/ Model-based learning for location-to-channel mapping /

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

† Univ Rennes, INSA Rennes, CNRS, IETR-UMR 6164, Rennes, France
‡ Mitsubishi Electric R&D Centre Europe, Rennes, France
⋆ b<>com, Rennes, France

baptiste.chatelier@insa-rennes.fr

- Typical data processing setting:
  - We observe a ***large*** number of ***correlated*** variables, explained by a ***small*** number of ***independent*** factors.

- 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:

- 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
- **ML/AI**
  - Data based
  - Low bias
  - High 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
- **ML/AI**
  - Data based
  - Low bias
  - High complexity

Hybrid approach: Model-based AI
Use models to structure, initialize and train 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
- **ML/AI**
  - Data based
  - Low bias
  - High complexity

Hybrid approach: Model-based AI
Use models to structure, initialize and train learning methods

- Make models more flexible: reduce bias of signal processing 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
- **ML/AI**
  - Data based
  - Low bias
  - High complexity

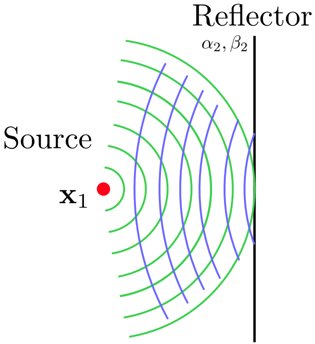
Hybrid approach: Model-based AI
Use models to structure, initialize and train learning methods

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

- In a SISO-monocarrier setting, the channel can be expressed as:

- In a SISO-monocarrier setting, the channel can be expressed as:
  - Hypothesis: attenuation/phase proportional to propagation distance

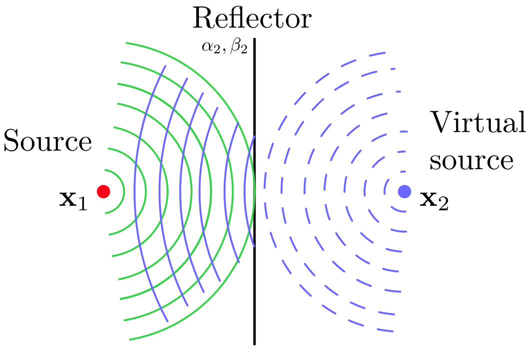
$$h(\mathbf{x}) = \sum_{l=1}^{L_p} \frac{\alpha_l e^{j\beta_l}}{d_l} e^{-j\frac{2\pi}{\lambda} d_l} \quad (1)$$



- In a SISO-monocarrier setting, the channel can be expressed as:
  - Hypothesis: attenuation/phase proportional to propagation distance

$$h(\mathbf{x}) = \sum_{l=1}^{L_p} \frac{\alpha_l e^{j\beta_l}}{d_l} e^{-j\frac{2\pi}{\lambda} d_l} \quad (1)$$

$$= \sum_{l=1}^{L_p} \frac{\alpha_l e^{j\beta_l}}{\|\mathbf{x} - \mathbf{x}_l\|_2} e^{-j\frac{2\pi}{\lambda}\|\mathbf{x}-\mathbf{x}_l\|_2} \quad (2)$$



- How to learn the location-to-channel mapping ?

- How to learn the location-to-channel mapping ?
- Use of the Implicit Neural Representation (INR) concept:

- How to learn the location-to-channel mapping ?
- 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”.

- How to learn the location-to-channel mapping ?

- 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 $h(\mathbf{x})$

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

- How to learn the location-to-channel mapping ?

- 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 $h(\mathbf{x})$

- Goal: learn

$$\begin{aligned} f_{\boldsymbol{\theta}}: \mathbb{R}^2 &\longrightarrow \mathbb{C} \\ \mathbf{x} &\longrightarrow h(\mathbf{x}), \end{aligned} \tag{3}$$

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

- How to learn the location-to-channel mapping ?
- Use of the Implicit Neural Representation (INR) concept:
  - Neural networks are universal function approximators
  - Using $\mathbf{x}$, one can design and train a neural network in a supervised manner to learn a representation of $h(\mathbf{x})$
- Goal: learn

$$\begin{aligned} f_{\boldsymbol{\theta}} : \mathbb{R}^2 &\longrightarrow \mathbb{C} \\ \mathbf{x} &\longrightarrow h(\mathbf{x}), \end{aligned} \tag{3}$$

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

- Classical architecture (MLPs) are biased towards learning low frequency content[1,2]

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

- Classical architecture (MLPs) are biased towards learning low frequency content[1],[2]

$$h\left(\mathbf{x}\right) = \sum_{l=1}^{L_p} \frac{\alpha_l \mathrm{e}^{\mathrm{j}\beta_l}}{\left\|\mathbf{x} - \mathbf{x}_l\right\|_2} \mathrm{e}^{-\mathrm{j}\frac{2\pi}{\lambda}\left\|\mathbf{x}-\mathbf{x}_l\right\|_2} \quad (4)$$

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

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

- Classical architecture (MLPs) are biased towards learning low frequency content[1,2]

$$h(\mathbf{x}) = \sum_{l=1}^{L_p} \frac{\alpha_l e^{j\beta_l}}{\|\mathbf{x} - \mathbf{x}_l\|_2} e^{-j\frac{2\pi}{\lambda}\|\mathbf{x}-\mathbf{x}_l\|_2} \quad (4)$$

