

# Wireless Channel Charting for Next-Generation Radio Access Networks

Maxime Guillaud and Christoph Studer



*International Symposium on Wireless Communication Systems*  
*July 14, 2024*

# Acknowledgements



Victoria Palhares  
(ETHZ)



A. Decurninge  
(Huawei)



Paul Ferrand  
(Huawei)



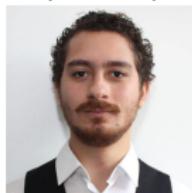
L. García Ordóñez  
(Huawei)



O. Tirkkonen  
(Aalto)



Sueda Taner  
(ETHZ)



Yamil Vindas  
(Inria)



J. Torres Sospedra  
(Univ. Valencia)



EC HORIZON EUROPE  
Grant #101139616



EC CHIST-ERA  
Grant CHIST-ERA-22-WAI-01

# Tutorial outline

- **Part 1:** Channel charting basics
- **Part 2:** Dimensionality reduction (DR)
- **Part 3:** Neural-network-based DR
- **Part 4:** CC pipeline (incl. feature design)
- **Part 5:** Performance metrics
- **Part 6:** Experimental results
- **Part 7:** Applications
- **Part 8:** Extensions
- **Part 9:** Conclusions and additional resources

# **Part 1:**

## Channel Charting Basics



## MIMO baseband model:

$$\mathbf{r}_n = \mathbf{H}_n \mathbf{s}_n + \mathbf{w}_n$$

- $n$  is a discrete-time index (time, frequency, basestation, or user)
- $\mathbf{s}_n$  is the  $U$ -dimensional transmit vector; contains pilots or data
- $\mathbf{H}_n$  is the channel matrix; **models narrowband propagation**
- $\mathbf{w}_n$  is additive disturbance (noise, interference, etc.)
- $\mathbf{r}_n$  is the  $B$ -dimensional received signal vector

# What is channel state information (CSI)?

- CSI describes signal propagation from every Tx-DAC to every Rx-DAC, including the wireless channel:



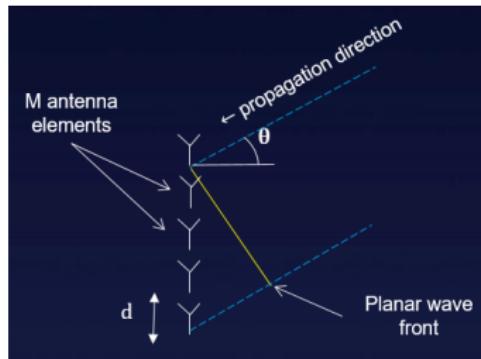
$$r_b(t) = \int_{-\infty}^{+\infty} h_{b,u}(t, \tau) s_u(t - \tau) d\tau + w_b(t)$$

- Channel matrix  $\mathbf{H}_n$  contains  $B \times U$  complex scalars
- For data detection (and beamforming), basestation estimates CSI  $\{\mathbf{H}_n\}_{n=1}^N$  using pilots

CSI is typically discarded after data detection (or beamforming)

The entries in  $\mathbf{H}_n$  are governed by Maxwell's equations

- **Toy example:** Simple plane-wave model with receiver equipped with uniform linear array (ULA) and  $\lambda/2$  antenna spacing



- Transmitter (Tx) in far-field
- Small number  $L$  of arriving paths; e.g., line-of-sight path & scatterers
- Only one UE and one antenna at Tx  
→  $\mathbf{H}_n$  **reduces to a column**  $\mathbf{h}_n$
- Channel vector model:

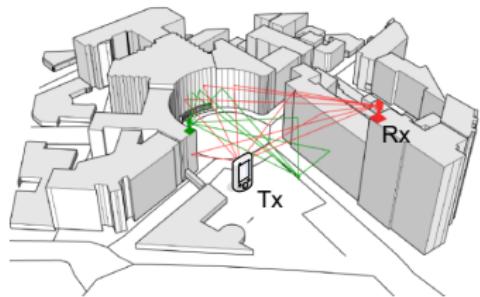
$$\mathbf{h}_n = \sum_{\ell=0}^{L-1} \alpha_\ell \mathbf{a}(\omega_\ell) \quad \text{with} \quad \mathbf{a}(\omega) = [e^{j0\omega}, e^{j1\omega}, \dots, e^{j(B-1)\omega}]^T,$$

where  $\omega_\ell = 2\pi \frac{d}{\lambda} \cos(\theta_\ell)$  with  $\ell$ th incident angle  $\theta_\ell$

# We are wasting valuable information!

Rx-RF chains are some of the most accurate sensors

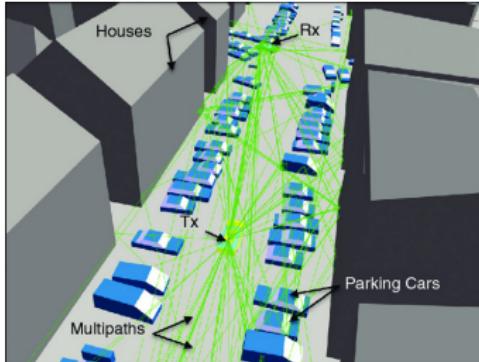
- After mitigation of RF impairments, CSI mostly describes wireless channel, which is determined by
  - transmitter position  $\mathbf{p}_{Tx}$
  - physical environment (scatterers, materials, etc.)
  - receiver position  $\mathbf{p}_{Rx}$



Idea: recycle estimated CSI to extract Tx position information

- Positioning using geometric models under line-of-sight conditions possible, but fails to work indoors or in dense urban scenarios and requires calibration and synchronization

# Real-world channels can be extremely complicated

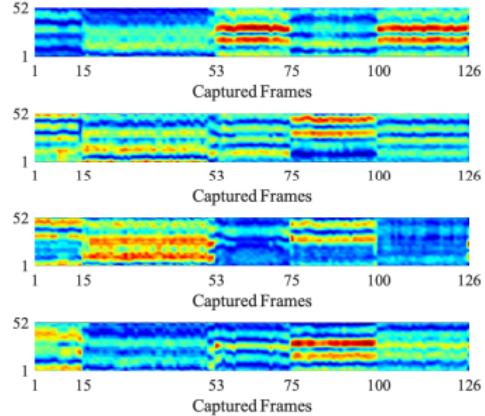


In reality, many geometric propagation models are intractable:

- The **set of channel parameters explodes**
- Parameters difficult/impossible to estimate; e.g., scatterers at unknown position, unknown electromagnetic properties, etc.
- Moving objects in environment (people, cars, foliage, etc.) further complicate things...

# A data-driven approach is preferable!

- CSI is high dimensional:  $B \times U \times N$
- CSI is estimated at fast rates ( $> 100\times$  per second)
- CSI is effectively low-dimensional  
→ **propagation characteristics usually determined by few parameters**



We can use a data-driven approach relying on machine learning to extract Tx position information solely from CSI dataset

Unfortunately: ground-truth position information of transmitters typically unknown (e.g., GNSS coordinates not available to BS)  
→ **we must resort to self-supervised machine learning**

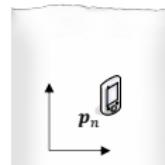
# Channel Charting to the Rescue!

