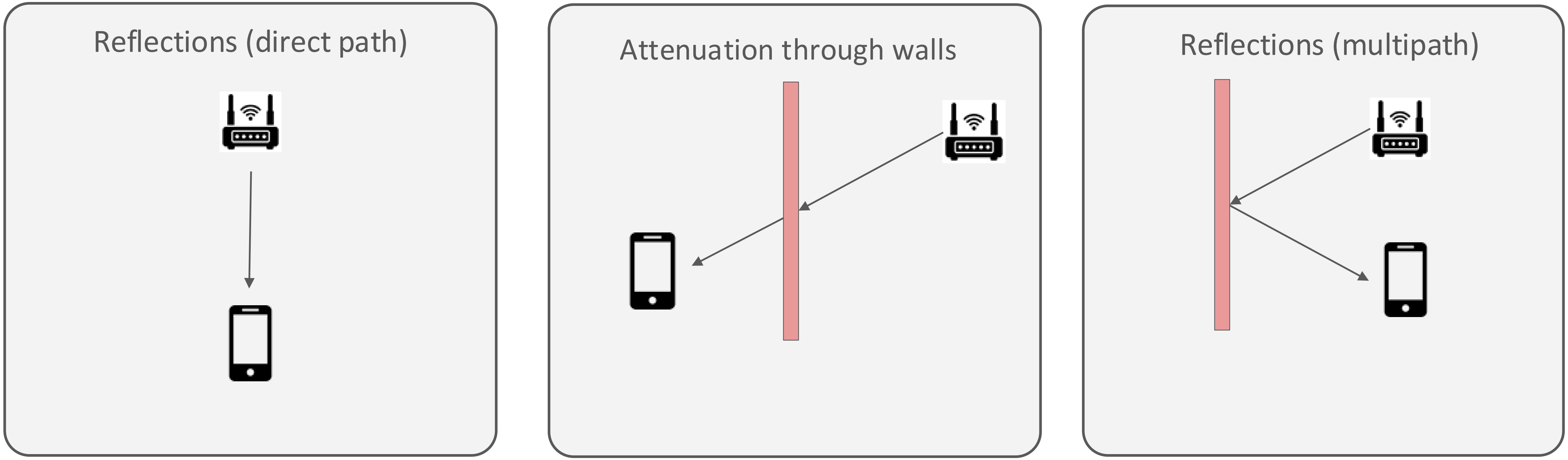


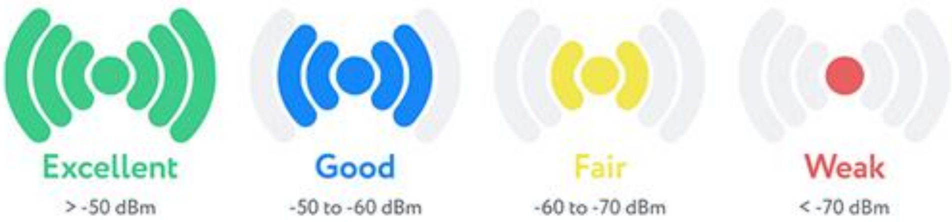






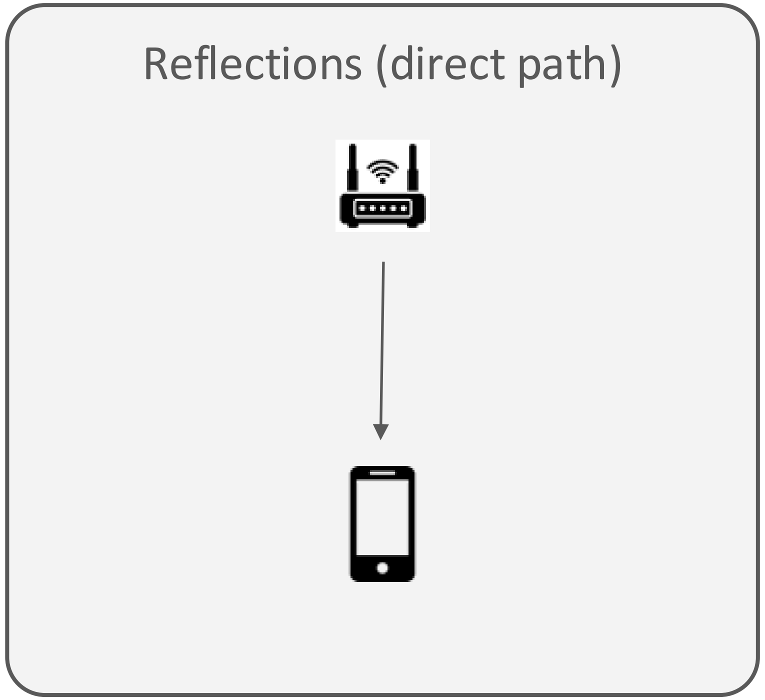
Indoor Signal Propagation

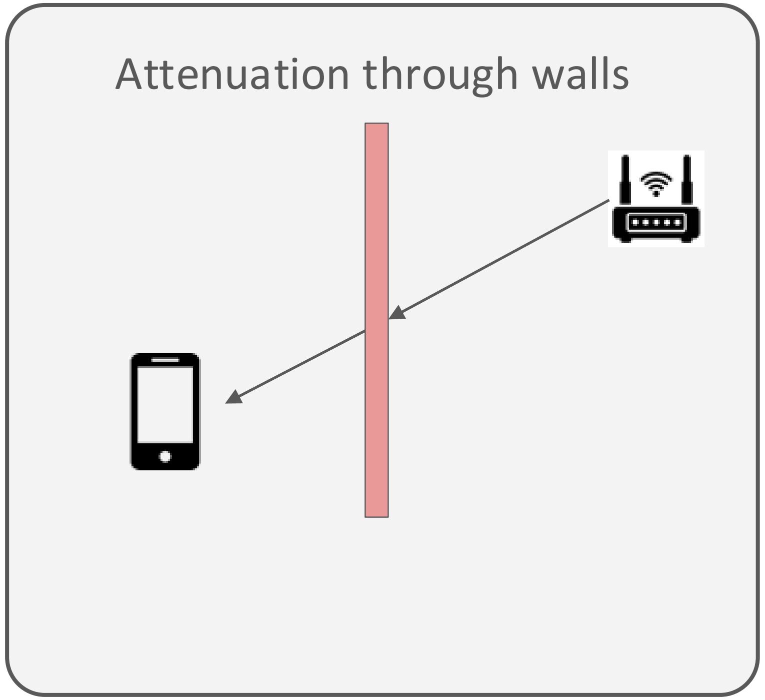


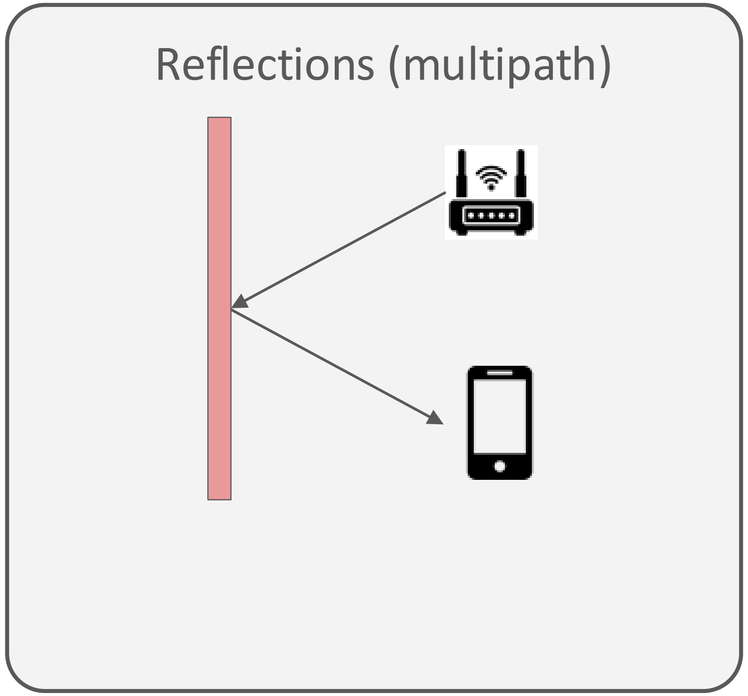


RSSI only provides a single number of signal strength

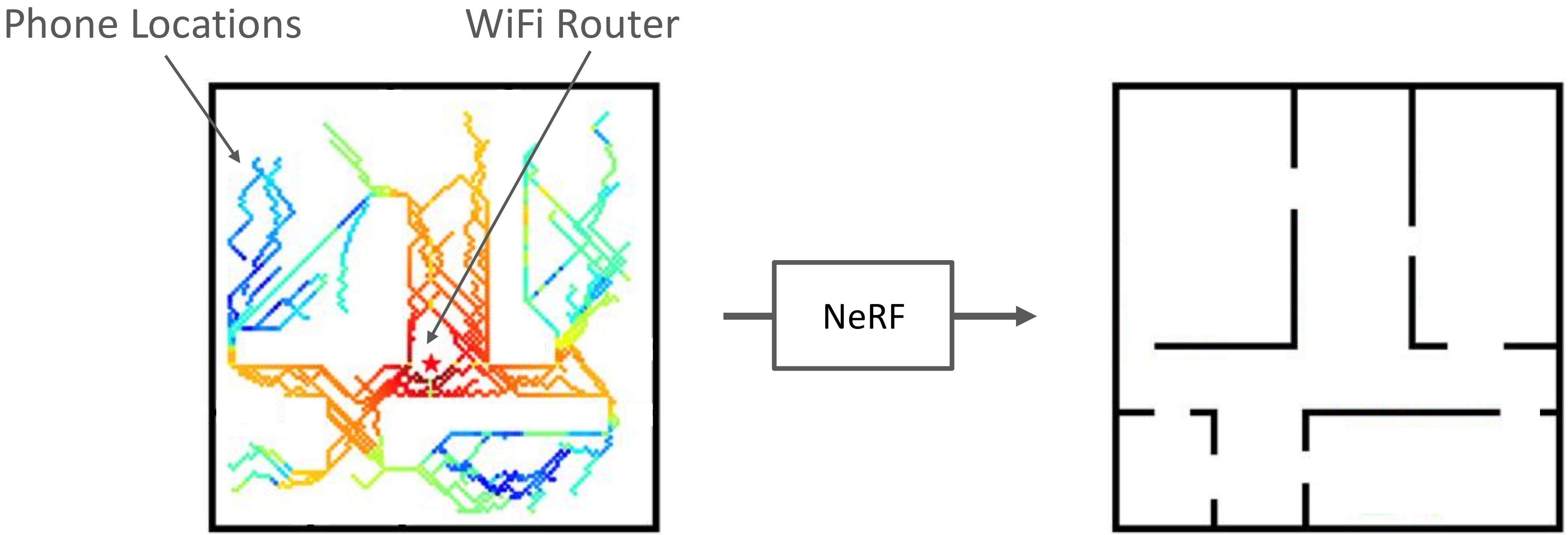
How does EchoNeRF model RSSI?







