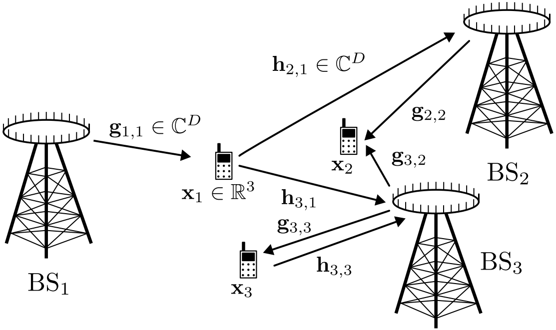
/ Model-based Deep Learning for Beam Prediction based on a Channel Chart /

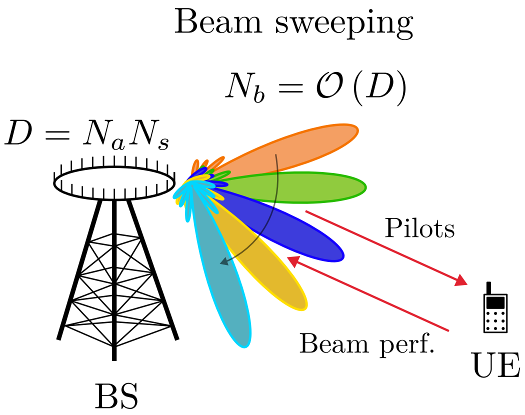
Taha YASSINE[⋆,†], **Baptiste CHATELIER**[‡,†,⋆], Vincent CORLAY[‡,⋆], Matthieu CRUSSIERE[†,⋆], Stéphane PAQUELET[⋆], Olav TIRKKONEN[§], Luc LE MAGOAROU[†,⋆]

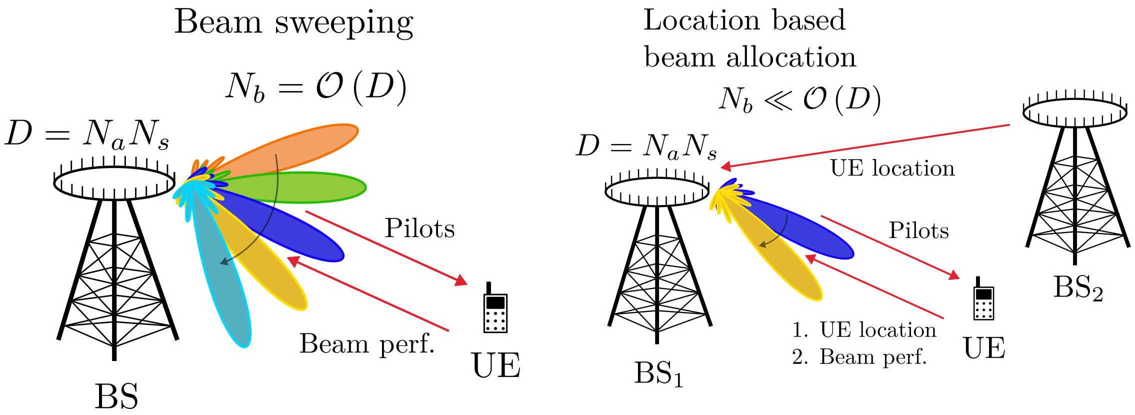
† Univ Rennes, INSA Rennes, CNRS, IETR-UMR 6164, Rennes, France
‡ Mitsubishi Electric R&D Centre Europe, Rennes, France
§ Department of Information and Communications Engineering, Aalto University, Finland
⋆ b<>com, Rennes, France

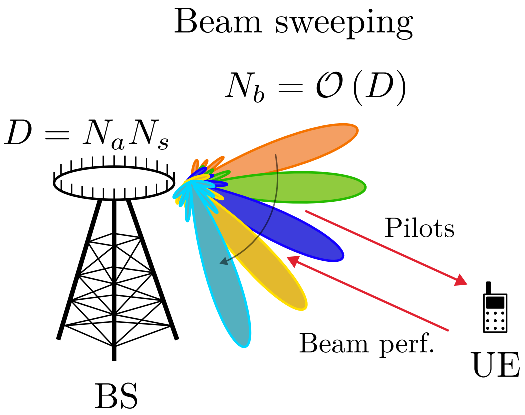
baptiste.chatelier@insa-rennes.fr

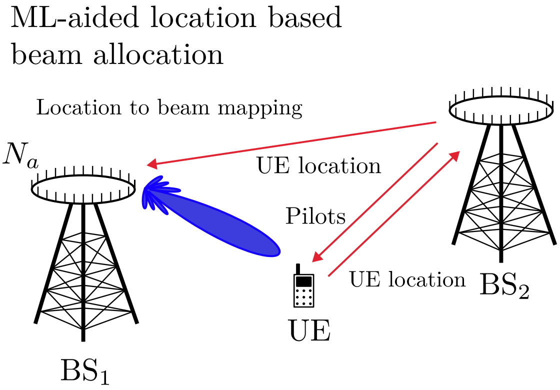
- In Cell Free Massive MIMO communication systems, with different uplink and downlink frequencies, how to attribute the best BS beam to a given UE ?



