LEARNING THE LOCATION-TO-CHANNEL MAPPING

Baptiste CHATELIER[‡,†], Vincent CORLAY[‡], Matthieu CRUSSIERE[†], Luc LE MAGOAROU[†]

† Univ Rennes, INSA Rennes, CNRS, IETR-UMR 6164, Rennes, France
‡ Mitsubishi Electric R&D Centre Europe, Rennes, France

GDR IASIS - Représentations Neuronales Implicites : des NeRF aux PINN

Paris – November 4, 2025

LOCATION-TO-CHANNEL MAPPING

- The wireless propagation channel characterizes how the propagation environment affects transmitted electromagnetic waves

LOCATION-TO-CHANNEL MAPPING

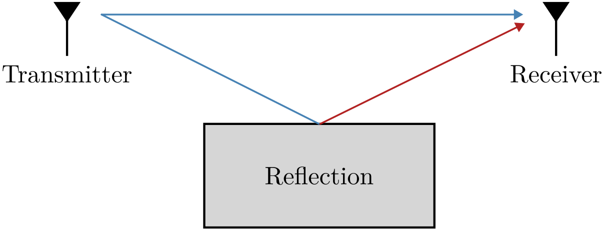
- The wireless propagation channel characterizes how the propagation environment affects transmitted electromagnetic waves

LOCATION-TO-CHANNEL MAPPING

- The wireless propagation channel characterizes how the propagation environment affects transmitted electromagnetic waves

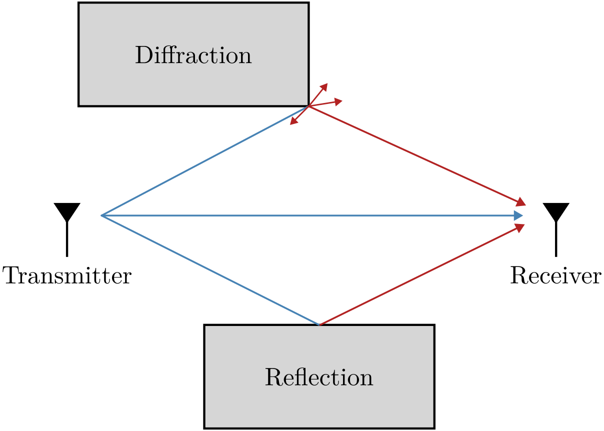
LOCATION-TO-CHANNEL MAPPING

- The wireless propagation channel characterizes how the propagation environment affects transmitted electromagnetic waves



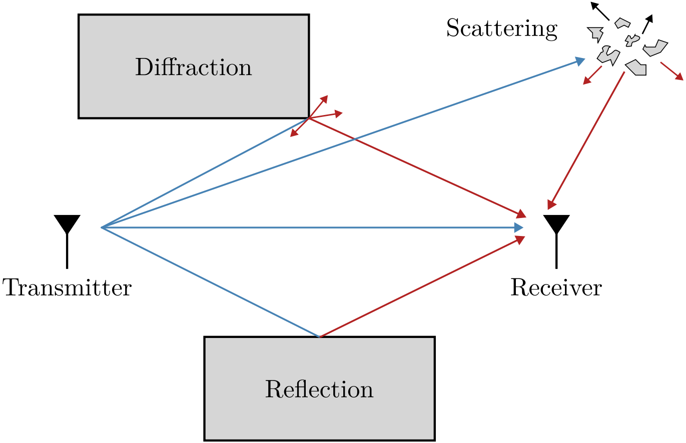
LOCATION-TO-CHANNEL MAPPING

- The wireless propagation channel characterizes how the propagation environment affects transmitted electromagnetic waves



LOCATION-TO-CHANNEL MAPPING

- The wireless propagation channel characterizes how the propagation environment affects transmitted electromagnetic waves



LOCATION-TO-CHANNEL MAPPING

- Channel impulse response ($\mathbf{x} \in \mathbb{R}^3$: receiver location):

$$h(\tau, \mathbf{x}) = \sum_{l=1}^{L_p} \gamma_l(\mathbf{x}) \, \delta(\tau - \tau_l(\mathbf{x})) \tag{1}$$

LOCATION-TO-CHANNEL MAPPING

- Channel impulse response ($\mathbf{x} \in \mathbb{R}^3$: receiver location):

$$h(\tau, \mathbf{x}) = \sum_{l=1}^{L_p} \gamma_l(\mathbf{x})\, \delta(\tau - \tau_l(\mathbf{x})) \tag{1}$$

- Channel frequency response (f_k: considered frequency):

$$h(f_k, \mathbf{x}) = \sum_{l=1}^{L_p} \gamma_l(\mathbf{x})\, \mathrm{e}^{-\mathrm{j}2\pi f_k \tau_l(\mathbf{x})} \tag{2}$$

LOCATION-TO-CHANNEL MAPPING

- Channel impulse response ($\mathbf{x} \in \mathbb{R}^3$: receiver location):

$$h(\tau, \mathbf{x}) = \sum_{l=1}^{L_p} \gamma_l(\mathbf{x}) \, \delta(\tau - \tau_l(\mathbf{x})) \tag{1}$$

- Channel frequency response (f_k: considered frequency):

$$\begin{aligned} h(f_k, \mathbf{x}) &= \sum_{l=1}^{L_p} \gamma_l(\mathbf{x}) \, \mathrm{e}^{-\mathrm{j}2\pi f_k \tau_l(\mathbf{x})} && (2) \\ &= \sum_{l=1}^{L_p} \frac{\alpha_l \mathrm{e}^{\mathrm{j}\beta_l}}{d_l(\mathbf{x})} \mathrm{e}^{-\mathrm{j}\frac{2\pi}{\lambda_k} d_l(\mathbf{x})} && (3) \end{aligned}$$

LOCATION-TO-CHANNEL MAPPING

- Channel impulse response ($\mathbf{x} \in \mathbb{R}^3$: receiver location):

$$h(\tau, \mathbf{x}) = \sum_{l=1}^{L_p} \gamma_l(\mathbf{x}) \, \delta(\tau - \tau_l(\mathbf{x})) \tag{1}$$

- Channel frequency response (f_k: considered frequency):

$$h(f_k, \mathbf{x}) = \sum_{l=1}^{L_p} \gamma_l(\mathbf{x}) \, \mathrm{e}^{-\mathrm{j}2\pi f_k \tau_l(\mathbf{x})} \tag{2}$$

$$= \sum_{l=1}^{L_p} \frac{\alpha_l \mathrm{e}^{\mathrm{j}\beta_l}}{d_l(\mathbf{x})} \mathrm{e}^{-\mathrm{j}\frac{2\pi}{\lambda_k} d_l(\mathbf{x})} \tag{3}$$

- How to learn this mapping in a system with N_a antennas operating on N_s frequencies?

TOWARDS INR

- Use of the Implicit Neural Representation (INR) concept:

TOWARDS INR

- Use of the Implicit Neural Representation (INR) concept:
    - Neural networks are universal function approximators[1,2]

---

[1] Hornik, Stinchcombe, and White, “Multilayer feedforward networks are universal approximators”.
[2] Cybenko, “Approximation by superpositions of a sigmoidal function”.

TOWARDS INR

- Use of the Implicit Neural Representation (INR) concept:
  - Neural networks are universal function approximators[1,2]
  - Using $\mathbf{x}$, one can design and train a neural network in a supervised manner to learn a representation of $\mathbf{H}\left(\mathbf{x}\right)$

[1] Hornik, Stinchcombe, and White, "Multilayer feedforward networks are universal approximators".
[2] Cybenko, "Approximation by superpositions of a sigmoidal function".

