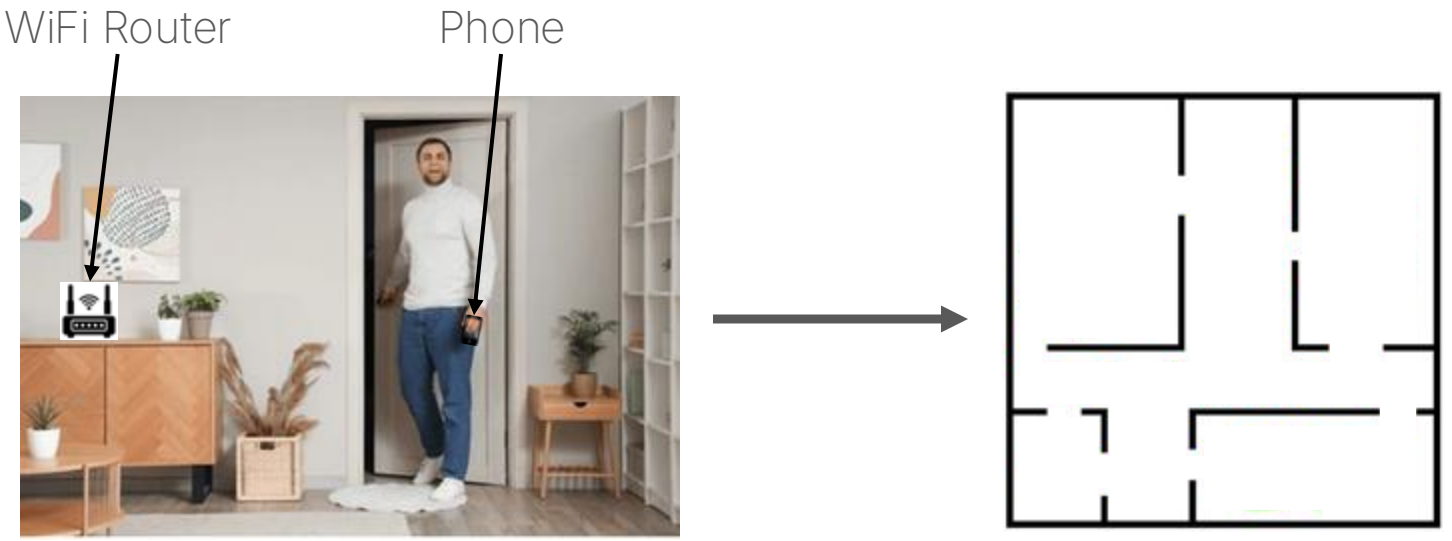
Can NeRFs See without Cameras?

Chaitanya Amballa, Sattwik Basu, Yu-Lin Wei, Zhijian Yang, Mehmet Ergezer, Romit Roy Choudhury

