CSI COMPRESSION USING CHANNEL CHARTING

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
⋆ b<>com, Rennes, France

baptiste.chatelier@insa-rennes.fr

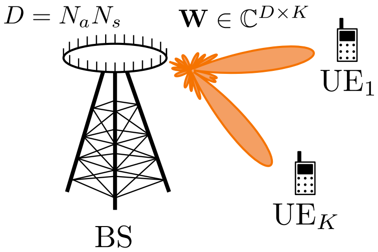
ASILOMAR 2024, Pacific Grove – October 30, 2024

SUM-RATE MAXIMIZATION IN FDD MMIMO SYSTEMS

- How to maximize the sum-rate with K parallel UEs in FDD mMIMO systems?

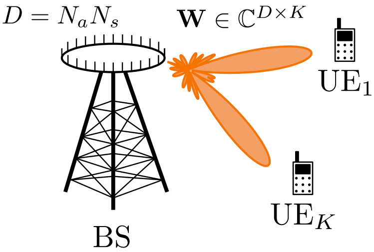
SUM-RATE MAXIMIZATION IN FDD MMIMO SYSTEMS

- How to maximize the sum-rate with K parallel UEs in FDD mMIMO systems?



SUM-RATE MAXIMIZATION IN FDD MMIMO SYSTEMS

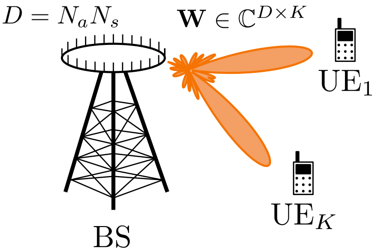
- How to maximize the sum-rate with K parallel UEs in FDD mMIMO systems?



$$\max_{\mathbf{W} \in \mathbb{C}^{D \times K}} \sum_{k=1}^{K} \log_2 \left(1 + \frac{\left|\mathbf{h}_k^{\mathsf{H}} \mathbf{w}_k\right|^2}{\sigma_k^2 + \sum_{j \neq k} \left|\mathbf{h}_k^{\mathsf{H}} \mathbf{w}_j\right|^2} \right) \quad \text{s.t.} \quad \mathcal{C}_p : \forall k \in [\![1, K]\!], \|\mathbf{w}_k\|_2 = \frac{1}{\sqrt{K}} \tag{1}$$

SUM-RATE MAXIMIZATION IN FDD MMIMO SYSTEMS

- How to maximize the sum-rate with K parallel UEs in FDD mMIMO systems?

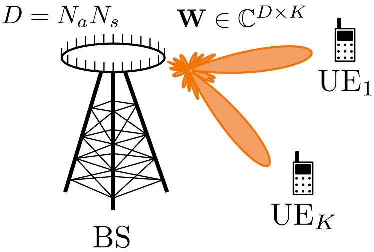


$$\max_{\mathbf{W} \in \mathbb{C}^{D\times K}} \quad \sum_{k=1}^{K} \log_2\left(1 + \frac{\left|\mathbf{h}_k^{\mathsf{H}}\mathbf{w}_k\right|^2}{\sigma_k^2 + \sum_{j\neq k}\left|\mathbf{h}_k^{\mathsf{H}}\mathbf{w}_j\right|^2}\right) \tag{1}$$
$$\text{s.t.} \quad \mathcal{C}_p : \forall k \in [\![1, K]\!], \|\mathbf{w}_k\|_2 = \frac{1}{\sqrt{K}}$$

- $\mathbf{w}_k$: precoding vector for UEk

SUM-RATE MAXIMIZATION IN FDD MMIMO SYSTEMS

- How to maximize the sum-rate with K parallel UEs in FDD mMIMO systems?



$$\max_{\mathbf{W} \in \mathbb{C}^{D \times K}} \quad \sum_{k=1}^{K} \log_2 \left(1 + \frac{\left|\mathbf{h}_k^{\mathsf{H}} \mathbf{w}_k\right|^2}{\sigma_k^2 + \sum_{j \neq k} \left|\mathbf{h}_k^{\mathsf{H}} \mathbf{w}_j\right|^2} \right) \tag{1}$$
$$\text{s.t.} \quad \mathcal{C}_p : \forall k \in [\![1, K]\!], \|\mathbf{w}_k\|_2 = \frac{1}{\sqrt{K}}$$

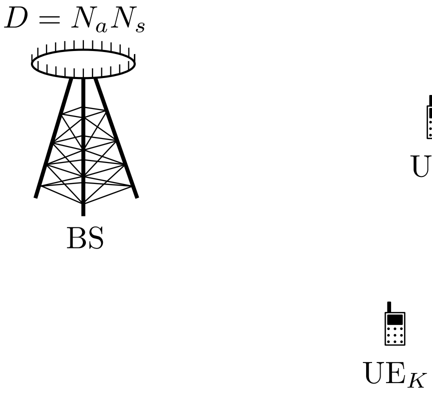
- $\mathbf{w}_k$: precoding vector for UEk
- $\mathcal{C}_p$: uniform power allocation policy

CSI COMPRESSION IN FDD SYSTEMS

- Classical beamforming methods rely on the full CSI knowledge: $\mathbf{H} \in \mathbb{C}^{D \times K}$

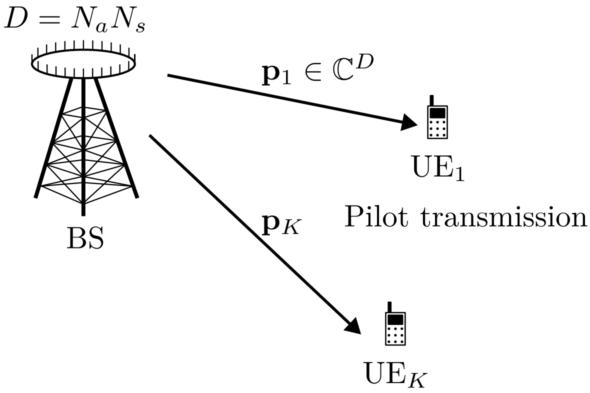
CSI COMPRESSION IN FDD SYSTEMS

- Classical beamforming methods rely on the full CSI knowledge: $\mathbf{H} \in \mathbb{C}^{D\times K}$
- In FDD systems, how to minimize the CSI reporting overhead?