TOWARDS INR

- Use of the Implicit Neural Representation (INR) concept:
  - Neural networks are universal function approximators[1,2]
  - Using $\mathbf{x}$, one can design and train a neural network in a supervised manner to learn a representation of $\mathbf{H}\left(\mathbf{x}\right)$

- Goal: learn

$$\begin{aligned} f_{\boldsymbol{\theta}} \colon \mathbb{R}^3 &\longrightarrow \mathbb{C}^{N_a \times N_s} \\ \mathbf{x} &\longrightarrow \mathbf{H}\left(\mathbf{x}\right) \end{aligned} \tag{4}$$

[1] Hornik, Stinchcombe, and White, "Multilayer feedforward networks are universal approximators".
[2] Cybenko, "Approximation by superpositions of a sigmoidal function".

TOWARDS INR

- Use of the Implicit Neural Representation (INR) concept:
  - Neural networks are universal function approximators[1,2]
  - Using $\mathbf{x}$, one can design and train a neural network in a supervised manner to learn a representation of $\mathbf{H}(\mathbf{x})$

- Goal: learn

$$\begin{aligned} f_{\boldsymbol{\theta}} : \mathbb{R}^3 &\longrightarrow \mathbb{C}^{N_a \times N_s} \\ \mathbf{x} &\longrightarrow \mathbf{H}(\mathbf{x}) \end{aligned} \tag{4}$$

How to structure and learn $f_{\boldsymbol{\theta}}(\mathbf{x})$?

[1] Hornik, Stinchcombe, and White, "Multilayer feedforward networks are universal approximators".
[2] Cybenko, "Approximation by superpositions of a sigmoidal function".

SPECTRAL BIAS

- Classical architecture (MLPs) are biased towards learning low frequency content[3,4]

[3] Rahaman et al., “On the spectral bias of neural networks”.
[4] Cao et al., “Towards Understanding the Spectral Bias of Deep Learning”.

SPECTRAL BIAS

- Classical architecture (MLPs) are biased towards learning low frequency content[3,4]

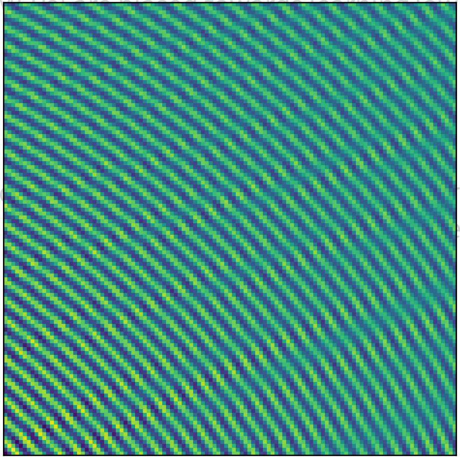
$$h\left(f_k, \mathbf{x}\right) = \sum_{l=1}^{L_p} \frac{\alpha_l \mathrm{e}^{\mathrm{j}\beta_l}}{d_l\left(\mathbf{x}\right)} \mathrm{e}^{-\mathrm{j}\frac{2\pi}{\lambda_k} d_l(\mathbf{x})} \tag{5}$$

- High frequency spatial dependence due to the exponential argument: small change in $\mathbf{x}$ leads to a huge change in $\mathbf{H}\left(\mathbf{x}\right) \rightarrow$ on the order of the wavelength

[3] Rahaman et al., "On the spectral bias of neural networks".
[4] Cao et al., "Towards Understanding the Spectral Bias of Deep Learning".

SPECTRAL BIAS

- Classical architecture (... frequency content[3,4]



(5)

- High frequency spatial ... gument: small change in $\mathbf{x}$ leads to a huge change ... ngth

[3] Rahaman et al., “On the spectral bias of neural networks”.
[4] Cao et al., “Towards Understanding the Spectral Bias of Deep Learning”.

SPECTRAL BIAS

- Classical architecture (MLPs) are biased towards learning low frequency content[3,4]

$$h\left(f_k, \mathbf{x}\right) = \sum_{l=1}^{L_p} \frac{\alpha_l \mathrm{e}^{\mathrm{j}\beta_l}}{d_l\left(\mathbf{x}\right)} \mathrm{e}^{-\mathrm{j}\frac{2\pi}{\lambda_k} d_l(\mathbf{x})} \tag{5}$$

- High frequency spatial dependence due to the exponential argument: small change in $\mathbf{x}$ leads to a huge change in $\mathbf{H}\left(\mathbf{x}\right) \rightarrow$ on the order of the wavelength

How to learn $f_{\boldsymbol{\theta}}\left(\mathbf{x}\right)$ without suffering from the spectral bias?

[3] Rahaman et al., "On the spectral bias of neural networks".
[4] Cao et al., "Towards Understanding the Spectral Bias of Deep Learning".

Typical data processing setting:

- We observe a ***large*** number of ***correlated*** variables, explained by a ***small*** number of ***independent*** factors.

MODEL-BASED MACHINE LEARNING

Typical data processing setting:

- We observe a ***large*** number of ***correlated*** variables, explained by a ***small*** number of ***independent*** factors.

There are two complementary approaches to handle this situation:

MODEL-BASED MACHINE LEARNING

Typical data processing setting:

- We observe a ***large*** number of ***correlated*** variables, explained by a ***small*** number of ***independent*** factors.

There are two complementary approaches to handle this situation:

- **Signal processing**
  - Model based
  - Large bias
  - Low complexity

Typical data processing setting:

- We observe a ***large*** number of ***correlated*** variables, explained by a ***small*** number of ***independent*** factors.

There are two complementary approaches to handle this situation:

- **Signal processing**
  - Model based
  - Large bias
  - Low complexity
- **Machine learning**
  - Data based
  - Low bias
  - High complexity

MODEL-BASED MACHINE LEARNING

Typical data processing setting:

- We observe a ***large*** number of ***correlated*** variables, explained by a ***small*** number of ***independent*** factors.

There are two complementary approaches to handle this situation:

- **Signal processing**
  - Model based
  - Large bias
  - Low complexity
- **Machine learning**
  - Data based
  - Low bias
  - High complexity

Hybrid approach: Model-based machine learning

Use models to structure, initialize or optimize learning methods

Typical data processing setting:

- We observe a ***large*** number of ***correlated*** variables, explained by a ***small*** number of ***independent*** factors.

There are two complementary approaches to handle this situation:

- **Signal processing**
  - Model based
  - Large bias
  - Low complexity
- **Machine learning**
  - Data based
  - Low bias
  - High complexity

Hybrid approach: Model-based machine learning
Use models to structure, initialize or optimize learning methods

- Make models more flexible: reduce bias of signal processing methods

MODEL-BASED MACHINE LEARNING

Typical data processing setting:

- We observe a ***large*** number of ***correlated*** variables, explained by a ***small*** number of ***independent*** factors.

There are two complementary approaches to handle this situation:

- **Signal processing**
  - Model based
  - Large bias
  - Low complexity
- **Machine learning**
  - Data based
  - Low bias
  - High complexity

Hybrid approach: Model-based machine learning
Use models to structure, initialize or optimize learning methods

- Make models more flexible: reduce bias of signal processing methods
- Guide machine learning methods: reduce their complexity

MODEL-BASED MACHINE LEARNING

Typical data processing setting:

- We observe a ***large*** number of ***correlated*** variables, explained by a ***small*** number of ***independent*** factors.