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

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

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

- **Derive a model-based architecture for the location-to-channel mapping learning**

- **Derive a model-based architecture for the location-to-channel mapping learning**
  - Where the model does not have to learn high frequency spatial content

- **Derive a model-based architecture for the location-to-channel mapping learning**
  - Where the model does not have to learn high frequency spatial content
- Show that this model-based approach overcomes the spectral bias, and successfully learns the location-to-channel mapping

- The mapping is hard to learn due to the high frequency spatial content

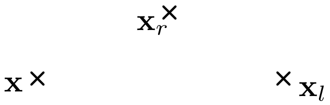
- The mapping is hard to learn due to the high frequency spatial content
- Idea: **split high frequency from low frequency spatial content with a Taylor expansion**

- The mapping is hard to learn due to the high frequency spatial content
- Idea: **split high frequency from low frequency spatial content with a Taylor expansion**
- Around a reference point $\mathbf{x}_r \in \mathbb{R}^2$:

$$\|\mathbf{x} - \mathbf{x}_l\|_2 \simeq \|\mathbf{x}_r - \mathbf{x}_l\|_2 + \mathbf{u}_{(\mathbf{x}_r - \mathbf{x}_l)} \cdot (\mathbf{x} - \mathbf{x}_r) \quad (5)$$

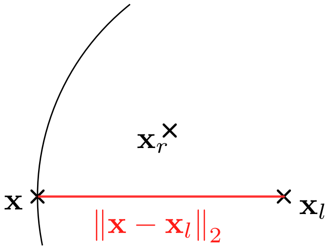
- The mapping is hard to learn due to the high frequency spatial content
- Idea: **split high frequency from low frequency spatial content with a Taylor expansion**
- Around a reference point $\mathbf{x}_r \in \mathbb{R}^2$:

$$\|\mathbf{x} - \mathbf{x}_l\|_2 \simeq \|\mathbf{x}_r - \mathbf{x}_l\|_2 + \mathbf{u}_{(\mathbf{x}_r - \mathbf{x}_l)} \cdot (\mathbf{x} - \mathbf{x}_r) \quad (5)$$



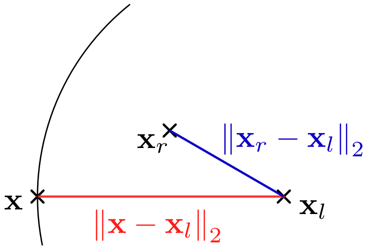
- The mapping is hard to learn due to the high frequency spatial content
- Idea: **split high frequency from low frequency spatial content with a Taylor expansion**
- Around a reference point $\mathbf{x}_r \in \mathbb{R}^2$:

$$\|\mathbf{x} - \mathbf{x}_l\|_2 \simeq \|\mathbf{x}_r - \mathbf{x}_l\|_2 + \mathbf{u}_{(\mathbf{x}_r - \mathbf{x}_l)} \cdot (\mathbf{x} - \mathbf{x}_r) \quad (5)$$



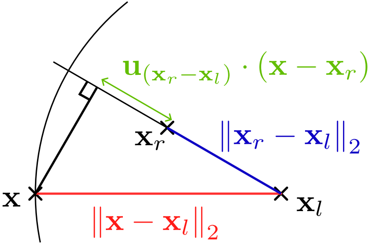
- The mapping is hard to learn due to the high frequency spatial content
- Idea: **split high frequency from low frequency spatial content with a Taylor expansion**
- Around a reference point $\mathbf{x}_r \in \mathbb{R}^2$:

$$\|\mathbf{x} - \mathbf{x}_l\|_2 \simeq \|\mathbf{x}_r - \mathbf{x}_l\|_2 + \mathbf{u}_{(\mathbf{x}_r - \mathbf{x}_l)} \cdot (\mathbf{x} - \mathbf{x}_r) \tag{5}$$



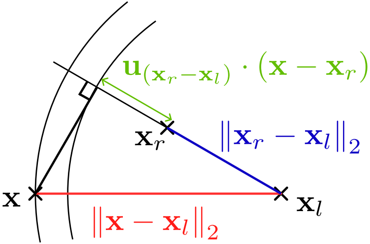
- The mapping is hard to learn due to the high frequency spatial content
- Idea: **split high frequency from low frequency spatial content with a Taylor expansion**
- Around a reference point $\mathbf{x}_r \in \mathbb{R}^2$:

$$\|\mathbf{x} - \mathbf{x}_l\|_2 \simeq \|\mathbf{x}_r - \mathbf{x}_l\|_2 + \mathbf{u}_{(\mathbf{x}_r - \mathbf{x}_l)} \cdot (\mathbf{x} - \mathbf{x}_r) \tag{5}$$



- The mapping is hard to learn due to the high frequency spatial content
- Idea: **split high frequency from low frequency spatial content with a Taylor expansion**
- Around a reference point $\mathbf{x}_r \in \mathbb{R}^2$:

$$\|\mathbf{x} - \mathbf{x}_l\|_2 \simeq \|\mathbf{x}_r - \mathbf{x}_l\|_2 + \mathbf{u}_{(\mathbf{x}_r - \mathbf{x}_l)} \cdot (\mathbf{x} - \mathbf{x}_r) \quad (5)$$

