NERF2: Neural Radio-Frequency Radiance Fields

Presenter: Xuyang Cao

Background

Neural Radiance fields (NeRF) by Google uses a few shots of photos to produce view synthesis.

I.e., train a neural network using a few ray traces of light to predict the light paths of other TX/RX pair in the same environment.

Question: can we do the similar things for communication radio waves?

Challenges from Light to Radio Waves

- Microwaves are more prone to be reflected, diffracted, and scattered.
- For light, only amplitude/real-value signal is considered. For radio waves, phase/complex-value signal is needed.
- The RX model is different: antenna (array) vs million-pixel camera

Need to update the underlying tracing model – nontrivial new design.

NeRF2: Neural Radio-Frequency Radiance Fields

- A new tracing model for radio waves
- Single antenna & multi antenna RX model
- “Turbo-learning”: data augmentation using NeRF2
- NeRF2 application: BLE RSSI/localization prediction and MIMO CSI prediction

Design: Modeling the Radiance Field

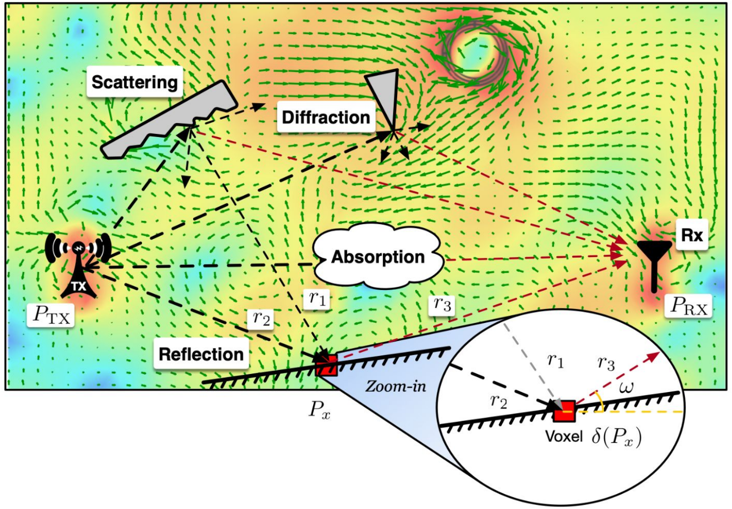


Fig. 1: Illustration of a radio-frequency radiance field. The ideal distribution of RF radiance is disturbed by the obstacles, which cause the RF signals to be reflected, scattered, diffracted, or absorbed.

P: location
δ: attenuation index of the voxel

Then, given a fixed RX node, what it multipath profile will look like (output) for a chosen TX location (input variable)?

Radiance Field Modeling Key Principles

- Discretize the 3D space into small pixels: “voxels”
- A radio wave hits voxels, undergoes attenuation, and breaks into new radio waves (unevenly towards new directions)
    - Attenuation: a function of the voxel itself (e.g., its material)
    - Radiance: a function of the voxel and the incoming wave