# In an ideal world...

- ... each Tx position in space  $\mathbf{p}_n \in \mathbb{R}^d$  with  $d = 3$  leads to unique CSI  $\mathbf{H}(\mathbf{p}_n)$  at BS side
- ... CSI is high-dimensional, but lives on a low-dimensional manifold

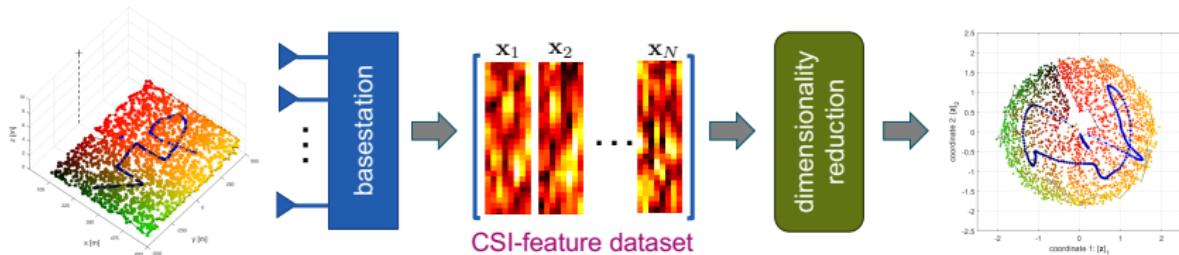
This enables **Channel Charting [1]!**

- learns low-dimensional manifold structure in CSI
- **manifold coordinates associated with CSI will be tied to Tx position**
- Advantages: self-supervised → no ground-truth position information necessary
- data-driven → nearly unlimited CSI data available
- recycles CSI → acquired anyway at the BS



[1] C. Studer, S. Medjkouh, E. Gönültaş, T. Goldstein, and O. Tirkkonen, "Channel Charting: Locating Users within the Radio Environment using Channel State Information," IEEE Access, 2018

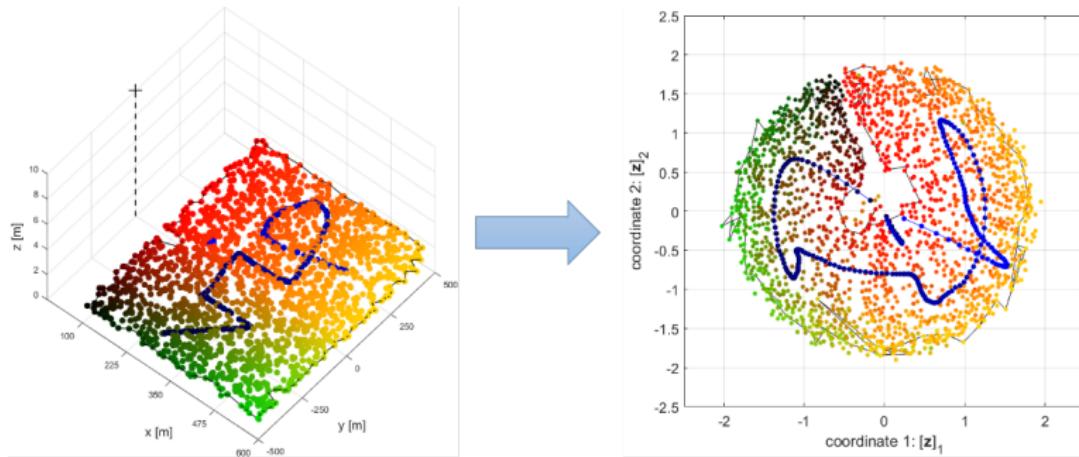
# Typical channel charting (CC) pipeline [1]



- Offline phase:
  - Transmit pilots from positions  $\mathbf{p}_n \in \mathbb{R}^d$  generates high-dimensional CSI matrices  $\mathbf{H}(\mathbf{p}_n)$
  - Extract **CSI features**  $\mathbf{H}(\mathbf{p}_n) \rightarrow \mathbf{x}_n$  and collect large CSI-feature dataset  $\{\mathbf{x}_n\}_{n=1}^N$
  - Apply **dimensionality reduction** to dataset and learn **low-dimensional embedding = channel chart**
- Online phase: use new CSI matrix  $\mathbf{H}(\mathbf{p}_{\text{new}})$  to estimate “position” in channel chart (coordinate of embedding)

[1] C. Studer, S. Medjkouh, E. Gönültaş, T. Goldstein, and O. Tirkkonen, “Channel Charting: Locating Users within the Radio Environment using Channel State Information,” IEEE Access, 2018

# An idealistic view on the capabilities of channel charting



32 BS antennas; uniform linear array with  $\lambda/2$  antenna spacing; 2 GHz; narrowband; simulated at 0 dB SNR

- We can extract relative UE location in a self-supervised manner, without GNSS, antenna calibration, or LoS connectivity
- We can predict future actions of UEs that are tied to location

**Unfortunately, reality is a bit more tricky...**

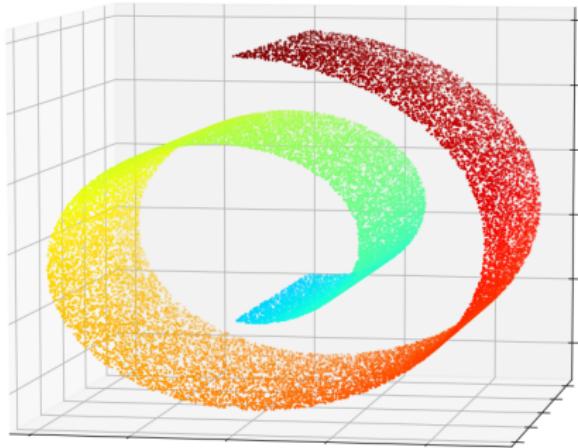
# **Part 2:**

## Dimensionality Reduction

Consider a Swiss roll



# Consider a Swiss roll



- Data-points (samples) in  $\mathbb{R}^3$  (color gradient is arbitrary)
- However, the position of any data-point on the Swiss roll manifold can be parametrized by only 2 variables...

**Dimensionality reduction** aims at identifying the **low-dimensional** parametric description of such sample distributions

# What is a manifold?

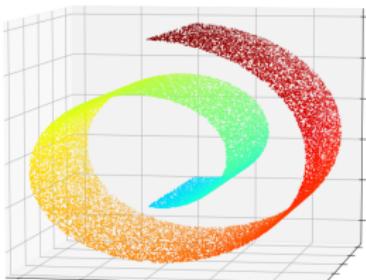


- Ex.: Earth is roughly a 3D sphere, but appears flat (2D) to us
- Manifold: a space that **locally resembles an Euclidean space**
- Each point of a  $d$ -dimensional manifold has a neighborhood that can be mapped to Euclidean space of dimension  $d$

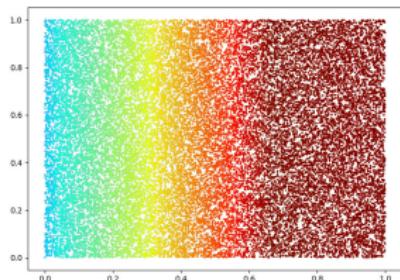
# Dimensionality reduction I

- Real-world data, such as speech signals, images, videos, fMRI scans, channel measurements are usually high dimensional
- Dimensionality reduction (DR): **mapping** of high-dimensional data into a **meaningful representation** of low dimension
- Ideally: Mapping should **preserve neighborhoods** in dataset with dimension corresponding to **intrinsic dimensionality**