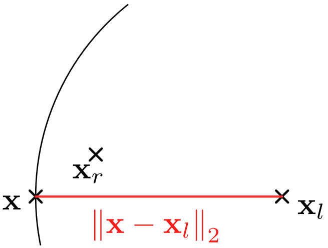


- The mapping is hard to learn due to the high frequency spatial content
- Idea: **split high frequency from low frequency spatial content with a Taylor expansion**
- Around a reference point $\mathbf{x}_r \in \mathbb{R}^2$:

$$\|\mathbf{x} - \mathbf{x}_l\|_2 \simeq \|\mathbf{x}_r - \mathbf{x}_l\|_2 + \mathbf{u}_{(\mathbf{x}_r - \mathbf{x}_l)} \cdot (\mathbf{x} - \mathbf{x}_r) \tag{5}$$

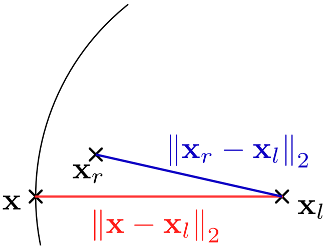
- The mapping is hard to learn due to the high frequency spatial content
- Idea: **split high frequency from low frequency spatial content with a Taylor expansion**
- Around a reference point $\mathbf{x}_r \in \mathbb{R}^2$:

$$\|\mathbf{x} - \mathbf{x}_l\|_2 \simeq \|\mathbf{x}_r - \mathbf{x}_l\|_2 + \mathbf{u}_{(\mathbf{x}_r - \mathbf{x}_l)} \cdot (\mathbf{x} - \mathbf{x}_r) \quad (5)$$



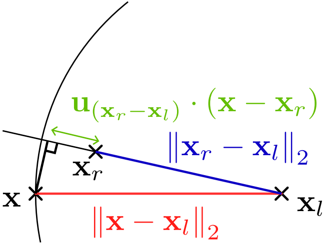
- The mapping is hard to learn due to the high frequency spatial content
- Idea: **split high frequency from low frequency spatial content with a Taylor expansion**
- Around a reference point $\mathbf{x}_r \in \mathbb{R}^2$:

$$\|\mathbf{x} - \mathbf{x}_l\|_2 \simeq \|\mathbf{x}_r - \mathbf{x}_l\|_2 + \mathbf{u}_{(\mathbf{x}_r - \mathbf{x}_l)} \cdot (\mathbf{x} - \mathbf{x}_r) \quad (5)$$



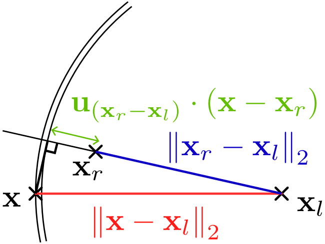
- The mapping is hard to learn due to the high frequency spatial content
- Idea: **split high frequency from low frequency spatial content with a Taylor expansion**
- Around a reference point $\mathbf{x}_r \in \mathbb{R}^2$:

$$\|\mathbf{x} - \mathbf{x}_l\|_2 \simeq \|\mathbf{x}_r - \mathbf{x}_l\|_2 + \mathbf{u}_{(\mathbf{x}_r - \mathbf{x}_l)} \cdot (\mathbf{x} - \mathbf{x}_r) \quad (5)$$



- The mapping is hard to learn due to the high frequency spatial content
- Idea: **split high frequency from low frequency spatial content with a Taylor expansion**
- Around a reference point $\mathbf{x}_r \in \mathbb{R}^2$:

$$\|\mathbf{x} - \mathbf{x}_l\|_2 \simeq \|\mathbf{x}_r - \mathbf{x}_l\|_2 + \mathbf{u}_{(\mathbf{x}_r - \mathbf{x}_l)} \cdot (\mathbf{x} - \mathbf{x}_r) \quad (5)$$



- The mapping is hard to learn due to the high frequency spatial content
- Idea: **split high frequency from low frequency spatial content with a Taylor expansion**
- Around a reference point $\mathbf{x}_r \in \mathbb{R}^2$:

$$\|\mathbf{x} - \mathbf{x}_l\|_2 \simeq \|\mathbf{x}_r - \mathbf{x}_l\|_2 + \mathbf{u}_{(\mathbf{x}_r - \mathbf{x}_l)} \cdot (\mathbf{x} - \mathbf{x}_r) \tag{5}$$

- This yields:

$$h(\mathbf{x}) \simeq \sum_{l=1}^{L_p} \underbrace{\frac{\alpha_l e^{j\beta_l} h_l(\mathbf{x}_r)\, e^{j\frac{2\pi}{\lambda}\mathbf{u}_{(\mathbf{x}_r-\mathbf{x}_l)}\cdot\mathbf{x}_r}}{1 + \frac{\mathbf{u}_{(\mathbf{x}_r-\mathbf{x}_l)}\cdot(\mathbf{x}-\mathbf{x}_r)}{\|\mathbf{x}_r-\mathbf{x}_l\|_2}}}_{\text{Slowly varying}} \underbrace{e^{-j\frac{2\pi}{\lambda}\mathbf{u}_{(\mathbf{x}_r-\mathbf{x}_l)}\cdot\mathbf{x}}}_{\text{Fastly varying}} \tag{6}$$