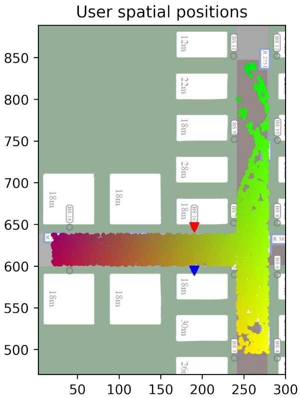

Beam sweeping
$N_b = \mathcal{O}(D)$
$D = N_a N_s$
Pilots
Beam perf.
UE
BS

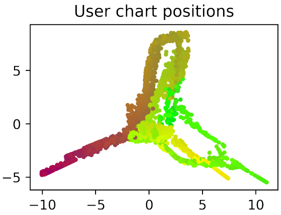

Beam sweeping
$N_b = \mathcal{O}(D)$
$D = N_a N_s$
Pilots
Beam perf.
UE
BS
Location based
beam allocation
$N_b \ll \mathcal{O}(D)$
$D = N_a N_s$
UE location
Pilots
1. UE location
2. Beam perf.
UE
BS_1
BS_2

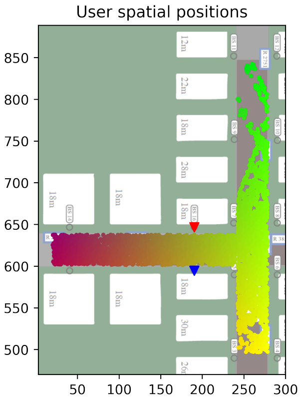

Beam sweeping
$N_b = \mathcal{O}(D)$
$D = N_a N_s$
Pilots
Beam perf.
UE
BS

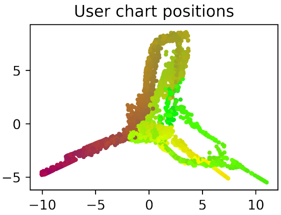

ML-aided location based beam allocation
Location to beam mapping
N_a
UE location
Pilots
UE location
BS_2
UE
BS_1

What happens if the location becomes a chart location ?

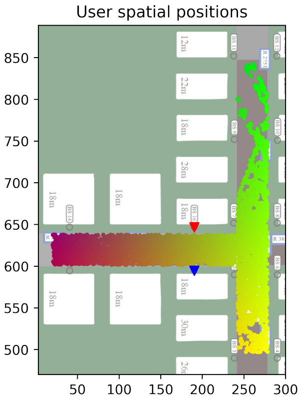

User spatial positions
850
800
750
700
650
600
550
500
50
100
150
200
250
300

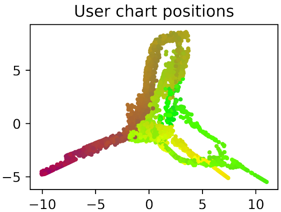

User chart positions
5
0
−5
−10
−5
0
5
10





- Locations: GNSS



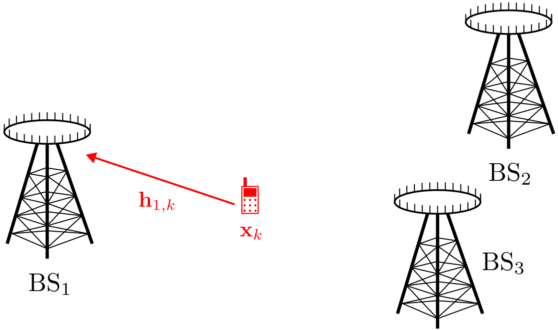


- Locations: GNSS
- Chart/Pseudo-locations: dim. reduction of the channel

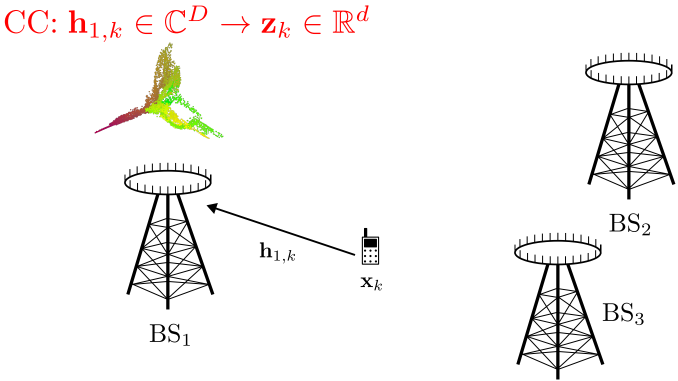
Contributions

- **New neural architecture for the pseudo-location to beam mapping**

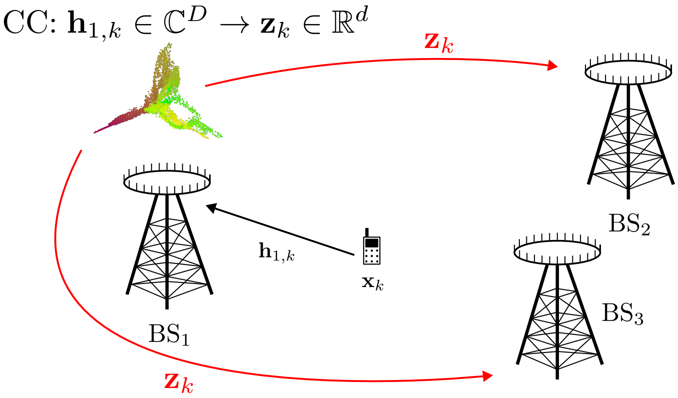
- **New neural architecture for the pseudo-location to beam mapping**
- **Assessment of codebook performance versus precoder learning in cell-free systems**