There are two complementary approaches to handle this situation:

- **Signal processing**
  - Low complexity
- **Machine learning**
  - High complexity

Use the physical channel model to structure a neural network that overcomes the spectral bias

Hybrid approach: Model-based machine learning
Use models to structure, initialize or optimize learning methods

- Make models more flexible: reduce bias of signal processing methods
- Guide machine learning methods: reduce their complexity

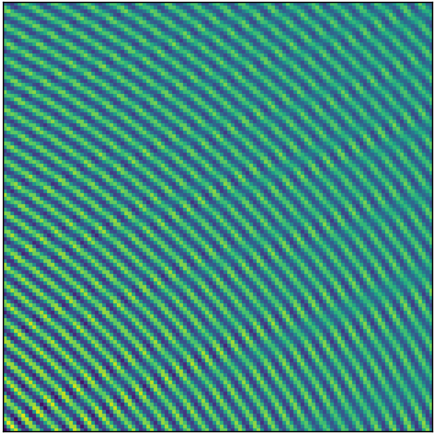
OVERCOMING THE SPECTRAL BIAS

- Main idea[5,6]: planar approximation of spherical wavefronts using Taylor expansions

[5] Chatelier et al., “Model-Based Learning for Location-to-Channel Mapping”.
[6] Chatelier et al., “Model-Based Learning for Multi-Antenna Multi-Frequency Location-to-Channel Mapping”.

OVERCOMING THE SPECTRAL BIAS

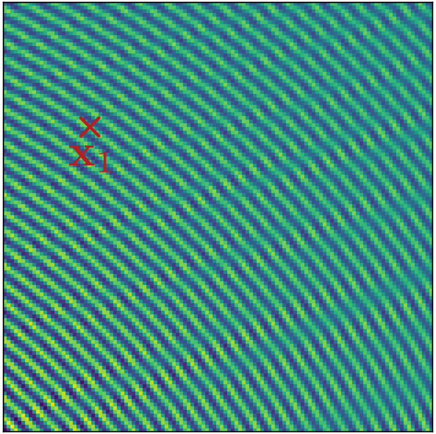
- Main idea[5,6]: planar approximation of spherical wavefronts using Taylor expansions



[5] Chatelier et al., "Model-Based Learning for Location-to-Channel Mapping".
[6] Chatelier et al., "Model-Based Learning for Multi-Antenna Multi-Frequency Location-to-Channel Mapping".

OVERCOMING THE SPECTRAL BIAS

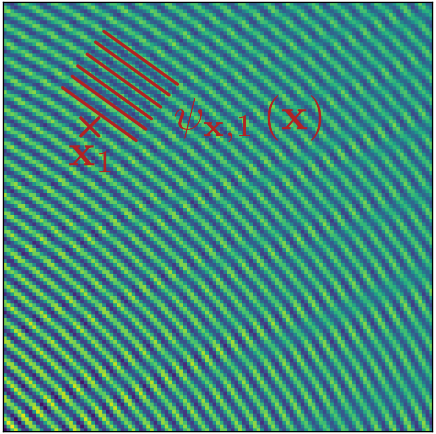
- Main idea[5,6]: planar approximation of spherical wavefronts using Taylor expansions



[5] Chatelier et al., “Model-Based Learning for Location-to-Channel Mapping”.
[6] Chatelier et al., “Model-Based Learning for Multi-Antenna Multi-Frequency Location-to-Channel Mapping”.

OVERCOMING THE SPECTRAL BIAS

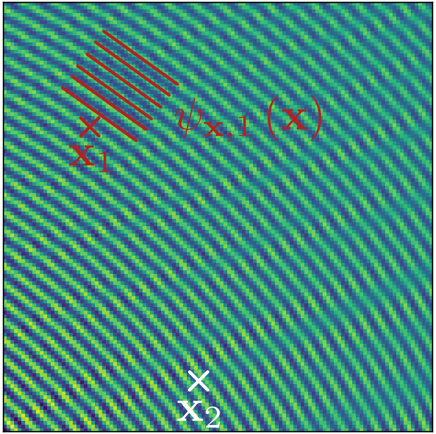
- Main idea[5,6]: planar approximation of spherical wavefronts using Taylor expansions



[5] Chatelier et al., "Model-Based Learning for Location-to-Channel Mapping".
[6] Chatelier et al., "Model-Based Learning for Multi-Antenna Multi-Frequency Location-to-Channel Mapping".

OVERCOMING THE SPECTRAL BIAS

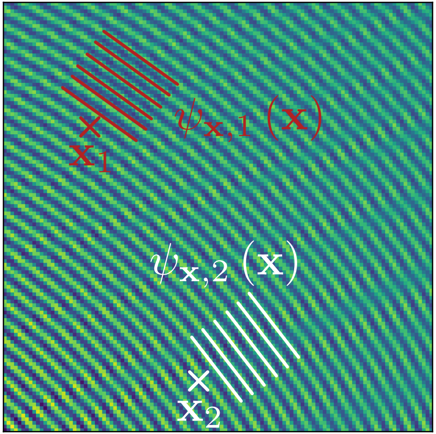
- Main idea[5,6]: planar approximation of spherical wavefronts using Taylor expansions



[5] Chatelier et al., "Model-Based Learning for Location-to-Channel Mapping".
[6] Chatelier et al., "Model-Based Learning for Multi-Antenna Multi-Frequency Location-to-Channel Mapping".

OVERCOMING THE SPECTRAL BIAS

- Main idea[5,6]: planar approximation of spherical wavefronts using Taylor expansions



[5] Chatelier et al., "Model-Based Learning for Location-to-Channel Mapping".
[6] Chatelier et al., "Model-Based Learning for Multi-Antenna Multi-Frequency Location-to-Channel Mapping".

GENERALIZATION

- This results in a sparse recovery problem with a slowly varying coefficients vector $\mathbf{w}(\mathbf{x})$:

GENERALIZATION

- This results in a sparse recovery problem with a slowly varying coefficients vector $\mathbf{w}(\mathbf{x})$:

$$\forall \mathbf{x} \in \mathbb{R}^3,\ \mathbf{H}(\mathbf{x}) \simeq \sum_{i=1}^{D} \left(\mathbf{w}(\mathbf{x}) \odot \tilde{\boldsymbol{\psi}}_{\mathbf{x}}(\mathbf{x}) \right) \tilde{\boldsymbol{\psi}}_{\mathbf{a},i} \tilde{\boldsymbol{\psi}}_{\mathbf{f},i}^{\mathsf{T}} \quad (6)$$
$$\text{with } \|\mathbf{w}(\mathbf{x})\|_0 = L_p$$

  - $\tilde{\boldsymbol{\psi}}_{\mathbf{a},i}$ and $\tilde{\boldsymbol{\psi}}_{\mathbf{f},i}$ allow antenna and frequency extrapolation

GENERALIZATION

- This results in a sparse recovery problem with a slowly varying coefficients vector $\mathbf{w}(\mathbf{x})$:

$$\forall \mathbf{x} \in \mathbb{R}^3,\ \mathbf{H}(\mathbf{x}) \simeq \sum_{i=1}^{D} \left(\mathbf{w}(\mathbf{x}) \odot \tilde{\boldsymbol{\psi}}_{\mathbf{x}}(\mathbf{x})\right) \tilde{\boldsymbol{\psi}}_{\mathbf{a},i} \tilde{\boldsymbol{\psi}}_{\mathbf{f},i}^{\mathsf{T}} \quad (6)$$
$$\text{with } \|\mathbf{w}(\mathbf{x})\|_0 = L_p$$