Radiance Field Modeling Using Neural Network

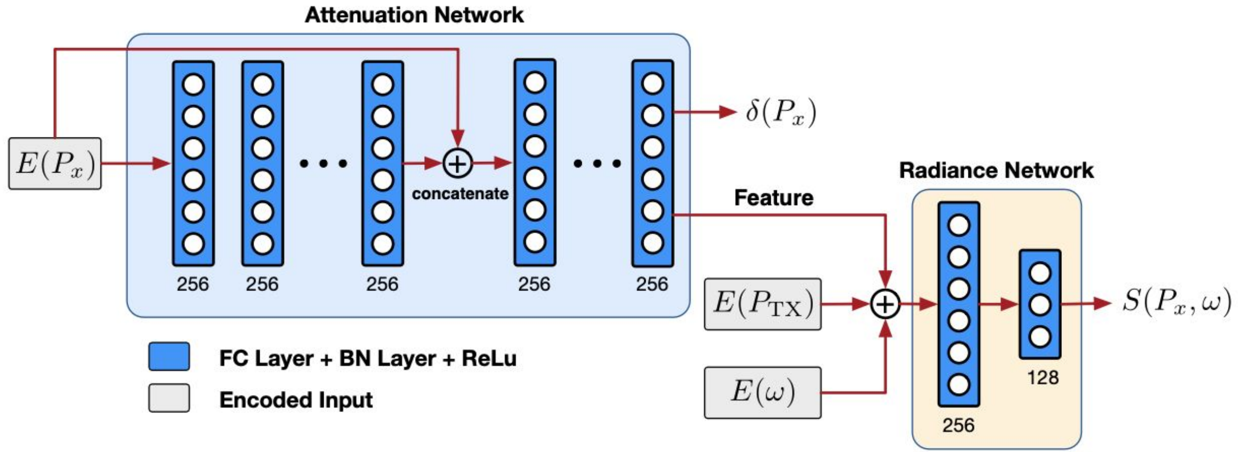


Fig. 3: Architecture of the neural network. NeRF^2 consists of two MLPs, the attenuation network, and the radiance network. The attenuation network can predict the attenuation δ of any voxel. Given the TX position and a measuring direction, the radiance network can predict the signal transmitting from an arbitrary voxel.

After knowing how to derive δ and S, building the trace

The propagation of an RF signal S from a transmitter (TX) to a receiver (RX) conforms to the Friis equation as follows:

$$R = H_{\mathrm{TX}\to\mathrm{RX}} S = a_{\mathrm{TX}\to\mathrm{RX}} e^{J\theta_{\mathrm{TX}\to\mathrm{RX}}} S \tag{2}$$

from the TX to the RX. Mathematically, a direction ω related to the RX can be modeled as a ray, which starts from the RX and directs toward ω. The points on this ray are correspondingly described as follows:

$$P(r, \omega) = P_{\mathrm{RX}} + r\omega \tag{3}$$

Electromagnetic Ray Tracing Model

Aggregated signal from the trace:

ray. Namely, the received signal at the RX from the direction ω can be expressed as:

$$R(\omega) = \int_0^D H_{P(r,\omega)\rightarrow P_{\mathrm{RX}}} S\Big(P(r,\omega), -\omega\Big)\,\mathrm{d}r \tag{4}$$

In the above equation, $S(P(r,\omega), -\omega)$ represents the signal transmitted from the voxel at $P(r,\omega)$ to the RX at P_{RX}. Its

Expand H:

$$\begin{aligned} H_{P(r,\omega)\rightarrow P_{\mathrm{RX}}} &= \prod_{\tilde{r}=0}^{r} \delta\big(P(\tilde{r},\omega)\big) \\ &= \left(\prod_{\tilde{r}=0}^{r} \Delta a_{P(\tilde{r},\omega)}\, e^{\mathrm{J}\Delta\theta_{P(\tilde{r},\omega)}}\right) \end{aligned}$$

Electromagnetic Ray Tracing Model

Convert to log scale for easy computation:

$$H_{P(r,\omega)\to P_{\mathrm{RX}}} = \exp\left(\ln\left(\prod_{\tilde{r}=0}^{r} \delta(P(\tilde{r},\omega))\right)\right) = \exp\left(\int_0^r \ln\left(\delta(P(\tilde{r},\omega))\right) \mathrm{d}\tilde{r}\right)$$
$$= \exp\left(\underbrace{\int_0^r \hat{\delta}\left(P(\tilde{r},\omega)\right) \mathrm{d}\tilde{r}}_{\text{Sum of attenuations}}\right) \tag{6}$$

$$\hat{\delta}(P(\tilde{r},\omega)) = \ln \delta(P(\tilde{r},\omega)) \tag{7}$$

Therefore, we have (to a single RX antenna):

$$R(\omega) = \int_0^D \underbrace{\exp\left(\int_0^r \hat{\delta}(P(\tilde{r},\omega))\, \mathrm{d}\tilde{r}\right)}_{\text{Attenuation Network}} \overbrace{S(P(r,\omega), -\omega)}^{\text{Radiance Network}}\ \mathrm{d}r \tag{8}$$