Samples in  $\mathcal{H}^D$   
(ambient space)



Low-dimensional embedding  
in  $\mathcal{H}^d$  (latent space)



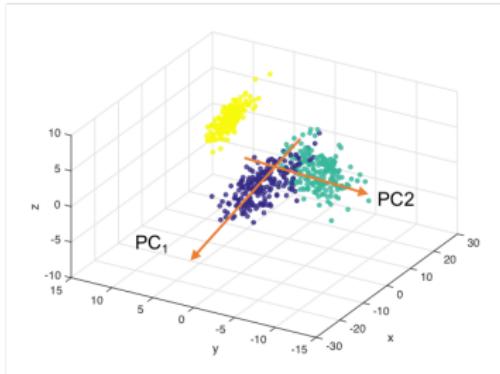
## Dimensionality reduction II

- Each sample maps to a point in the low-dimensional latent space ( $d \ll D$ ):  
$$\begin{array}{ccc} \mathcal{H}^D & \xrightarrow{\quad} & \mathcal{H}^d \\ \mathbf{x}_i & \xmapsto{f} & \mathbf{y}_i \end{array}$$
- Function  $f$  is **optimized based on dataset** such that the resulting low-dimensional **embedding**  $\mathcal{Y} = \{\mathbf{y}_i\}_{i=1}^N$  respects local geometry (neighborhoods) of the high-dimensional dataset
- Example: locally Euclidean manifold so that  
 $\|\mathbf{y}_i - \mathbf{y}_j\| \approx \|\mathbf{x}_i - \mathbf{x}_j\|$  for nearby points, i.e., small  $\|\mathbf{x}_i - \mathbf{x}_j\|$
- Dimensionality reduction is **self-supervised** → label free
- Dimensionality reduction is **continuous version of clustering**
- Dimensionality reduction is ill-posed and requires assumptions on the data: **dimension  $d$ , similarity measures, etc.**

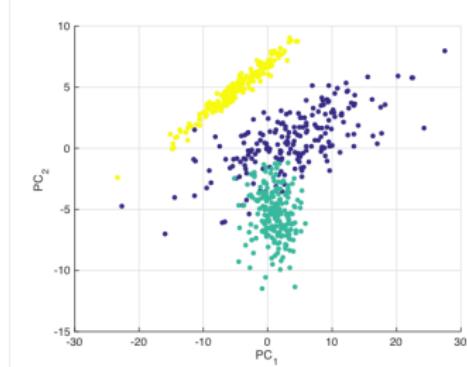
# Classical DR methods

# Principal component analysis (PCA) I

3D data in original representation



2D data in PCA representation



- PCA-based dimensionality reduction proceeds as follows:
  - Preprocess data-points  $\bar{\mathbf{x}}_n = \mathbf{x}_n - \mathbf{m}$  by subtracting mean  $\mathbf{m}$
  - Extract matrix  $\mathbf{M}$  from dominant left singular vectors of zero-mean data set matrix  $\bar{\mathbf{X}} = [\bar{\mathbf{x}}_1, \dots, \bar{\mathbf{x}}_N]$
  - $\mathbf{y}_n = \mathbf{M}(\mathbf{x}_n - \mathbf{m})$  where  $\mathbf{M} \in \mathcal{H}^{d \times D}$  and  $\mathbf{M}\mathbf{M}^H = \mathbf{I}_d$

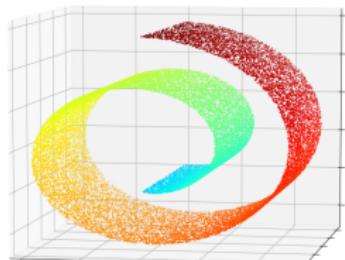
# Principal component analysis (PCA) II

- PCA is **affine**, and aims at maximizing variance (entropy, if data is Gaussian) in low dimensional space
- PCA is **parametric**, i.e., learns a function  $f : \mathcal{H}^D \rightarrow \mathcal{H}^d$  that can map new (unseen) points  $\mathbf{x}_n$  to low-dimensional space

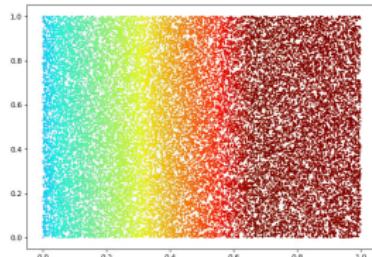
$$\mathbf{y}_n = f_{\theta}(\mathbf{x}_n) = \mathbf{M}(\mathbf{x}_n - \mathbf{m})$$

with function parameters  $\theta = \{\mathbf{M}, \mathbf{m}\}$

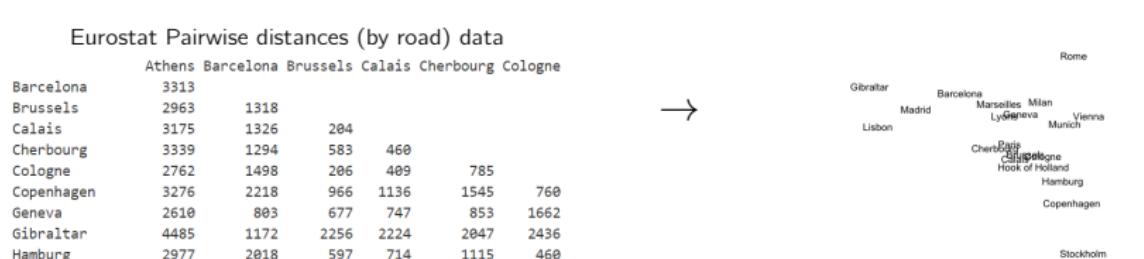
Key limitation: for most manifolds, **no affine transform exist**



???



# Multi-dimensional scaling (MDS) I



- MDS assigns coordinates to points for which **pairwise distances** are known
- Generate fictitious coordinates in an arbitrary (Euclidean) space that approximate the given pairwise distances
- Solution is **nonunique** up to a global translation and a unitary transform (rotation)

**Nonparametric:** generates embedding  $\mathcal{Y}$ , but no explicit mapping

# Multi-dimensional scaling (MDS) II

- Goal: Find low-dimensional embedding that preserves all **pairwise distances**, i.e.,  $\|\mathbf{x}_i - \mathbf{x}_j\| \approx \|\mathbf{y}_i - \mathbf{y}_j\|$
- Metric MDS [1] minimizes the following cost function:

$$L(\{\mathbf{y}_n\}_{n=1}^N) = \frac{2}{N^2 - N} \sum_{i < j} (\|\mathbf{x}_i - \mathbf{x}_j\| - \|\mathbf{y}_i - \mathbf{y}_j\|)^2$$

- One typically includes a constraint that embedding is zero-mean

The MDS cost function is nonconvex but embeddings can be computed approximately and efficiently using gradient descent