  - $\tilde{\boldsymbol{\psi}}_{\mathbf{a},i}$ and $\tilde{\boldsymbol{\psi}}_{\mathbf{f},i}$ allow antenna and frequency extrapolation

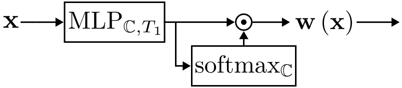
- Alternative formulation:

$$\forall \mathbf{x} \in \mathbb{R}^3,\ \text{vec}\left(\mathbf{H}(\mathbf{x})\right) \simeq \left(\tilde{\boldsymbol{\Psi}}_{\mathbf{f}}(\mathbf{x}) \otimes \tilde{\boldsymbol{\Psi}}_{\mathbf{a}}(\mathbf{x})\right) \text{vec}\left(\text{diag}\left(\mathbf{w}(\mathbf{x}) \odot \tilde{\boldsymbol{\psi}}_{\mathbf{x}}(\mathbf{x})\right)\right) \quad (7)$$
$$\text{with } \|\mathbf{w}(\mathbf{x})\|_0 = L_p$$

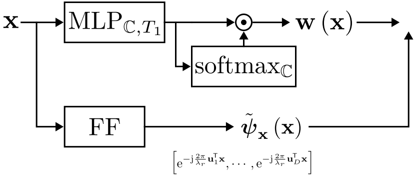
MODEL-BASED NEURAL ARCHITECTURE

$\mathbf{x}$

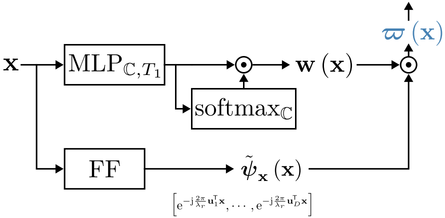
MODEL-BASED NEURAL ARCHITECTURE



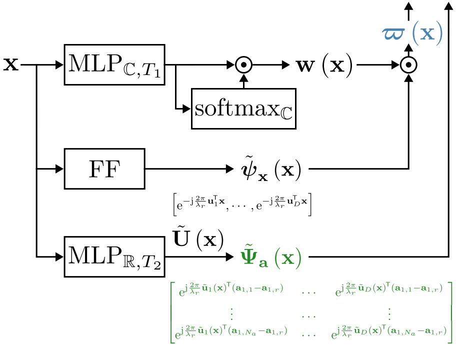
MODEL-BASED NEURAL ARCHITECTURE

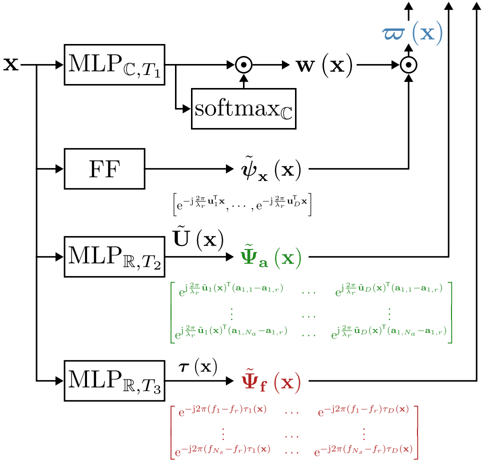


MODEL-BASED NEURAL ARCHITECTURE

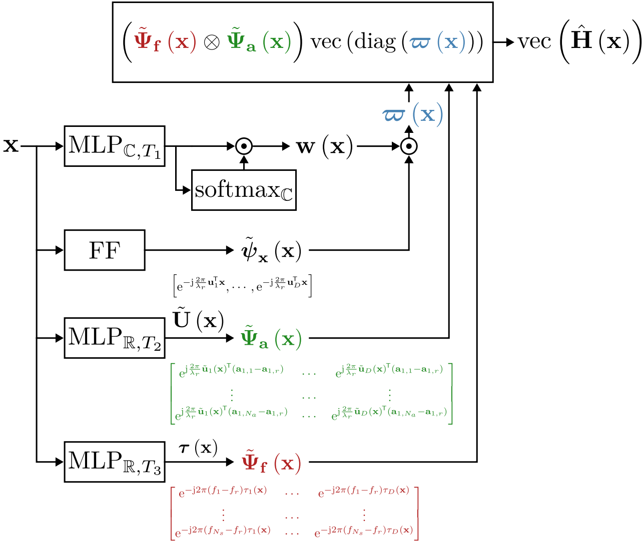


MODEL-BASED NEURAL ARCHITECTURE




$\mathbf{x}$
$\mathrm{MLP}_{\mathbb{C},T_1}$
$\mathrm{softmax}_{\mathbb{C}}$
$\mathbf{w}(\mathbf{x})$
$\boldsymbol{\varpi}(\mathbf{x})$
FF
$\tilde{\boldsymbol{\psi}}_{\mathbf{x}}(\mathbf{x})$
$\left[e^{-j\frac{2\pi}{\lambda_r}\mathbf{u}_1^\mathsf{T}\mathbf{x}}, \dots, e^{-j\frac{2\pi}{\lambda_r}\mathbf{u}_D^\mathsf{T}\mathbf{x}}\right]$
$\mathrm{MLP}_{\mathbb{R},T_2}$
$\tilde{\mathbf{U}}(\mathbf{x})$
$\tilde{\boldsymbol{\Psi}}_{\mathbf{a}}(\mathbf{x})$
$\begin{bmatrix} e^{j\frac{2\pi}{\lambda_r}\tilde{\mathbf{u}}_1(\mathbf{x})^\mathsf{T}(\mathbf{a}_{1,1}-\mathbf{a}_{1,r})} & \dots & e^{j\frac{2\pi}{\lambda_r}\tilde{\mathbf{u}}_D(\mathbf{x})^\mathsf{T}(\mathbf{a}_{1,1}-\mathbf{a}_{1,r})} \\ \vdots & \dots & \vdots \\ e^{j\frac{2\pi}{\lambda_r}\tilde{\mathbf{u}}_1(\mathbf{x})^\mathsf{T}(\mathbf{a}_{1,N_a}-\mathbf{a}_{1,r})} & \dots & e^{j\frac{2\pi}{\lambda_r}\tilde{\mathbf{u}}_D(\mathbf{x})^\mathsf{T}(\mathbf{a}_{1,N_a}-\mathbf{a}_{1,r})} \end{bmatrix}$
$\mathrm{MLP}_{\mathbb{R},T_3}$
$\boldsymbol{\tau}(\mathbf{x})$
$\tilde{\boldsymbol{\Psi}}_{\mathbf{f}}(\mathbf{x})$
$\begin{bmatrix} e^{-j2\pi(f_1-f_r)\tau_1(\mathbf{x})} & \dots & e^{-j2\pi(f_1-f_r)\tau_D(\mathbf{x})} \\ \vdots & \dots & \vdots \\ e^{-j2\pi(f_{N_s}-f_r)\tau_1(\mathbf{x})} & \dots & e^{-j2\pi(f_{N_s}-f_r)\tau_D(\mathbf{x})} \end{bmatrix}$