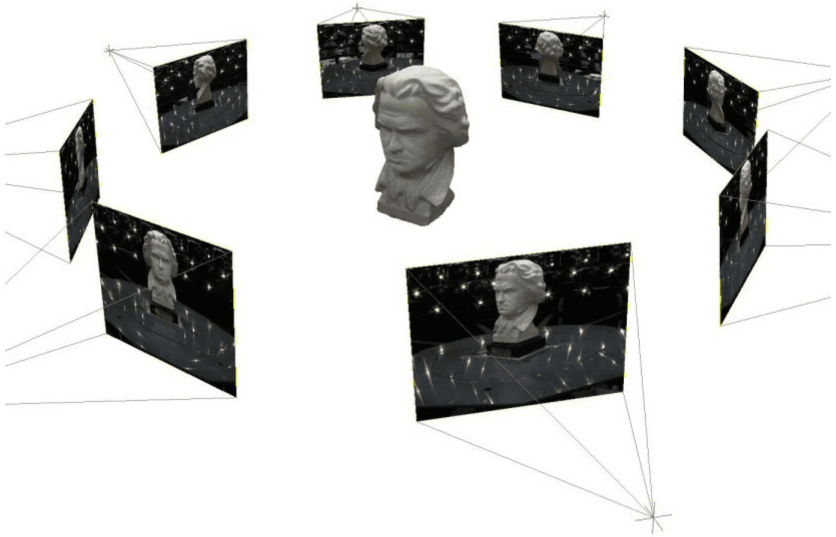
Problem Setup: Inputs and Outputs



User walks around in a building with phone, measuring WiFi signals

Floor plan of the room

Let’s pause and take a detour into computer vision …

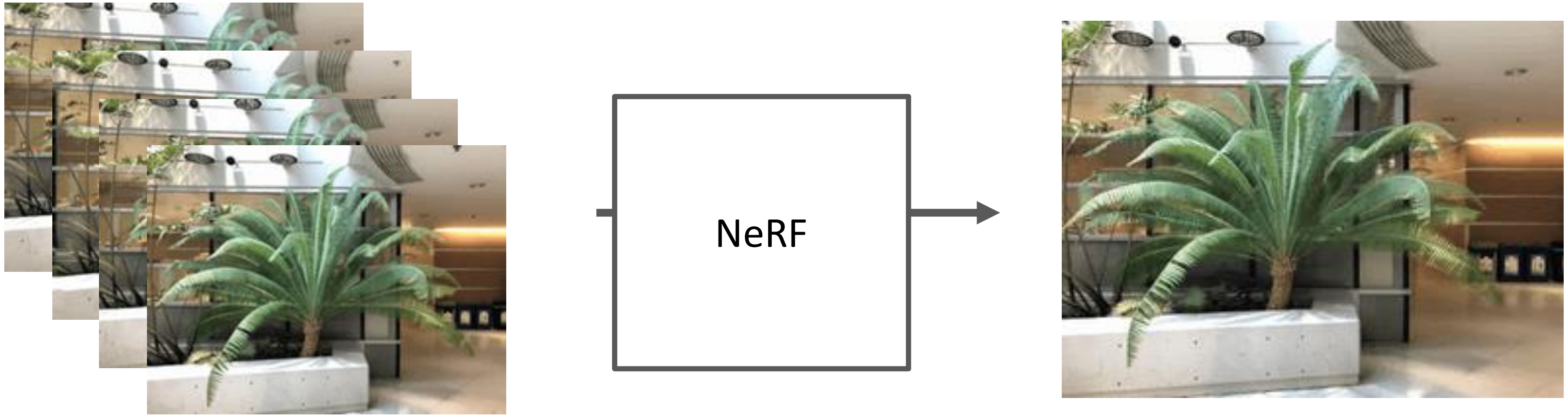


EchoNeRF

Vision NeRFs

Vision NeRFs

Goal: Generate novel views of a scene using set of input images



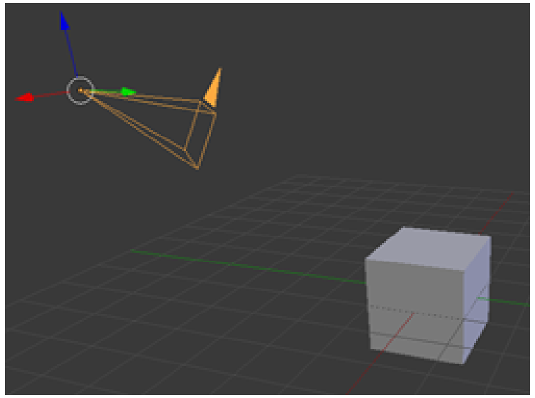
Discrete measurements

Novel views

Radiance Fields

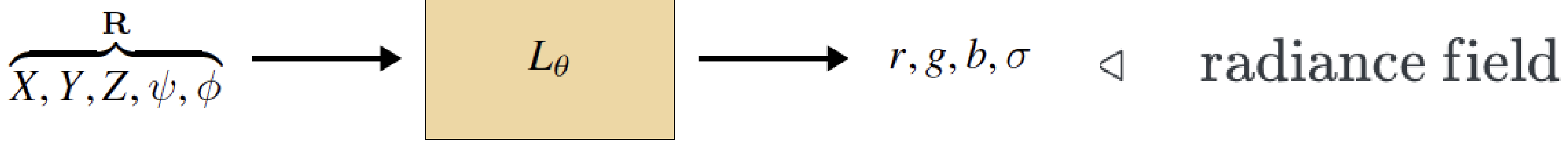
Function that maps coordinates in a scene to color and density (transparency)

X, Y, Z **viewing angle**



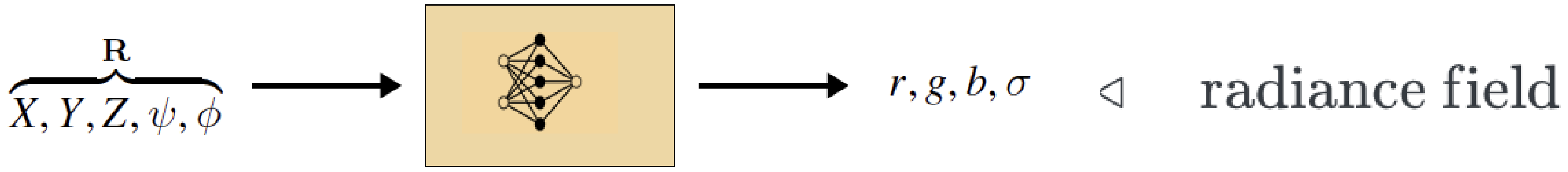
Radiance Fields

Function that maps coordinates in a scene to color and density (transparency)



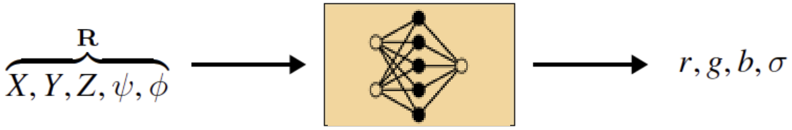
Neural Radiance Fields

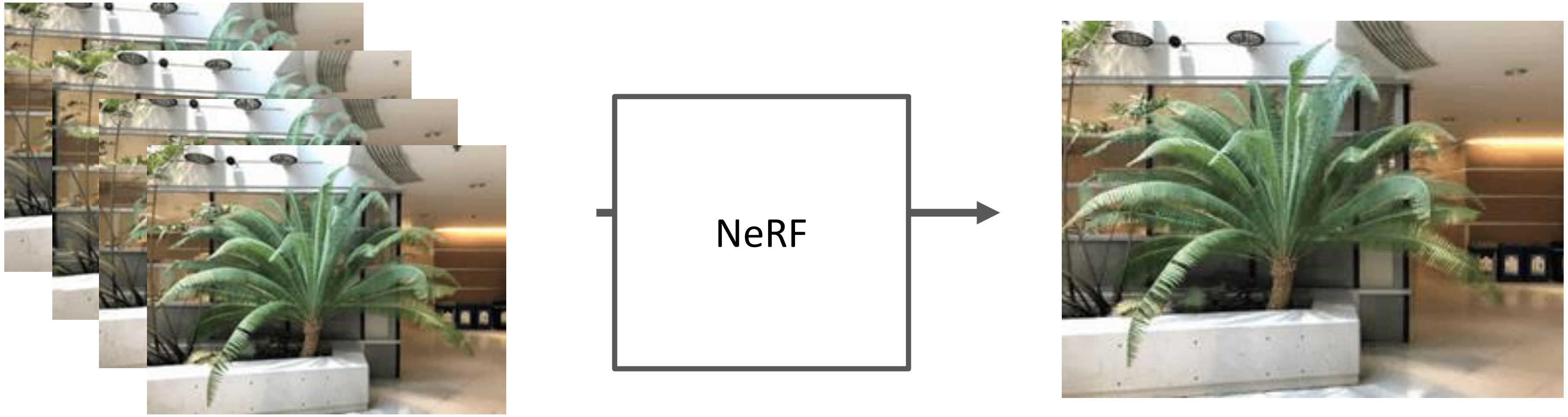
Use a neural network to map coordinates in a scene to color and density (transparency)



Neural Radiance Fields model L as a neural network (MLP)

How can we use this MLP to achieve our goal?

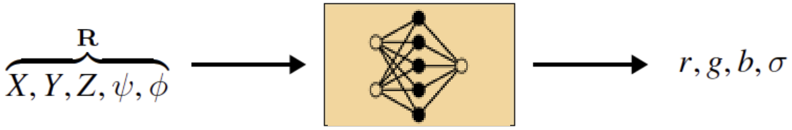




Discrete measurements

Novel views

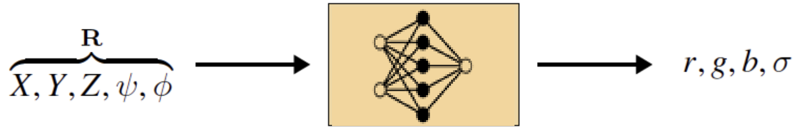
How can we use this MLP to achieve our goal?



1. Train the MLP on input images

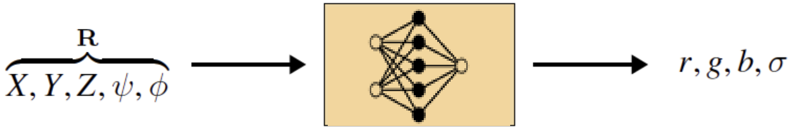
2. Use MLP to generate novel views

How can we use this MLP to achieve our goal?



1. **Train the MLP on input images**
2. Use MLP to generate novel views

Training the MLP

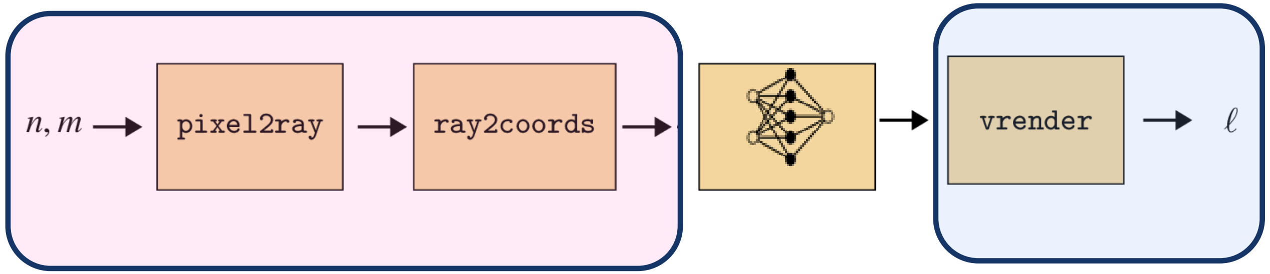


For every pixel in the set of input images:

Input Images

Pixel --> 3D coordinates

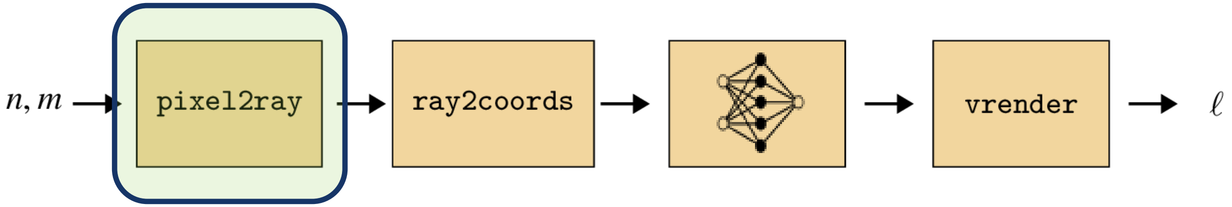
3D coordinate data --> Pixel value

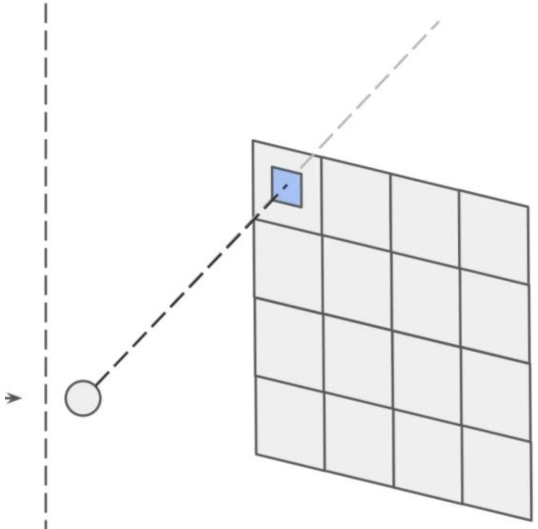


Predicted Images

Compute Loss

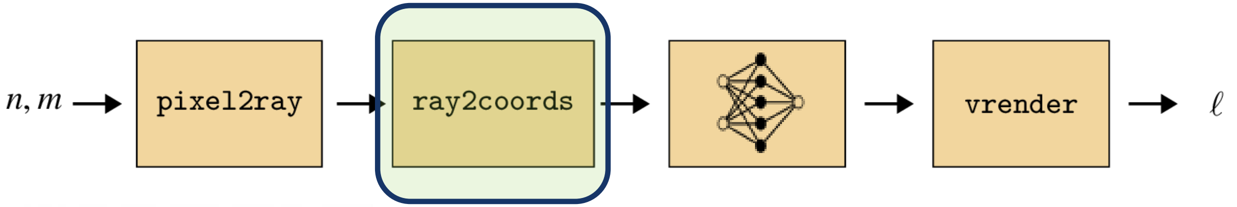
Step 1: Pixel to Ray

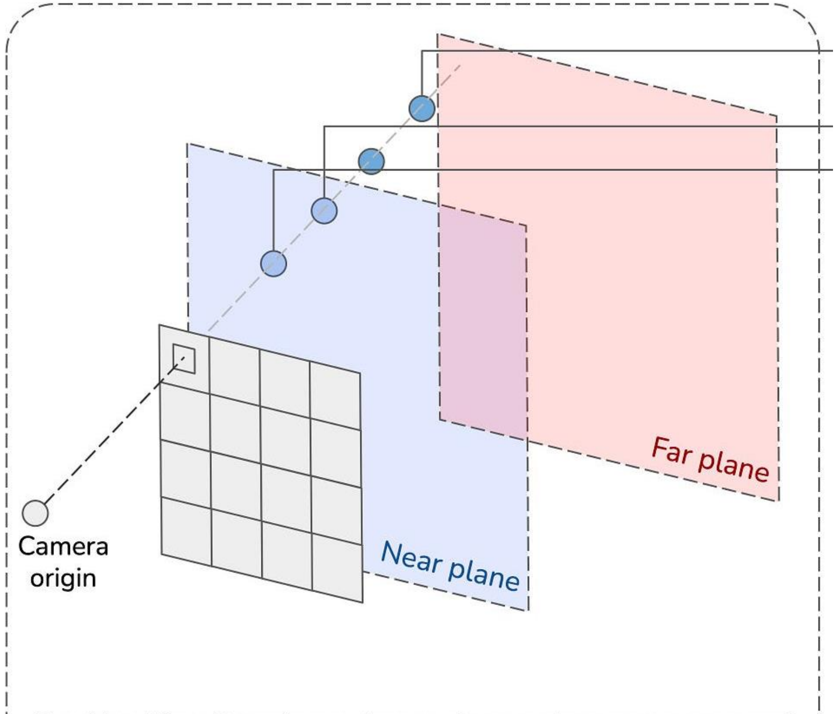




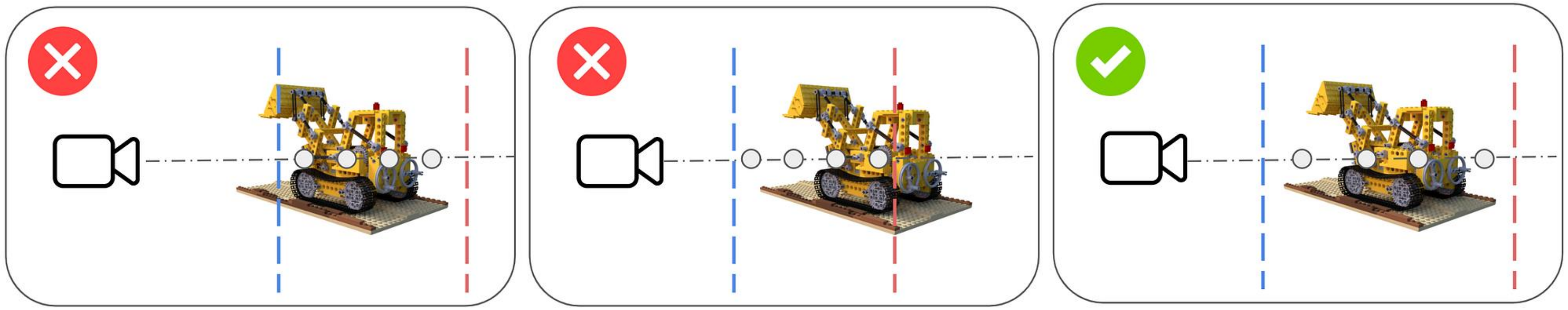
<u>Ray casting:</u> Draw a ray **r** passing through the center of the camera and the pixel (camera matrix)