Visual Illustration

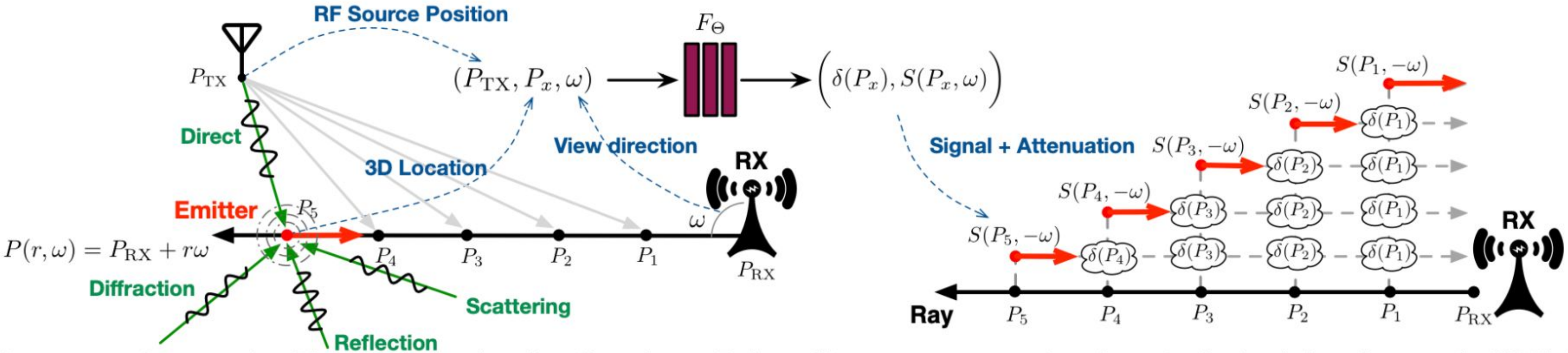


Fig. 4: Electromagnetic ray tracing. There are five voxels at $P_1 - P_5$ on the ray. Each voxel becomes a new transmitter that emits the signal along the ray to the RX. Their signals are attenuated by the other voxels between the new transmitters and the RX.

RX Antenna Model: Single Antenna

Ω: directions that the antenna can cover
GRX (ω): the antenna directivity

$$
\begin{aligned}
R &= \int_{\Omega} \sqrt{G_{\mathrm{RX}}(\omega)} R(\omega)\, \mathrm{d}\omega \\
&= \int_{\Omega} \int_{0}^{D} \exp\left(\int_{0}^{r} \hat{\delta}(P(\tilde{r}, \omega))\, \mathrm{d}\tilde{r} \right) S(P(r, \omega), -\omega)\, \mathrm{d}\tilde{r}
\end{aligned} \tag{9}
$$

Loss Function: L

$$
\mathcal{L} = |R - \widetilde{R}|^2 \tag{10}
$$

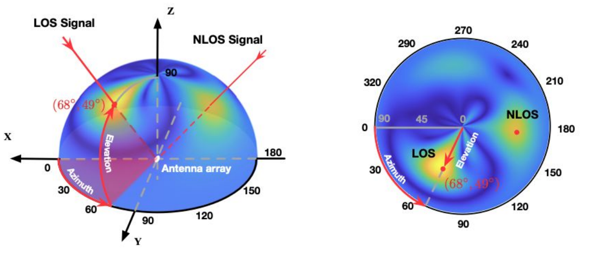
R: from the ray tracing
R hat: the measured ground truth

RX Antenna Model: Phase Array

“Sensitivity” of hearing from a particular direction (relative power)

$$\Psi(\omega) = \frac{1}{(K^2 - 1)} \left| \sum_{i=1}^{K} \sum_{j=1}^{K} w_{i,j}(\omega) \cdot e^{\mathrm{J}\Delta\widetilde{\theta}_{\mathrm{i,j}}} \right| \qquad (11)$$

$$\Psi = \begin{pmatrix} \Psi(0°, 0°) & \Psi(1°, 0°) & \cdots & \Psi(360°, 0°) \\ \Psi(0°, 1°) & \Psi(1°, 1°) & \cdots & \Psi(360°, 1°) \\ \vdots & \vdots & \vdots & \vdots \\ \Psi(0°, 90°) & \Psi(1°, 90°) & \cdots & \Psi(360°, 90°) \end{pmatrix} \qquad (12)$$