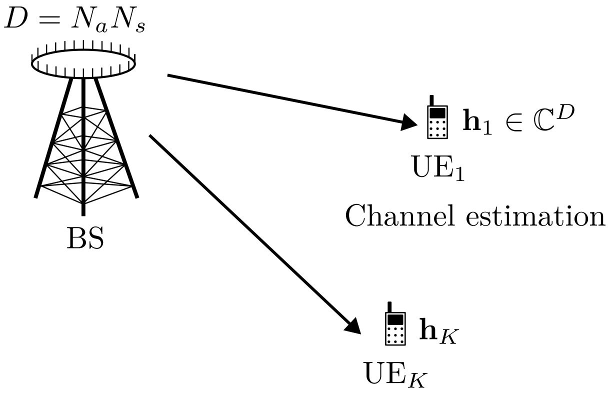
CSI COMPRESSION IN FDD SYSTEMS

- Classical beamforming methods rely on the full CSI knowledge: $\mathbf{H} \in \mathbb{C}^{D \times K}$
- In FDD systems, how to minimize the CSI reporting overhead?



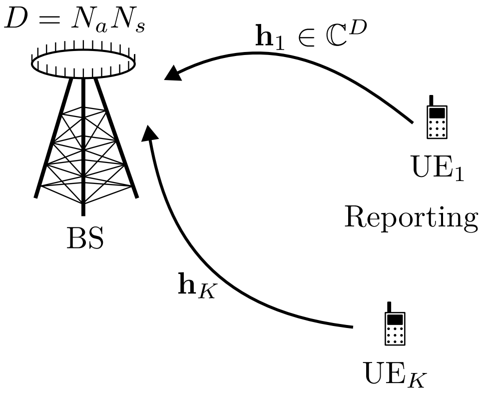
CSI COMPRESSION IN FDD SYSTEMS

- Classical beamforming methods rely on the full CSI knowledge: $\mathbf{H} \in \mathbb{C}^{D \times K}$
- In FDD systems, how to minimize the CSI reporting overhead?



CSI COMPRESSION IN FDD SYSTEMS

- Classical beamforming methods rely on the full CSI knowledge: $\mathbf{H} \in \mathbb{C}^{D \times K}$
- In FDD systems, how to minimize the CSI reporting overhead?



CSI COMPRESSION IN FDD SYSTEMS

- Classical beamforming methods rely on the full CSI knowledge: $\mathbf{H} \in \mathbb{C}^{D \times K}$
- In FDD systems, how to minimize the CSI reporting overhead?

- Classical CSI reporting:

CSI COMPRESSION IN FDD SYSTEMS

- Classical beamforming methods rely on the full CSI knowledge: $\mathbf{H} \in \mathbb{C}^{D \times K}$
- In FDD systems, how to minimize the CSI reporting overhead?

- Classical CSI reporting:
  - UEs report the full CSI: $\mathbf{h} \in \mathbb{C}^{D}$

CSI COMPRESSION IN FDD SYSTEMS

- Classical beamforming methods rely on the full CSI knowledge: $\mathbf{H} \in \mathbb{C}^{D \times K}$
- In FDD systems, how to minimize the CSI reporting overhead?

- Classical CSI reporting:
  - UEs report the full CSI: $\mathbf{h} \in \mathbb{C}^{D}$
  - $\mathcal{O}\left(D\right)$ complexity

CSI COMPRESSION IN FDD SYSTEMS

- Classical beamforming methods rely on the full CSI knowledge: $\mathbf{H} \in \mathbb{C}^{D \times K}$
- In FDD systems, how to minimize the CSI reporting overhead?

- Classical CSI reporting:
  - UEs report the full CSI: $\mathbf{h} \in \mathbb{C}^{D}$
  - $\mathcal{O}(D)$ complexity
- CSI compression:

CSI COMPRESSION IN FDD SYSTEMS

- Classical beamforming methods rely on the full CSI knowledge: $\mathbf{H} \in \mathbb{C}^{D \times K}$
- In FDD systems, how to minimize the CSI reporting overhead?

- Classical CSI reporting:
  - UEs report the full CSI: $\mathbf{h} \in \mathbb{C}^{D}$
  - $\mathcal{O}(D)$ complexity
- CSI compression:
  - UEs report a compressed CSI: $\mathbf{z} \in \mathbb{R}^{d}$

CSI COMPRESSION IN FDD SYSTEMS

- Classical beamforming methods rely on the full CSI knowledge: $\mathbf{H} \in \mathbb{C}^{D \times K}$
- In FDD systems, how to minimize the CSI reporting overhead?

- Classical CSI reporting:
  - UEs report the full CSI: $\mathbf{h} \in \mathbb{C}^{D}$
  - $\mathcal{O}(D)$ complexity

- CSI compression:
  - UEs report a compressed CSI: $\mathbf{z} \in \mathbb{R}^{d}$
  - BS reconstructs the full CSI from $\mathbf{z}$

CSI COMPRESSION IN FDD SYSTEMS

- Classical beamforming methods rely on the full CSI knowledge: $\mathbf{H} \in \mathbb{C}^{D \times K}$
- In FDD systems, how to minimize the CSI reporting overhead?

- Classical CSI reporting:
  - UEs report the full CSI: $\mathbf{h} \in \mathbb{C}^{D}$
  - $\mathcal{O}(D)$ complexity
- CSI compression:
  - UEs report a compressed CSI: $\mathbf{z} \in \mathbb{R}^{d}$
  - BS reconstructs the full CSI from $\mathbf{z}$
  - $\mathcal{O}(d)$ complexity ($d \ll D$)

CSI COMPRESSION IN FDD SYSTEMS

- Classical beamforming methods rely on the full CSI knowledge: $\mathbf{H} \in \mathbb{C}^{D \times K}$
- In FDD systems, how to minimize the CSI reporting overhead?