[1] J. B. Kruskal, "Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis," Psychometrika, Mar. 1964

# Sammon mapping (SM)

Metric MDS tries to match all pairwise distances between nearby and also far away points, which is **often infeasible**

- Sammon's mapping attempts to preserve only **small pairwise distances** → manifolds should be **locally** Euclidean
- Sammon mapping [1] minimizes the following cost function:

$$L(\{\mathbf{y}_n\}_{n=1}^N) = \frac{2}{N^2 - N} \sum_{n < m} \frac{1}{\|\mathbf{x}_n - \mathbf{x}_m\|} (\|\mathbf{x}_n - \mathbf{x}_m\| - \|\mathbf{y}_n - \mathbf{y}_m\|)^2$$

- Pairwise distances for which  $\|\mathbf{x}_n - \mathbf{x}_m\|$  is large are less important; other de-weighting functions are possible

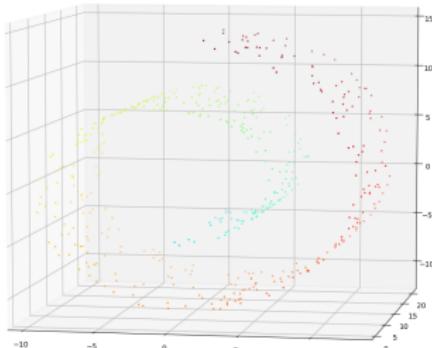
The SM cost function is nonconvex but solutions can be found approximately and efficiently using (projected) gradient descent

[1] J. W. Sammon, "A nonlinear mapping for data structure analysis," IEEE Trans. Comput., May 1969

# Isomap

Isomap [1] tries to recover the manifold structure using a graph

- Construct a graph of “neighbor” samples (based on distance threshold)
- Use pairwise distances on graph to **approximate the geodesic distances** on the manifold
- Use MDS on the geodesic distances  $D_{i,j}$  to assign Euclidean coordinates

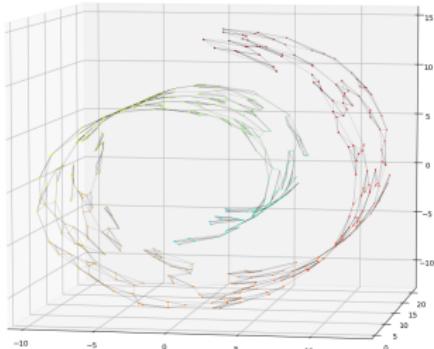


[1] J. B. Tenenbaum, V. de Silva and J. C. Langford, “A Global Geometric Framework for Nonlinear Dimensionality Reduction,” Science, 2000.

# Isomap

Isomap [1] tries to recover the manifold structure using a graph

- Construct a graph of “neighbor” samples (based on distance threshold)
- Use pairwise distances on graph to **approximate the geodesic distances** on the manifold
- Use MDS on the geodesic distances  $D_{i,j}$  to assign Euclidean coordinates

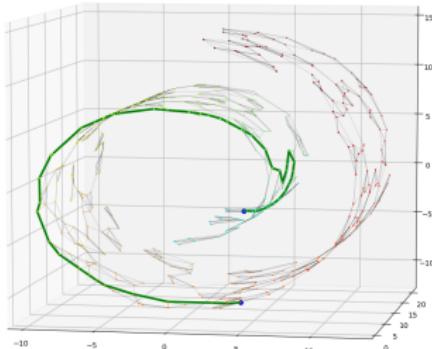


[1] J. B. Tenenbaum, V. de Silva and J. C. Langford, “A Global Geometric Framework for Nonlinear Dimensionality Reduction,” Science, 2000.

# Isomap

Isomap [1] tries to recover the manifold structure using a graph

- Construct a graph of “neighbor” samples (based on distance threshold)
- Use pairwise distances on graph to **approximate the geodesic distances** on the manifold
- Use MDS on the geodesic distances  $D_{i,j}$  to assign Euclidean coordinates



[1] J. B. Tenenbaum, V. de Silva and J. C. Langford, “A Global Geometric Framework for Nonlinear Dimensionality Reduction,” Science, 2000.

# $t$ -distributed stochastic neighbor embedding ( $t$ -SNE)

- $t$ -SNE [1] is a widespread dimensionality reduction method used for data visualization (mostly 2D embeddings)
- Learns a low-dimensional representation by minimizing the KL divergence between the distributions of original pairwise similarities, and the low-dimensional pairwise similarities:

$$(t\text{-SNE}) \quad \text{minimize } \sum_i \sum_j P_{i,j} \log \frac{P_{i,j}}{Q_{i,j}},$$

$\{P_{i,j}\}$  original probabilities with Gaussian kernels

$\{Q_{i,j}\}$  representation probabilities with Student-t kernels

Solution can approximately be found via gradient descent

[1] L. J. P. van der Maaten and G. E. Hinton, "Visualizing High-Dimensional Data Using t-SNE," J. Machine Learning Research , Nov. 2008

## Other dimensionality reduction methods

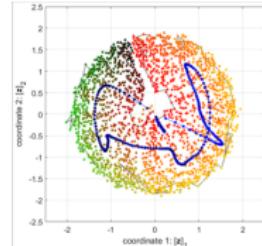
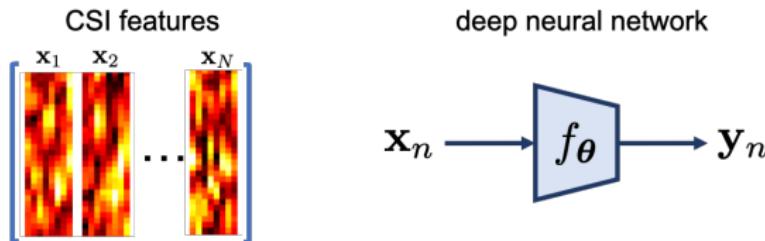
- **Laplacian eigenmaps:** Spectral DR applied to graph Laplacian based on k-NN or heat kernel
- **UMAP:** Related to *t*-SNE but avoids scaling issues to reduce complexity for (stochastic) gradient descent
- **Diffusion maps:** Spectral DR applied to diffusion distance graph simulating a Markov random walk between samples
- **Maximum variance unfolding:** Spectral DR applied to feature matrix from optimization problem with local isometry constraint

All of the methods so far are based on **pairwise distances** in  $\mathcal{H}^D$

## **Part 3:**

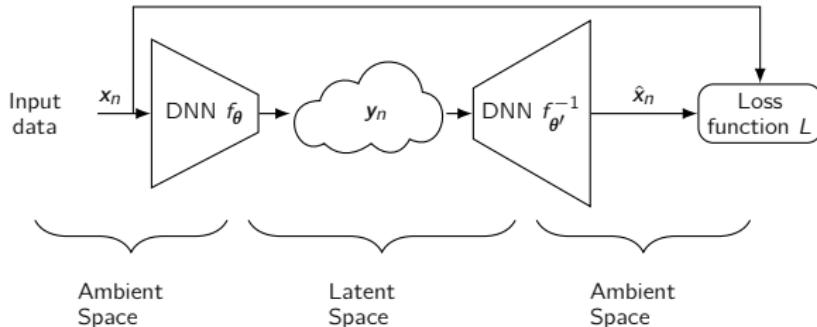
Neural-network-based  
dimensionality reduction