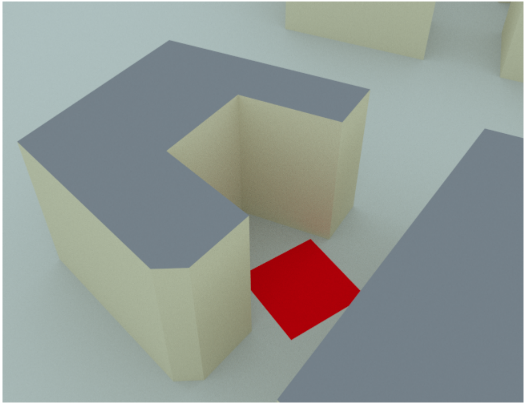
MODEL-BASED NEURAL ARCHITECTURE



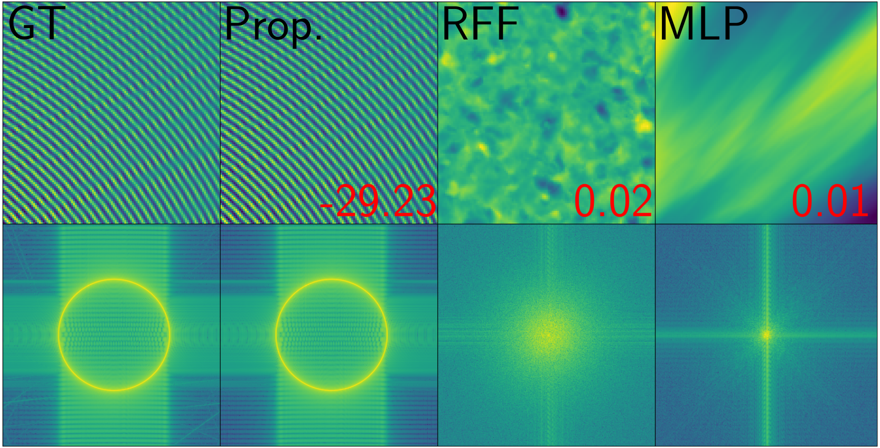
MB-ML: we used the channel model to structure a neural network

LEARNING FRAMEWORK

- Scene:
  - 10m by 10m square plane
  - Uniformly dropped train/test locations
  - Performance evaluation on $\lambda/4$ uniform grid $(210$k locs.)



RESULTS



- Top row: real part of the reconstructed channels with NMSE in dB (in red)
- Bottom row: 2D Fourier transform of the reconstruction

POTENTIAL APPLICATIONS

- The objective was to learn:

$$\begin{aligned} f_{\boldsymbol{\theta}} \colon \mathbb{R}^3 &\longrightarrow \mathbb{C}^{N_a \times N_s} \\ \mathbf{x} &\longrightarrow \mathbf{H}(\mathbf{x}) \end{aligned} \tag{8}$$

POTENTIAL APPLICATIONS

- The objective was to learn:

$$\begin{aligned} f_{\boldsymbol{\theta}} : \mathbb{R}^3 &\longrightarrow \mathbb{C}^{N_a \times N_s} \\ \mathbf{x} &\longrightarrow \mathbf{H}(\mathbf{x}) \end{aligned} \tag{8}$$

- Once trained, this model achieves near-perfect channel reconstruction, *at the sub-wavelength level*

POTENTIAL APPLICATIONS

- The objective was to learn:

$$
\begin{aligned}
f_{\boldsymbol{\theta}} : \mathbb{R}^3 &\longrightarrow \mathbb{C}^{N_a \times N_s} \\
\mathbf{x} &\longrightarrow \mathbf{H}(\mathbf{x})
\end{aligned}
\tag{8}
$$

- Once trained, this model achieves near-perfect channel reconstruction, *at the sub-wavelength level*

f_θ can be seen as a generative neural channel model in the considered scene

POTENTIAL APPLICATIONS

- The objective was to learn:

$$\begin{aligned} f_{\boldsymbol{\theta}}: \mathbb{R}^3 &\longrightarrow \mathbb{C}^{N_a \times N_s} \\ \mathbf{x} &\longrightarrow \mathbf{H}(\mathbf{x}) \end{aligned} \tag{8}$$

- Once trained, this model achieves near-perfect channel reconstruction, *at the sub-wavelength level*

f_θ can be seen as a generative neural channel model in the considered scene

- Potential applications include:

POTENTIAL APPLICATIONS

- The objective was to learn:

$$\begin{aligned} f_{\boldsymbol{\theta}} : \mathbb{R}^3 &\longrightarrow \mathbb{C}^{N_a \times N_s} \\ \mathbf{x} &\longrightarrow \mathbf{H}(\mathbf{x}) \end{aligned} \tag{8}$$

- Once trained, this model achieves near-perfect channel reconstruction, *at the sub-wavelength level*

f_θ can be seen as a generative neural channel model in the considered scene

- Potential applications include:
  - Channel prediction

POTENTIAL APPLICATIONS

- The objective was to learn:

$$\begin{aligned} f_{\boldsymbol{\theta}} : \mathbb{R}^3 &\longrightarrow \mathbb{C}^{N_a \times N_s} \\ \mathbf{x} &\longrightarrow \mathbf{H}\left(\mathbf{x}\right) \end{aligned} \tag{8}$$

- Once trained, this model achieves near-perfect channel reconstruction, *at the sub-wavelength level*

f_{θ} can be seen as a generative neural channel model in the considered scene

- Potential applications include:
  - Channel prediction
  - **Precise localization**

RADIO LOCALIZATION

- Given $\mathbf{H}(\mathbf{x})$, how to estimate $\mathbf{x}$?

RADIO LOCALIZATION

- Given $\mathbf{H}\left(\mathbf{x}\right)$, how to estimate $\mathbf{x}$?

- Fingerprinting-based localization:

RADIO LOCALIZATION

- Given $\mathbf{H}(\mathbf{x})$, how to estimate $\mathbf{x}$?

- Fingerprinting-based localization:
    - Estimate $\mathbf{x}$ as the position in a dictionary $\mathcal{G}$ whose associated channel coefficients are the most similar with $\mathbf{H}(\mathbf{x})$.

$$\hat{\mathbf{x}}(\mathbf{H}(\mathbf{x})) = \underset{\tilde{\mathbf{x}} \in \mathcal{G}}{\arg\max}\, \mathsf{sim}(\mathbf{H}(\mathbf{x}), \mathbf{H}(\tilde{\mathbf{x}})) \tag{9}$$

RADIO LOCALIZATION

- Given $\mathbf{H}(\mathbf{x})$, how to estimate $\mathbf{x}$?

- Fingerprinting-based localization:
  - Estimate $\mathbf{x}$ as the position in a dictionary $\mathcal{G}$ whose associated channel coefficients are the most similar with $\mathbf{H}(\mathbf{x})$.

$$\hat{\mathbf{x}}(\mathbf{H}(\mathbf{x})) = \arg\max_{\tilde{\mathbf{x}} \in \mathcal{G}} \mathsf{sim}(\mathbf{H}(\mathbf{x}), \mathbf{H}(\tilde{\mathbf{x}})) \tag{9}$$

Localization accuracy is limited by the dictionary resolution

RADIO LOCALIZATION

- Given $\mathbf{H}(\mathbf{x})$, how to estimate $\mathbf{x}$?