- Classical CSI reporting:
  - UEs report the full CSI: $\mathbf{h} \in \mathbb{C}^{D}$
  - $\mathcal{O}(D)$ complexity
- CSI compression:
  - UEs report a compressed CSI: $\mathbf{z} \in \mathbb{R}^{d}$
  - BS reconstructs the full CSI from $\mathbf{z}$
  - $\mathcal{O}(d)$ complexity ($d \ll D$)

Huge complexity reduction

TOWARDS TASK-BASED CSI COMPRESSION

- Is it necessary to reconstruct the full CSI from its compressed version?

TOWARDS TASK-BASED CSI COMPRESSION

- Is it necessary to reconstruct the full CSI from its compressed version?
- CSI is mainly used for beamforming:

TOWARDS TASK-BASED CSI COMPRESSION

- Is it necessary to reconstruct the full CSI from its compressed version?

- CSI is mainly used for beamforming:

Is it possible to decode a beamforming matrix from the compressed CSI?

CONTRIBUTIONS

- **Use of channel charting for task-based CSI compression**

CONTRIBUTIONS

- **Use of channel charting for task-based CSI compression**
- **Heuristic task-based learning strategy**

CONTRIBUTIONS

- **Use of channel charting for task-based CSI compression**
- **Heuristic task-based learning strategy**
- **Learnable parameter optimization**

CHANNEL CHARTING AS AN ENCODING FUNCTION

- Channel charting: dimensionality reduction method that preserves local neighborhoods

CHANNEL CHARTING AS AN ENCODING FUNCTION

- Channel charting: dimensionality reduction method that preserves local neighborhoods

- Channel $\mathbf{h}_i \in \mathbb{C}^D$

CHANNEL CHARTING AS AN ENCODING FUNCTION

- Channel charting: dimensionality reduction method that preserves local neighborhoods

- Channel $\mathbf{h}_i \in \mathbb{C}^D$
- Pseudo-loc. $\mathbf{z}_i = \mathcal{E}\left(\mathbf{h}_i\right) \in \mathbb{R}^d$,
  $d \ll D$

CHANNEL CHARTING AS AN ENCODING FUNCTION

- Channel charting: dimensionality reduction method that preserves local neighborhoods

- Channel $\mathbf{h}_i \in \mathbb{C}^D$
- Pseudo-loc. $\mathbf{z}_i = \mathcal{E}\left(\mathbf{h}_i\right) \in \mathbb{R}^d$,
  $d \ll D$
- CC objective:

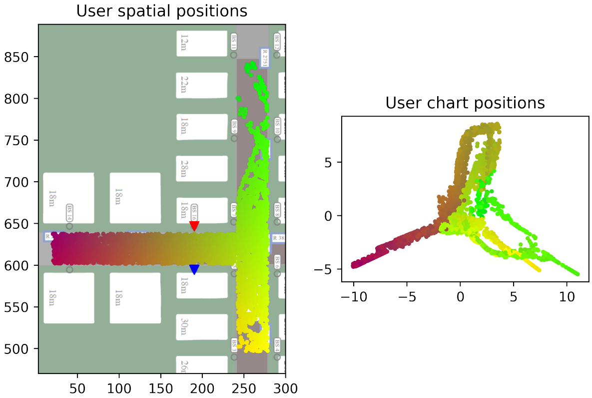
$$\left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2 \approx \left\|\mathbf{z}_i - \mathbf{z}_j\right\|_2 \quad (2)$$

CHANNEL CHARTING AS AN ENCODING FUNCTION

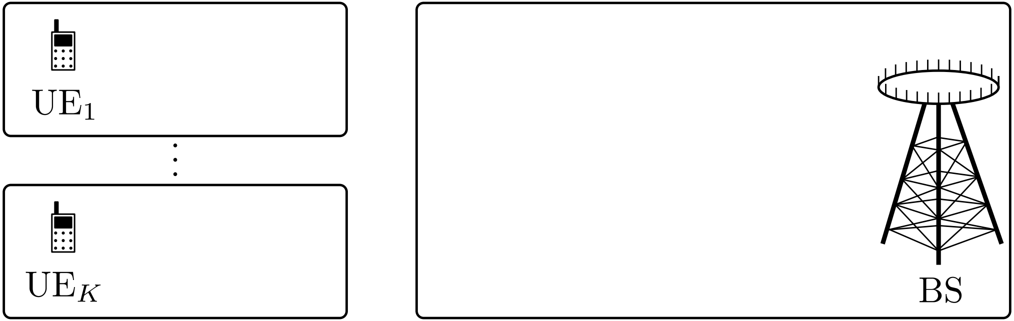
- Channel charting: dimensionality reduction method that preserves local neighborhoods

- Channel $\mathbf{h}_i \in \mathbb{C}^D$
- Pseudo-loc. $\mathbf{z}_i = \mathcal{E}(\mathbf{h}_i) \in \mathbb{R}^d$, $d \ll D$
- CC objective:

$$\|\mathbf{x}_i - \mathbf{x}_j\|_2 \approx \|\mathbf{z}_i - \mathbf{z}_j\|_2 \quad (2)$$