(a) 3D Spatial Spectrum (b) 2D Spatial Spectrum

Fig. 5: Illustration of spatial spectrum

A Mini Benchmark

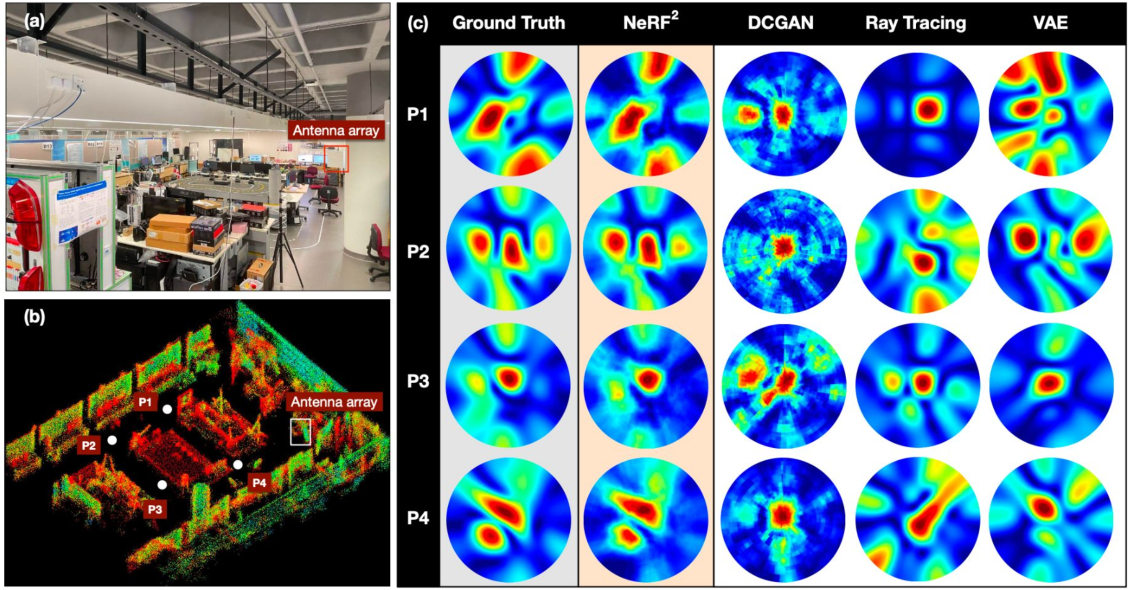


Fig. 2: Synthesis of spatial spectrums. The spatial spectrums, also known as the multipath profile, show how strong the RF signal is from a particular direction composed of the azimuthal and elevation angles. The formal definition refers to Eqn. 12. (a) shows the scene, in which the TX may be located at any position, but the RX equipped with a 4×4 antenna array is fixed at a corner; (b) shows the point cloud created by LiDAR, which is only used for the conventional ray-tracing algorithm; (c) compares the synthesis spectrums generated by different algorithms when the TX is located at four different positions. The ground truth is obtained using the antenna array.