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

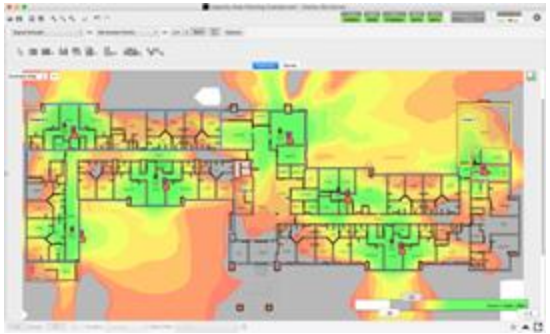


User walks around a building with phone, measuring WiFi signals

Floor plan of the room

Applications

Wireless Network Planning



Can help us choose locations for wireless APs

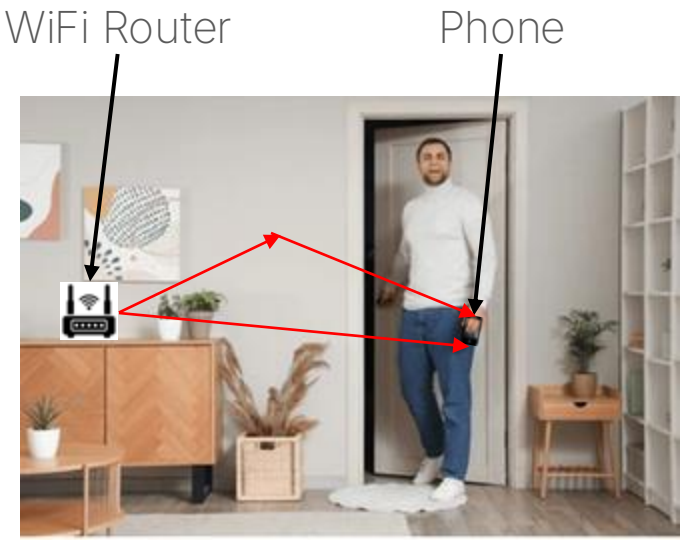
Drones and Robotics

Can inform robots with an initial map of the environment

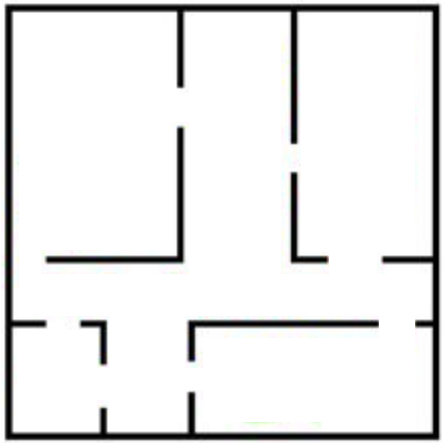
AR/VR

Can inform where to place virtual walls and objects

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

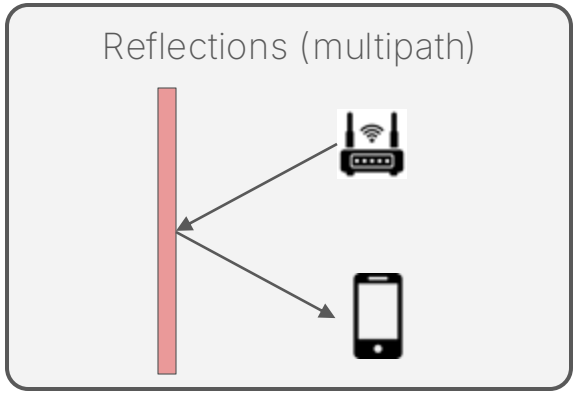


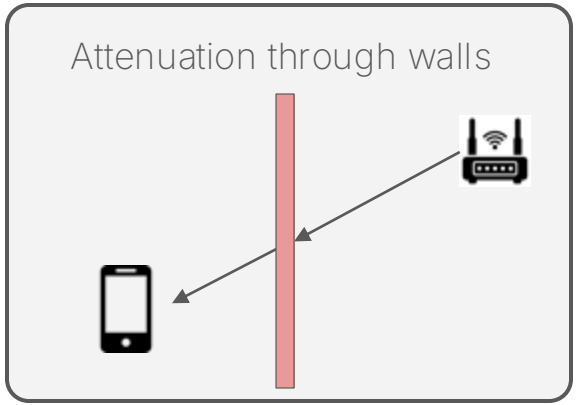
User walks around in a building with phone, measuring WiFi signals