- The mapping is hard to learn due to the high frequency spatial content
- Idea: **split high frequency from low frequency spatial content with a Taylor expansion**
- Around a reference point $\mathbf{x}_r \in \mathbb{R}^2$:

$$\|\mathbf{x} - \mathbf{x}_l\|_2 \simeq \|\mathbf{x}_r - \mathbf{x}_l\|_2 + \mathbf{u}_{(\mathbf{x}_r - \mathbf{x}_l)} \cdot (\mathbf{x} - \mathbf{x}_r) \quad (5)$$

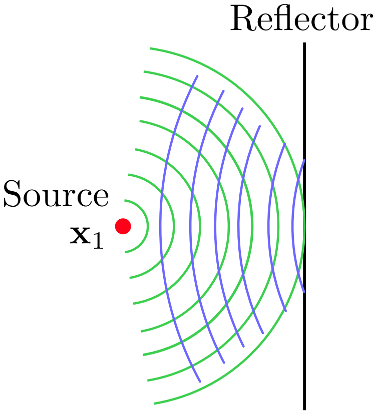
- This yields:

$$h(\mathbf{x}) \simeq \sum_{l=1}^{L_p} \underbrace{\frac{\alpha_l e^{j\beta_l} h_l(\mathbf{x}_r)\, e^{j\frac{2\pi}{\lambda}\mathbf{u}_{(\mathbf{x}_r-\mathbf{x}_l)}\cdot\mathbf{x}_r}}{1 + \frac{\mathbf{u}_{(\mathbf{x}_r-\mathbf{x}_l)}\cdot(\mathbf{x}-\mathbf{x}_r)}{\|\mathbf{x}_r-\mathbf{x}_l\|_2}}}_{\text{Slowly varying}} \underbrace{e^{-j\frac{2\pi}{\lambda}\mathbf{u}_{(\mathbf{x}_r-\mathbf{x}_l)}\cdot\mathbf{x}}}_{\text{Fastly varying}} \quad (6)$$

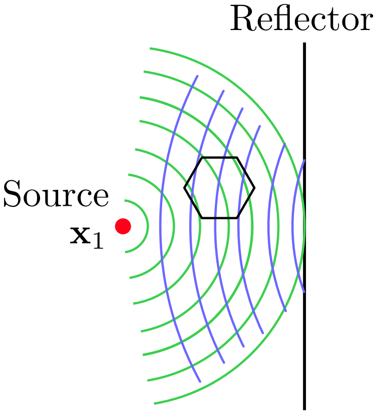
$h(\mathrm{x})$ is locally approximated as a linear combination of planar wavefronts