Step 2: Ray 2 Coordinates



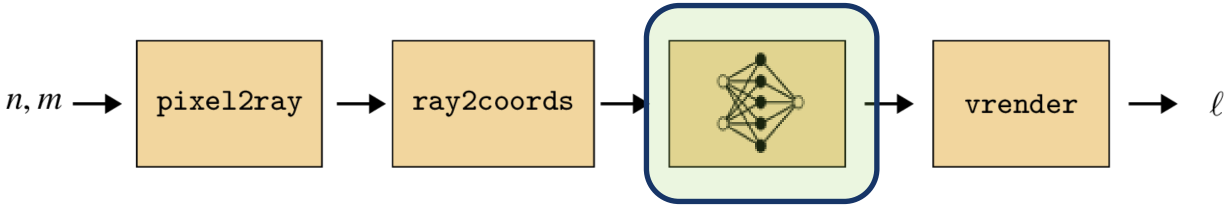


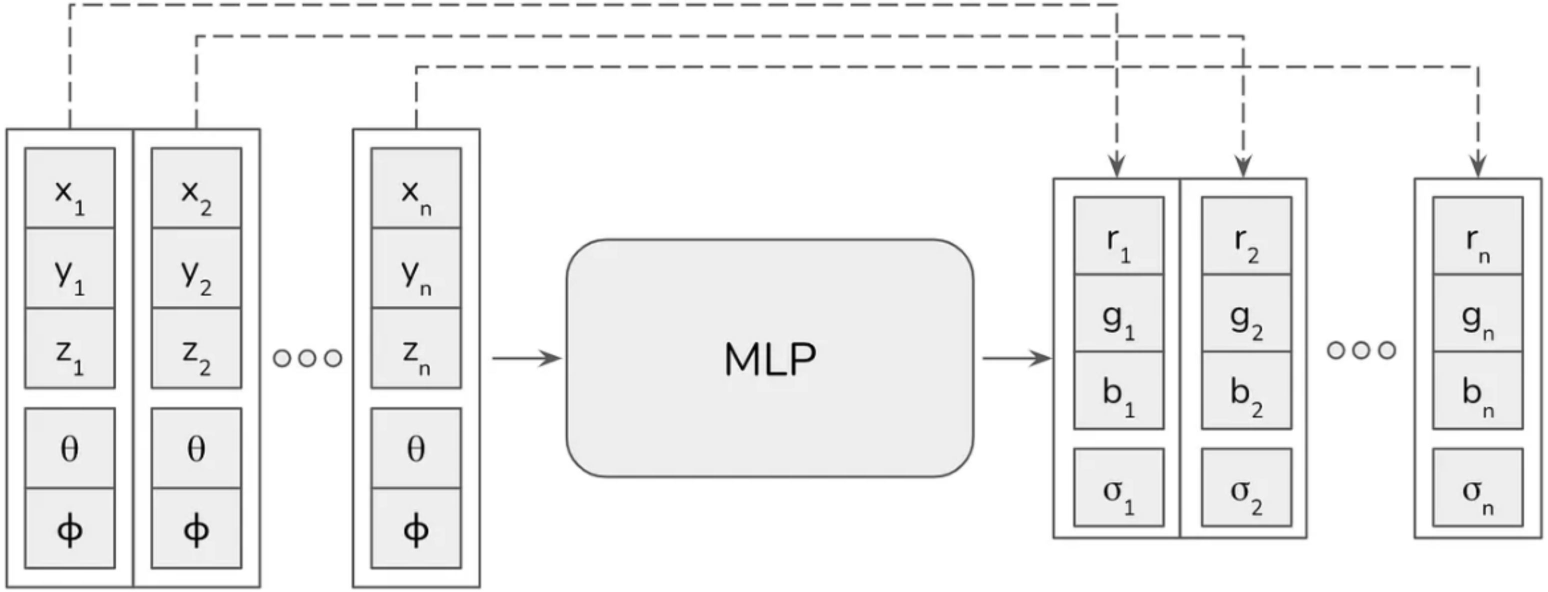
Ray Marching: Sample n points on the ray **r** between a near and a far plane



-- Near plane -- Far plane

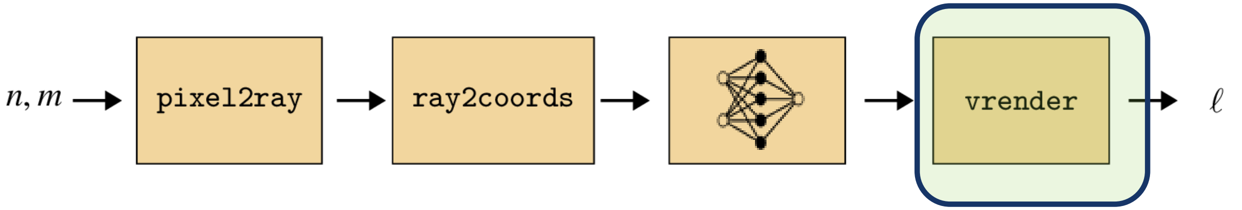
Step 3: Neural Radiance Field



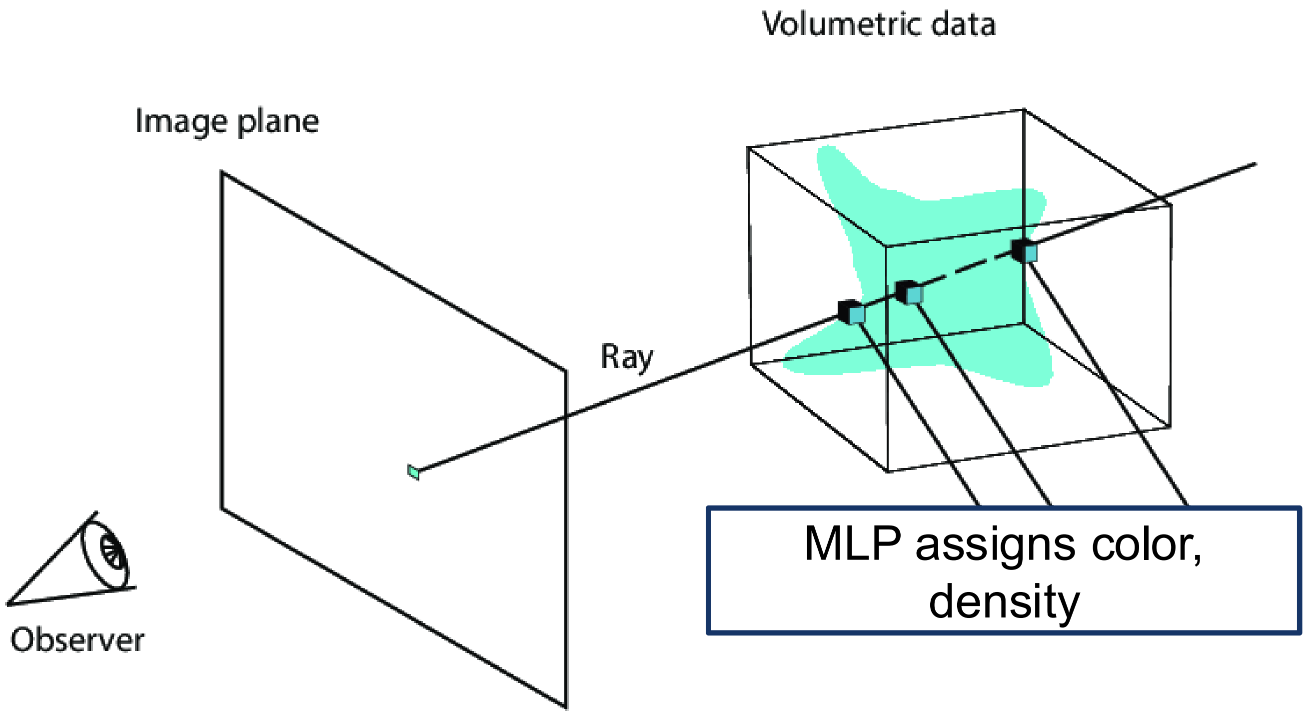


Neural Network: Introduce an MLP to estimate the color (r, g, b) and density σ for each sampled position (x_i, y_i, z_i) when viewed from a certain direction (θ,ϕ)

Step 4: Volume Rendering



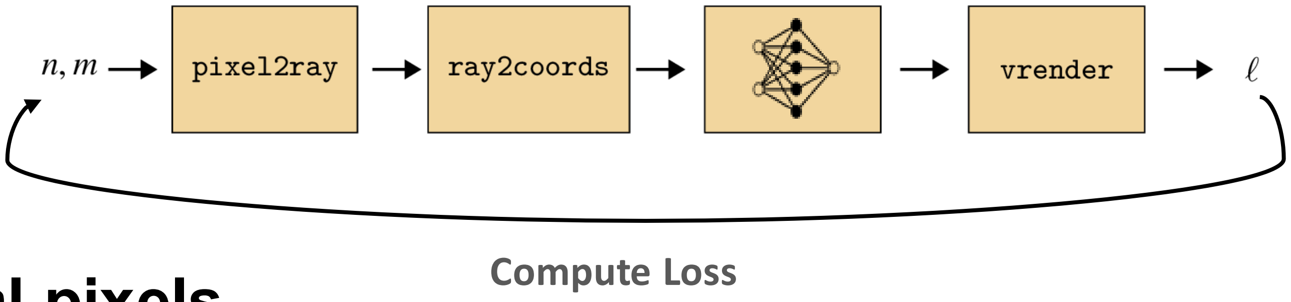
From the camera, a pixel is the integral over a ray through the 3D scene



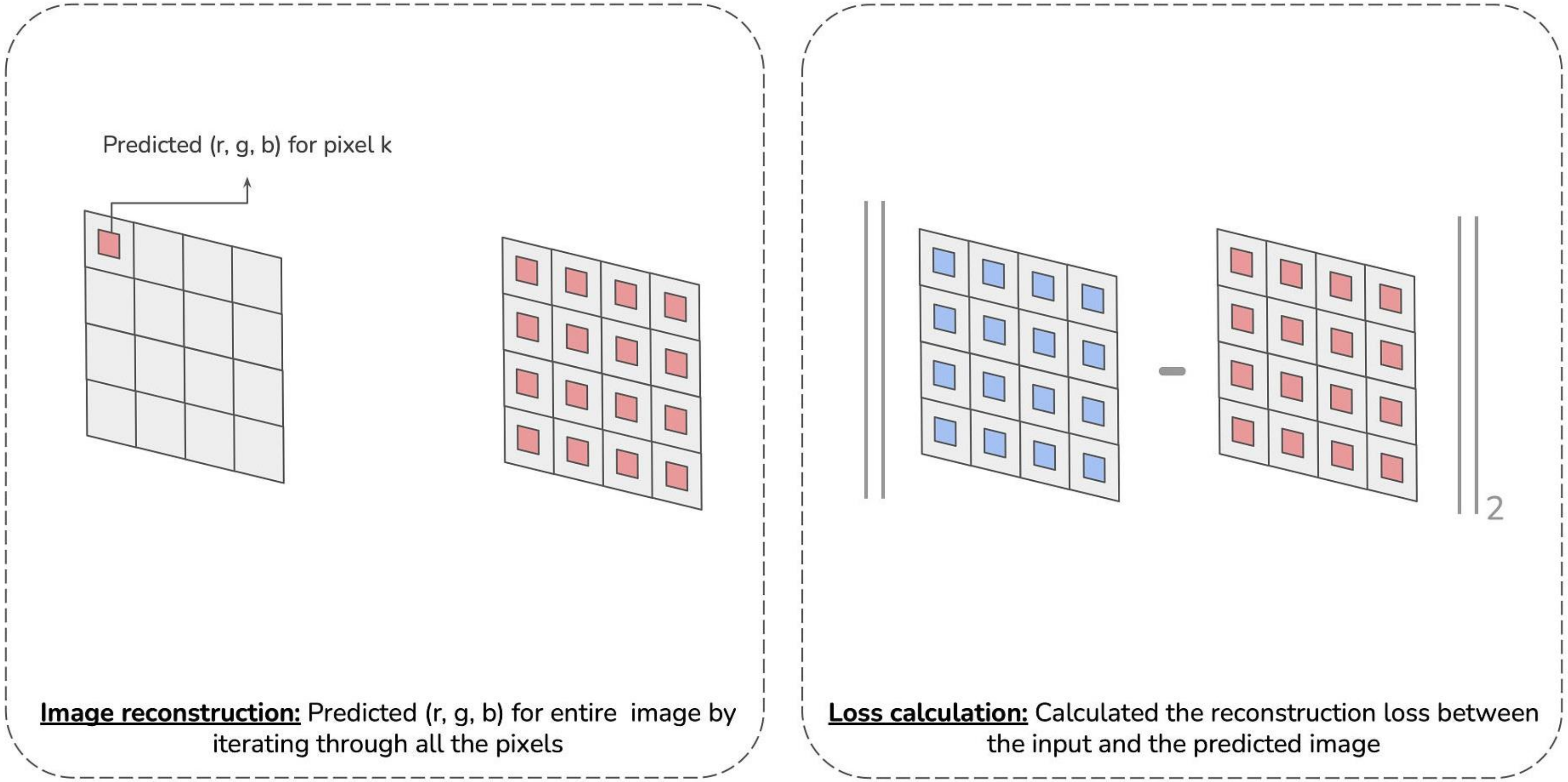
$$\ell(r) = \int_{t_n}^{t_f} \alpha(t) \overbrace{L^\sigma(r(t))}^{\text{density}} \overbrace{L^c(r(t), \mathbf{D})}^{\text{color}} \, dt$$

$$\alpha(t) = \exp\left(- \int_{t_n}^{t} \underbrace{L^\sigma(r(t)) dt}_{\text{density}} \right)$$