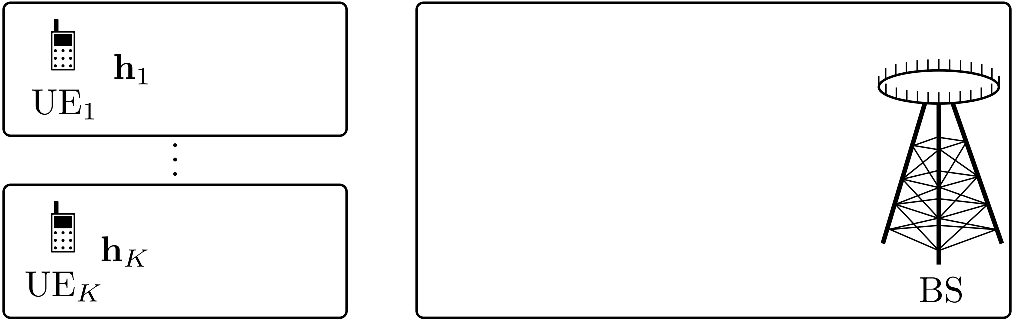


PROPOSED APPROACH



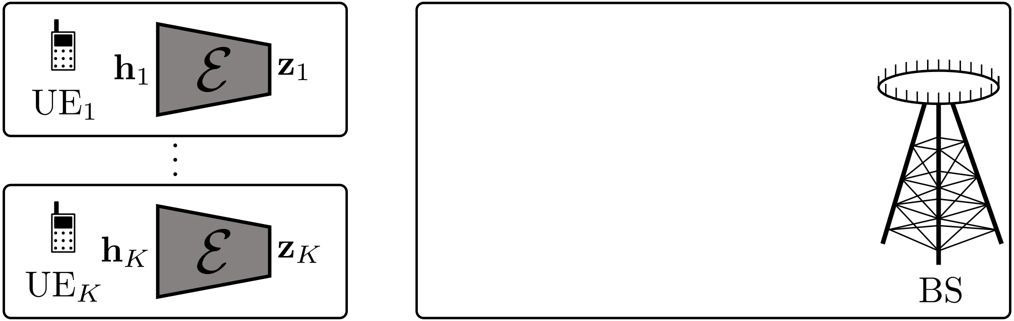
- System model

PROPOSED APPROACH



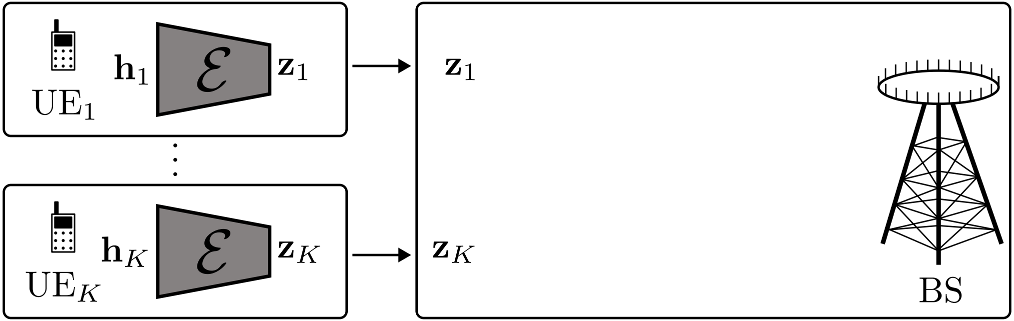
- Downlink channel estimation

PROPOSED APPROACH



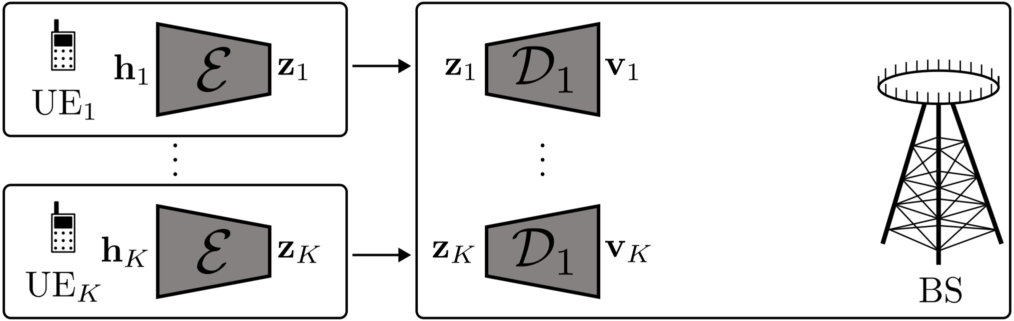
- Channel compression

PROPOSED APPROACH



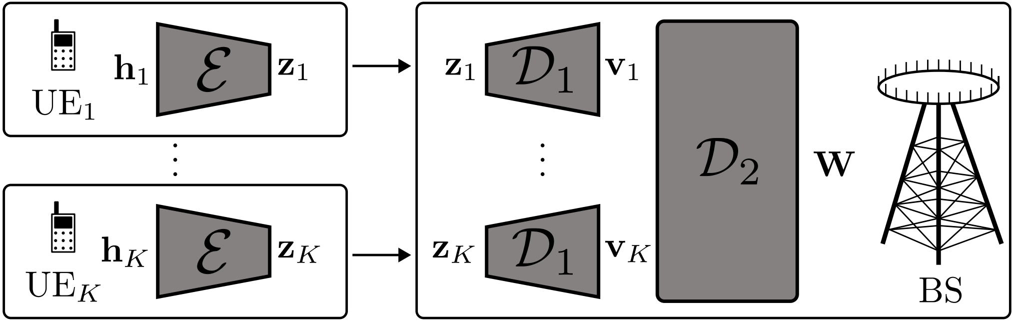
- Encoded channel transmission

PROPOSED APPROACH



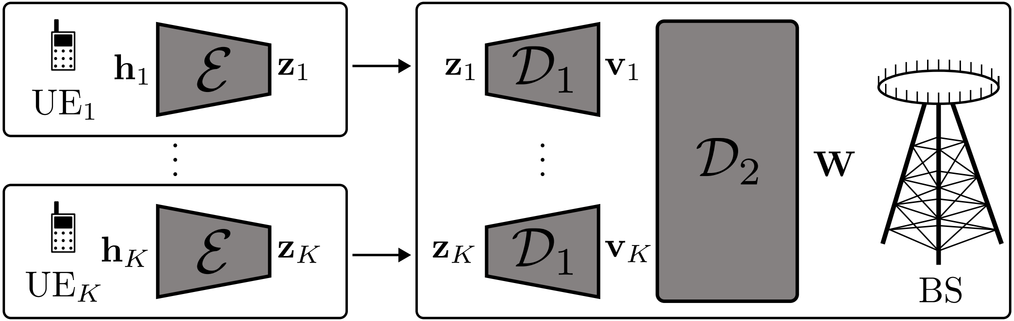
- Individual decoding: mono-UE processing