# Why neural networks?



- Deep neural networks are excellent function approximators
- Deep neural networks can be trained efficiently from very large datasets using stochastic gradient descent
- **Hope 1:** We can train a neural network  $\mathbf{y}_n = f_{\theta}(\mathbf{x}_n)$  that performs parametric dimensionality reduction
- **Hope 2:** We can train such neural networks in a self-supervised manner, without labels or ground-truth location information

# Autoencoders



- Autoencoders are **parametric** and learn two functions:
  - Encoder function  $\mathbf{y}_n = f_\theta(\mathbf{x}_n)$
  - Decoder function  $\hat{\mathbf{x}}_n = f_{\theta'}^{-1}(\mathbf{y}_n)$
- Goal: learn weights  $\{\theta, \theta'\}$  of two functions that minimize **pairwise** reconstruction error  $\|\hat{\mathbf{x}}_n - \mathbf{x}_n\|$

$$L(\theta, \theta') = \frac{1}{N} \sum_{n=1}^N \|\mathbf{x}_n - f_{\theta'}^{-1}(f_\theta(\mathbf{x}_n))\|^2$$

# How to constrain the geometry in the latent space?

Embeddings of autoencoders do, in general,  
**not preserve local Euclidean geometry**

- Recall: Sammon's mapping preserves **small pairwise distances**

$$L(\{\mathbf{y}_n\}_{n=1}^N) = \frac{2}{N^2 - N} \sum_{n < m} \frac{1}{\|\mathbf{x}_n - \mathbf{x}_m\|} (\|\mathbf{x}_n - \mathbf{x}_m\| - \|\mathbf{y}_n - \mathbf{y}_m\|)^2$$

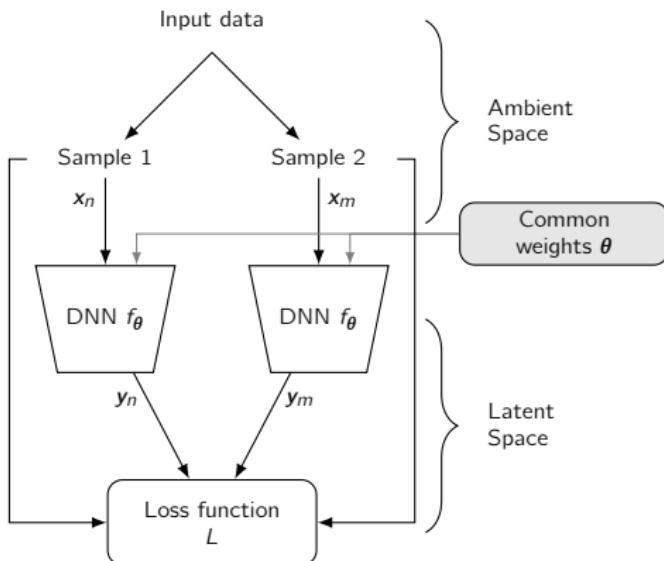
- Instead of computing nonparametric embedding  $\{\mathbf{y}_n\}_{n=1}^N$ , directly learn function  $\mathbf{y}_n = f_\theta(\mathbf{x})$  that performs DR:

$$L(\boldsymbol{\theta}) = \frac{2}{N^2 - N} \sum_{n < m} \frac{1}{\|\mathbf{x}_n - \mathbf{x}_m\|} (\|\mathbf{x}_n - \mathbf{x}_m\| - \|f_\theta(\mathbf{x}_n) - f_\theta(\mathbf{x}_m)\|)^2$$

# “Siamese” networks: parametric Sammon-type Mapping

- Minimizing  $L(\theta)$  yields structure of a “Siamese” network [1]:

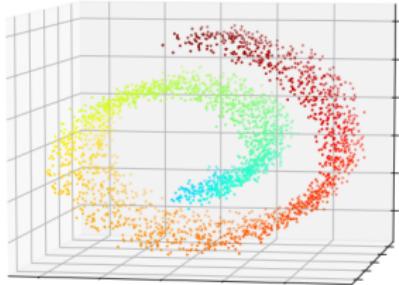
$$L(\theta) = \frac{2}{N^2 - N} \sum_{n < m} \frac{1}{\|\mathbf{x}_n - \mathbf{x}_m\|} (\|\mathbf{x}_n - \mathbf{x}_m\| - \|f_\theta(\mathbf{x}_n) - f_\theta(\mathbf{x}_m)\|)^2$$



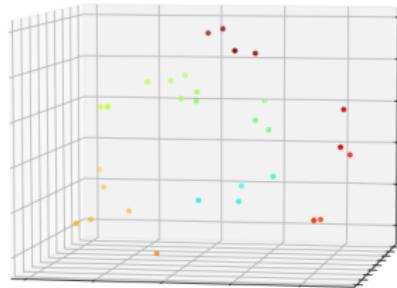
[1] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, “Signature verification using a “siamese” time delay neural network,” Advances in Neural Information Processing Systems (NIPS), 1994

# The issues with selecting a distance I

- The cost functions considered so far all require the evaluation of Euclidean distances  $\|\mathbf{x} - \mathbf{x}'\|$  in  $\mathcal{H}^D$



3000 samples in  $\mathbb{R}^3$



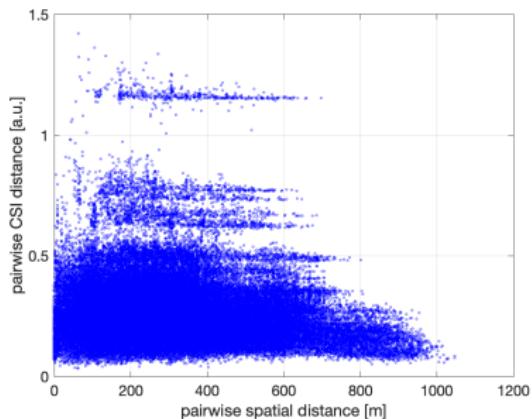
30 samples in  $\mathbb{R}^3$

## Issue 1: Curse of dimensionality

- The number of samples required to accurately describe a manifold increases exponentially with the dimension of  $\mathcal{H}^D$
- In high dimensions, **pairwise distance is often dominated by noise** and not by proximity

# The issues with selecting a distance II

- Small-scale fading effects and hardware impairments affect CS → different CSI even from same Tx location
- It is **not obvious** what distance to use to measure CSI dissimilarity



## Issue 2: Euclidean distance of CSI vectors is unreliable

- Two CSI samples from nearby Tx locations may **not** have small Euclidean distance between resulting CSI vectors

Both of these issues can be addressed!

# Distance Learning

# Distance learning using triplets

- Inspired by an approach to learn a Mahalanobis distance [1], similar to FaceNet [2]
- Assumption:
  - The distance  $d_{\text{CSI}}$  belongs to a (known) **parametric family of functions**
  - We have a **training set  $\mathcal{T}_T$  of sample triplets**  $(x_i, x_j, x_k)$  for which we know that  $d_{\text{CSI}}$  should fulfill

$$d_{\text{CSI}}(x_i, x_j) \leq d_{\text{CSI}}(x_i, x_k),$$

where  $i$  is an anchor,  $j$  is a near sample, and  $k$  a far sample

- [1] C. Shen, J. Kim, L. Wang, and A. Hengel, "Positive semidefinite metric learning with boosting," Proc. Advances in Neural Information Processing Systems, 2009.
- [2] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2015.