Step 5: Compute Loss

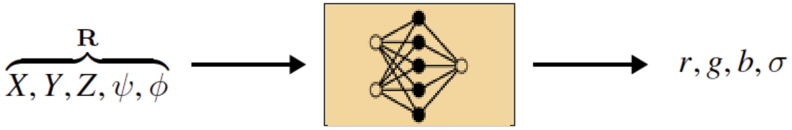


Loss computed: Predicted pixels - real pixels



MLP is overfitted to only work for the given scene

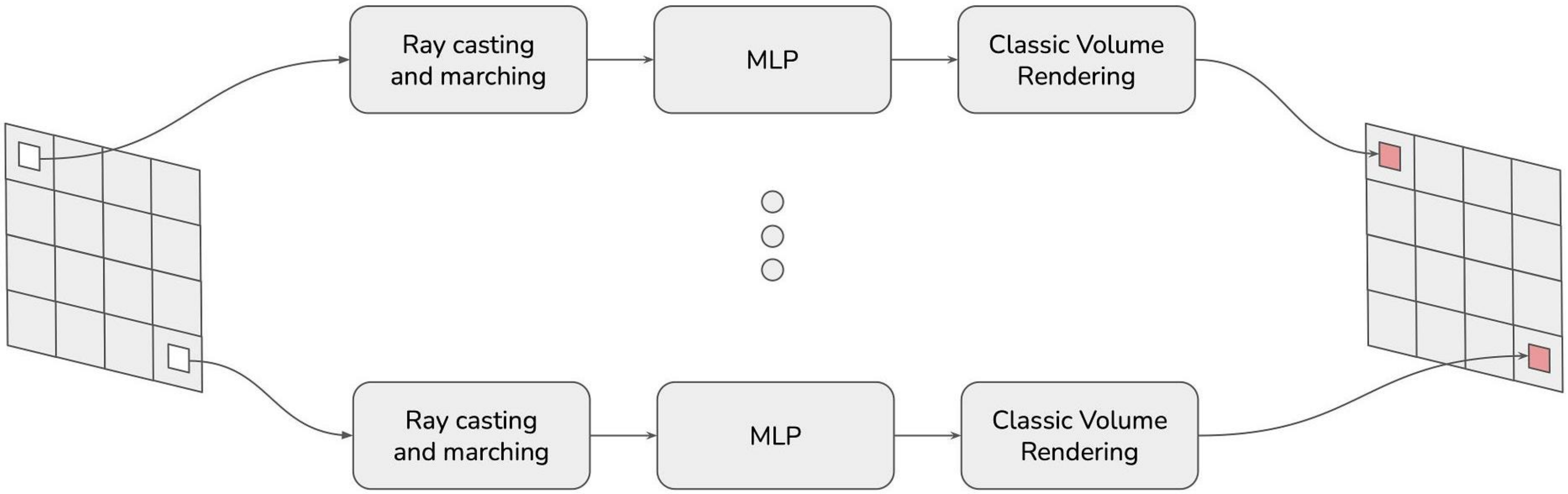
How can we use this MLP to achieve our goal?



1. Train the MLP on input images

2. **Use MLP to generate novel views**

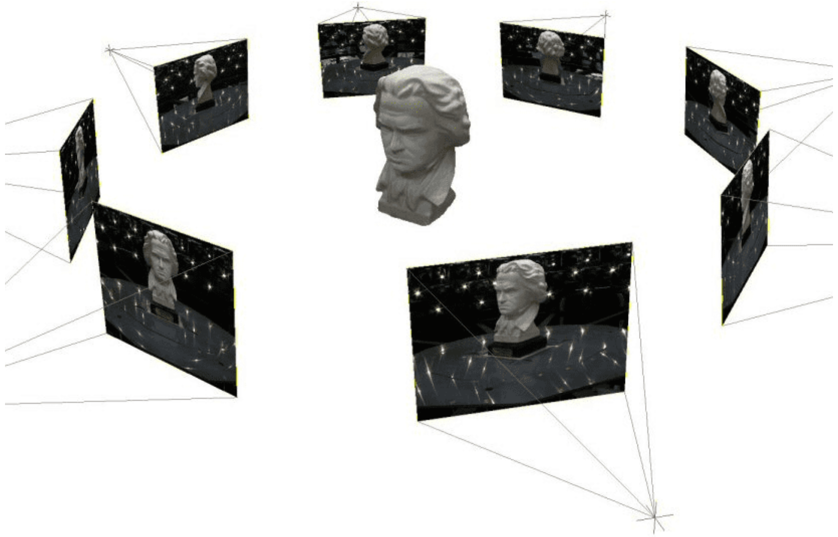
Use MLP to generate novel views



Rendering image from different viewpoint: Move to the viewpoint and render each pixel of the 2D image independently. For each pixel,

a. Find the view angle of the ray (θ,ϕ), and sample n points (x, y, z) along the ray between a near and far field.
b. Pass these n # of 5D tensors to the trained MLP and get the estimated (r,g,b) for the pixel using classic volume rendering

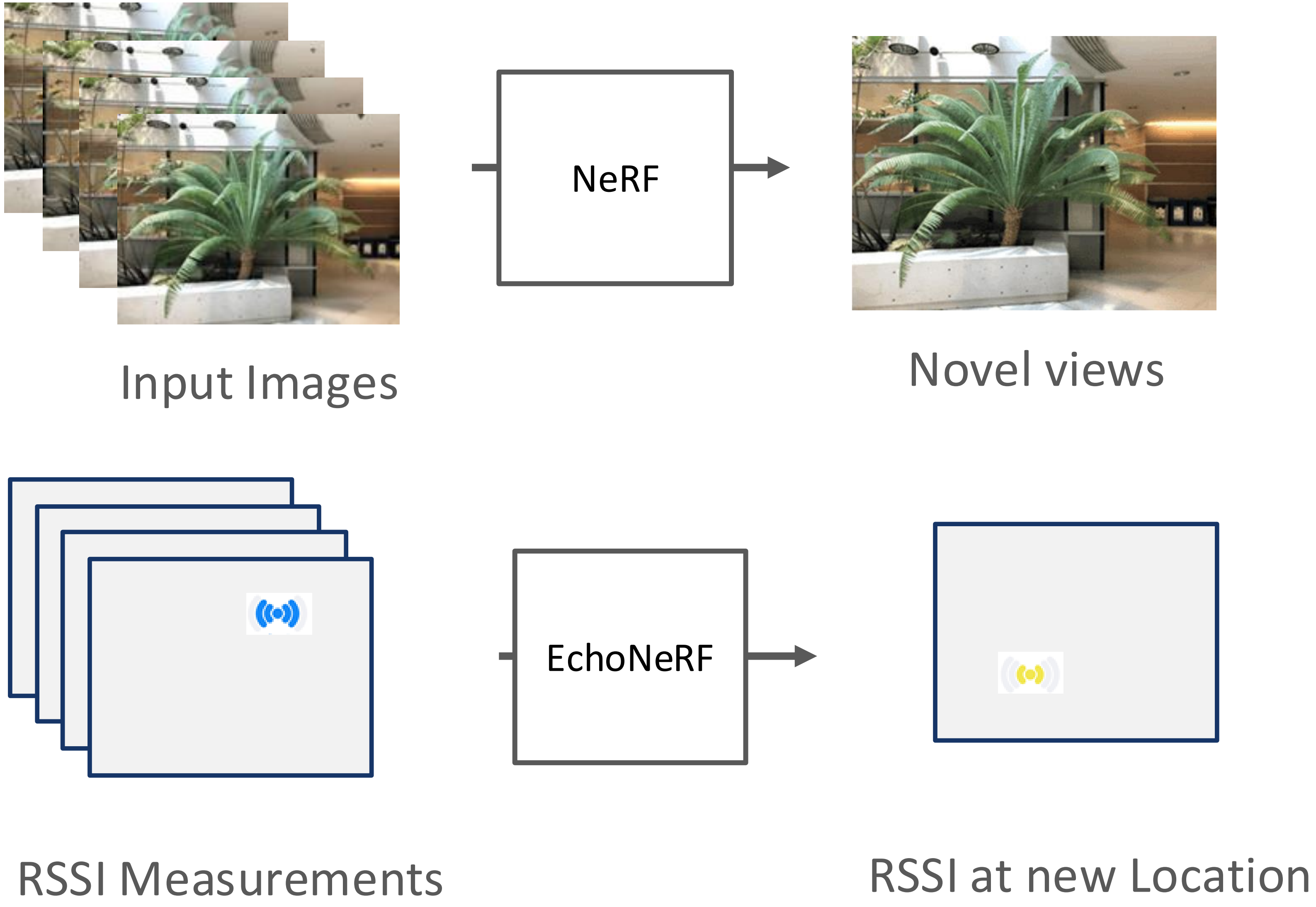
Let's go back to our Wireless Case



Vision NeRFs

EchoNeRF

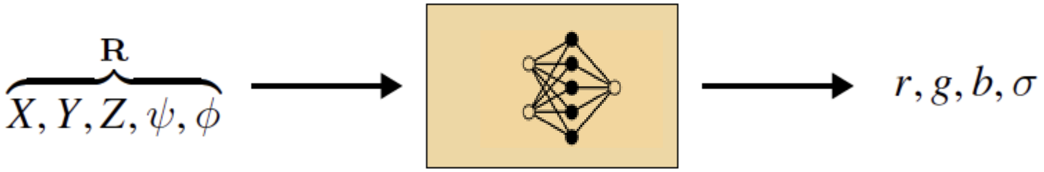
How can we remodel this for wireless



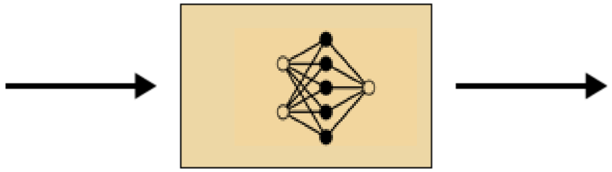
How can we remodel this for wireless

Neural Radiance Field Mapping

NeRF: Map coordinates in a scene to color and density (transparency)



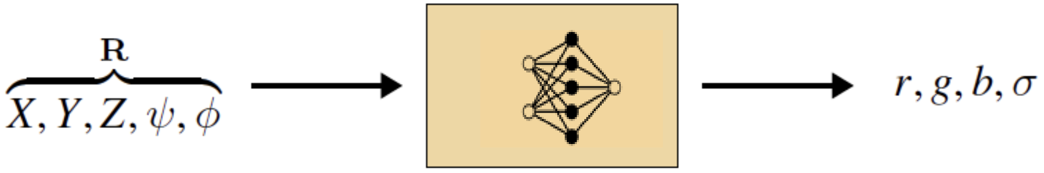
EchoNeRF: ??



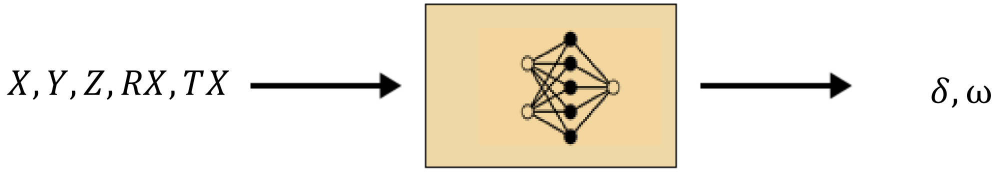
How can we remodel this for wireless

Neural Radiance Field Mapping

NeRF: Map coordinates in a scene to color and density (transparency)



EchoNeRF: Map coordinates to **density and orientation**



$$\delta \in [0,1]$$

$$\omega \in [-\pi, \pi]$$

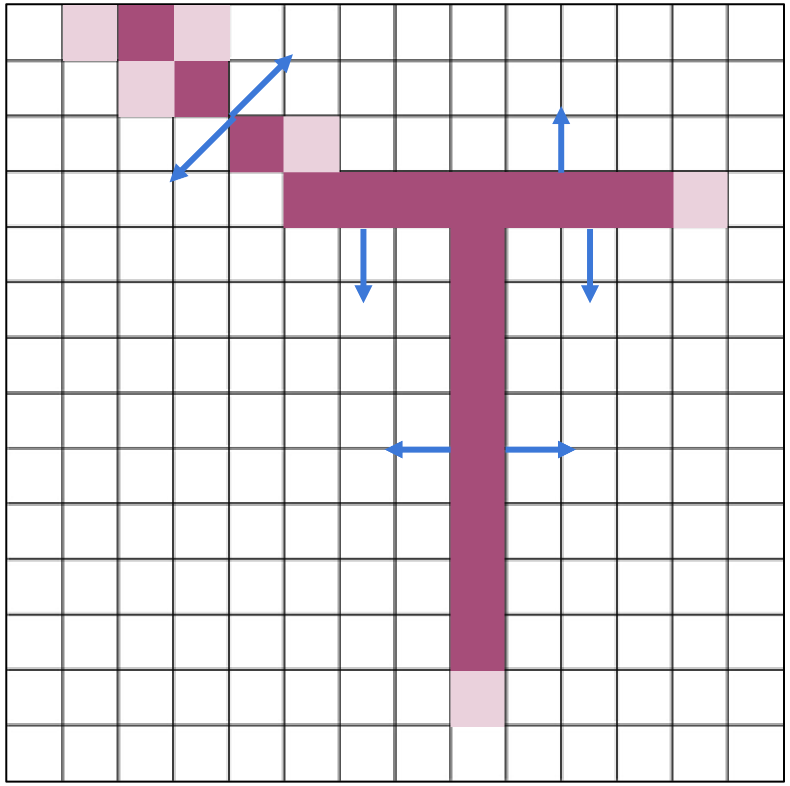
EchoNeRF - Physical Model

Their NeRF represents the space as a **2D** grid of cells.