PROPOSED APPROACH



- Parallel decoding: multi-UE processing

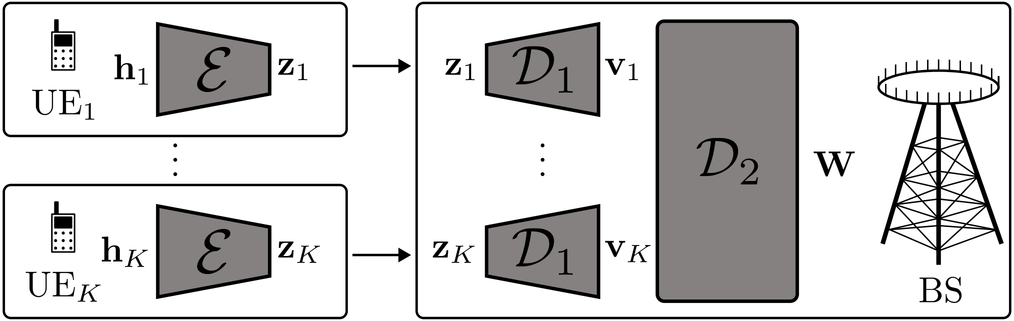
PROPOSED APPROACH



- Loss function: $\mathbf{v} = (\mathcal{D}_1 \circ \mathcal{E})(\mathbf{h})$

$$\mathcal{L} = 1 - \frac{1}{|\mathcal{B}|} \sum_{\mathbf{h} \in \mathcal{B}} \frac{\left|\mathbf{v}^{\mathsf{H}} \mathbf{h}\right|^2}{\|\mathbf{h}\|_2^2}, \qquad (3)$$

PROPOSED APPROACH

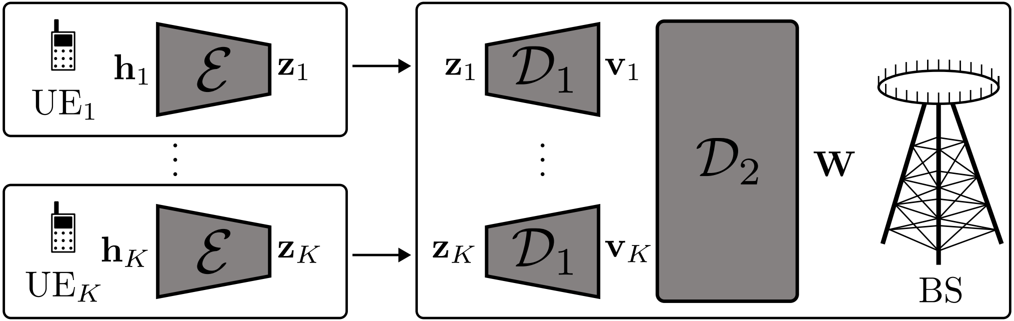


- Loss function: $\mathbf{v} = (\mathcal{D}_1 \circ \mathcal{E})(\mathbf{h})$

$$\mathcal{L} = 1 - \frac{1}{|\mathcal{B}|} \sum_{\mathbf{h} \in \mathcal{B}} \frac{\left|\mathbf{v}^{\mathsf{H}}\mathbf{h}\right|^2}{\|\mathbf{h}\|_2^2}, \qquad (3)$$

- $\mathcal{L}$ is minimized when $\mathbf{v} \to \mathbf{h}e^{\mathrm{j}\phi} / \|\mathbf{h}\|_2$

PROPOSED APPROACH



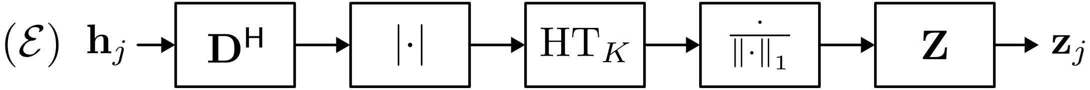
- Loss function: $\mathbf{v} = (\mathcal{D}_1 \circ \mathcal{E})(\mathbf{h})$

$$\mathcal{L} = 1 - \frac{1}{|\mathcal{B}|} \sum_{\mathbf{h} \in \mathcal{B}} \frac{\left|\mathbf{v}^{\mathsf{H}}\mathbf{h}\right|^2}{\|\mathbf{h}\|_2^2}, \tag{3}$$

- $\mathcal{L}$ is minimized when $\mathbf{v} \rightarrow \mathbf{h}\mathrm{e}^{\mathrm{j}\phi} / \|\mathbf{h}\|_2$
- After denormalization, $\mathbf{v}$ is an imperfect channel estimate, that can be used in linear precoders

ENCODER ARCHITECTURE

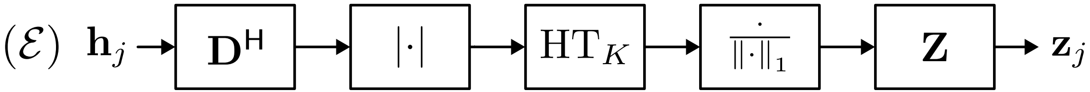
- Architecture presented in[1,2,3]



[1]Le Magoarou et al., “Channel charting based beamforming”.
[2]Yassine, Le Magoarou, et al., “Leveraging triplet loss and nonlinear dimensionality reduction for on-the-fly channel charting”.
[3]Yassine, Chatelier, et al., “Model-Based Deep Learning for Beam Prediction Based on a Channel Chart”.

ENCODER ARCHITECTURE

- Architecture presented in[1,2,3]



- $\mathbf{D} = \{\mathbf{h}_i\}_{i=1}^{N_c}$ and $\mathbf{Z} = \{\mathbf{z}_i\}_{i=1}^{N_c}$ are initialized through the `ISOMAP` algorithm

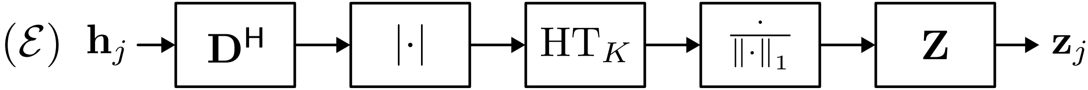
[1] Le Magoarou et al., “Channel charting based beamforming”.