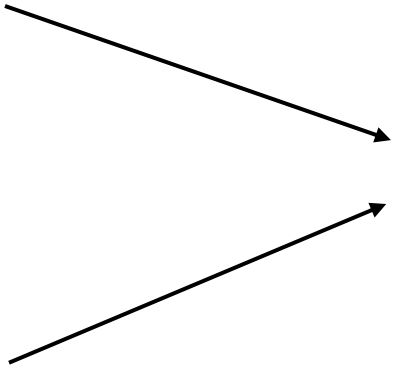


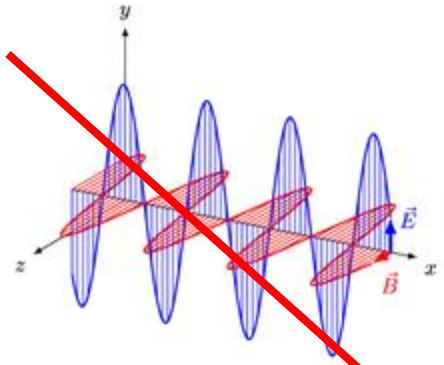
Floor plan of the room

Indoor Signal Propagation and RSSI



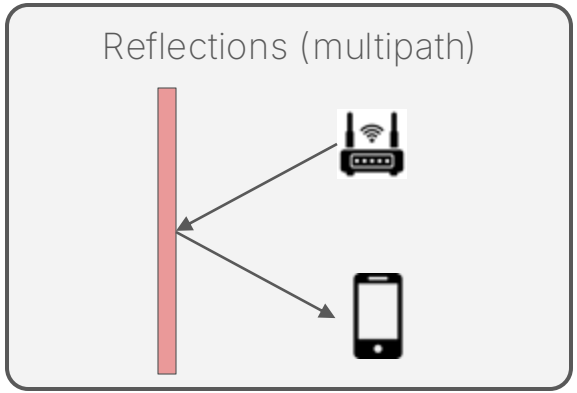


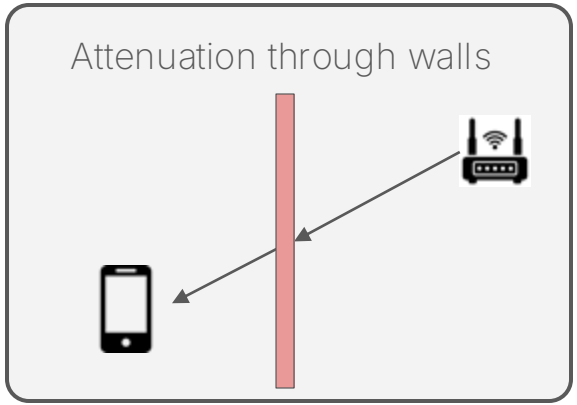


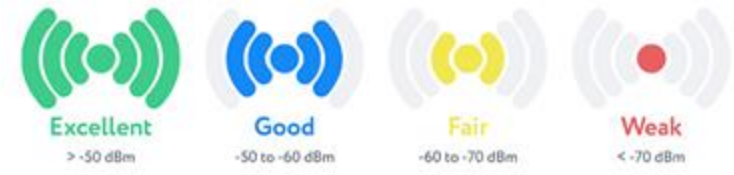


Full magnitude and phase data for radio signals

Indoor Signal Propagation and RSSI

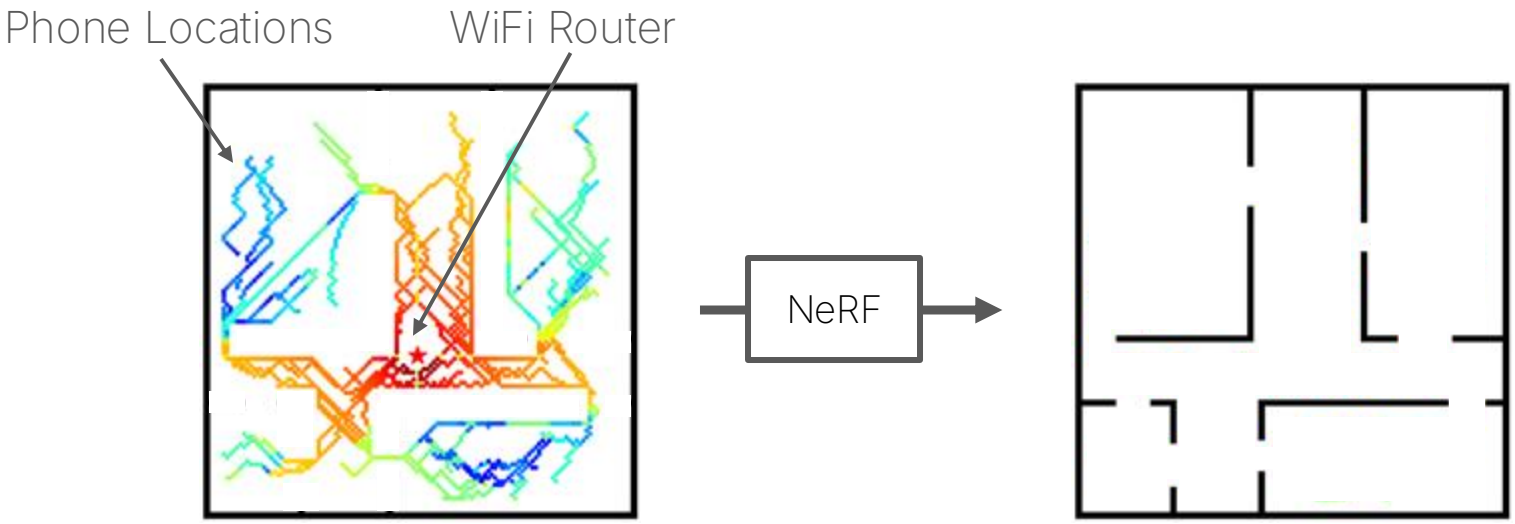






RSSI only provides a single number of signal strength

Problem Setup: Inputs and Outputs

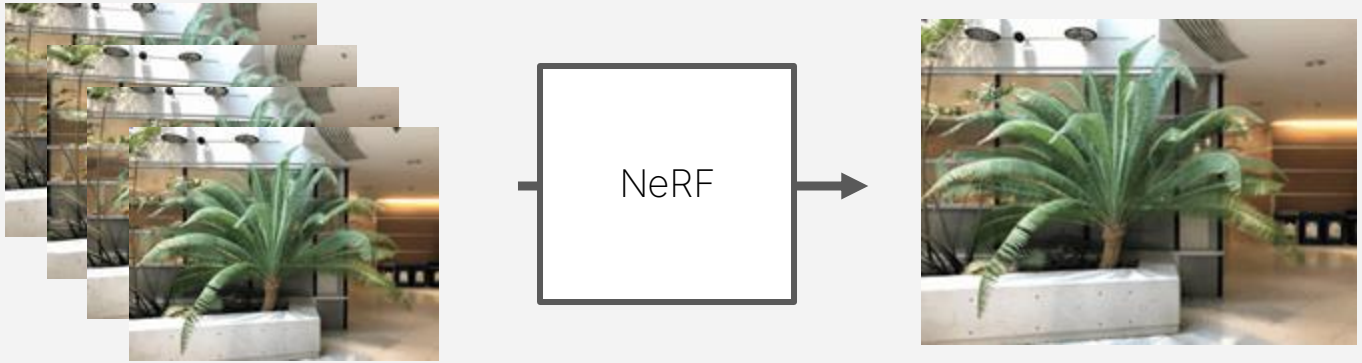


User walks around in a building with phone, measuring WiFi signals

Floor plan of the room

Vision NeRFs

Goal: Encode a complex 3D scene implicitly in the weights of a neural network



Discrete measurements

Continuous representation of the scene

Steps to build a NeRF

1. Define a (simplified) physical representation of the world
2. Train a neural network using real measurements
3. Sample the neural network as a prediction

Vision NeRFs - Physical Modeling

Imagine representing this 3D scene as a continuous function:

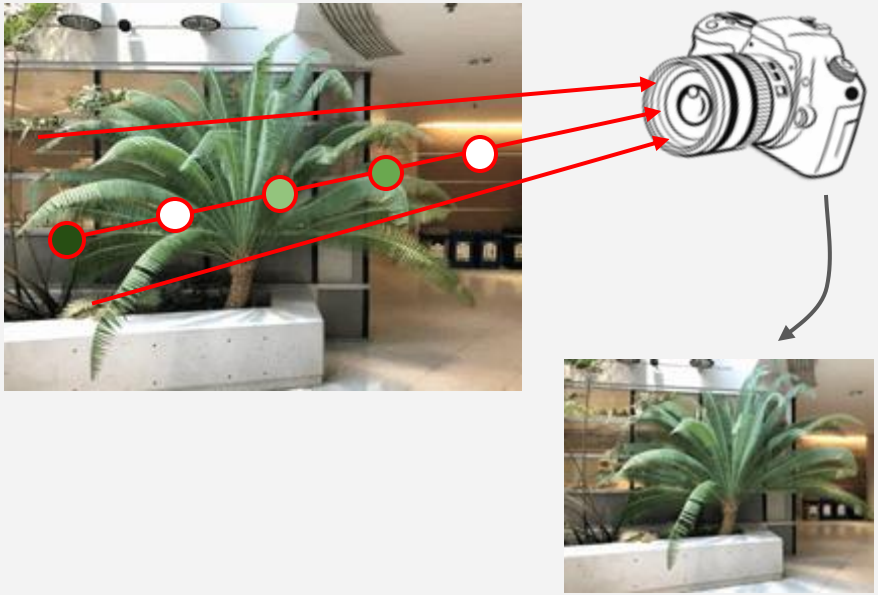
Neural Network

f(**x**, **y**, **z**, **θ**, **φ**) -> (**c**, **σ**)

position, **direction** -> **color**, **opacity**

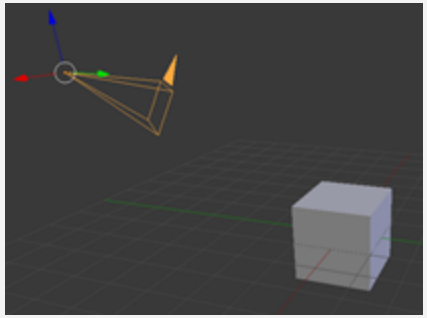
Physics Model

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

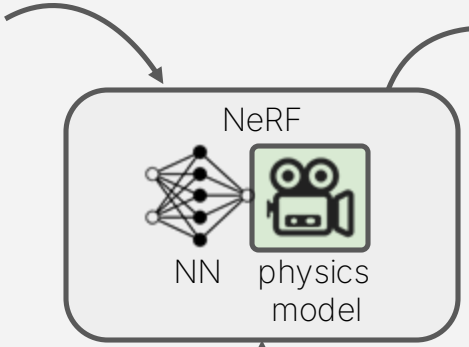


Vision NeRFs - Training Process

Input: *new view location and orientation*

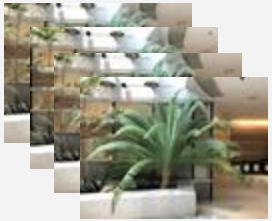


(x, y, z, θ, φ)



Output: *an image from the new view*

...
(x, y, z, θ, φ)
(x, y, z, θ, φ)
(x, y, z, θ, φ)



Train *a NeRF using many images of a scene and their location/orientation*

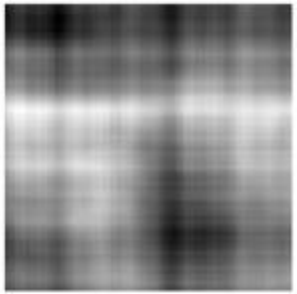
NeRFs for Wireless Signals is Hard

An RF antenna receives input from many different rays, not a single one

Previous work (NeRF2, NeWRF) shows:

- NeRFs are good at predicting the channel response over a 3D space
- **Trained to predict wireless signals, not geometry**
- **NeRF2 is used as a baseline**

Ground Truth



NeRF2

Steps to build a NeRF

1. **Define a (simplified) physical representation of the world**
2. Train a neural network using real measurements
3. Sample the neural network as a prediction