Each cell contains:

1. **δ - opacity value** (closer to air or wall)
2. **ω - normal direction** (angle)

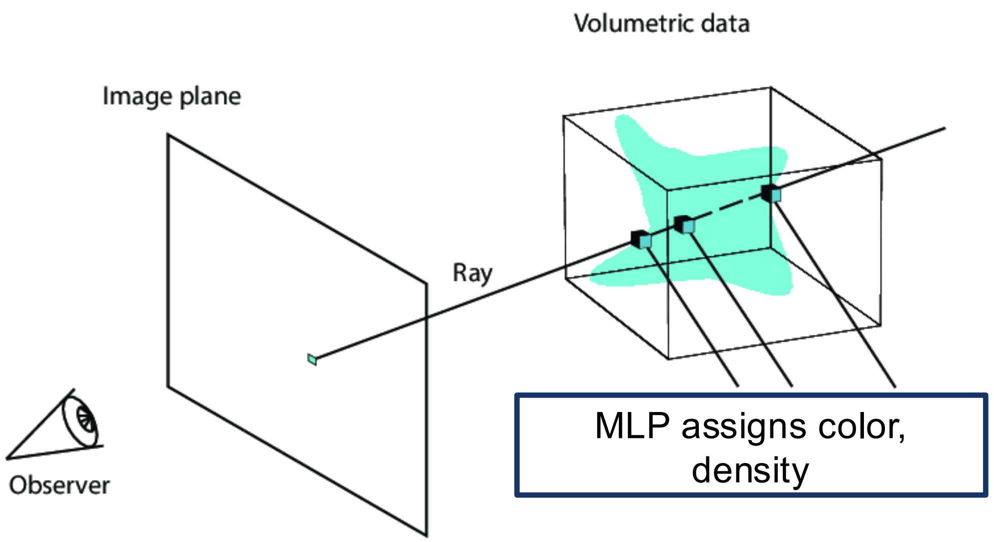
EchoNeRF's NN outputs (δ, ω) for every point in the grid



How can we remodel this for wireless

Rendering

NeRF: Uses volume rendering gets pixel from sampling and integrating 3D space

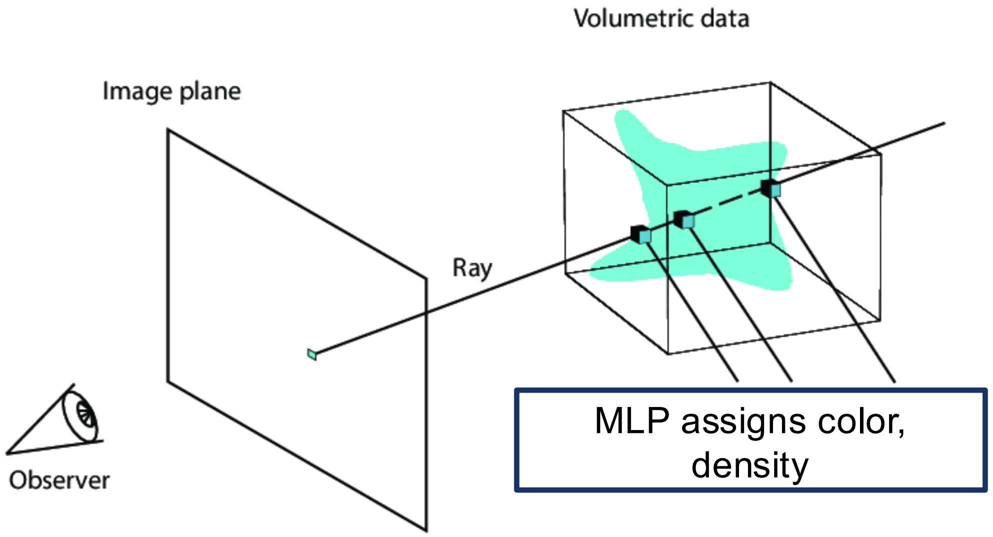


EchoNeRF: Uses …

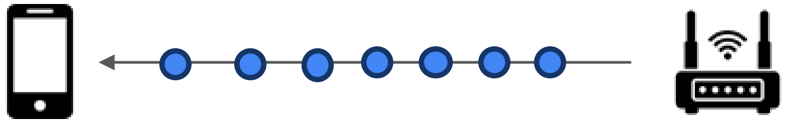
How can we remodel this for wireless

Rendering

NeRF: Uses volume rendering gets pixel from sampling and integrating 3D space



EchoNeRF: Uses signal models to get RSSI value from sampling 3D space

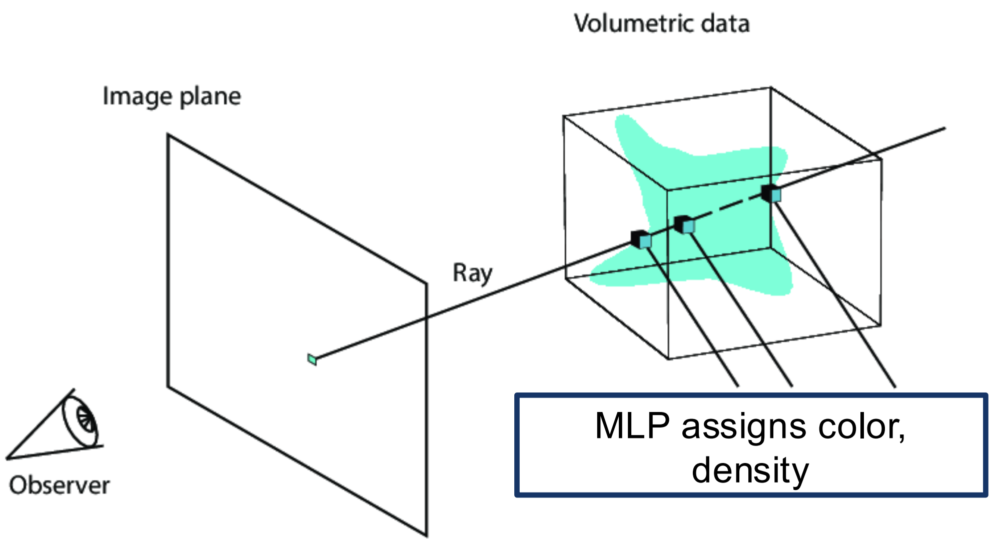


An RX receives input from many different rays!

How can we remodel this for wireless

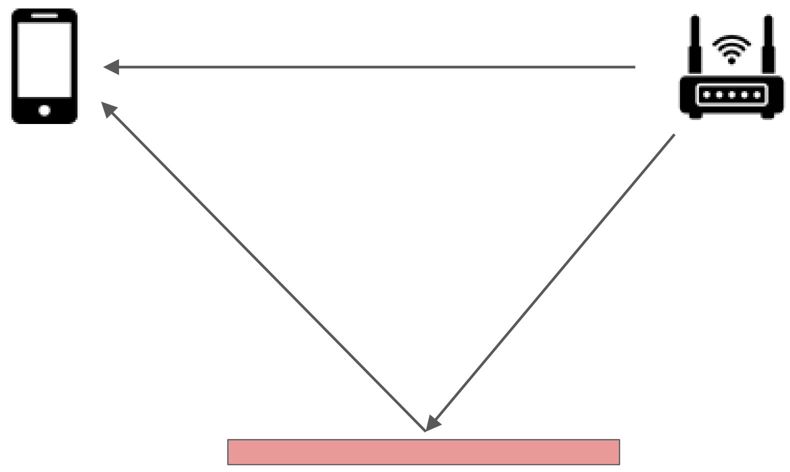
Rendering

NeRF: Uses volume rendering gets pixel from sampling and integrating 3D space



Cameras only have 1 primary ray

EchoNeRF: Uses signal models to get RSSI value from sampling 3D space



An RF antenna receives input from many different rays!

Modeling RSSI

Modeling RSSI by sampling space

Line of Sight:

$$\psi_{LoS} = K \frac{\prod_{\{i|v_i \in LoS\}} (1 - \delta_i)}{d^2}$$

First Order Reflection:

$$\psi_{ref}(v_j) = \delta_j f(\theta, \beta) \frac{\prod_{k \in \{Rx:v_j\}} (1 - \delta_k) \prod_{l \in \{v_j:Tx\}} (1 - \delta_l)}{\left(d_{Tx:v_j} + d_{v_j:Rx}\right)^2}$$

$$\psi_{ref_1} = \sum_{\{j|v_j \in \mathcal{V}\}} \psi_{ref}(v_j)$$

Final Modeled RSSI:

$$\tilde{\psi} = \psi_{LoS} + \psi_{ref_1}$$

EchoNeRF - Physical Model

Rules for Radio Propagation:

1. Line of sight propagation is proportional to $1/d^2$

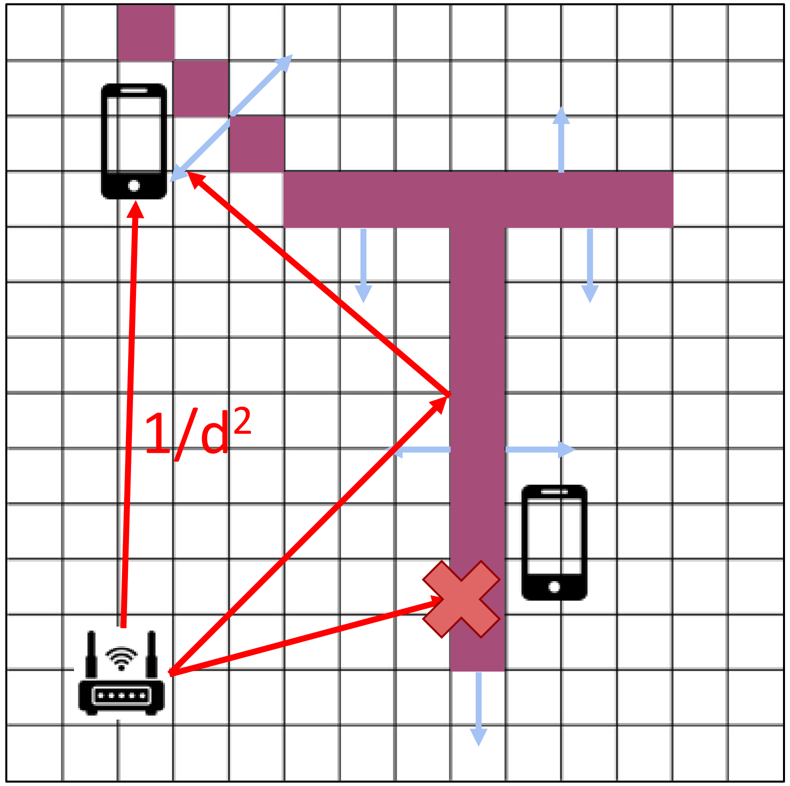
2. Walls completely block LoS propagation

3. Reflections off walls are specular

4. Signals *powers* from different paths add

   Real signals could constructively or destructively interfere.