[2] Yassine, Le Magoarou, et al., “Leveraging triplet loss and nonlinear dimensionality reduction for on-the-fly channel charting”.

[3] Yassine, Chatelier, et al., “Model-Based Deep Learning for Beam Prediction Based on a Channel Chart”.

ENCODER ARCHITECTURE

- Architecture presented in[1,2,3]



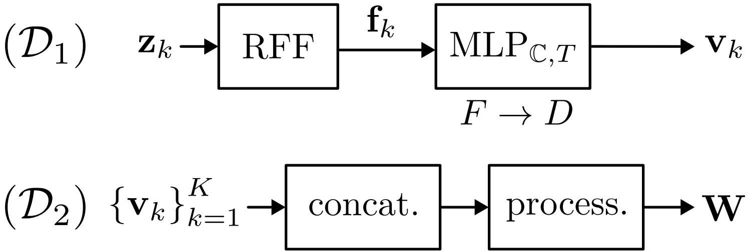
- $\mathbf{D} = \{\mathbf{h}_i\}_{i=1}^{N_c}$ and $\mathbf{Z} = \{\mathbf{z}_i\}_{i=1}^{N_c}$ are initialized through the `ISOMAP` algorithm
- $\mathbf{z}_j$ can be seen as a wisely chosen convex combination of calibration chart locations

[1] Le Magoarou et al., “Channel charting based beamforming”.

[2] Yassine, Le Magoarou, et al., “Leveraging triplet loss and nonlinear dimensionality reduction for on-the-fly channel charting”.

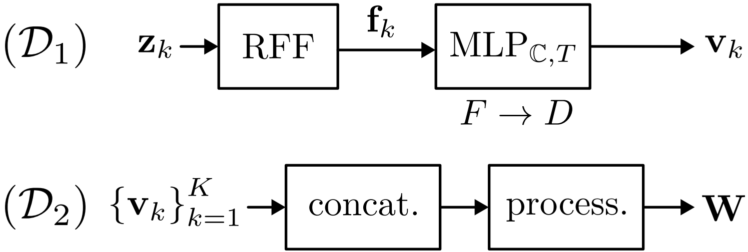
[3] Yassine, Chatelier, et al., “Model-Based Deep Learning for Beam Prediction Based on a Channel Chart”.

DECODER ARCHITECTURE



- RFF: helps to learn fast variations in the chart-location-to-precoder mapping

DECODER ARCHITECTURE



- RFF: helps to learn fast variations in the chart-location-to-precoder mapping

- $\mathbf{f}_k = \begin{bmatrix} \cos(2\pi \mathbf{F}\mathbf{z}_k) \\ \sin(2\pi \mathbf{F}\mathbf{z}_k) \end{bmatrix}, \mathbf{F} \sim \mathcal{N}\left(\mathbf{0}_F, \sigma_F^2 \mathrm{Id}_F\right)$

LEARNABLE PARAMETER OPTIMIZATION

- Main parameter complexity comes from $\mathbf{D} \in \mathbb{C}^{D \times N_c}$ in the encoder

LEARNABLE PARAMETER OPTIMIZATION

- Main parameter complexity comes from $\mathbf{D} \in \mathbb{C}^{D \times N_c}$ in the encoder
    - N_c needs to be high to have a good init. with `ISOMAP`, $D = N_a N_s$

LEARNABLE PARAMETER OPTIMIZATION

- Main parameter complexity comes from $\mathbf{D} \in \mathbb{C}^{D \times N_c}$ in the encoder
  - N_c needs to be high to have a good init. with `ISOMAP`, $D = N_a N_s$
  - $N_c = 5$k, $D = 1024 \Rightarrow D \times N_c \simeq 5$M complex params.

LEARNABLE PARAMETER OPTIMIZATION

- Main parameter complexity comes from $\mathbf{D} \in \mathbb{C}^{D \times N_c}$ in the encoder
    - N_c needs to be high to have a good init. with `ISOMAP`, $D = N_a N_s$
    - $N_c = 5$k, $D = 1024 \Rightarrow D \times N_c \simeq 5$M complex params.
- Use the subsampling strategy of[4]:

[4] Taner et al., "Channel Charting for Streaming CSI Data".

LEARNABLE PARAMETER OPTIMIZATION

- Main parameter complexity comes from $\mathbf{D} \in \mathbb{C}^{D \times N_c}$ in the encoder
  - N_c needs to be high to have a good init. with `ISOMAP`, $D = N_a N_s$
  - $N_c = 5$k, $D = 1024 \Rightarrow D \times N_c \simeq 5$M complex params.
- Use the subsampling strategy of[4]:
  - Discard channels that produces too similar chart locations, i.e. solve:

$$\min \max_{(\mathbf{h}_i, \mathbf{h}_j) \in \tilde{\mathbf{D}}} \operatorname{sim}\left(\mathbf{h}_i, \mathbf{h}_j\right), \quad (4)$$

[4] Taner et al., “Channel Charting for Streaming CSI Data”.

LEARNABLE PARAMETER OPTIMIZATION

- Main parameter complexity comes from $\mathbf{D} \in \mathbb{C}^{D \times N_c}$ in the encoder
  - N_c needs to be high to have a good init. with `ISOMAP`, $D = N_a N_s$
  - $N_c = 5$k, $D = 1024 \Rightarrow D \times N_c \simeq 5$M complex params.
- Use the subsampling strategy of[4]:
  - Discard channels that produces too similar chart locations, i.e. solve:

$$\min \max_{(\mathbf{h}_i, \mathbf{h}_j) \in \tilde{\mathbf{D}}} \operatorname{sim}\left(\mathbf{h}_i, \mathbf{h}_j\right), \tag{4}$$

where $\operatorname{sim}\left(\mathbf{h}_i, \mathbf{h}_j\right)$ is a similarity metric:

$$\operatorname{sim}\left(\mathbf{h}_i, \mathbf{h}_j\right) = \frac{\left|\mathcal{E}\left(\mathbf{h}_i\right)^{\mathsf{H}} \mathcal{E}\left(\mathbf{h}_j\right)\right|}{\left\|\mathcal{E}\left(\mathbf{h}_i\right)\right\|_2 \left\|\mathcal{E}\left(\mathbf{h}_j\right)\right\|_2}. \tag{5}$$

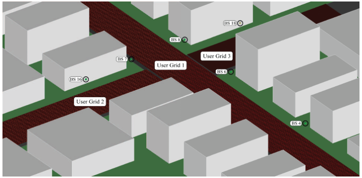
[4] Taner et al., “Channel Charting for Streaming CSI Data”.