- Fingerprinting-based localization:
  - Estimate $\mathbf{x}$ as the position in a dictionary $\mathcal{G}$ whose associated channel coefficients are the most similar with $\mathbf{H}(\mathbf{x})$.

$$\hat{\mathbf{x}}(\mathbf{H}(\mathbf{x})) = \underset{\tilde{\mathbf{x}} \in \mathcal{G}}{\arg\max}\, \mathsf{sim}(\mathbf{H}(\mathbf{x}), \mathbf{H}(\tilde{\mathbf{x}})) \tag{9}$$

Localization accuracy is limited by the dictionary resolution

- Idea: use the trained f_{θ} to generate channel coefficients at wanted locations to enhance localization accuracy

PROPOSED LOCALIZATION METHOD

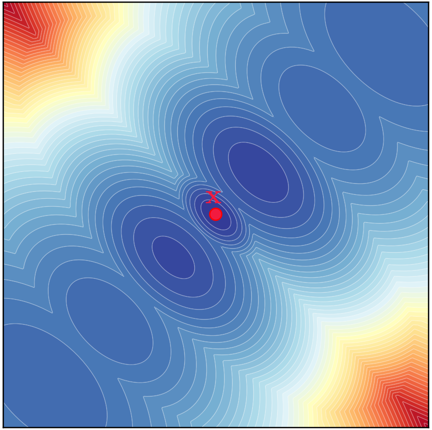
- Based on grid-search and gradient descent[7], using a Frobenius norm similarity measure:

$$\mu_{\mathsf{PS}}\left(\mathbf{H}\left(\mathbf{x}\right), \tilde{\mathbf{x}} | \boldsymbol{\theta}\right) = \left\| \mathbf{H}\left(\mathbf{x}\right) - f_{\boldsymbol{\theta}}\left(\tilde{\mathbf{x}}\right) \right\|_{\mathsf{F}} \tag{10}$$

[7] Chatelier et al., *Model-based Implicit Neural Representation for sub-wavelength Radio Localization*.

PROPOSED LOCALIZATION METHOD

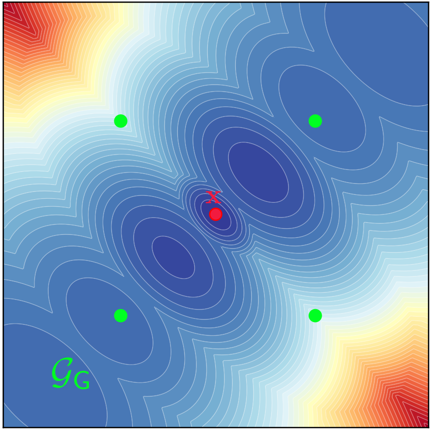
- How to estimate $\mathbf{x}$?
- Background: $\left\|\mathbf{H}\left(\mathbf{x}\right) - f_{\boldsymbol{\theta}}\left(\tilde{\mathbf{x}}\right)\right\|_{\mathsf{F}}$



PROPOSED LOCALIZATION METHOD

- Generate the global grid $\mathcal{G}_{\mathsf{G}}$ based on topological knowledge of the scene

Global grid: $\mathcal{O}\left(\left|\mathcal{G}_{\mathsf{G}}\right| \kappa_{f_{\boldsymbol{\theta}}}\right)$

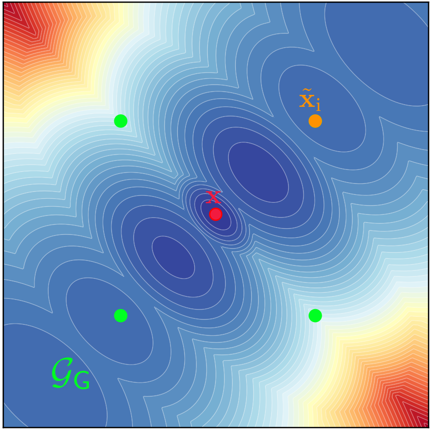


PROPOSED LOCALIZATION METHOD

- Generate the global grid $\mathcal{G}_\mathsf{G}$ based on topological knowledge of the scene
- Using $f_{\boldsymbol{\theta}}$, solve:

$$\tilde{\mathbf{x}}_\mathrm{i} = \underset{\tilde{\mathbf{x}} \in \mathcal{G}_\mathsf{G}}{\arg\min} \left\| \mathbf{H}\left(\mathbf{x}\right) - f_{\boldsymbol{\theta}}\left(\tilde{\mathbf{x}}\right) \right\|_\mathsf{F} \tag{10}$$

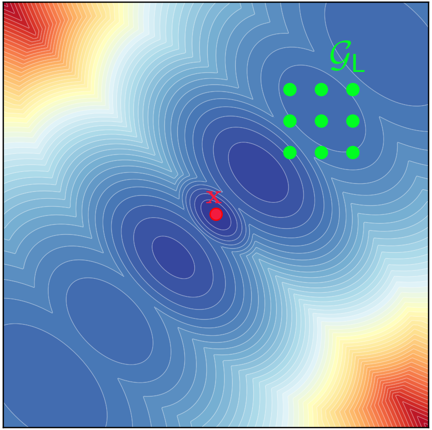
Global grid: $\mathcal{O}\left(\left|\mathcal{G}_\mathsf{G}\right| \kappa_{f_{\boldsymbol{\theta}}}\right)$



PROPOSED LOCALIZATION METHOD

- Generate the local grid $\mathcal{G}_\mathsf{L}$ around the obtained location

Local grid: $\mathcal{O}\left(|\mathcal{G}_\mathsf{L}|\, \kappa_{f_{\boldsymbol{\theta}}}\right)$

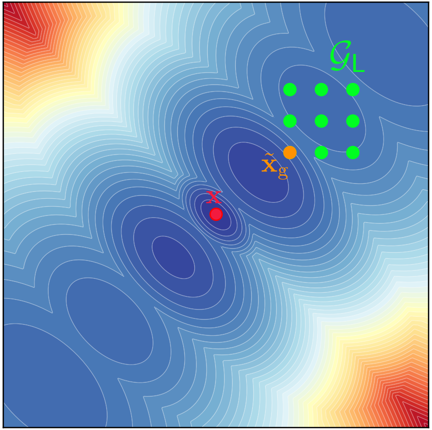


PROPOSED LOCALIZATION METHOD

- Generate the local grid $\mathcal{G}_\mathsf{L}$ around the obtained location
- Using $f_{\boldsymbol{\theta}}$, solve:

$$\tilde{\mathbf{x}}_\mathrm{g} = \underset{\tilde{\mathbf{x}} \in \mathcal{G}_\mathsf{L}}{\arg\min} \left\| \mathbf{H}\left(\mathbf{x}\right) - f_{\boldsymbol{\theta}}\left(\tilde{\mathbf{x}}\right) \right\|_\mathsf{F} \tag{10}$$

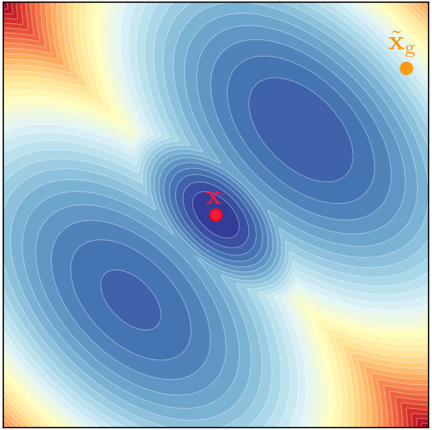
Local grid: $\mathcal{O}\left(\left|\mathcal{G}_\mathsf{L}\right| \kappa_{f_{\boldsymbol{\theta}}}\right)$