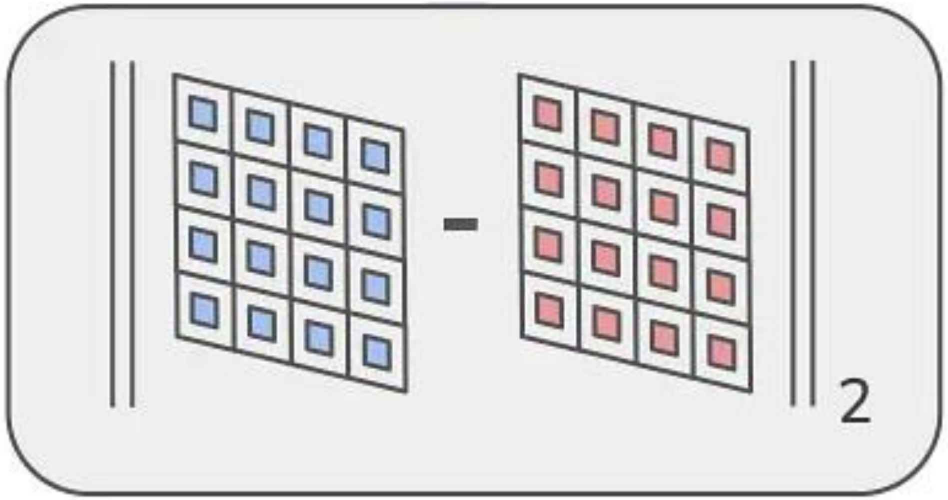
   Without phase information, this is the best we can do. The authors argue it holds up in this scenario (many paths, wideband signal)



How can we remodel this for wireless

Loss Function

NeRF: Pixel based loss

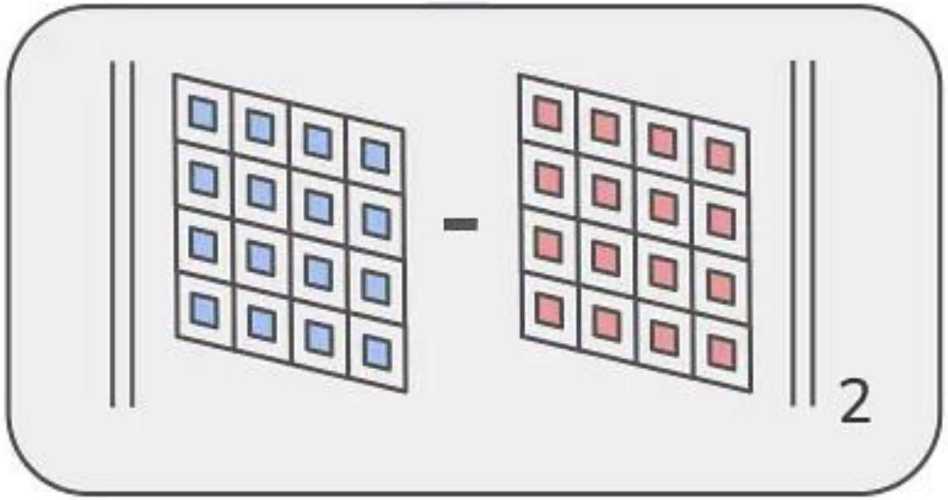


EchoNeRF: ???? based loss

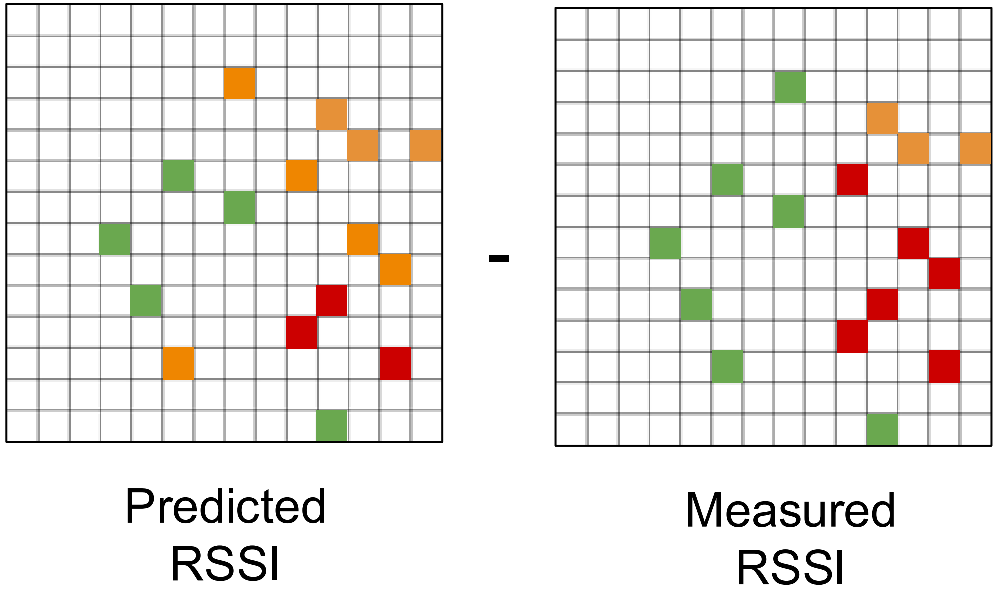
How can we remodel this for wireless

Loss Function

NeRF: Pixel based loss



EchoNeRF: RSSI based loss



Naïve loss doesn't work well

Loss Function

This naive loss doesn't work:

- **LoS dominates RSSI measurements (and NN gradients)**
- Training optimizes for LoS paths and reflections are ignored
  - **Doesn't learn walls in the process**

They propose a two-stage training solution:

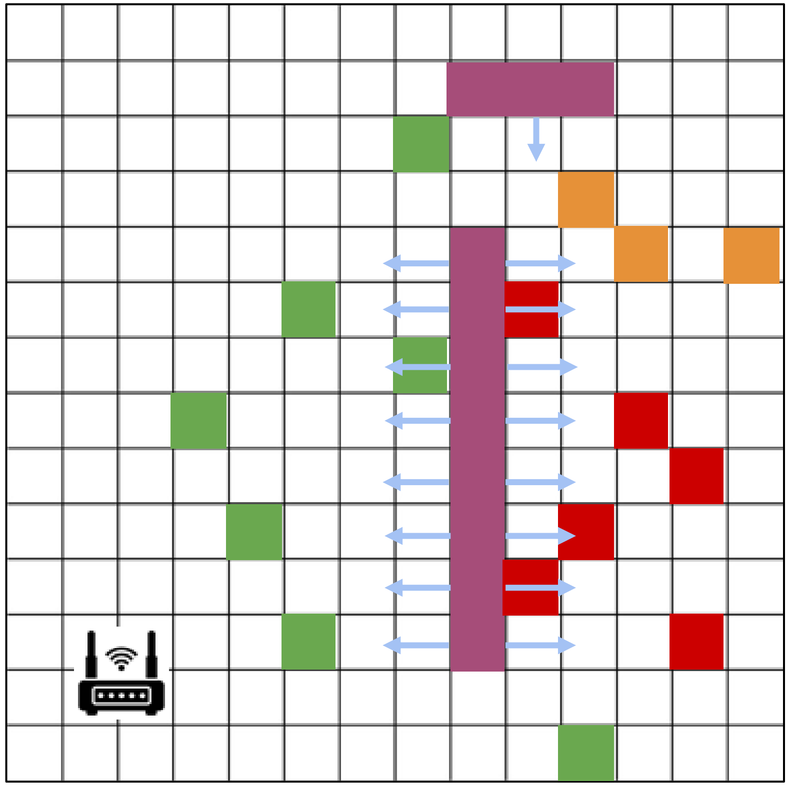
1. Train on LoS paths only
   - Generates reasonable areas of air and wall
2. Freeze LoS-only RSSI estimates over the space

1. Reintroduce reflections
2. **Optimize the network to explain the difference between the real measurements and the LoS-only RSSI map**

EchoNeRF –Training

1. NN proposes a layout
2. Physical model simulates the RSSIs the layout would produce
3. Loss measures how compatible the layout is with the observed RSSIs
4. Gradient descent nudges the NN toward a more reasonable layout

Optimize the **occupancy** and **normal directions** that result in best fit to the real measurements



Evaluation

Simulated evaluation:

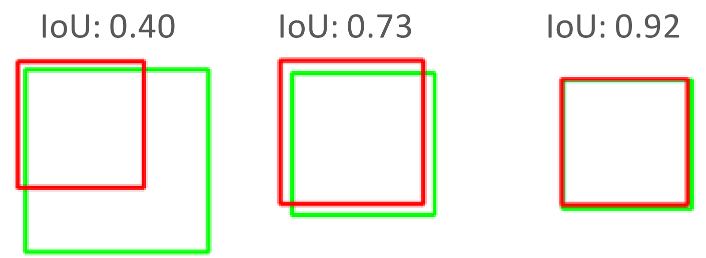
- **Floor plans** - Zillow's Indoor Dataset
- **Signal simulation** - NVIDIA Sionna Ray-Tracing
- About 1 router per room

Walk-like trajectories with 1000 or 2000 RSSI samples each

Results

Ground
Truth

Quantitative Results



Metric: **Wall IoU**

- Measures fraction of wall pixels agreed on by the method and the ground truth

Wall IoU (0-1, higher is better)

Technique	**1000 RSSI Samples**	**2000 RSSI Samples**
Heatmap Segmentation	0.09 ± 0.02	0.12 ± 0.03
NeRF2	0.12 ± 0.02	0.14 ± 0.02
EchoNeRF LoS	0.25 ± 0.04	0.27 ± 0.07
EchoNeRF	**0.32 ± 0.06**	**0.38 ± 0.06**

Robustness

Until now, TX (router) and RX (phone) locations have been perfectly known

Estimating TX location based on RSSI map (maximum likelihood estimator)
- Works well, average TX location error is ~15cm

Localization noise added to RX positions
- Up to 2 meter gaussian noise added to RX locations

Error σ(m)	0 meters	0.5 meters	1 meter	2 meters
Wall IoU	0.38	0.35	0.33	0.29

Future Work

- Use this technique to build a system tested in the real world
  - RF propagation is messier in the real world
  - Localizing the phone will be more challenging

- Extend the system to 3D
  - Extracting 3D data (e.g. furniture, stairs, etc.) from a scene would enable many more applications
  - Will likely require higher order reflection modeling

- Consider mitigations to prevent the misuse of this technique
  - RSSI permissions are easier to obtain than camera, microphone
  - How can AP/phone makers prevent a bad app from obtaining information about your home unknowingly.

Takeaways

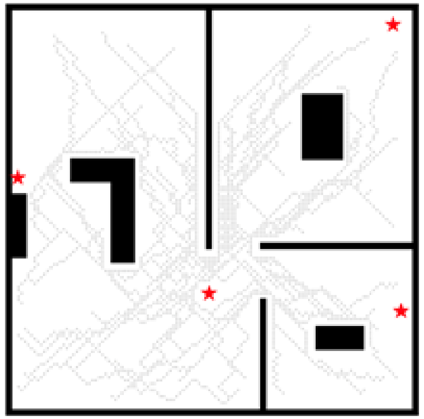
- Use physics-based models to condition data before doing learning
  - Inference on raw signals is much harder

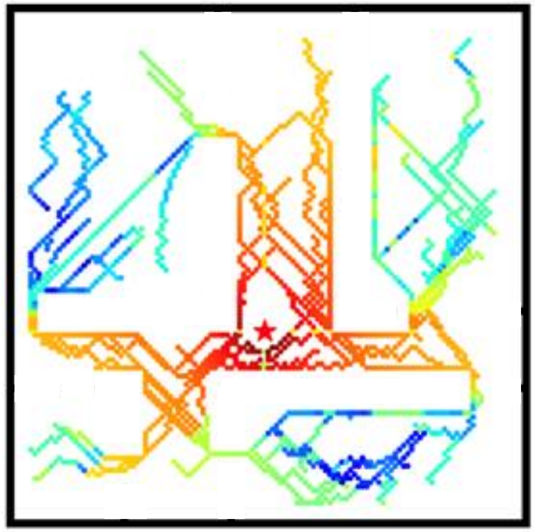
- NeRFs work well across different modalities
  - Vision NeRFs do image->image transformations, but that's not a limit of the technique!

- No need to make a general predictor
  - You can train a NN just for your scene using observations you collect!

Limitations

- Simulated only (not real world)
- Assumptions about signal propagation
  - not passing through walls
- Requires one TX per room
- Requires localization of RX positions
- 2D only





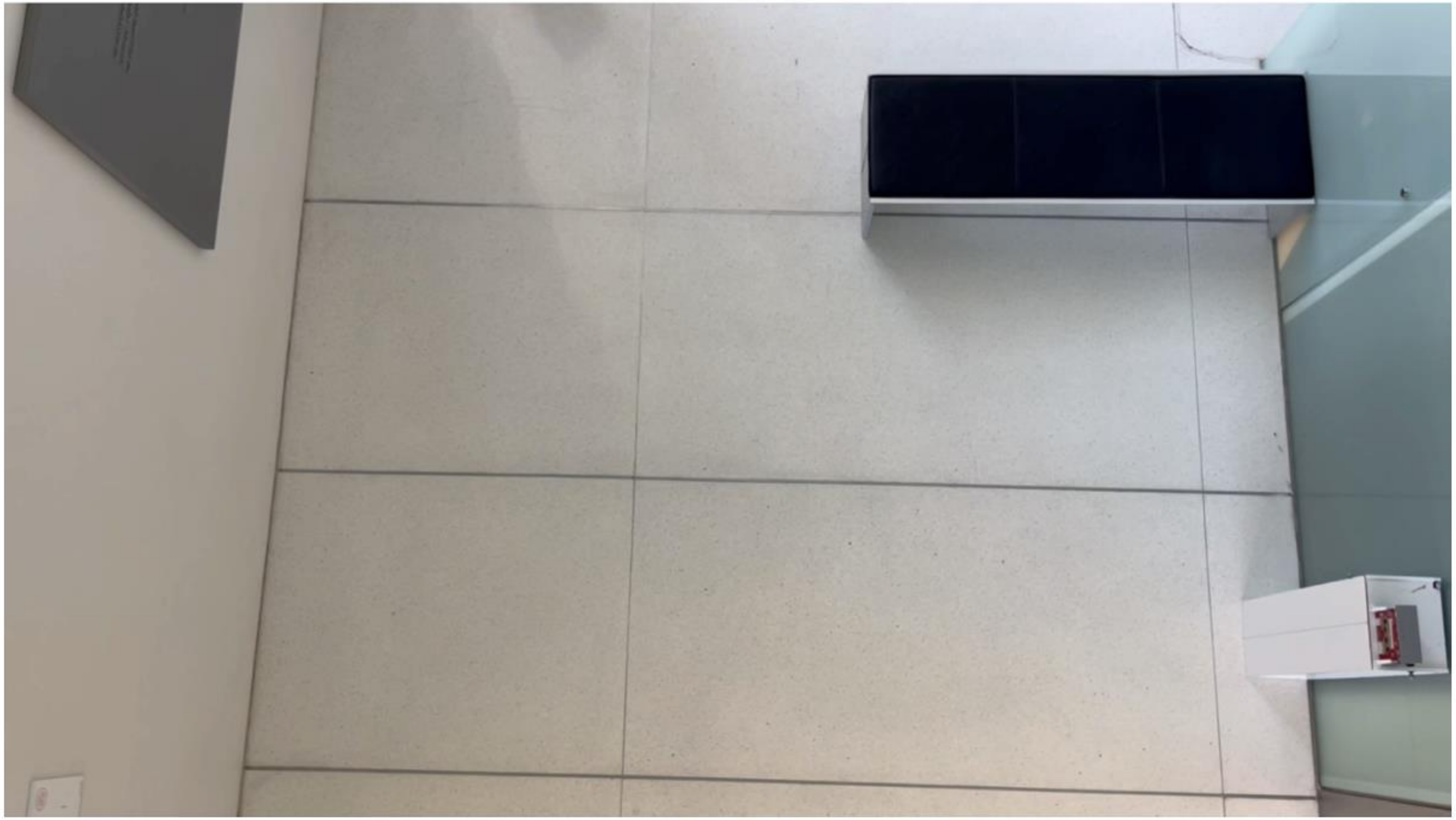
BONUS

Can we reconstruct indoor scenes?