- Taylor expansion only valid locally

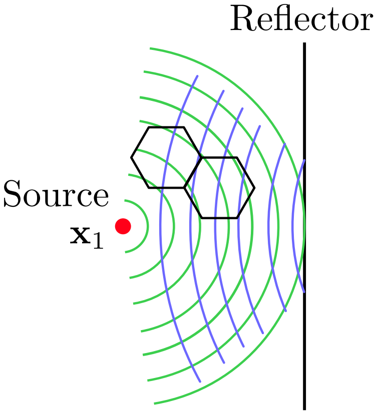
- Taylor expansion only valid locally



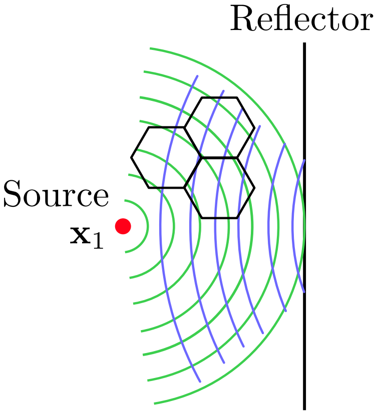
- Taylor expansion only valid locally



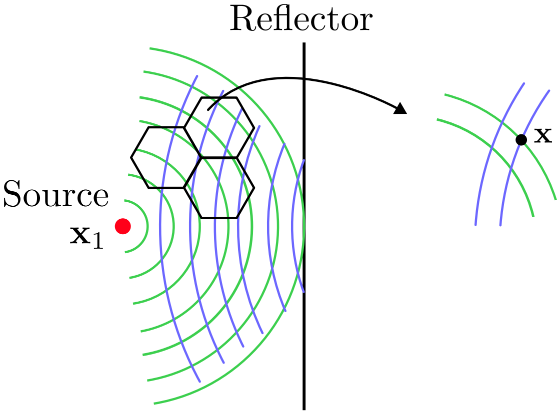
- Taylor expansion only valid locally



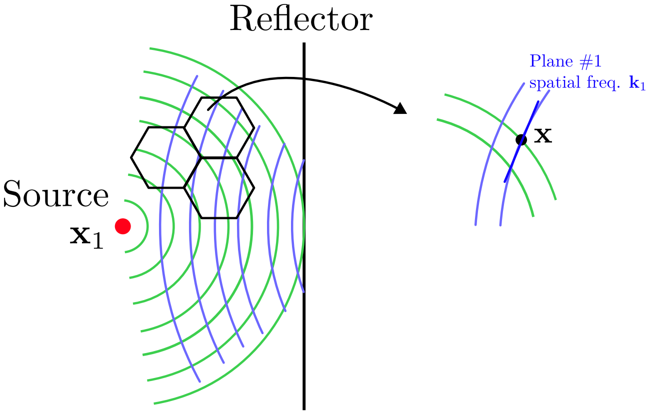
- Taylor expansion only valid locally



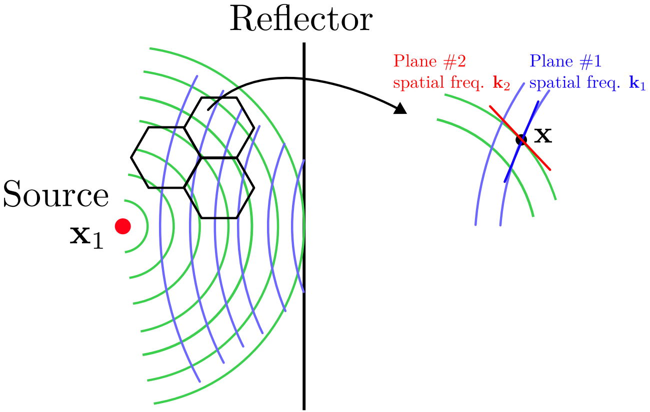
- Taylor expansion only valid locally



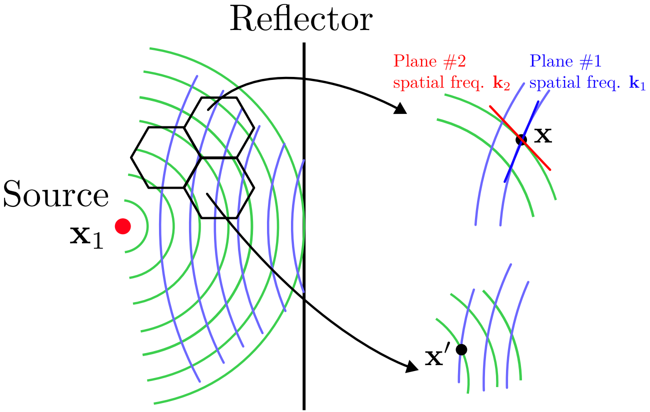
- Taylor expansion only valid locally



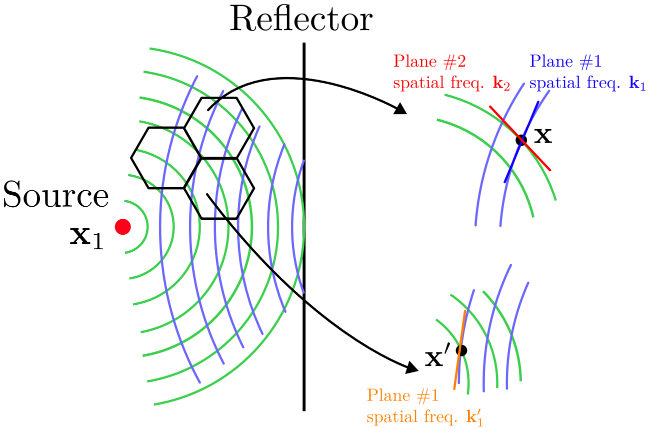
- Taylor expansion only valid locally



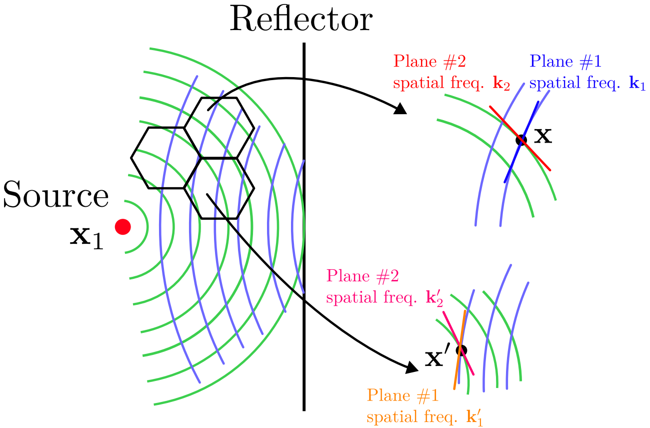
- Taylor expansion only valid locally



- Taylor expansion only valid locally



- Taylor expansion only valid locally



- One needs a set of spatial frequencies per hexagon:

< From local to global spatial validity >

- One needs a set of spatial frequencies per hexagon:
  - $\mathbf{\Psi}(\mathbf{x}) = \{\psi_i(\mathbf{x})\}_{i=1}^{D} = \left\{\mathrm{e}^{-\mathrm{j}\mathbf{k}_i\cdot\mathbf{x}}\right\}_{i=1}^{D}$: dictionary containing well-chosen planar wavefronts

- One needs a set of spatial frequencies per hexagon:
  - $\boldsymbol{\Psi}(\mathbf{x}) = \{\psi_i(\mathbf{x})\}_{i=1}^{D} = \left\{\mathrm{e}^{-\mathrm{j}\mathbf{k}_i \cdot \mathbf{x}}\right\}_{i=1}^{D}$: dictionary containing well-chosen planar wavefronts
    - Can be constructed by sampling the unit circle with D spatial frequencies