Turbo Learning – Application of NeRF2

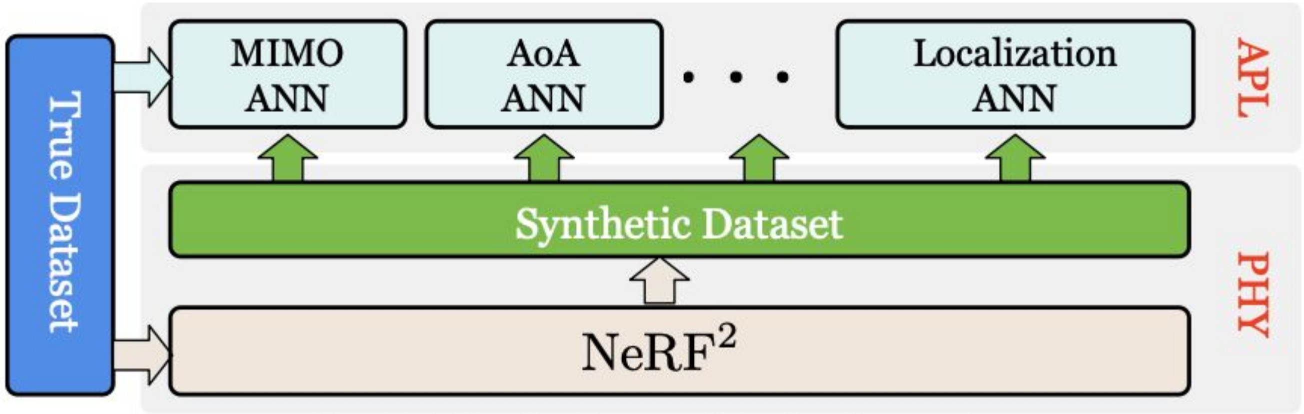


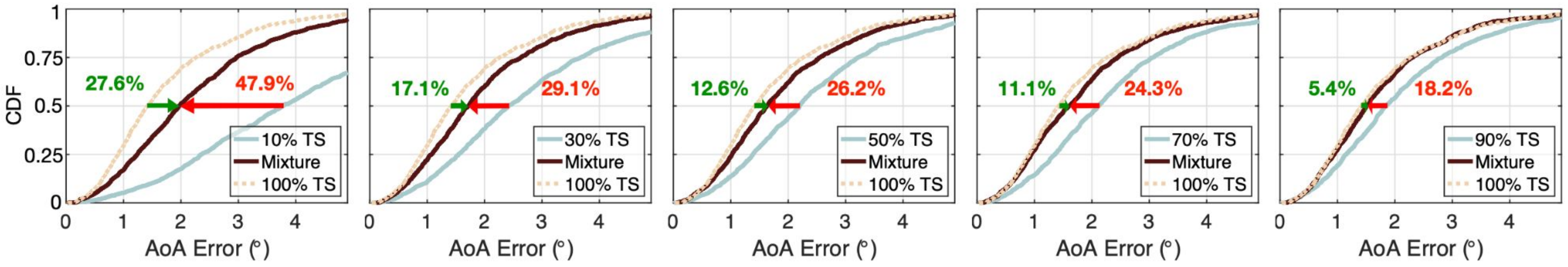
Fig. 6: Illustration of turbo-learning

AoA Estimation Evaluation

- **Naive Learning**. We use 10% of the true training dataset (TS, total 8 K) to train the AANN straightforwardly. In this approach, $NeRF^2$ is not involved.
- **Turbo-Learning**. We use the same 10% of the true dataset to train the $NeRF^2$. Then, we use the well-trained $NeRF^2$ to generate the rest 90% synthetic dataset (SS). Finally, the 10% true dataset and the 90% synthetic dataset (i.e., turbocharger) are mixed to train the AANN.

Input: TX location → NN → Spatial Spectrum → AOA Profile

AoA Estimation Evaluation



(a) 10% TS + 90% SS (b) 30% TS + 70% SS (c) 50% TS + 50% SS (d) 70% TS + 30% SS (e) 90% TS + 10% SS

Fig. 7: CDFs of AoA error. The ANN is trained by the naive-learning (in light blue) and the turbo-learning (in dark red), respectively. We quantify the benefits of $NeRF^2$ with different mixture percentages.

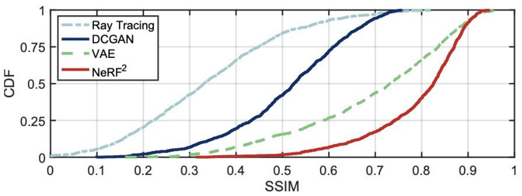


Fig. 8: SSIM Comparison

BLE Localization

Input: TX location → NN → Spatial Spectrum → RSSI

Multiple RSSIs → BLE Localization

BLE Localization

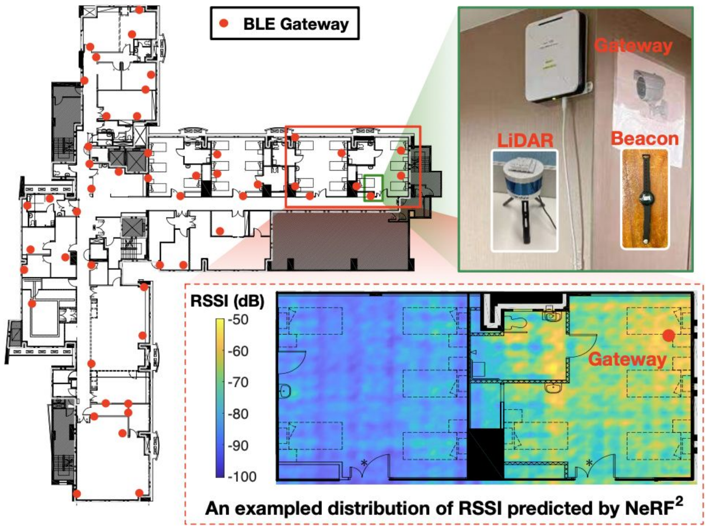


Fig. 12: The floor plan of the nursing home and deployment of BLE gateways.