# DNN-based distance learning

DNN-based parametric approach with trainable parameters  $\theta$



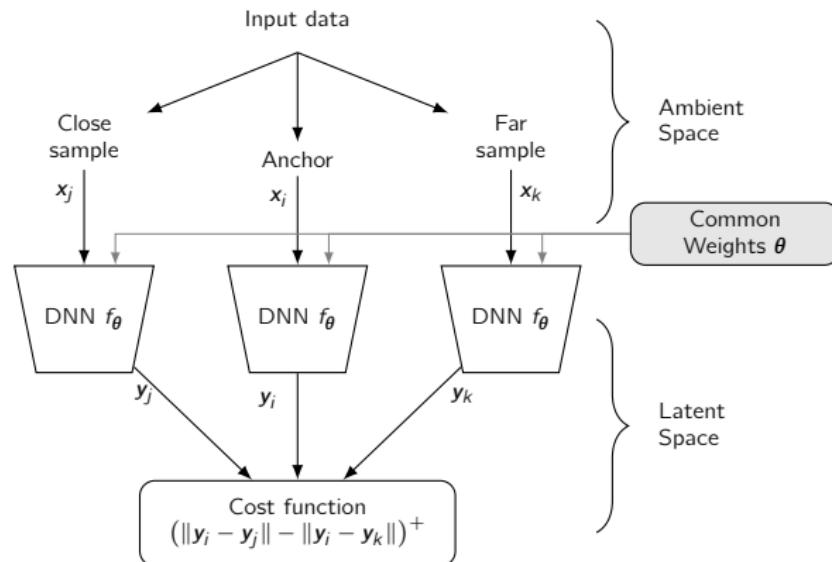
- Triplets should fulfill:  $\|f_\theta(\mathbf{x}_i) - f_\theta(\mathbf{x}_j)\| \leq \|f_\theta(\mathbf{x}_i) - f_\theta(\mathbf{x}_k)\|$
- Learn  $d_{\text{CSI}}$  by optimization over  $\theta$  of a **cost function that promotes the inequality**:

$$\min_{\theta} \sum_{(x_i, x_j, x_k) \in \mathcal{T}_T} \left( \|f_\theta(x_i) - f_\theta(x_j)\| - \|f_\theta(x_i) - f_\theta(x_k)\| \right)^+$$

Joint distance learning and dimensionality reduction

# Triplet networks

- This defines a triplet network:



- The cost function involves **only norms** in  $\mathcal{H}^d$

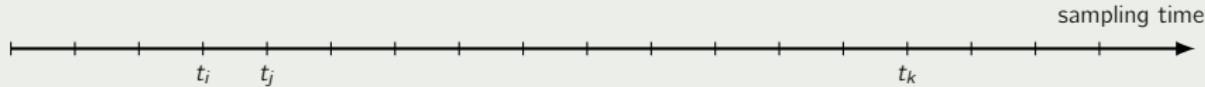
# How can we determine the triplets in practice?

## ■ Successive CSI samples are not independent

- The speed at which the physical parameters of the system change is bounded: the **meaningful** latent parameters should change continuously in time
- Select sample triplets  $(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k)$  acquired at time instants  $t_i, t_j, t_k$ , where  $|t_i - t_j| \leq T \leq |t_i - t_k|$  with coherence time  $T$

### Distance Learning using short-term CSI time correlation [1]

- Almost successive samples  $i$  and  $j$  should have smaller distances than to a randomly picked sample  $k$  in the dataset:  
 $d_{\text{CSI}}(\mathbf{x}_i, \mathbf{x}_j) \leq d_{\text{CSI}}(\mathbf{x}_i, \mathbf{x}_k)$

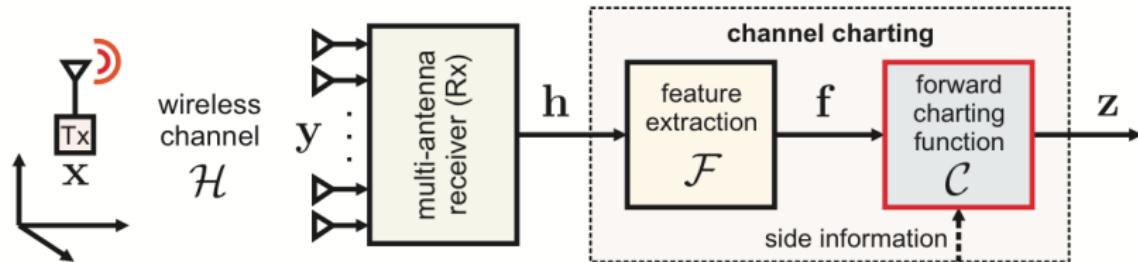


[1] P. Ferrand, A. Decurninge, L. Garcia Ordoñez and M. Guillaud, "Triplet-Based Wireless Channel Charting: Architecture and Experiments," IEEE JSAC, Aug. 2021.

# **Part 4:**

## CC pipeline and feature design

# Details of a CC pipeline

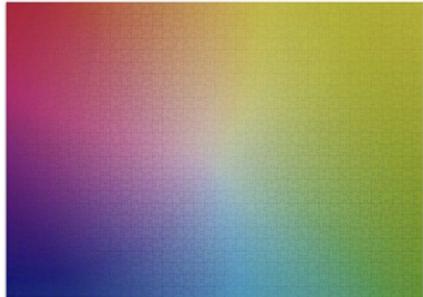


- 1 UEs transmit data to the multi-antenna receiver (Rx)
- 2 Rx estimates the wireless channel for data detection
- 3 CSI is recycled → **CSI feature extraction**
- 4 Perform dimensionality reduction for channel charting

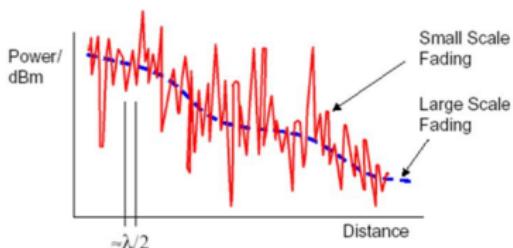
Requires no true location information → **self-supervised**

# Feature engineering

In principle, one could learn channel chart directly from CSI  
→ astronomically large data set and excessive training complexity



- CSI features corresponding to nearby locations must be “sufficiently similar” to recover local geometry efficiently



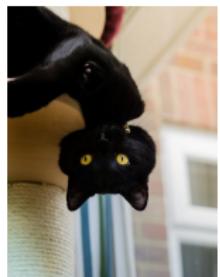
- Suitable CSI features are robust to small-scale fading, noise, and system/hardware impairments
- Suitable CSI features capture large-scale fading properties

# Role of the features

Features are functions of the input samples which should

- **Remove (=be invariant to) spurious transformations** of the data due to sampling, noise, etc.
- Preserve the relevant part of the signal