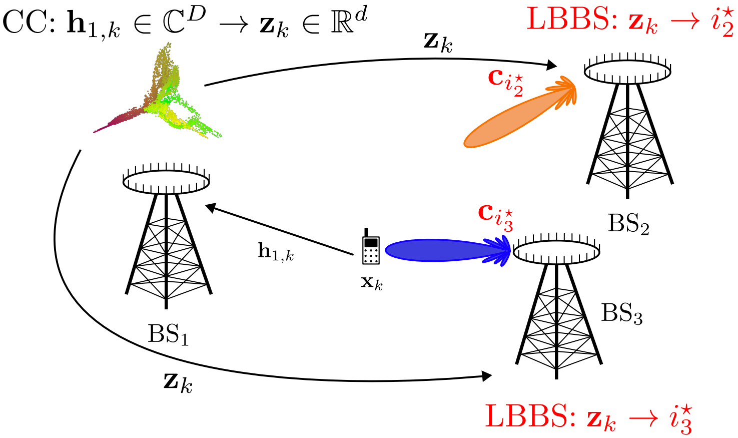
- Inference: get uplink channels at BS1



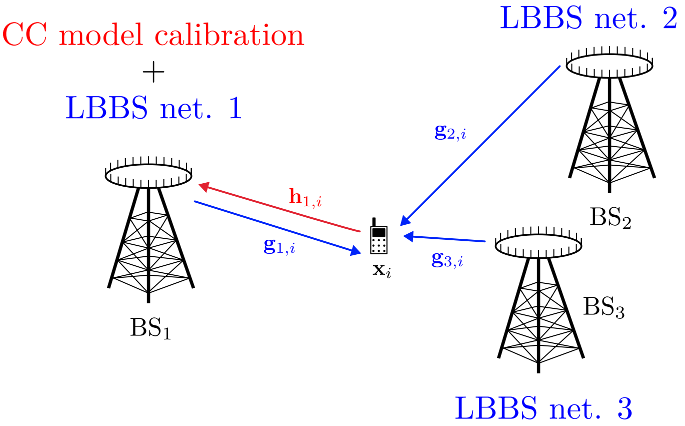
- Inference: channel charting



- Inference: send pseudo-locations to other BSs

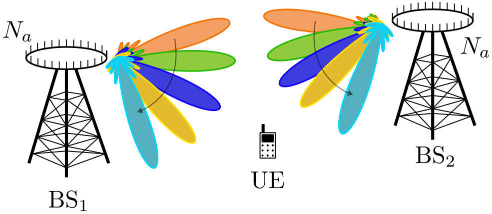


- Inference: beam selection from pseudo-locations

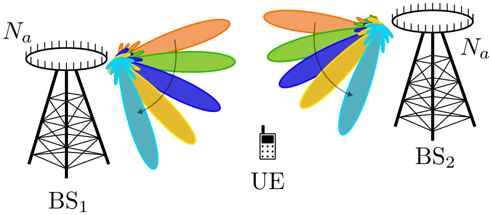


- Training: CC only at BS1, LBBS networks at all BSs

- Classical approach:



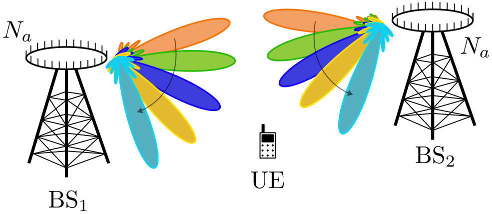
- Classical approach:



  - All BSs perform beam sweeping: $\mathcal{O}(BD)$

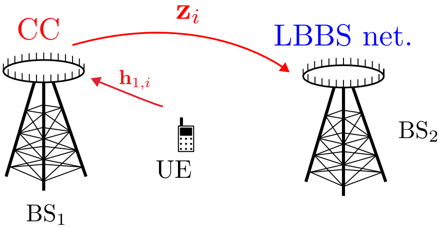
< Beam allocation complexity >

- Classical approach:



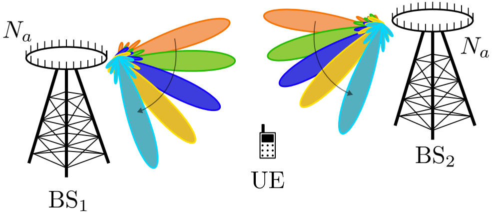
  - All BSs perform beam sweeping: $\mathcal{O}(BD)$

- CC-based approach:



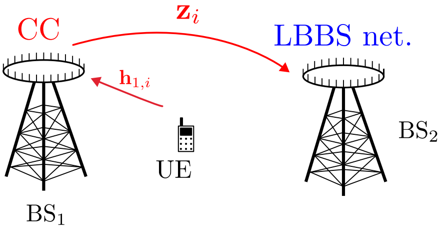
< Beam allocation complexity >

- Classical approach:



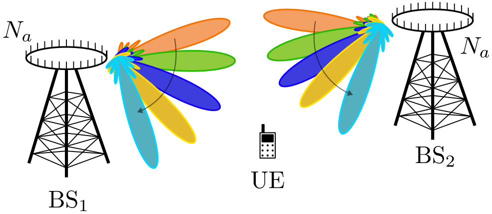
  - All BSs perform beam sweeping: $\mathcal{O}\left(BD\right)$

- CC-based approach:



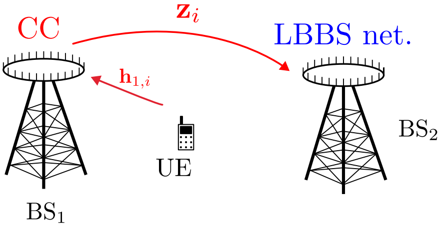
  - One BS performs channel estimation and channel charting

- Classical approach:



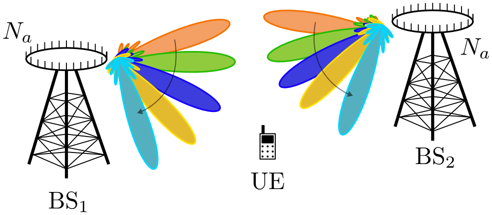
  - All BSs perform beam sweeping: $\mathcal{O}(BD)$

- CC-based approach:



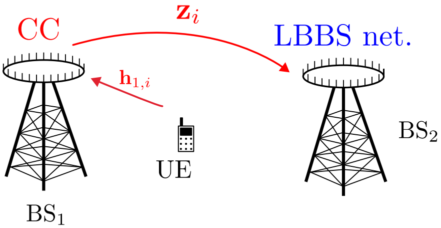
  - One BS performs channel estimation and channel charting
  - Then it sends the pseudo-loc. to other BSs

- Classical approach:



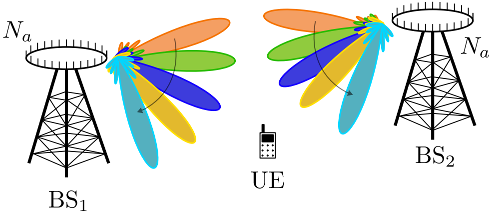
  - All BSs perform beam sweeping: $\mathcal{O}\left(BD\right)$

- CC-based approach:



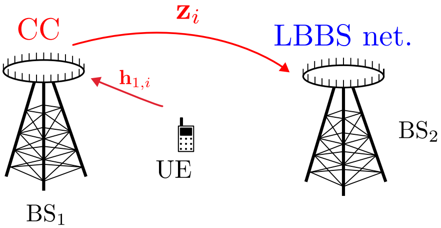
  - One BS performs channel estimation and channel charting
  - Then it sends the pseudo-loc. to other BSs
  - Total complexity: $\mathcal{O}\left(D + Bd\right), d \ll D$

- Classical approach:



  - All BSs perform beam sweeping: $\mathcal{O}\left(BD\right)$

- CC-based approach:



  - One BS performs channel estimation and channel charting
  - Then it sends the pseudo-loc. to other BSs
  - Total complexity: $\mathcal{O}\left(D + Bd\right), d \ll D$

Huge complexity reduction

- Manifold learning methods such as `ISOMAP` are computationally intensive in out-of-sample scenarios

- Manifold learning methods such as ISOMAP are computationally intensive in out-of-sample scenarios → use of the method proposed in[1] with a phase insensitive distance[2].

[1] Yassine et al., "Leveraging triplet loss and nonlinear dimensionality reduction for on-the-fly channel charting".
[2] Le Magoarou, "Efficient Channel Charting via Phase-Insensitive Distance Computation".

- Manifold learning methods such as ISOMAP are computationally intensive in out-of-sample scenarios → use of the method proposed in[1] with a phase insensitive distance[2].

$$\{\mathbf{h}_{1,n}\}_{n=1}^{N} \xrightarrow{\texttt{ISOMAP}} \{\mathbf{z}_{1,n}\}_{n=1}^{N}$$

$$\left(\mathbf{D} \triangleq (\mathbf{h}_{1,1} \cdots \mathbf{h}_{1,N}) \in \mathbb{C}^{N_a \times N}\right) \qquad \left(\mathbf{Z} \triangleq (\mathbf{z}_{1,1} \cdots \mathbf{z}_{1,N}) \in \mathbb{C}^{d \times N}\right)$$

[1] Yassine et al., “Leveraging triplet loss and nonlinear dimensionality reduction for on-the-fly channel charting”.
[2] Le Magoarou, “Efficient Channel Charting via Phase-Insensitive Distance Computation”.