PROPOSED LOCALIZATION METHOD

- Perform N_∇ gradient descent steps

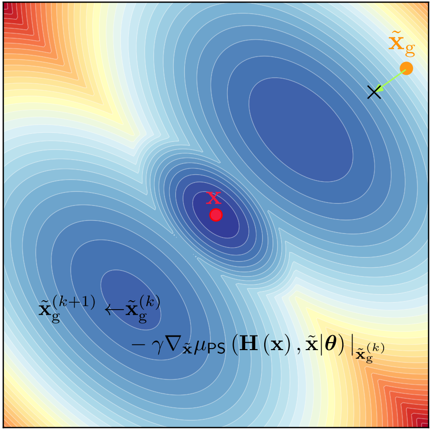
$$\mathrm{GD}_1\text{: } \mathcal{O}\left(N_\nabla \kappa_{f_{\boldsymbol{\theta}}}\right)$$



PROPOSED LOCALIZATION METHOD

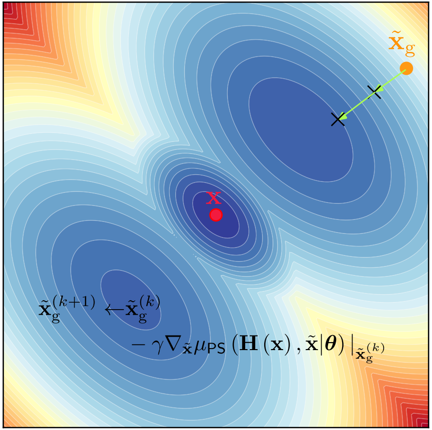
- Perform N_∇ gradient descent steps

$$\mathrm{GD}_1\colon\ \mathcal{O}\left(N_\nabla \kappa_{f_{\boldsymbol{\theta}}}\right)$$



- Perform N_∇ gradient descent steps

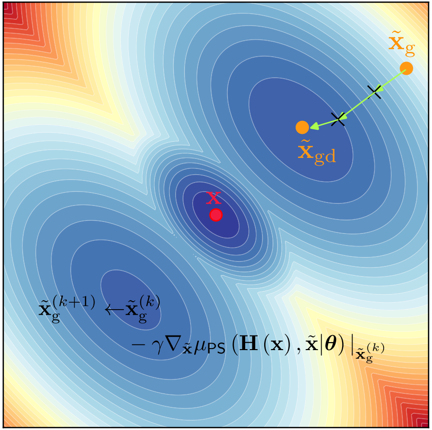
$$\mathrm{GD}_1\text{:}\ \mathcal{O}\left(N_\nabla \kappa_{f_{\boldsymbol{\theta}}}\right)$$



PROPOSED LOCALIZATION METHOD

- Perform N_∇ gradient descent steps

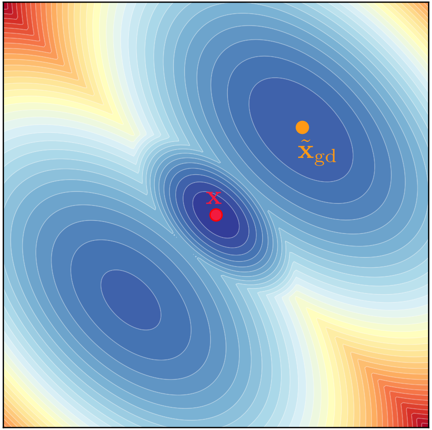
$$\mathrm{GD}_1\colon\ \mathcal{O}\left(N_\nabla \kappa_{f_{\boldsymbol{\theta}}}\right)$$



PROPOSED LOCALIZATION METHOD

- Perform N_∇ gradient descent steps
- Local minima issue

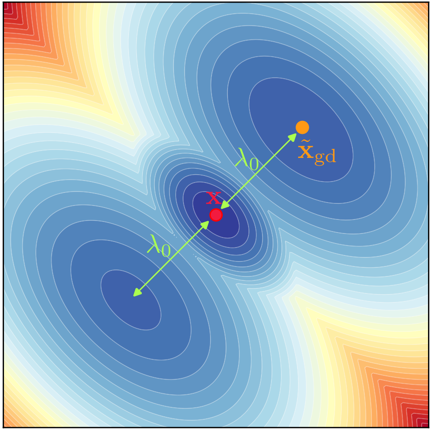
Circles: $\mathcal{O}\left(|\mathcal{G}_{\mathsf{C}}|\,\kappa_{f_{\boldsymbol{\theta}}}\right)$



PROPOSED LOCALIZATION METHOD

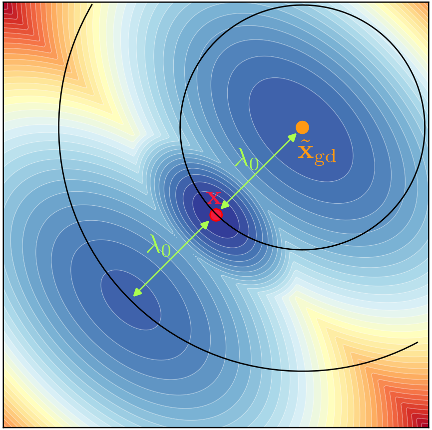
- Perform N_∇ gradient descent steps
- Local minima issue
- Spacing between minima derived from μ_{PS}

Circles: $\mathcal{O}\left(|\mathcal{G}_{\mathsf{C}}|\,\kappa_{f_{\boldsymbol{\theta}}}\right)$



- Perform N_∇ gradient descent steps
- Local minima issue
- Spacing between minima derived from μ_{PS}
- Generate circles of radius $k\lambda_0$, $k \in \mathbb{N}^*$

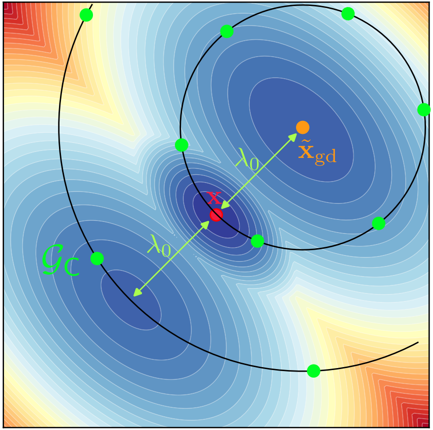
Circles: $\mathcal{O}\left(|\mathcal{G}_{\mathsf{C}}|\,\kappa_{f_{\boldsymbol{\theta}}}\right)$



PROPOSED LOCALIZATION METHOD

- Perform N_∇ gradient descent steps
- Local minima issue
- Spacing between minima derived from μ_{PS}
- Generate circles of radius $k\lambda_0$, $k \in \mathbb{N}^*$
- Generate $\mathcal{G}_{\mathrm{C}}$ by sampling from the circles

Circles: $\mathcal{O}\left(|\mathcal{G}_{\mathrm{C}}|\,\kappa_{f_{\boldsymbol{\theta}}}\right)$

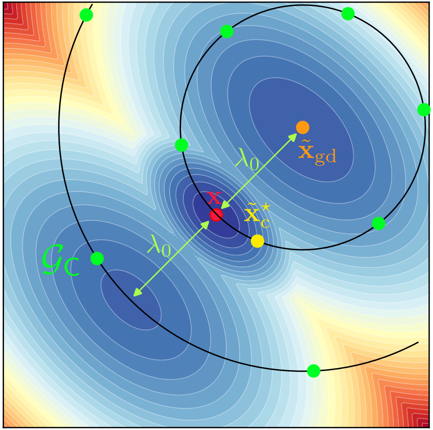


PROPOSED LOCALIZATION METHOD

- Perform N_∇ gradient descent steps
- Local minima issue
- Spacing between minima derived from μ_{PS}
- Generate circles of radius $k\lambda_0$, $k \in \mathbb{N}^*$
- Generate $\mathcal{G}_\text{C}$ by sampling from the circles
- Using $f_{\boldsymbol{\theta}}$, solve:

$$\tilde{\mathbf{x}}_{\text{c}^\star} = \underset{\tilde{\mathbf{x}} \in \mathcal{G}_\text{C}}{\arg\min} \left\| \mathbf{H}(\mathbf{x}) - f_{\boldsymbol{\theta}}(\tilde{\mathbf{x}}) \right\|_\text{F} \tag{10}$$