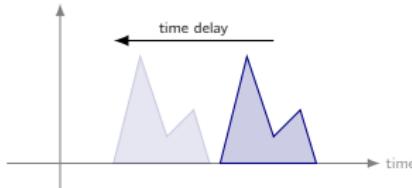
Example: rotation is a spurious transformation for the purpose of object classification



Expert-designed features have the advantage (e.g. over data augmentation) of **reducing complexity and dimensionality** of the self-supervised DR problem

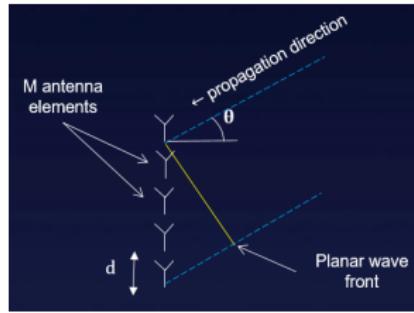
# Common impairments in CSI estimation

- Small-scale fading (e.g., caused by foliage or moving objects) can cause arbitrary phase shifts at wavelength-scale
- Some hardware impairments can make CSI samples appear more dissimilar than they actually are:
  - **multiplicative impairments** (gain+phase shift) per antenna port, due to RF components manufacturing variations
  - **clock and frequency offsets, jitter** between the Tx and Rx
  - **imperfect time synchronization**



Solution: Extract **CSI features** that are resilient to system and hardware impairments!

# Example: simple CSI feature extraction pipeline

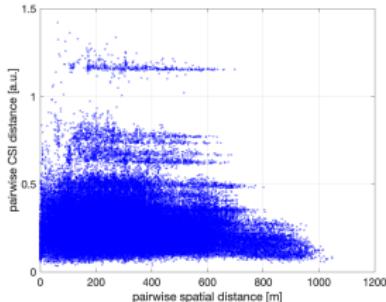


Assume a  $B$ -antenna BS and narrowband propagation

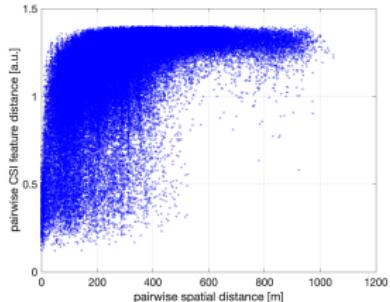
$$\mathbf{h}_n = \sum_{\ell=0}^{L-1} \alpha_\ell \mathbf{a}(\omega_\ell)$$
$$\mathbf{a}(\omega) = [e^{j0\omega}, e^{j1\omega}, \dots, e^{j(B-1)\omega}]^T$$

where  $\omega_\ell = 2\pi \frac{d}{\lambda} \cos(\theta_\ell)$

- Beamspace transform  $\hat{\mathbf{h}}_n = \mathbf{F}\mathbf{h}_n$  with DFT matrix  $\mathbf{F}$
- Remove phase shifts and global scale:  $\mathbf{f}_n = |\hat{\mathbf{h}}_n| / \|\hat{\mathbf{h}}_n\|$



CSI  
features  
→



# More CSI features I

- Time, frequency, and/or antenna-domain covariance matrix: robust to phase shifts, statistical averages [1]
- Frequency-domain autocorrelation: invariant to common phase and time synchronization shifts [2]
- Angle-delay-power (ADP) profile: yields sparse angular and delay-domain amplitude representation [3]

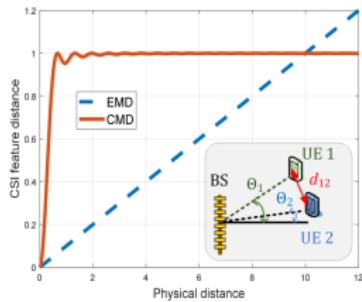
[1] C. Studer, S. Medjkouh, E. Gönültaş, T. Goldstein, and O. Tirkkonen, "Channel Charting: Locating Users within the Radio Environment using Channel State Information," IEEE Access, 2018

[2] P. Ferrand, A. Decurninge and M. Guillaud, "DNN-based Localization from Channel Estimates: Feature Design and Experimental Results," IEEE Globecom, 2020

[3] P. Stephan, F. Euchner, S. Ten Brink, "Angle-delay profile-based and timestamp-aided dissimilarity metrics for channel charting," IEEE T-COM, 2024

# More CSI features II

- Superresolution angle-delay profile (recover multi-path components) with Earth-mover distance: improved accuracy [1]
- Feature learning: includes tuning parameters into hand-crafted CSI feature extraction pipelines, data-driven parameter tuning, can combine several CSI measurements over time [2]



- [1] H. Al-Tous, P. Kazemi, C. Studer, O. Tirkkonen, "Channel Charting with Angle-Delay-Power-Profile Features and Earth-Mover Distance," Proc. Asilomar Conference, 2022
- [2] E. Gonultas, S. Taner, H. Huang, C. Studer, "Feature Learning for Neural-Network-Based Positioning with Channel State Information," Asilomar, 2021

# **Part 5:**

## Performance metrics

# Evaluating the performance of dimensionality reduction

How can we measure the performance of an embedding?

- Dimensionality reduction is agnostic to labels → **classification performance cannot be used** as a performance metric
- There is typically no ground truth available for the embedded data points → **mean-square error (MSE) cannot be used**

To evaluate the performance of dimensionality reduction, we need to specify measures of **distance or dissimilarity** on both  $\mathcal{X}$  and  $\mathcal{Y}$

- A common distance (or dissimilarity) measure is the **Euclidean distance**  $d(\mathbf{u}, \mathbf{u}') = \|\mathbf{u} - \mathbf{u}'\|$  between two points  $\mathbf{u}, \mathbf{u}' \in \mathcal{H}^K$

# The setup to measure neighborhood consistency



- Consider two abstract sets of  $N$  data-points:  $\{\mathbf{u}_n\}_{n=1}^N$  in original space and  $\{\mathbf{v}_n\}_{n=1}^N$  in representation space (embedding)
- Assume a relationship between pairs of points  $\mathbf{u}_n \sim \mathbf{v}_n$
- Two measures based on neighborhood ranking:
  - **Continuity (CT):** how similar is ranking of nearest- $K$  points in original space to ranking of points in representation space
  - **Trustworthiness (TW):** how similar is ranking of nearest- $K$  points of embedding in original space
  - Both metrics in  $[0, 1]$  and optimal if  $CT = TW = 1$

# Continuity (CT)



**Continuity** is penalized when one of the  $K$ -neighbors in original space is far away (dissimilar) in representation space

- Point-wise continuity for  $K$  neighbors defined as

$$CT_i(K) = 1 - \frac{1}{K(2N-3K-1)} \sum_{j \in \mathcal{V}_K(u_i)} (\hat{r}(i,j) - K)$$

where  $\mathcal{V}_K(u_i)$  is the  $K$ -neighborhood of  $\mathbf{u}_i$  and  $\hat{r}(i,j)$  the number of points more similar to  $\mathbf{v}_i$  than  $\mathbf{v}_j$  to  $\mathbf{v}_i$