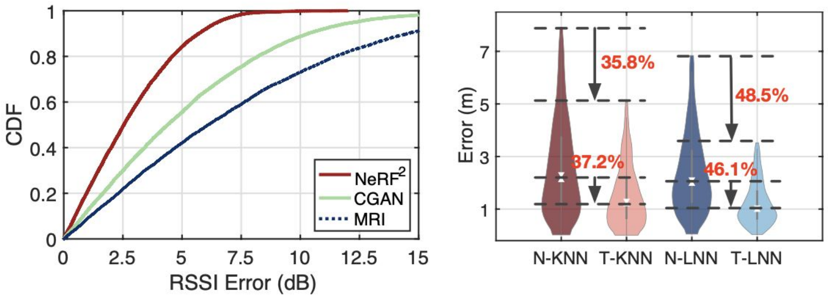


Fig. 13: RSSI Prediction

Fig. 14: Localization Result

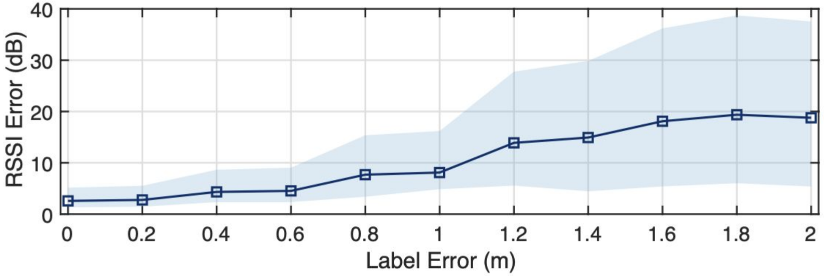


Fig. 15: RSSI vs. Label error

MIMO CSI Prediction (FDD)

Input: TX location → NN → Spatial Spectrum → Uplink CSIs

Uplink CSIs → Downlink CSIs

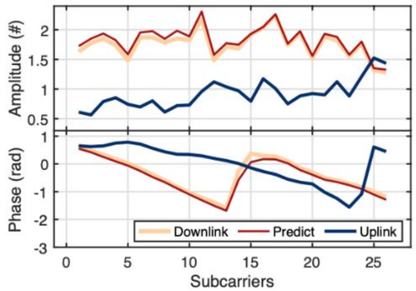


Fig. 16: Channel Amplitude & Phase

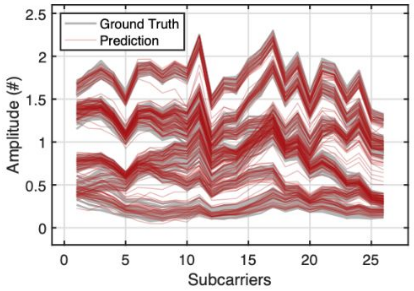


Fig. 17: Channel Amplitude in 2s

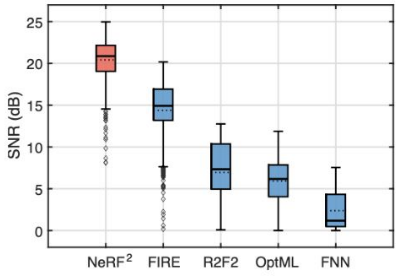


Fig. 18: Prediction SNR

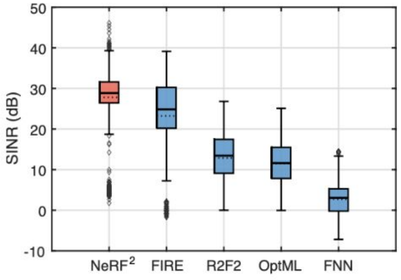


Fig. 19: MU-MIMO SINR

Questions

Do you think the complexity is tractable in complex environments? I think the radio wave number can go exponential after many scattering.

Let’s check Perusall.

My Opinion

Surely the work is beautiful and very educational.

I think the computational complexity is highly dependent on the environment. When the environment gets complicated, the computation will be very intense I guess.

Every training is only for a particular environment. Small change will invalidate the trained network.