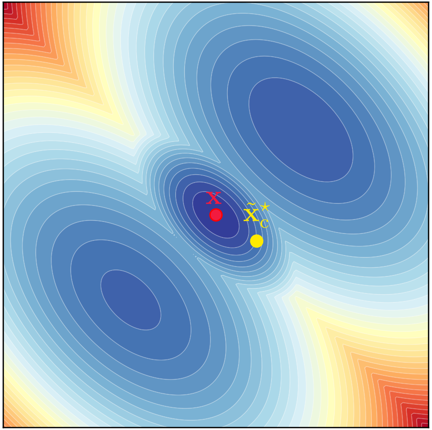
Circles: $\mathcal{O}\left(|\mathcal{G}_\text{C}|\,\kappa_{f_{\boldsymbol{\theta}}}\right)$



PROPOSED LOCALIZATION METHOD

- Perform N_∇ gradient descent steps

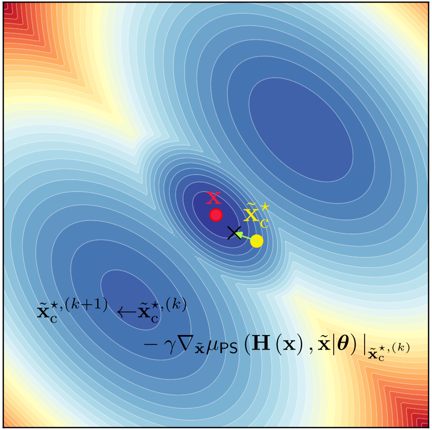
$$\mathrm{GD}_2\colon\ \mathcal{O}\left(N_\nabla \kappa_{f_{\boldsymbol{\theta}}}\right)$$



PROPOSED LOCALIZATION METHOD

- Perform N_∇ gradient descent steps

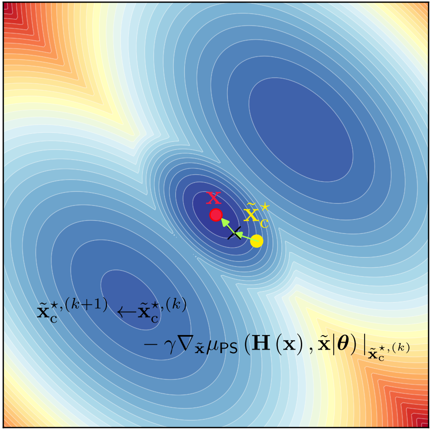
$$\mathrm{GD}_2\colon\ \mathcal{O}\left(N_\nabla \kappa_{f_{\boldsymbol{\theta}}}\right)$$



PROPOSED LOCALIZATION METHOD

- Perform N_∇ gradient descent steps

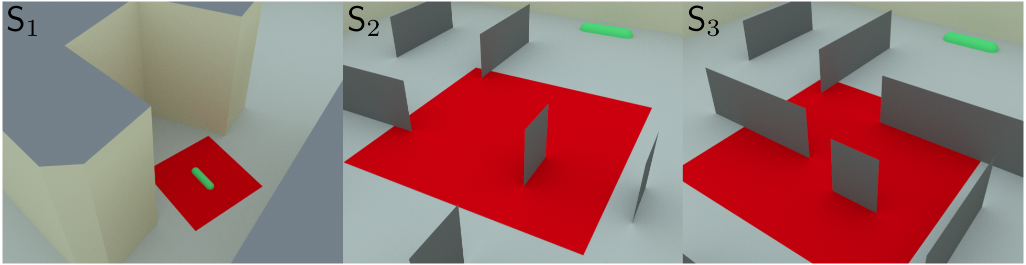
$$\mathrm{GD}_2\text{:}\ \mathcal{O}\left(N_\nabla \kappa_{f_{\boldsymbol{\theta}}}\right)$$



- Perform N_∇ gradient descent steps

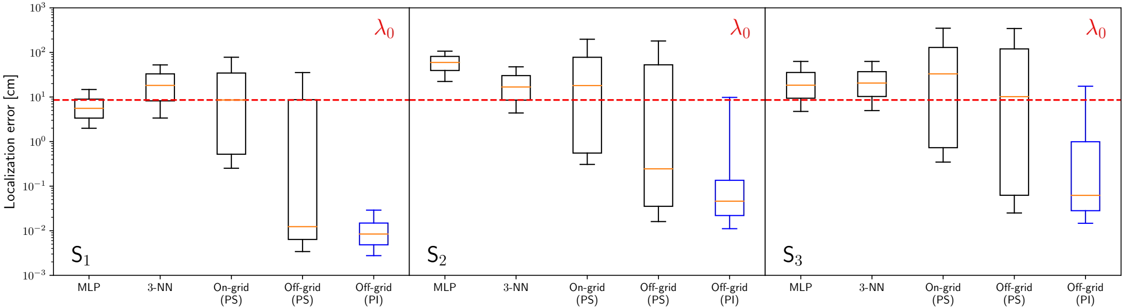
MB-ML: we used the channel model to structure a neural network and optimize a gradient descent process

SIMULATION SETUP



- Localization performance evaluated on 10k independent locations within the red plane

LOCALIZATION PERFORMANCE



- PI: phase insensitive similarity measure, used during the grid search on the global grid to mitigate the local minima issue
- Sub-wavelength median localization accuracy for the proposed method (in blue)

CONCLUSION

- Combining INR and MB-ML allowed us to:

CONCLUSION

- Combining INR and MB-ML allowed us to:
    - Learn the location-to-channel mapping and overcome the spectral bias issue

CONCLUSION

- Combining INR and MB-ML allowed us to:
  - Learn the location-to-channel mapping and overcome the spectral bias issue
  - Optimize a localization method based on the proposed network

CONCLUSION

- Combining INR and MB-ML allowed us to:
  - Learn the location-to-channel mapping and overcome the spectral bias issue
  - Optimize a localization method based on the proposed network

- General advantages of MB-ML:

CONCLUSION

- Combining INR and MB-ML allowed us to:
  - Learn the location-to-channel mapping and overcome the spectral bias issue
  - Optimize a localization method based on the proposed network

- General advantages of MB-ML:
  - Increased interpretability

CONCLUSION

- Combining INR and MB-ML allowed us to:
  - Learn the location-to-channel mapping and overcome the spectral bias issue
  - Optimize a localization method based on the proposed network

- General advantages of MB-ML:
  - Increased interpretability
  - Reduced complexity

CONCLUSION

- Combining INR and MB-ML allowed us to:
  - Learn the location-to-channel mapping and overcome the spectral bias issue
  - Optimize a localization method based on the proposed network

- General advantages of MB-ML:
  - Increased interpretability
  - Reduced complexity
  - Often better performance

CONCLUSION

- Combining INR and MB-ML allowed us to:
  - Learn the location-to-channel mapping and overcome the spectral bias issue
  - Optimize a localization method based on the proposed network

- General advantages of MB-ML:
  - Increased interpretability
  - Reduced complexity
  - Often better performance

(Mapping learning)

(Localization)

Thanks!