- One needs a set of spatial frequencies per hexagon:
  - $\boldsymbol{\Psi}(\mathbf{x}) = \{\psi_i(\mathbf{x})\}_{i=1}^{D} = \{\mathrm{e}^{-\mathrm{j}\mathbf{k}_i \cdot \mathbf{x}}\}_{i=1}^{D}$: dictionary containing well-chosen planar wavefronts
    - Can be constructed by sampling the unit circle with D spatial frequencies
  - $\mathbf{w}(\mathbf{x}) \in \mathbb{C}^D$: location-dependent activation vector

- One needs a set of spatial frequencies per hexagon:
  - $\boldsymbol{\Psi}(\mathbf{x}) = \{\psi_i(\mathbf{x})\}_{i=1}^{D} = \left\{\mathrm{e}^{-\mathrm{j}\mathbf{k}_i \cdot \mathbf{x}}\right\}_{i=1}^{D}$: dictionary containing well-chosen planar wavefronts
    - Can be constructed by sampling the unit circle with D spatial frequencies
  - $\mathbf{w}(\mathbf{x}) \in \mathbb{C}^D$: location-dependent activation vector

$$\forall \mathbf{x} \in \mathbb{R}^2,\ h(\mathbf{x}) \simeq \sum_{i=1}^{D} w_i(\mathbf{x})\, \psi_i(\mathbf{x})\,, \tag{7}$$

$$\text{with } \|\mathbf{w}(\mathbf{x})\|_0 = L_p$$

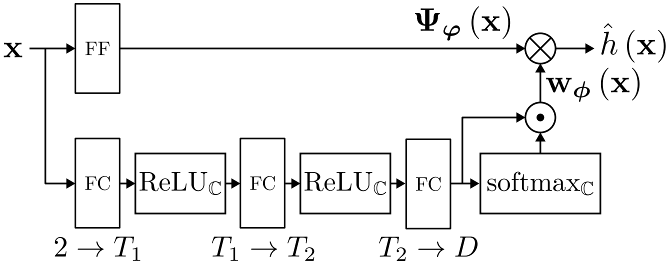
- One needs a set of spatial frequencies per hexagon:
  - $\mathbf{\Psi}(\mathbf{x}) = \{\psi_i(\mathbf{x})\}_{i=1}^{D} = \{\mathrm{e}^{-\mathrm{j}\mathbf{k}_i \cdot \mathbf{x}}\}_{i=1}^{D}$: dictionary containing well-chosen planar wavefronts
    - Can be constructed by sampling the unit circle with D spatial frequencies
  - $\mathbf{w}(\mathbf{x}) \in \mathbb{C}^D$: location-dependent activation vector

$$\forall \mathbf{x} \in \mathbb{R}^2,\ h(\mathbf{x}) \simeq \sum_{i=1}^{D} w_i(\mathbf{x})\, \psi_i(\mathbf{x}), \qquad (7)$$

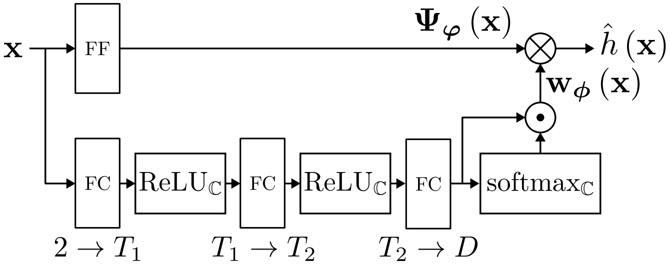
$$\text{with } \|\mathbf{w}(\mathbf{x})\|_0 = L_p$$

The local planar approximation becomes global with a well-chosen dictionary

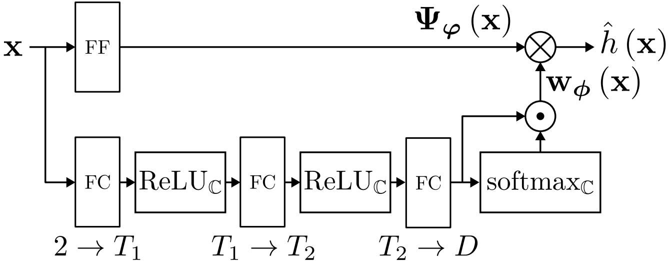
- Main idea: for a given input location $\mathbf{x} \in \mathbb{R}^2$



- Main idea: for a given input location $\mathbf{x} \in \mathbb{R}^2$
  - From fixed spatial frequencies $\{\mathbf{k}_i\}_{i=1}^{D}$ compute Fourier features $\left\{\mathrm{e}^{-\mathrm{j}\mathbf{k}_i \cdot \mathbf{x}}\right\}_{i=1}^{D}$



- Main idea: for a given input location $\mathbf{x} \in \mathbb{R}^2$
  - From fixed spatial frequencies $\{\mathbf{k}_i\}_{i=1}^{D}$ compute Fourier features $\{e^{-j\mathbf{k}_i \cdot \mathbf{x}}\}_{i=1}^{D}$
  - Compute the associated complex weights $\mathbf{w}(\mathbf{x})$, with the sparsity constraint



- Channel generation:

- Channel generation:
  - $f_0 = 3.5$GHz

- Channel generation:
  - $f_0 = 3.5$GHz
  - Synthetic, with hand-placed virtual sources

- Channel generation:
  - $f_0 = 3.5$GHz
  - Synthetic, with hand-placed virtual sources
  - Ray-tracing (Sionna) in Paris

- Channel generation:
  - $f_0 = 3.5$GHz
  - Synthetic, with hand-placed virtual sources
  - Ray-tracing (Sionna) in Paris
- Locations generation:

- Channel generation:
  - $f_0 = 3.5$GHz
  - Synthetic, with hand-placed virtual sources
  - Ray-tracing (Sionna) in Paris
- Locations generation:
  - 10m by 10m square scene

- Channel generation:
  - $f_0 = 3.5$GHz
  - Synthetic, with hand-placed virtual sources
  - Ray-tracing (Sionna) in Paris
- Locations generation:
  - 10m by 10m square scene
  - Train/test locations randomly dropped in the scene with a certain spatial density

- Channel generation:
  - $f_0 = 3.5$GHz
  - Synthetic, with hand-placed virtual sources
  - Ray-tracing (Sionna) in Paris
- Locations generation:
  - 10m by 10m square scene
  - Train/test locations randomly dropped in the scene with a certain spatial density
  - Evaluation locations: $\lambda/4$ uniform grid

- Channel generation:
  - $f_0 = 3.5$GHz
  - Synthetic, with hand-placed virtual sources
  - Ray-tracing (Sionna) in Paris
- Locations generation:
  - 10m by 10m square scene
  - Train/test locations randomly dropped in the scene with a certain spatial density
  - Evaluation locations: $\lambda/4$ uniform grid
- Train loss:

$$\mathcal{L} = \mathbb{E}\left[\|f_{\boldsymbol{\theta}}(\mathbf{x}) - h(\mathbf{x})\|_2^2\right], \mathbf{x} \in \mathcal{D} \subset \mathbb{R}^2, \tag{8}$$

with $\mathcal{D}$: batch locations set

- Channel generation:
  - $f_0 = 3.5$GHz
  - Synthetic, with hand-placed virtual sources
  - Ray-tracing (Sionna) in Paris
- Locations generation:
  - 10m by 10m square scene
  - Train/test locations randomly dropped in the scene with a certain spatial density
  - Evaluation locations: $\lambda/4$ uniform grid
- Train loss:

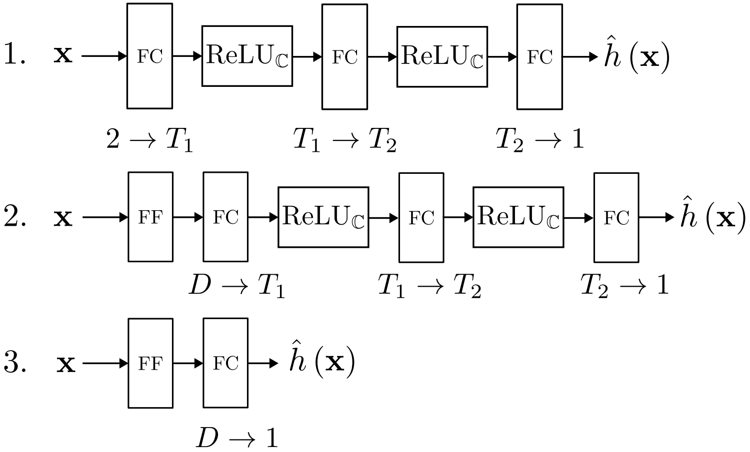
$$\mathcal{L} = \mathbb{E}\left[\left\|f_{\boldsymbol{\theta}}\left(\mathbf{x}\right) - h\left(\mathbf{x}\right)\right\|_2^2\right], \mathbf{x} \in \mathcal{D} \subset \mathbb{R}^2, \tag{8}$$

with $\mathcal{D}$: batch locations set

- Evaluation metric:

$$\text{NMSE} = 10\log_{10}\left(\frac{\left\|h\left(\mathbf{x}\right) - f_{\boldsymbol{\theta}}\left(\mathbf{x}\right)\right\|_2^2}{\left\|h\left(\mathbf{x}\right)\right\|_2^2}\right), \mathbf{x} \in \mathcal{E} \subset \mathbb{R}^2 \tag{9}$$

with $\mathcal{E}$: evaluation locations set



- 1. MLP, 2. RFF, 3. RFF lin.

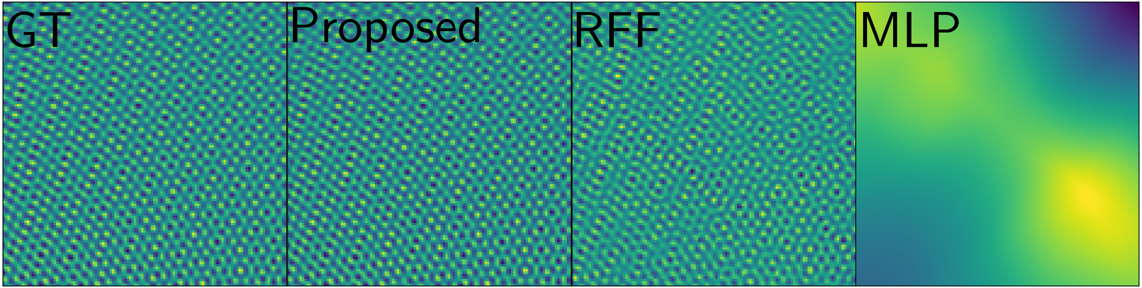
< Results: scene reconstruction >

- Synthetic channels, $L_p = 6$ propagation paths
- Train loc. density: 100locs./m$^2 \simeq 0.7$ locs./λ^2