- Manifold learning methods such as `ISOMAP` are computationally intensive in out-of-sample scenarios $\rightarrow$ use of the method proposed in[1] with a phase insensitive distance[2].

$$\begin{array}{ccc} \{\mathbf{h}_{1,n}\}_{n=1}^{N} & \xrightarrow{\quad\texttt{ISOMAP}\quad} & \{\mathbf{z}_{1,n}\}_{n=1}^{N} \\ \left(\mathbf{D} \triangleq (\mathbf{h}_{1,1} \cdots \mathbf{h}_{1,N}) \in \mathbb{C}^{N_a \times N}\right) & & \left(\mathbf{Z} \triangleq (\mathbf{z}_{1,1} \cdots \mathbf{z}_{1,N}) \in \mathbb{C}^{d \times N}\right) \end{array}$$

Out-of-sample channel: $\mathbf{h}_{1,j}$

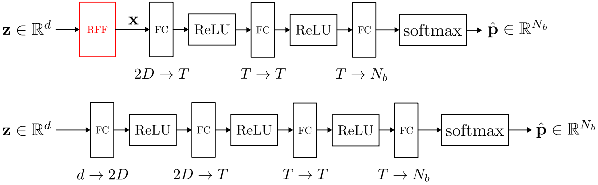
$$\mathbf{h}_{1,j} \rightarrow \boxed{\mathbf{D}^{\mathrm{H}}} \rightarrow \boxed{|\cdot|} \rightarrow \boxed{\mathrm{HT}_K} \rightarrow \boxed{\frac{\cdot}{\|\cdot\|_1}} \rightarrow \boxed{\mathbf{Z}} \rightarrow \mathbf{z}_{1,j}$$

- $\mathbf{z}_{1,j}$ can be seen as a convex combination of the pseudo-locations associated to the K most correlated channels with $\mathbf{h}_{1,j}$.

[1] Yassine et al., “Leveraging triplet loss and nonlinear dimensionality reduction for on-the-fly channel charting”.
[2] Le Magoarou, “Efficient Channel Charting via Phase-Insensitive Distance Computation”.

RFF

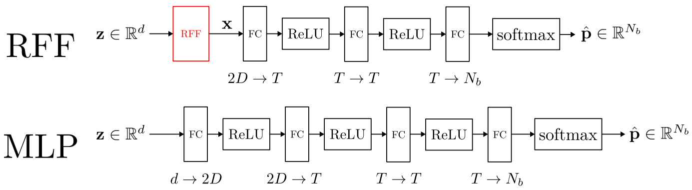
MLP



Training loss: multi-class cross entropy

$$\mathcal{L} = -\sum_{u=1}^{\mathcal{B}} \mathbf{p}_u^{\mathrm{T}} \log_2 \left(\hat{\mathbf{p}}_u\right)$$

$$\mathbf{p}_u \in \mathbb{R}^{N_b},\ \left(\mathbf{p}_u\right)_j = 1 \Leftrightarrow i^\star = j \text{ for UE } u.$$

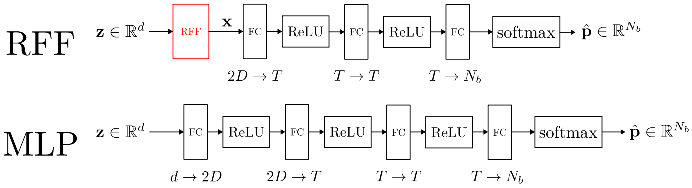


Training loss: multi-class cross entropy

$$\mathcal{L} = -\sum_{u=1}^{\mathcal{B}} \mathbf{p}_u^{\mathrm{T}} \log_2 (\hat{\mathbf{p}}_u)$$

$$\mathbf{p}_u \in \mathbb{R}^{N_b},\ (\mathbf{p}_u)_j = 1 \Leftrightarrow i^\star = j \text{ for UE } u.$$

- $\mathbf{x} = \begin{bmatrix} \cos(2\pi \mathbf{B}\mathbf{z}) \\ \sin(2\pi \mathbf{B}\mathbf{z}) \end{bmatrix}, \mathbf{B} \in \mathbb{R}^{F \times d}$
- $\mathbf{B} \sim \mathcal{N}\left(\mathbf{0}_F, \sigma^2 \mathbf{Id}_F\right)$
- $d = 5,\ F = 200,\ T = 64,\ N_b = 256$



Training loss: multi-class cross entropy

$$\mathcal{L} = -\sum_{u=1}^{\mathcal{B}} \mathbf{p}_u^{\mathrm{T}} \log_2 \left(\hat{\mathbf{p}}_u\right)$$

$$\mathbf{p}_u \in \mathbb{R}^{N_b},\ \left(\mathbf{p}_u\right)_j = 1 \Leftrightarrow i^\star = j \text{ for UE } u.$$

- $\mathbf{x} = \begin{bmatrix} \cos\left(2\pi\mathbf{B}\mathbf{z}\right) \\ \sin\left(2\pi\mathbf{B}\mathbf{z}\right) \end{bmatrix}, \mathbf{B} \in \mathbb{R}^{F\times d}$
- $\mathbf{B} \sim \mathcal{N}\left(\mathbf{0}_F, \sigma^2\mathbf{Id}_F\right)$
- $d = 5,\ F = 200,\ T = 64,\ N_b = 256$

- Baseline: 1-NN $\to$ the best beam for a given test pseudo-loc. is the optimal beam of the closest train pseudo-loc.

- Two different scenes:
  - Urban canyon with *DeepMIMO*[3]
  - Paris, Étoile neighborhood with *Sionna*[4]

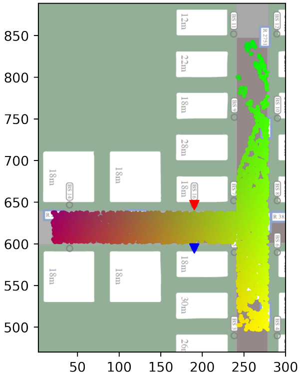
[3] Alkhateeb, “DeepMIMO: A Generic Deep Learning Dataset for Millimeter Wave and Massive MIMO Applications”.
[4] Hoydis et al., “Sionna: An Open-Source Library for Next-Generation Physical Layer Research”.