- Without mobile devices
- Real world environment
- Generalized training (not specific to a given environment)
- Single static radar

Indoor scene reconstruction from WiFi

Can we enable scene reconstruction with a single static device?

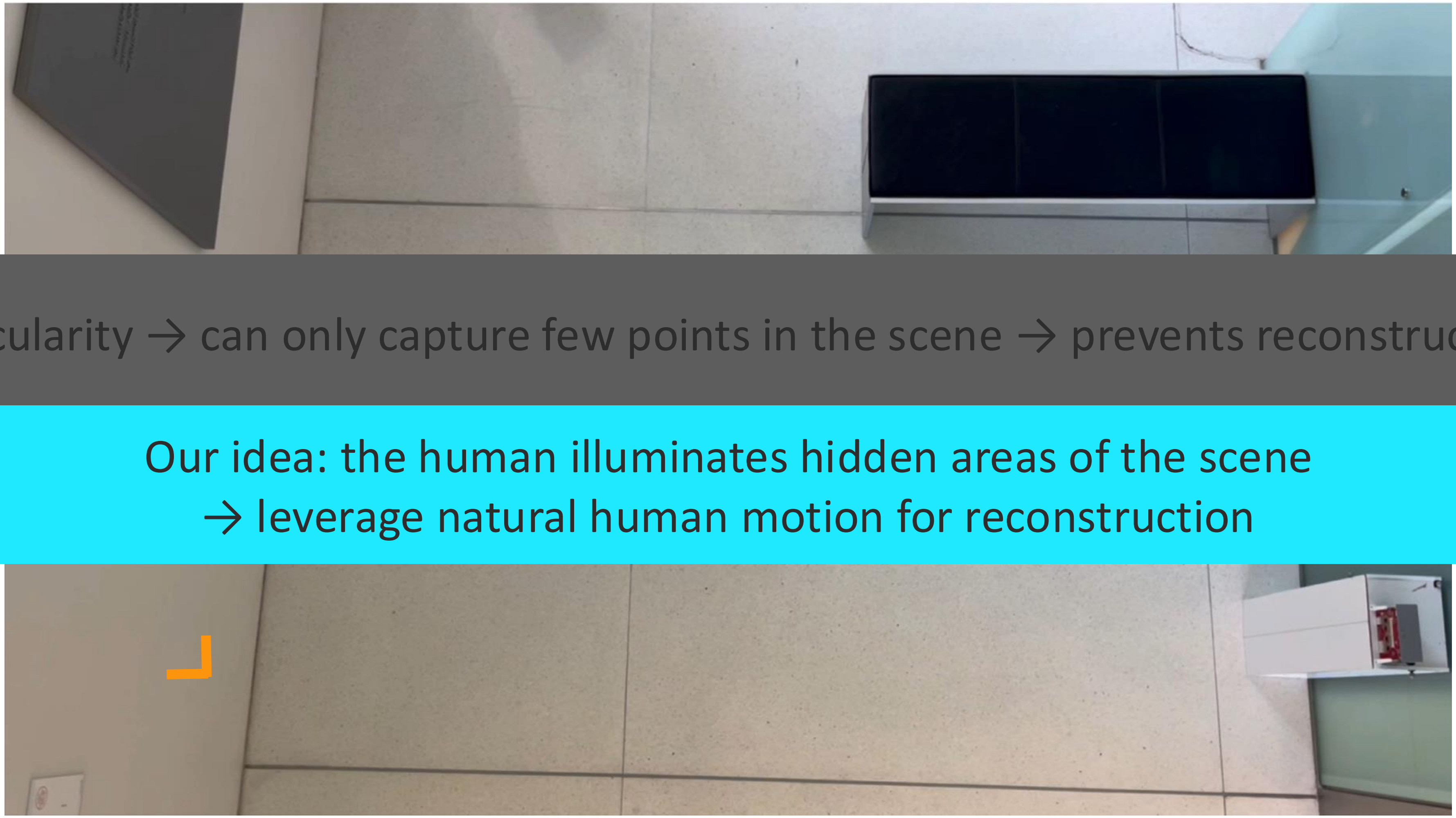


Indoor scene reconstruction from WiFi

Can we enable scene reconstruction with a single static device?

Indoor scene reconstruction from WiFi

Can we enable scene reconstruction with a single static device?



Specularity → can only capture few points in the scene → prevents reconstruction

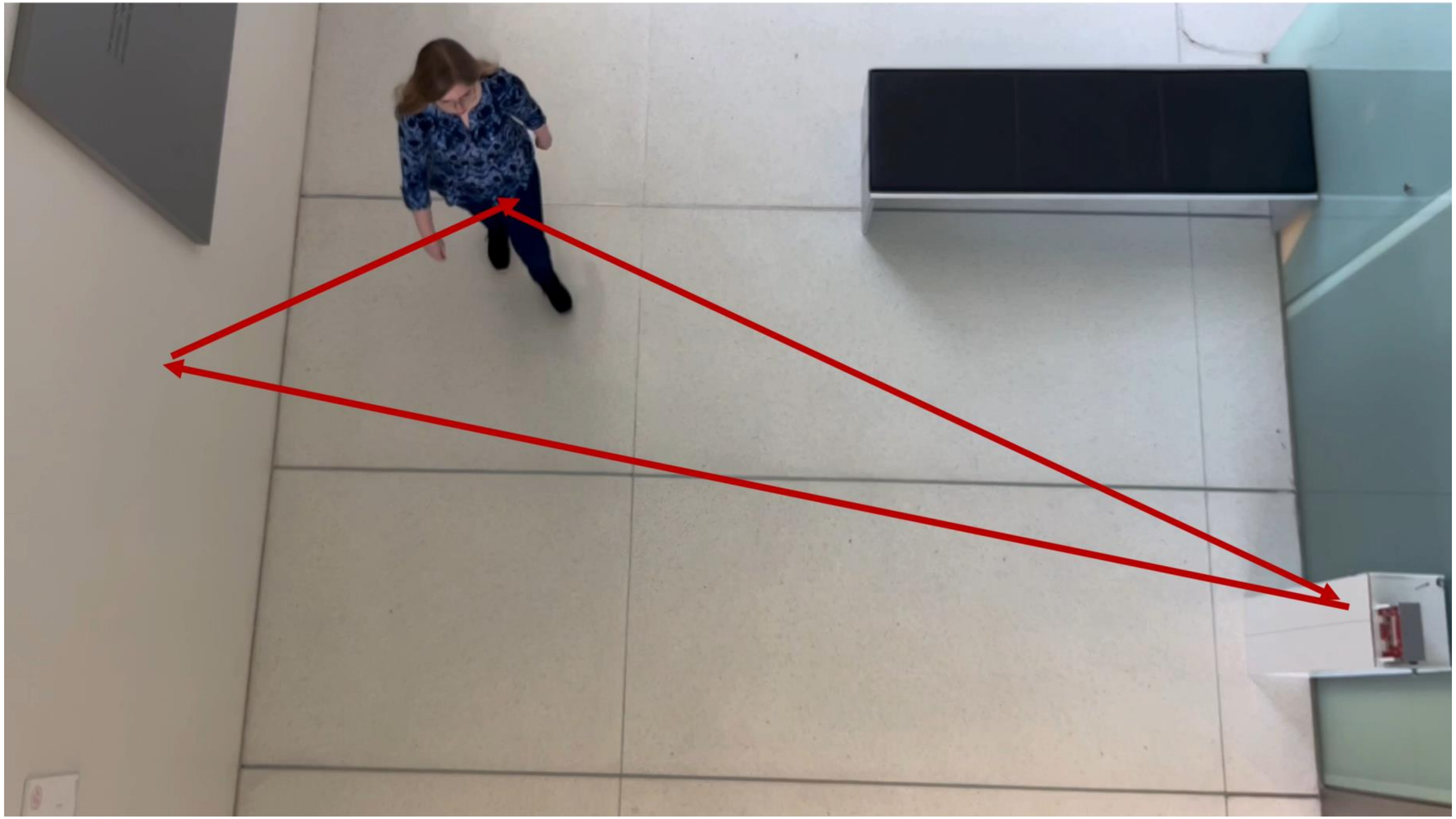
Our idea: the human illuminates hidden areas of the scene
→ leverage natural human motion for reconstruction