EXPERIMENTS: DATASETS AND METRICS

- Two scenes:

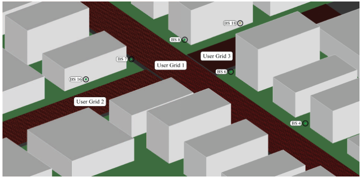
EXPERIMENTS: DATASETS AND METRICS

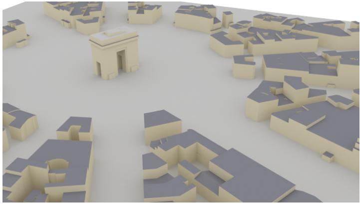
- Two scenes:
    - `DeepMIMO`: urban canyon, 3.5GHz



EXPERIMENTS: DATASETS AND METRICS

- Two scenes:
  - `DeepMIMO`: urban canyon, 3.5GHz
  - `Sionna`: Paris, 28GHz

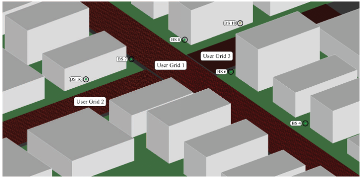


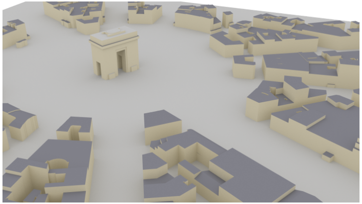


EXPERIMENTS: DATASETS AND METRICS

- Two scenes:
  - `DeepMIMO`: urban canyon, 3.5GHz
  - `Sionna`: Paris, 28GHz

- Radio parameters:

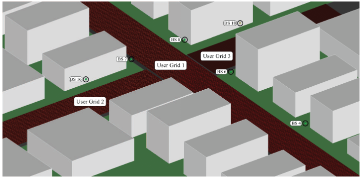


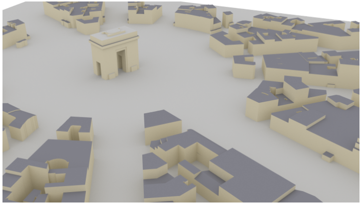


EXPERIMENTS: DATASETS AND METRICS

- Two scenes:
  - `DeepMIMO`: urban canyon, 3.5GHz
  - `Sionna`: Paris, 28GHz

- Radio parameters:
  - BS with 8x8 UPA $\Rightarrow N_a = 64$

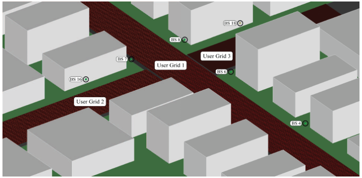


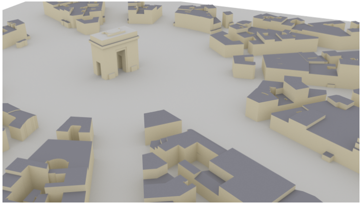


EXPERIMENTS: DATASETS AND METRICS

- Two scenes:
  - DeepMIMO: urban canyon, 3.5GHz
  - Sionna: Paris, 28GHz

- Radio parameters:
  - BS with 8x8 UPA $\Rightarrow N_a = 64$
  - $N_s = 16$ subcs. over 20MHz

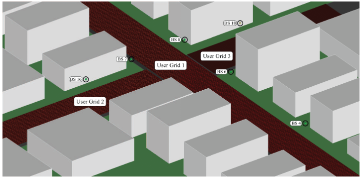


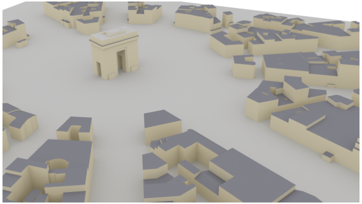


EXPERIMENTS: DATASETS AND METRICS

- Two scenes:
  - DeepMIMO: urban canyon, 3.5GHz
  - Sionna: Paris, 28GHz

- Radio parameters:
  - BS with 8x8 UPA $\Rightarrow N_a = 64$
  - $N_s = 16$ subcs. over 20MHz
  - Mono-ant. UEs





EXPERIMENTS: DATASETS AND METRICS

- Evaluation metrics:

EXPERIMENTS: DATASETS AND METRICS

- Evaluation metrics:
    - Mono-UE scenario, squared-cosine similarity:

$$\rho_k = \frac{\left|\mathbf{v}_k^{\mathsf{H}}\mathbf{h}_k\right|^2}{\|\mathbf{v}_k\|_2^2\,\|\mathbf{h}_k\|_2^2}, \tag{6}$$

$$\mathbf{v}_k = \left(\mathcal{D}_1 \circ \mathcal{E}\right)\left(\mathbf{h}_k\right).$$

EXPERIMENTS: DATASETS AND METRICS

- Evaluation metrics:
  - Mono-UE scenario, squared-cosine similarity:

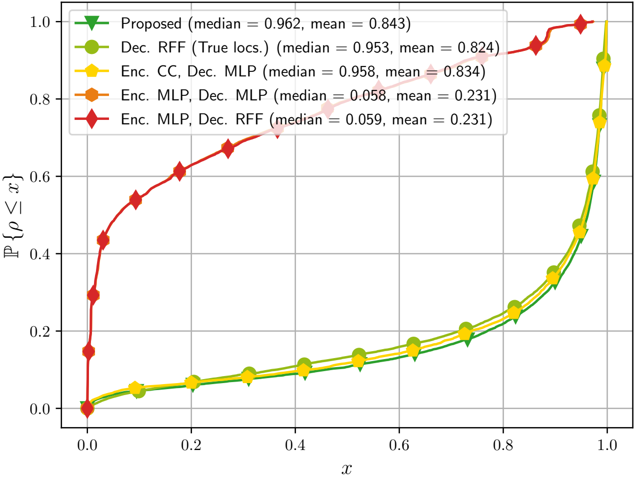
$$\rho_k = \frac{\left|\mathbf{v}_k^{\mathsf{H}}\mathbf{h}_k\right|^2}{\left\|\mathbf{v}_k\right\|_2^2 \left\|\mathbf{h}_k\right\|_2^2}, \tag{6}$$

    $\mathbf{v}_k = (\mathcal{D}_1 \circ \mathcal{E})(\mathbf{h}_k)$.
  - Multi-UE scenario, ergodic sum-rate:

$$\mathring{\mathcal{R}} = \frac{1}{B}\sum_{b=1}^{B}\sum_{k}\log_2\left(1 + \frac{\left|\mathbf{h}_{k,b}^{\mathsf{H}}\mathbf{w}_{k,b}\right|^2}{\sigma_{k,b}^2 + \sum_{j\neq k}\left|\mathbf{h}_{k,b}^{\mathsf{H}}\mathbf{w}_{j,b}\right|^2}\right), \tag{7}$$

    UE set partition into B groups of K UEs. $\mathbf{W}_b = \mathcal{D}_2 \circ \left(\{\mathbf{v}_{k,b}\}_{k=1}^{K}\right)$

EXPERIMENTS: PERFORMANCE WRT. BASELINES



EXPERIMENTS: ENC. LEARNING AND SUBSAMPLING IMPACT

	Network	a
	Params.	$101\mathrm{k}$
`Sionna`	Median ρ	0.88
	Mean ρ	0.79
`DeepMIMO`	Median ρ	0.91
	Mean ρ	0.73

- a. No encoder learning, no subsampling

EXPERIMENTS: ENC. LEARNING AND SUBSAMPLING IMPACT

	Network	a	b
	Params.	101k	10.3M
`Sionna`	Median ρ	0.88	**0.94**
	Mean ρ	0.79	0.86
`DeepMIMO`	Median ρ	0.91	0.96
	Mean ρ	0.73	0.85

- a. No encoder learning, no subsampling
- b. Encoder learning, no subsampling

EXPERIMENTS: ENC. LEARNING AND SUBSAMPLING IMPACT

	Network	a	b	c
	Params.	101k	10.3M	306k
Sionna	Median ρ	0.88	**0.94**	**0.94**
	Mean ρ	0.79	0.86	**0.88**
DeepMIMO	Median ρ	0.91	0.96	**0.97**
	Mean ρ	0.73	0.85	**0.87**

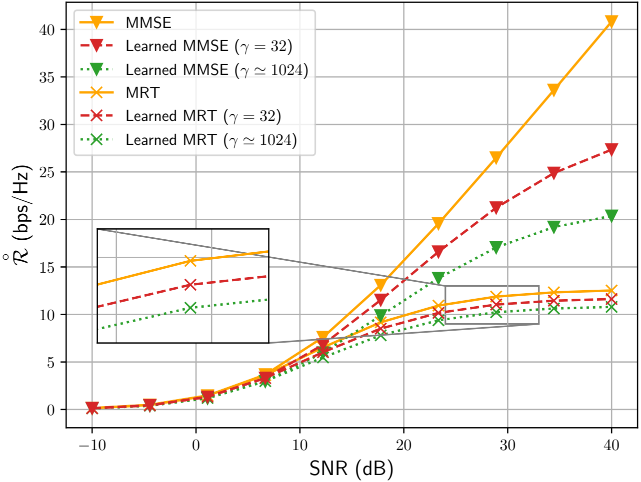
- a. No encoder learning, no subsampling
- b. Encoder learning, no subsampling
- c. Encoder learning, subsampling ($\tilde{N} = 100$)

EXPERIMENTS: ENC. LEARNING AND SUBSAMPLING IMPACT

	Network	a	b	c	d
	Params.	101k	10.3M	306k	121k
Sionna	Median ρ	0.88	**0.94**	**0.94**	**0.94**
	Mean ρ	0.79	0.86	**0.88**	**0.88**
DeepMIMO	Median ρ	0.91	0.96	**0.97**	0.96
	Mean ρ	0.73	0.85	**0.87**	0.84

- a. No encoder learning, no subsampling
- b. Encoder learning, no subsampling
- c. Encoder learning, subsampling ($\tilde{N} = 100$)
- d. Encoder learning, subsampling ($\tilde{N} = 10$)

EXPERIMENTS: MULTI-UE PERFORMANCE



CONCLUSION

- Contributions:

CONCLUSION

- Contributions:
  - CC can be used as an encoding method in a task-based CSI compression scenario

CONCLUSION

- Contributions:
  - CC can be used as an encoding method in a task-based CSI compression scenario
  - Better compression ratios than classical auto-encoders

CONCLUSION

- Contributions:
  - CC can be used as an encoding method in a task-based CSI compression scenario
  - Better compression ratios than classical auto-encoders
  - Inter-UE interference cancellation without considering it during training

CONCLUSION

- Contributions:
  - CC can be used as an encoding method in a task-based CSI compression scenario
  - Better compression ratios than classical auto-encoders
  - Inter-UE interference cancellation without considering it during training

- Future work:

CONCLUSION

- Contributions:
  - CC can be used as an encoding method in a task-based CSI compression scenario
  - Better compression ratios than classical auto-encoders
  - Inter-UE interference cancellation without considering it during training

- Future work:
  - Power allocation policy learning

CONCLUSION

- Contributions:
  - CC can be used as an encoding method in a task-based CSI compression scenario
  - Better compression ratios than classical auto-encoders
  - Inter-UE interference cancellation without considering it during training

- Future work:
  - Power allocation policy learning
  - SR-optimization through UE grouping policy learning

Thanks!