- Two different scenes:
  - Urban canyon with *DeepMIMO*[3]
  - Paris, Étoile neighborhood with *Sionna*[4]

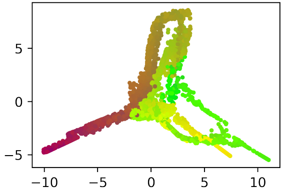
- Radio parameters:
  - 2 BSs: UPA $8\text{x}8 \Rightarrow N_a = 64$
  - 2D-DFT codebook: $N_b = 4N_a$
  - UEs: mono-antenna
  - Uplink: 3.5GHz
  - Downlink: 28GHz
  - Multicarrier: 16 subcarriers over 20MHz bandwidth

[3] Alkhateeb, “DeepMIMO: A Generic Deep Learning Dataset for Millimeter Wave and Massive MIMO Applications”.
[4] Hoydis et al., “Sionna: An Open-Source Library for Next-Generation Physical Layer Research”.

User spatial positions

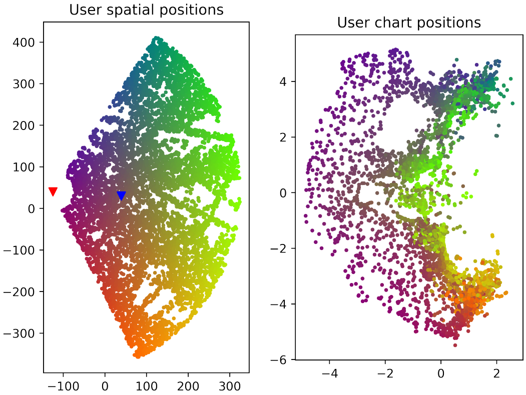


User chart positions



- Neighbours: 5% of dataset size

TW	CT	KS
0.973	0.929	0.471



- Neighbours: 5% of dataset size

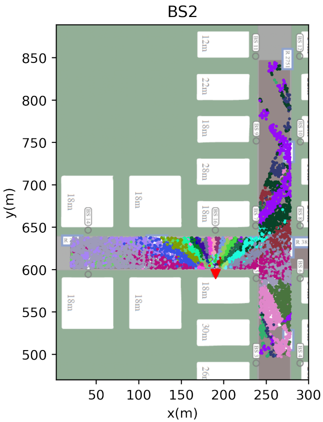
TW	CT	KS
0.960	0.952	0.292

- Chart shape can be explained[5]

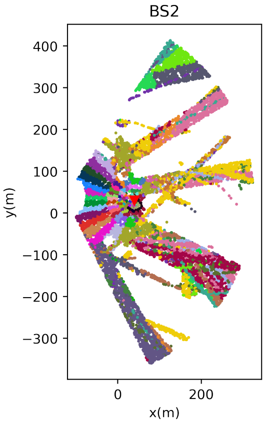
[5] Yassine et al., *Optimizing Multicarrier Multiantenna Systems for LoS Channel Charting*.

< Best beam spatial distribution: BS2 >

- DeepMIMO



- Sionna



- Pseudo-locs.

DeepMIMO	RFF	MLP	1-NN
Top 1 acc. (%)	**66.07**	56.06	61.40
Top 2 acc. (%)	**84.87**	76.97	81.31
Top 3 acc. (%)	**90.66**	85.09	88.77
Sionna	RFF	MLP	1-NN
Top 1 acc. (%)	66.07	54.07	**69.73**
Top 2 acc. (%)	75.13	65.00	**79.47**
Top 3 acc. (%)	78.27	69.07	**81.87**

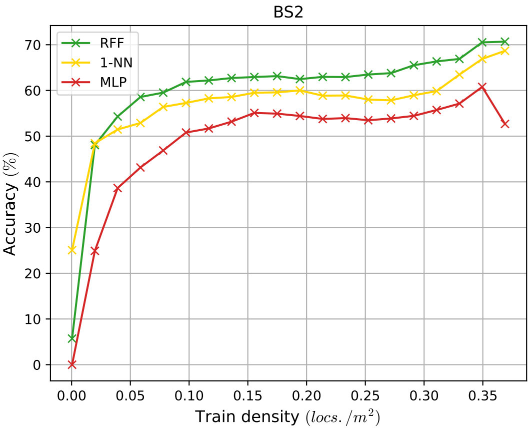
- Pseudo-locs.