Indoor scene reconstruction from WiFi

Our idea: human illuminates hidden areas of the scene

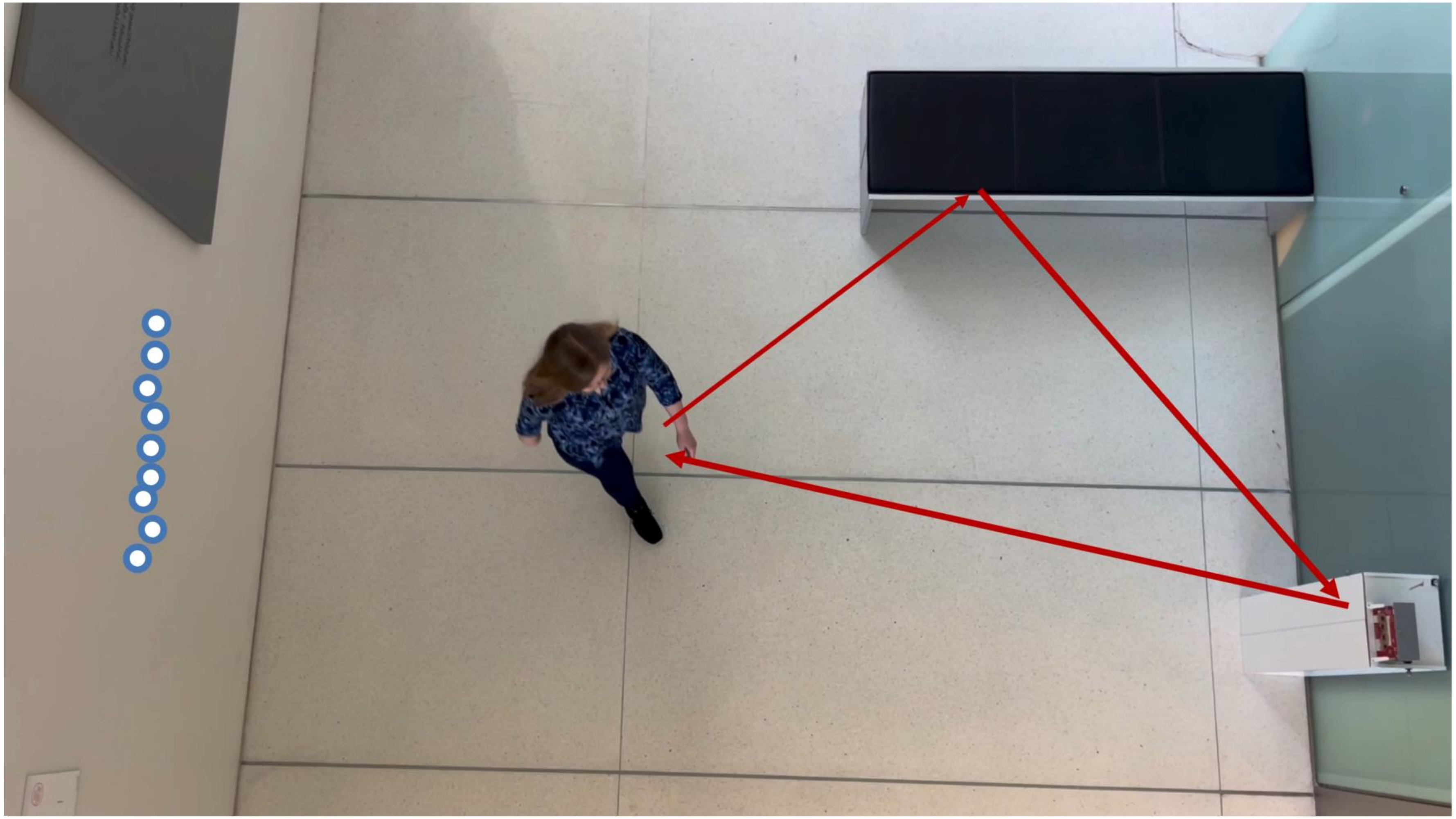
Indoor scene reconstruction from WiFi

Our idea: human illuminates hidden areas of the scene



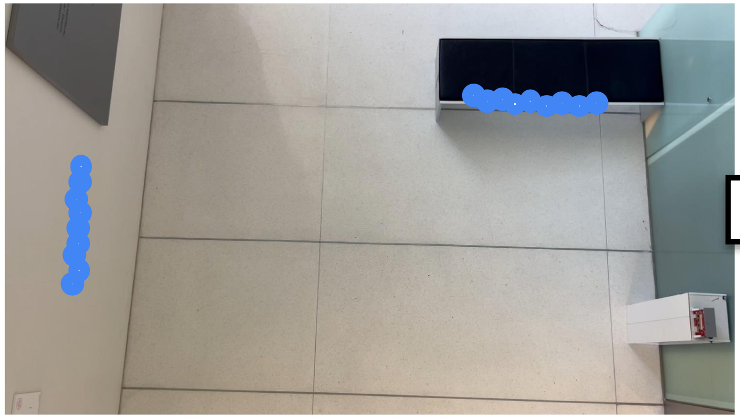
Indoor scene reconstruction from WiFi

Our idea: human illuminates hidden areas of the scene

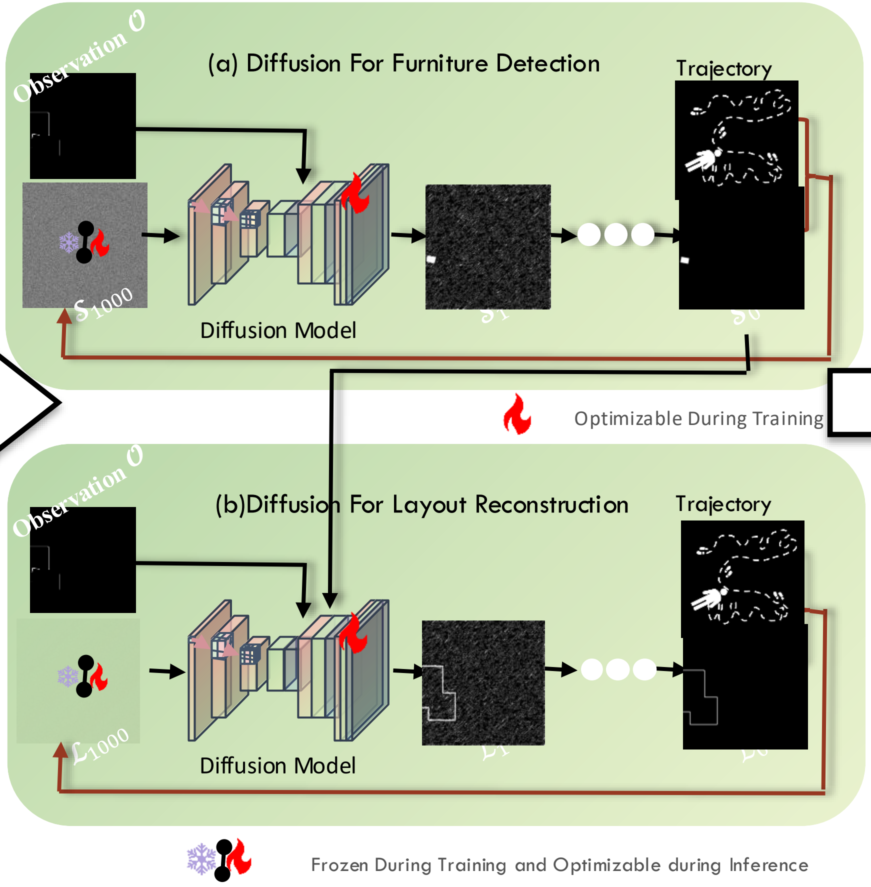


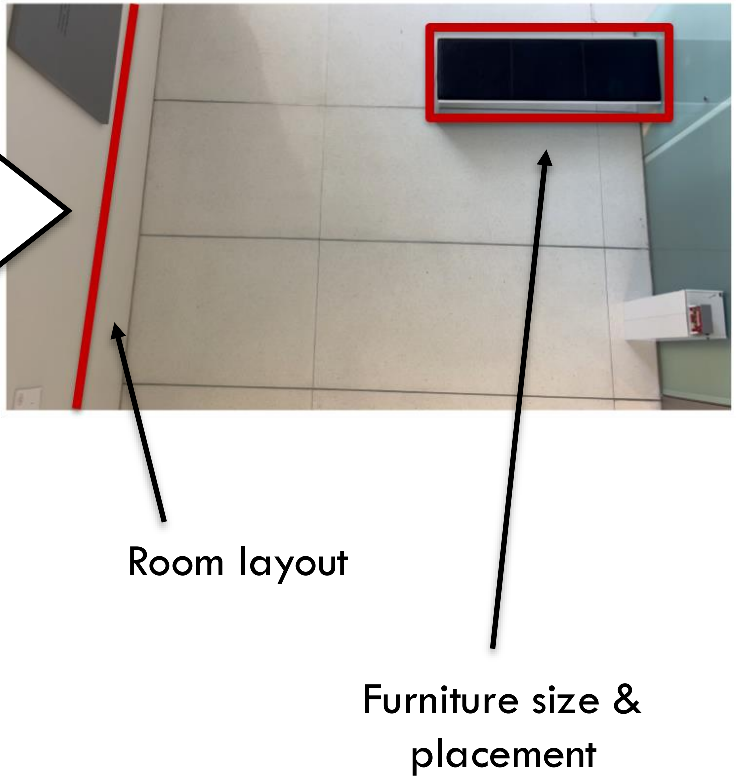
Indoor scene reconstruction from WiFi

Our idea: human illuminates hidden areas of the scene



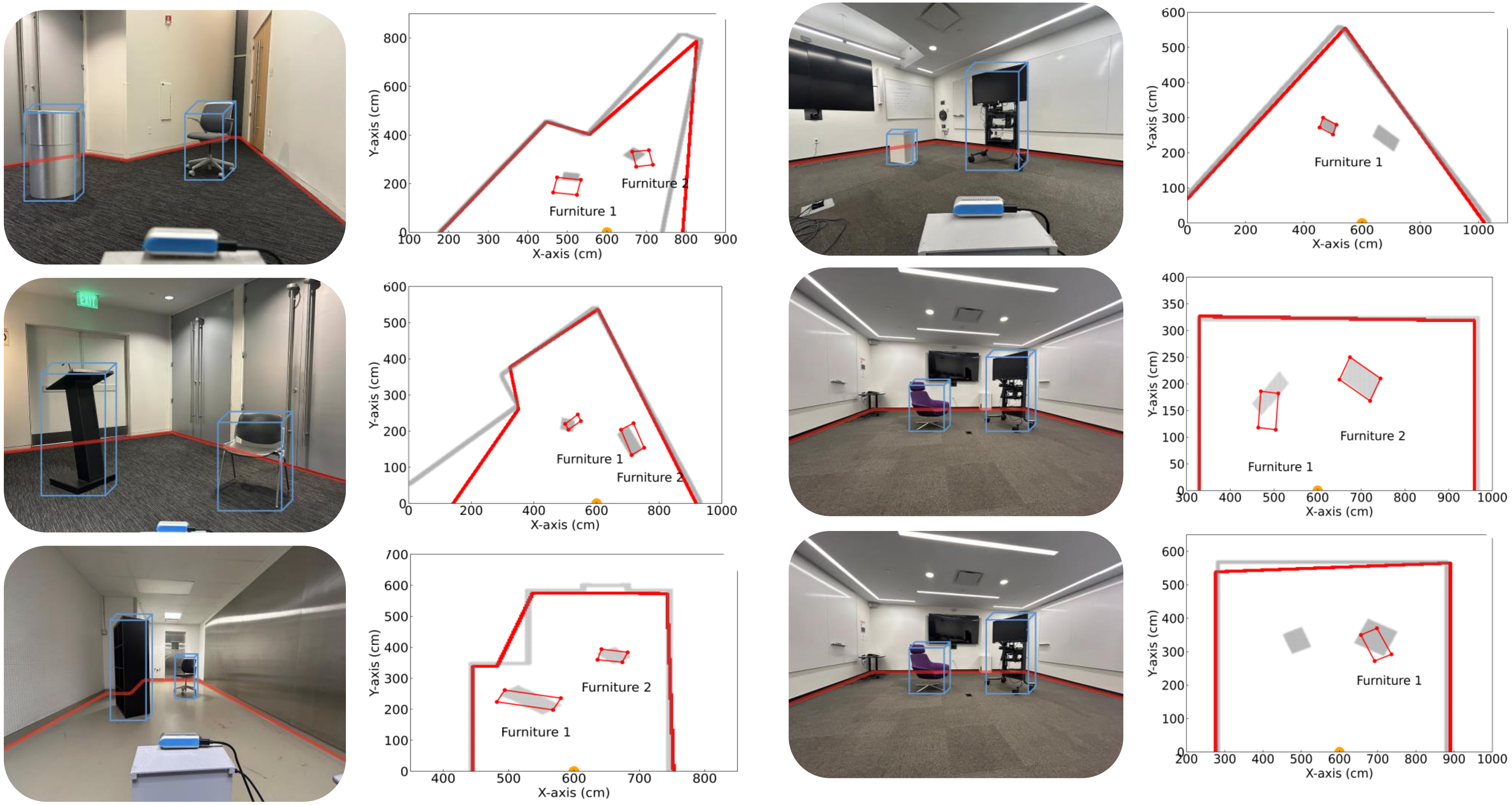
Two-stage diffusion model



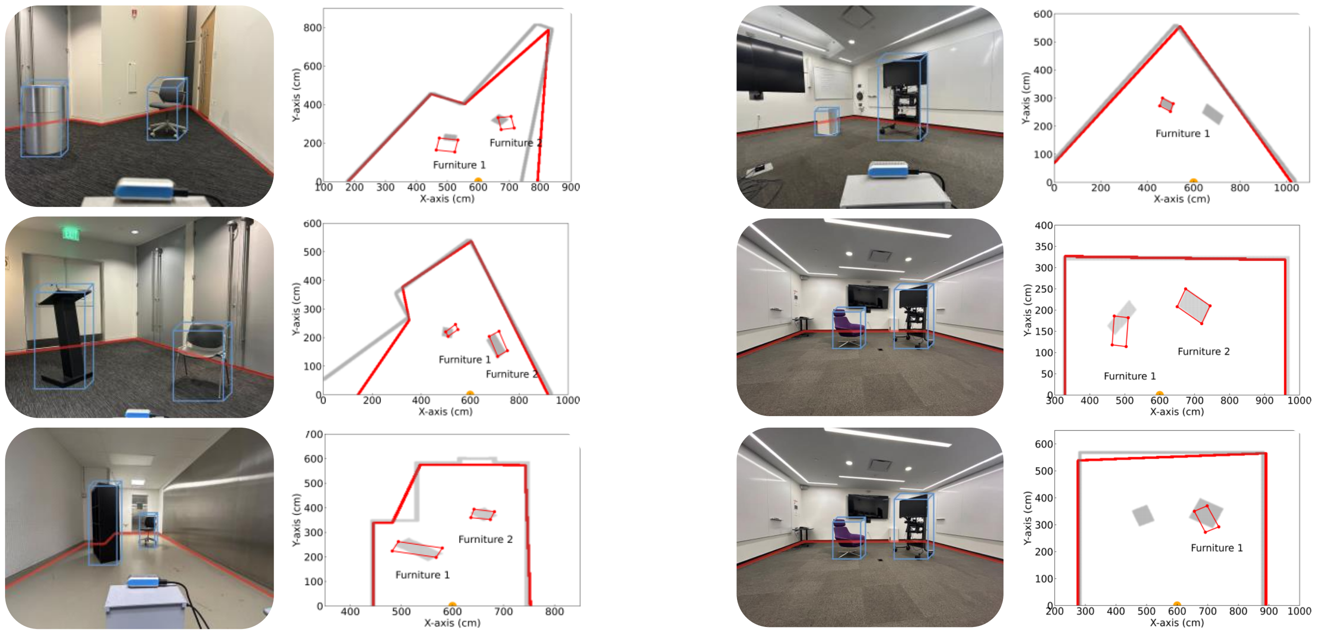


Indoor scene reconstruction from WiFi

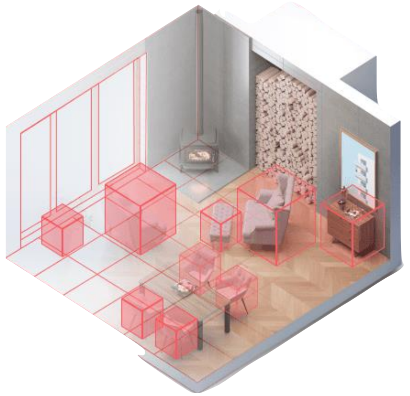
Preliminary results demonstrate exciting potential



Indoor scene reconstruction from WiFi



Full 3D Reconstruction



Novel Downstream Tasks

smart homes

scene-aware audio

indoor navigation

Privacy Preservation

RF-Protect: Privacy against Device-Free Human Tracking

Jayanth Shenoy†, Zikun Liu†, Bill Tao†, Zachary Kabelac‡, Deepak Vasisht†
†University of Illinois Urbana-Champaign, ‡ Analytical Space

ABSTRACT

The advent of radio sensing that works through walls & obstacles challenges the notion of indoor privacy. An eavesdropper can deploy such sensing to snoop on their neighbors and a smart sensor embedded with such sensing capabilities can perform large scale behavioral and health data mining. We present RF-Protect, a new framework that enables privacy by injecting fake humans in the sensed data. RF-Protect consists of a novel hardware reflector design that modifies radio waves to create reflections at arbitrary locations in the environment and a new generative mechanism to create realistic human trajectories. RF-Protect's design doesn't require any high bandwidth hardware or physical motion. We implement RF-Protect using commodity hardware and validate its ability to generate fake human trajectories.

CCS CONCEPTS

• **Security and privacy** → *Mobile and wireless security*; • **Computer systems organization** → *Embedded and cyber-physical systems.*

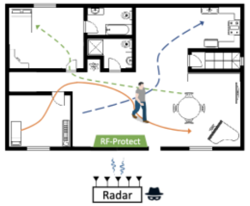


Figure 1: RF tracking systems can monitor an occupant's motion patterns (solid line) from *outside* the building. RF-Protect creates 'ghost' reflections (dashed lines above) that resemble human reflections and corrupt eavesdropper's information.