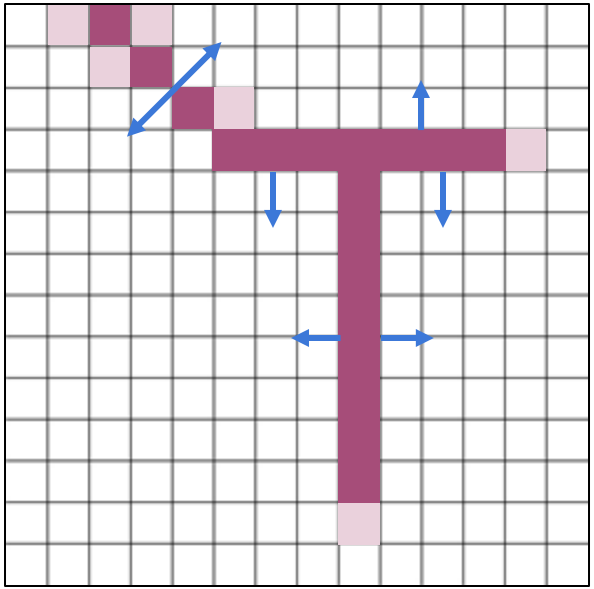
EchoNeRF - Physical Model

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



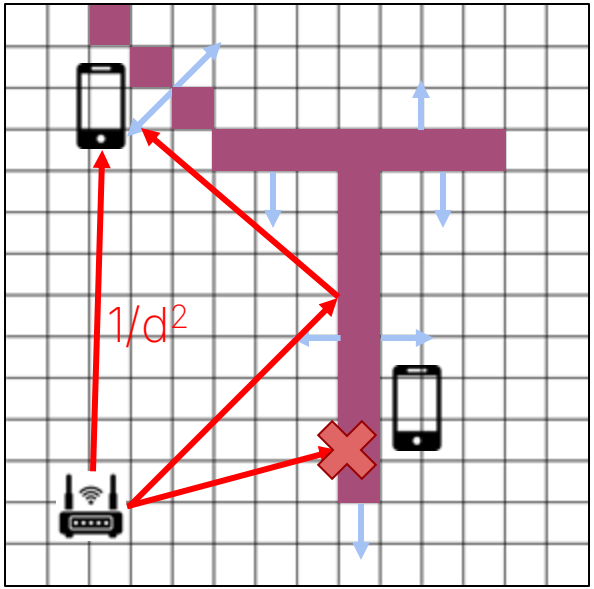
EchoNeRF - Physical Model

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)



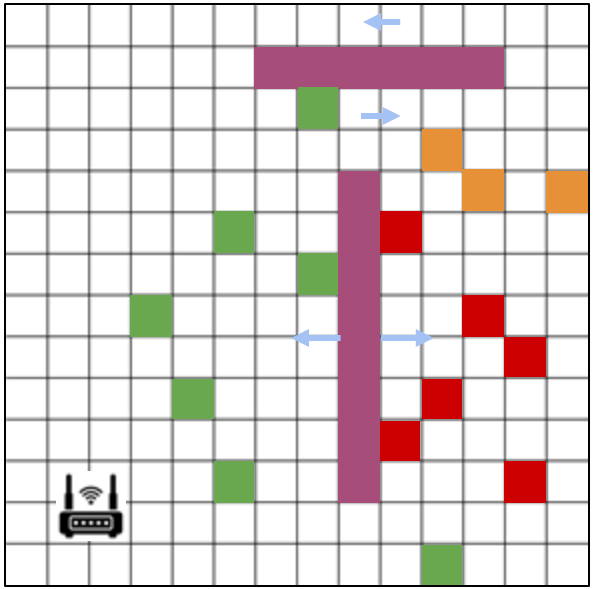
Steps to build a NeRF

1. ~~Define a (simplified) physical representation of the world~~
2. **Train a neural network using your real measurements**
3. Sample the neural network as a prediction

EchoNeRF - Training

Optimize the **occupancy** and **normal directions** that cause the best fit to the real measurements:

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



Training challenges

This naive training process doesn't work in reality:

- **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**

Steps to build a NeRF

1. ~~Define a (simplified) physical representation of the world~~
2. ~~Train a neural network using your real measurements~~
3. **Sample the neural network as a prediction**

Evaluation

Simulated evaluation:

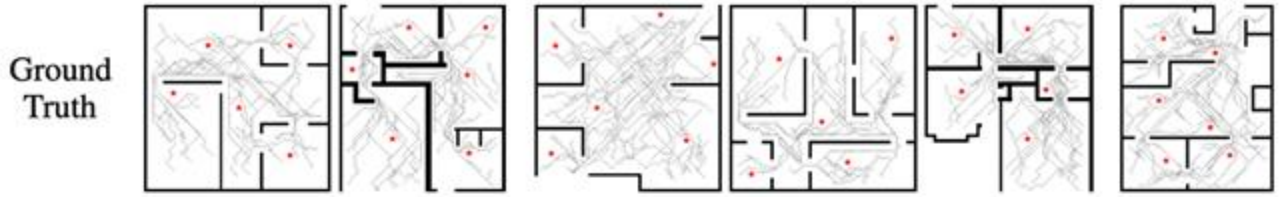
- **Floor plans** - Zillow's Indoor Dataset
- **Signal simulation** - NVIDIA Sionna Ray-Tracing
- About 1 router per room

Walk-like trajectories with 1000 or 2000 RSSI samples each

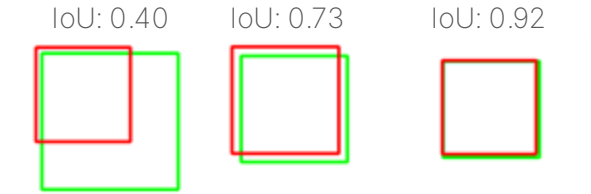
Tested against:

- **RSSI Interpolation** - Create a complete RSSI map and segment
- **NeRF2** - models reflections as "virtual transmitters" at reflection spots
- **EchoNeRF LoS** - ablated version without reflections

Results



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

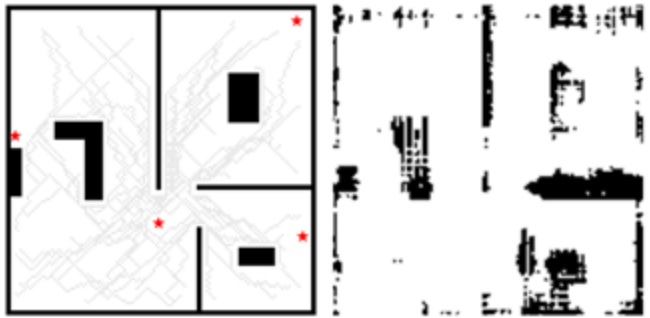
Estimating TX location based on RSSI map (maximum likelihood estimator)
- Works well, average TX location error is ~15cm

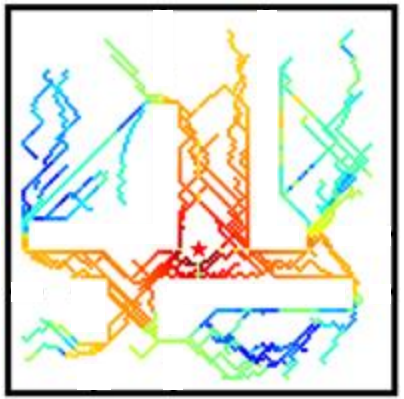
Localization noise added to RX positions
- Up to 2 meter gaussian noise added to RX locations

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

Limitations

- Simulated evaluation only
- Physical representation is 2D only
- Poor furniture reconstruction
  - Because of 2D-only perception
  - And, first-order reflections only
- Uses one TX (router) per room
  - Much more than typical home WiFi setups
- Requires localization of RX positions
  - How is this captured? Authors don't discuss options here
  - 802.11mc can achieve errors of 1-2 meters, may be usable





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!