- Synthetic channels, $L_p = 6$ propagation paths
- Train loc. density: 100locs./m$^2 \simeq 0.7$ locs./λ^2

	MLP	RFF	RFF lin.	Proposed
Params.	16.8M	33.1M	4k	0.5M
NMSE$_{\text{(dB)}}$	0.16	-3.30	-3.04	$\mathbf{-20.60}$

- Synthetic channels, $L_p = 6$ propagation paths
- Train loc. density: 100locs./m$^2 \simeq 0.7$ locs./λ^2



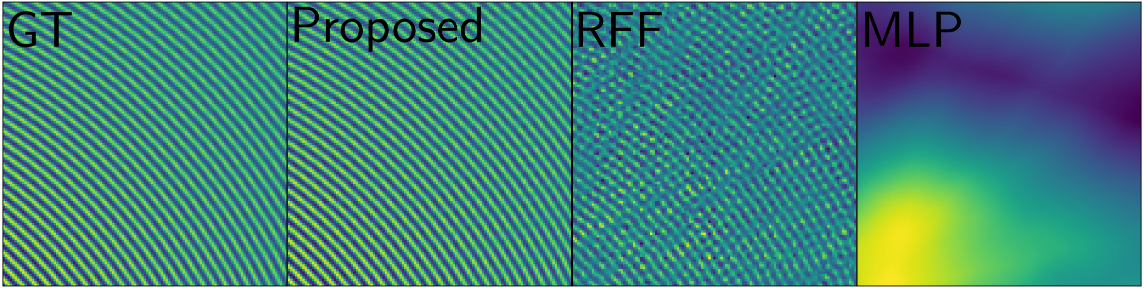
- Small zone (2.5m by 2.5m)

- Ray-tracing channels, $L_p = 11$ propagation paths
- Train loc. density: 150locs./m$^2 \simeq 1.1$ locs./λ^2

- Ray-tracing channels, $L_p = 11$ propagation paths
- Train loc. density: 150locs./m^2 $\simeq 1.1$ locs./λ^2

	MLP	RFF	RFF lin.	Proposed
Params.	16.8M	33.1M	4k	0.5M
NMSE$_{\text{(dB)}}$	0.14	-2.41	-2.21	$\mathbf{-23.41}$

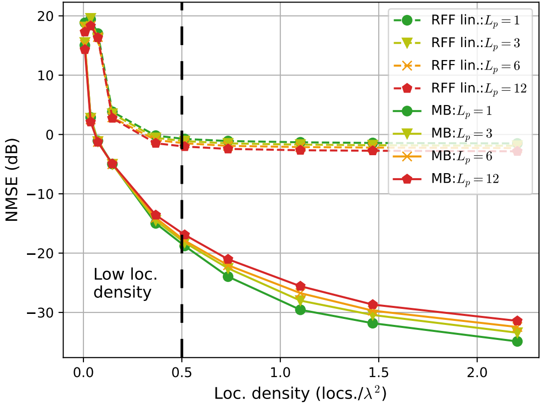
- Ray-tracing channels, $L_p = 11$ propagation paths
- Train loc. density: 150locs./m$^2 \simeq 1.1$ locs./λ^2



- Small zone (2.5m by 2.5m)

- Synthetic channels, variable training loc. density, variable propagation path number

- Synthetic channels, variable training loc. density, variable propagation path number
- For each point: 100 training with random virtual sources



- Contributions:

- Contributions:
  - Derive a model-based neural network to learn the location-to-channel mapping

- Contributions:
  - Derive a model-based neural network to learn the location-to-channel mapping
  - Show that the proposed model-based architecture allows to overcome the spectral bias

- Contributions:
  - Derive a model-based neural network to learn the location-to-channel mapping
  - Show that the proposed model-based architecture allows to overcome the spectral bias
  - Better performance than baselines, with less training parameters

- Contributions:
  - Derive a model-based neural network to learn the location-to-channel mapping
  - Show that the proposed model-based architecture allows to overcome the spectral bias
  - Better performance than baselines, with less training parameters
- Future work:

- Contributions:
  - Derive a model-based neural network to learn the location-to-channel mapping
  - Show that the proposed model-based architecture allows to overcome the spectral bias
  - Better performance than baselines, with less training parameters

- Future work:
  - Adapt the architecture to a more realistic scenario: multi-antenna/multicarrier

- Contributions:
  - Derive a model-based neural network to learn the location-to-channel mapping
  - Show that the proposed model-based architecture allows to overcome the spectral bias
  - Better performance than baselines, with less training parameters
- Future work:
  - Adapt the architecture to a more realistic scenario: multi-antenna/multicarrier
- Link to paper: https://arxiv.org/pdf/2308.14370.pdf

Thanks