DeepMIMO	RFF	MLP	1-NN
Top 1 acc. (%)	**66.07**	56.06	61.40
Top 2 acc. (%)	**84.87**	76.97	81.31
Top 3 acc. (%)	**90.66**	85.09	88.77
Sionna	RFF	MLP	1-NN
Top 1 acc. (%)	66.07	54.07	**69.73**
Top 2 acc. (%)	75.13	65.00	**79.47**
Top 3 acc. (%)	78.27	69.07	**81.87**

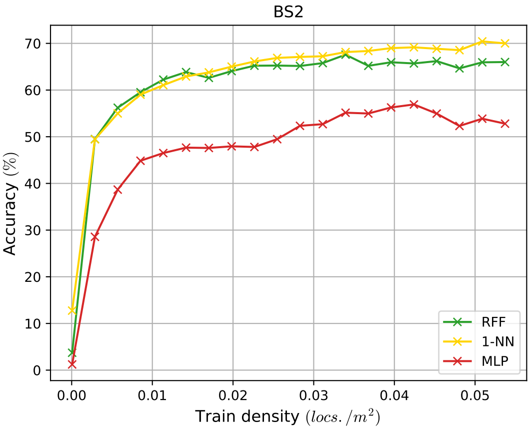
- True locs.

DeepMIMO	RFF	MLP	1-NN
Top 1 acc. (%)	**74.53**	34.15	71.08
Top 2 acc. (%)	**91.21**	46.61	88.32
Top 3 acc. (%)	**95.77**	54.39	94.33
Sionna	RFF	MLP	1-NN
Top 1 acc. (%)	**82.53**	42.40	82.07
Top 2 acc. (%)	**88.40**	49.93	88.27
Top 3 acc. (%)	**89.87**	53.80	**89.87**

- DeepMIMO



- Sionna



- Mean execution time for inference (1.5M pseudo-locs.)

	RFF	MLP	1-NN (ball-tree)	1-NN (brute force)
Execution time (ns)	602.6	$\mathbf{145.8}$	4928.2	10913.9

- Mean execution time for inference (1.5M pseudo-locs.)

	RFF	MLP	1-NN (ball-tree)	1-NN (brute force)
Execution time (ns)	602.6	$\mathbf{145.8}$	4928.2	10913.9

- GPU implementation of RFF/MLP: very fast inference times

< Time complexity: inference >

- Mean execution time for inference (1.5M pseudo-locs.)

	RFF	MLP	1-NN (ball-tree)	1-NN (brute force)
Execution time (ns)	602.6	$\mathbf{145.8}$	4928.2	10913.9

- GPU implementation of RFF/MLP: very fast inference times
- Optimized 1-NN is interesting: information in pseudo-locations work well with very simple ML methods

- Mean execution time for inference (1.5M pseudo-locs.)

	RFF	MLP	1-NN (ball-tree)	1-NN (brute force)
Execution time (ns)	602.6	$\mathbf{145.8}$	4928.2	10913.9

- GPU implementation of RFF/MLP: very fast inference times
- Optimized 1-NN is interesting: information in pseudo-locations work well with very simple ML methods

- When considering online learning, parametric methods (i.e. RFF/MLP) would outperform non-parametric methods (i.e. 1-NN) in terms of inference complexity

- From a pseudo-location learn a precoder $\mathbf{w} \in \mathbb{C}^{N_a}$
  - Pseudo-location based beamforming

- From a pseudo-location learn a precoder $\mathbf{w} \in \mathbb{C}^{N_a}$
  - Pseudo-location based beamforming

- Training loss: correlation-based (between precoder and downlink channel)

$$\mathcal{L} = 1 - \frac{1}{\mathcal{B}} \sum_{u=1}^{\mathcal{B}} \frac{\left| \mathbf{w}_u^{\mathsf{H}} \mathbf{g}_u \right|^2}{\left\| \mathbf{g}_u \right\|_2^2} \tag{1}$$

- From a pseudo-location learn a precoder $\mathbf{w} \in \mathbb{C}^{N_a}$
  - Pseudo-location based beamforming

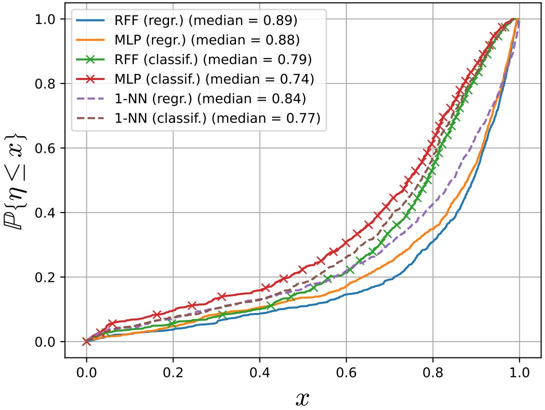
- Training loss: correlation-based (between precoder and downlink channel)

$$\mathcal{L} = 1 - \frac{1}{\mathcal{B}} \sum_{u=1}^{\mathcal{B}} \frac{\left|\mathbf{w}_u^{\mathsf{H}} \mathbf{g}_u\right|^2}{\|\mathbf{g}_u\|_2^2} \tag{1}$$

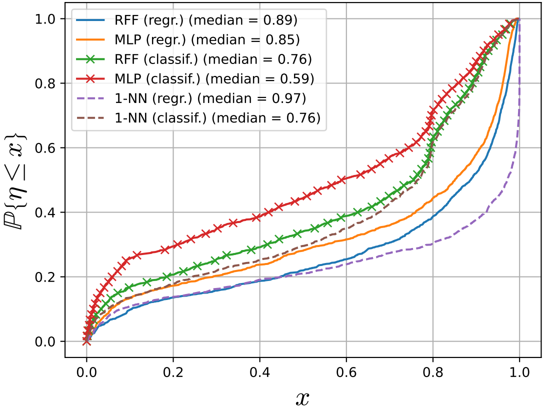
- Evaluation metric: normalized correlation between precoder and downlink channel:

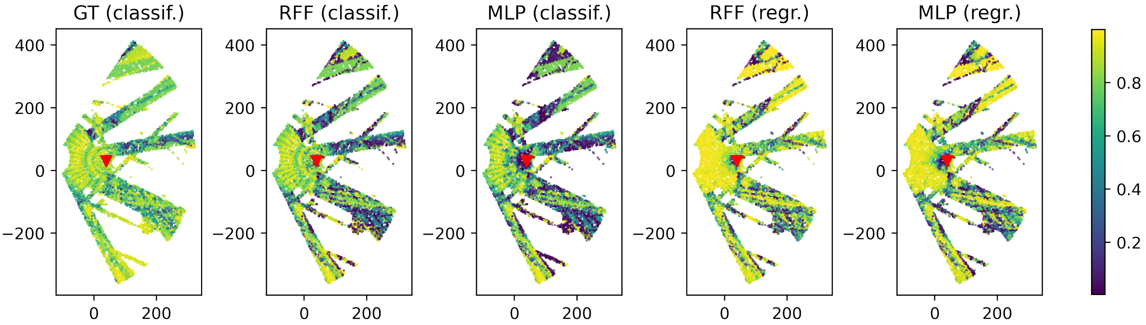
$$\eta = \frac{\left|\mathbf{w}^{\mathsf{H}} \mathbf{g}\right|^2}{\|\mathbf{g}\|_2^2} \tag{2}$$

- DeepMIMO



- Sionna




GT (classif.)
RFF (classif.)
MLP (classif.)
RFF (regr.)
MLP (regr.)
400
200
0
−200
0
200
0.8
0.6
0.4
0.2

- Contributions:

- Contributions:
  - Pseudo-locations can be used to select the best beam at **another** BS than the one used for charting.

- Contributions:
  - Pseudo-locations can be used to select the best beam at **another** BS than the one used for charting.
  - Better performance when using RFF networks in the pseudo-location to beam mapping.

- Contributions:
  - Pseudo-locations can be used to select the best beam at **another** BS than the one used for charting.
  - Better performance when using RFF networks in the pseudo-location to beam mapping.
  - Learning a precoder from pseudo-locations yields better performance than using a 2D-DFT codebook.

- Contributions:
  - Pseudo-locations can be used to select the best beam at **another** BS than the one used for charting.
  - Better performance when using RFF networks in the pseudo-location to beam mapping.
  - Learning a precoder from pseudo-locations yields better performance than using a 2D-DFT codebook.
  - Information in the chart also works very well with very simple machine learning methods (1-NN).

- Contributions:
  - Pseudo-locations can be used to select the best beam at **another** BS than the one used for charting.
  - Better performance when using RFF networks in the pseudo-location to beam mapping.
  - Learning a precoder from pseudo-locations yields better performance than using a 2D-DFT codebook.
  - Information in the chart also works very well with very simple machine learning methods (1-NN).
- Future work:

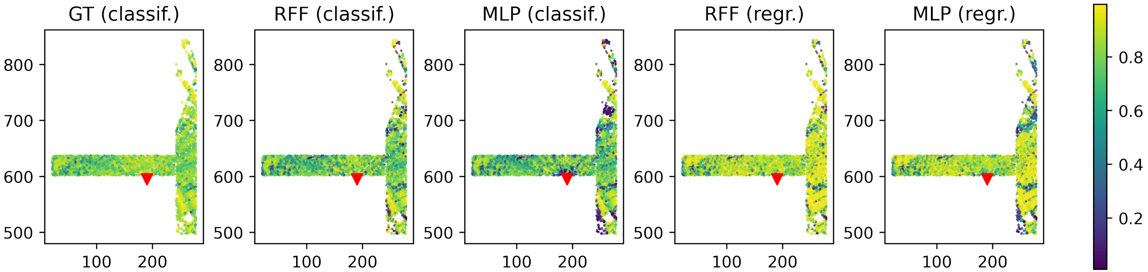
- Contributions:
  - Pseudo-locations can be used to select the best beam at **another** BS than the one used for charting.
  - Better performance when using RFF networks in the pseudo-location to beam mapping.
  - Learning a precoder from pseudo-locations yields better performance than using a 2D-DFT codebook.
  - Information in the chart also works very well with very simple machine learning methods (1-NN).

- Future work:
  - Pseudo-locations have only been obtained under the channel charting point of view.

- Contributions:
  - Pseudo-locations can be used to select the best beam at **another** BS than the one used for charting.
  - Better performance when using RFF networks in the pseudo-location to beam mapping.
  - Learning a precoder from pseudo-locations yields better performance than using a 2D-DFT codebook.
  - Information in the chart also works very well with very simple machine learning methods (1-NN).

- Future work:
  - Pseudo-locations have only been obtained under the channel charting point of view.
    - Would using an auto-encoders cause a performance drop ?
  - End-to-end training for channel charting and neural network.

Thank you!
Questions?



GT (classif.)
RFF (classif.)
MLP (classif.)
RFF (regr.)
MLP (regr.)
800
700
600
500
100
200
0.8
0.6
0.4
0.2

Thanks