- The (global) continuity is  $CT(K) = \frac{1}{N} \sum_{i=1}^N CT_i(K)$

# Trustworthiness (TW)



**Trustworthiness** is penalized when one of the  $K$ -neighbors in the representation space is far away (dissimilar) in the original space

- Point-wise trustworthiness for  $K$  neighbors defined as

$$TW_i(K) = 1 - \frac{1}{K(2N-3K-1)} \sum_{j \in \mathcal{U}_K(u_i)} (r(i,j) - K)$$

with the set  $\mathcal{U}_K(u_i)$  of false neighbors in the  $K$ -neighborhood of  $\mathbf{u}_i$  and  $r(i,j)$  the number of representations closer to  $\mathbf{u}_i$  than  $\mathbf{v}_j$

- The (global) trustworthiness is  $TW(K) = \frac{1}{N} \sum_{i=1}^N TW_i(K)$

# Limitations of CT and TW

- $CT$  and  $TW$  are based on **distances in  $\mathcal{H}^D$**
- Both metrics in  $[0, 1]$  and optimal if  $CT = TW = 1$
- These are local metrics, **oblivious to global errors**



Example: almost all points in the dataset have the same  $K$  neighbors in the ambient and latent space:  $CT \approx 1$  and  $TW \approx 1$

## Kruskal stress [1]

- Kruskal stress (KS) compares pairwise distances:

$$KS = \min_{\beta} \sqrt{\frac{\sum_{n,m} (\delta_{n,m} - \beta d_{n,m})^2}{\sum_{n,m} \delta_{n,m}^2}}$$

with  $\delta_{n,m} = \|\mathbf{u}_n - \mathbf{u}_m\|$  and  $d_{n,m} = \|\mathbf{v}_n - \mathbf{v}_m\|$

- Parameter  $\beta$  is selected to minimize  $KS$ ; this takes into account possible differences in global scale
- Kruskal stress is invariant to rotation and global shifts
- Kruskal stress in  $[0, 1]$  and optimal if  $KS = 0$

[1] J. B. Kruskal, "Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis," Psychometrika, Mar. 1964

## Rajsiki distance [1]

- **Rajsiki distance (RD)** measures mutual information between quantized pairwise distances in original space and embedding:

$$RD = 1 - \frac{I(V; Q)}{H(V, Q)}.$$

where  $V$  and  $Q$  are two discrete random variables and

- $I(V; Q)$  is the mutual information between  $V$  and  $Q$
- $H(V, Q)$  is the joint entropy of  $V$  and  $Q$
- RD is in range  $[0, 1]$  and optimal if  $RD = 0$
- Pairwise distances in original space and embedding must be quantized to calculate the joint probability mass  $P(V, Q)$

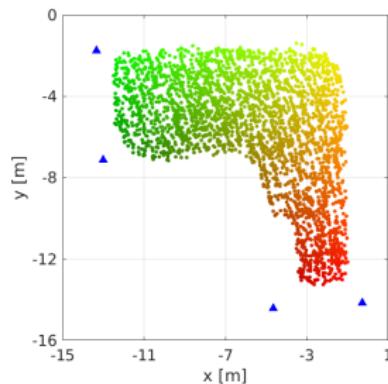
[1] C. Rajsiki, "A metric space of discrete probability distributions," Inf. Control, vol. 4, no. 4, pp. 371–377, 1961.

# **Part 6:**

## Experimental results

# Indoor CSI dataset

- DICHASUS indoor factory CSI dataset [1]:



- Four D-MIMO access points with  $B = 8$  antennas each
- 1.272 GHz carrier frequency
- 50 MHz bandwidth and 1024 subcarriers
- $N = 16k$  positions

- CSI feature extraction pipeline:

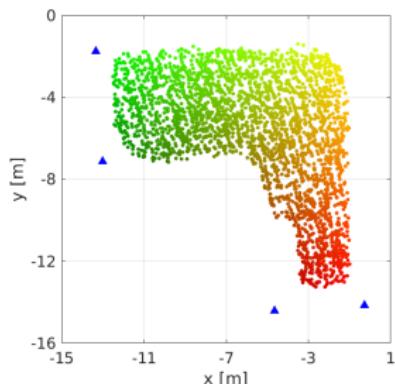
- Antenna-subcarrier CSI estimates  $\mathbf{H}_n \in \mathbb{C}^{32 \times 1024}$  transformed into delay domain  $\rightarrow$  only 8 first taps kept:  $\mathbf{f}_n \in \mathbb{C}^{256}$
- Entry-wise absolute value and normalized to unit norm

[1] F. Euchner and M. Gauger, "CSI Dataset dichasus-cf0x: Distributed Antenna Setup in Industrial Environment, Day 1", <https://doi.org/10.18419/darus-2854>, DaRUS, V2

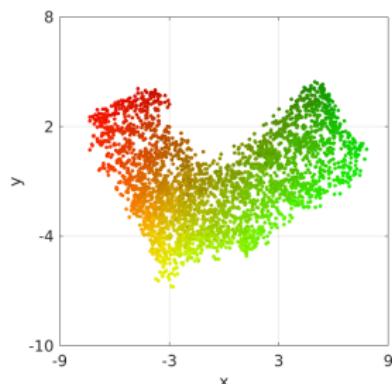
# Neural network architecture and results

- 6-layer fully-connected neural network with ReLU activations (except last layer, which is linear) to implement  $f_\theta$
- Training with time-stamp-based triplet loss and 20% of the samples ( $N = 3356$ )

ground-truth positions



channel chart (test set)



$TW = 0.989$ ,  $CT = 0.990$ ,  $KS = 0.179$ , and  $RD = 0.769$

# Outdoor CSI dataset



- Rooftop antenna array and pedestrian UE (both commercial 4G hardware)
- 64-element rectangular array ( $4 \times 8 \times 2$  polarizations)
- 20 MHz channel bandwidth in the 3.5 GHz band ( $\lambda = 8$  cm)
- Uplink CSI recorded at BTS, 200 samples/second
- Timestamps  $t_i$  and GNSS locations recorded at the UE

# Key dataset and neural network characteristics

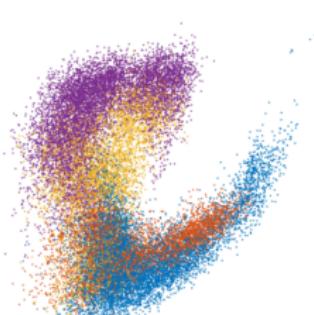


- CSI data in  $\mathbb{C}^{1843^2}$  (64 antenna elements  $\times$  288 subcarriers)
- Approximately 2.8 million samples collected over 4 hours
- We use a five-layer fully-connected neural network with batch normalization and ReLU to implement charting function  $f_\theta$

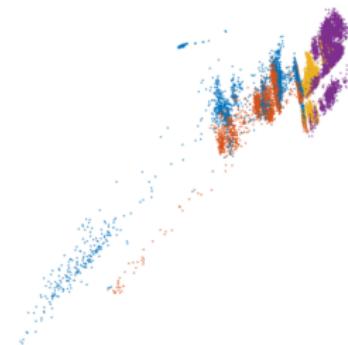
# Experimental channel charting results ( $d = 2$ )



(a) Geographic position



(b) Principal Component Analysis (PCA)



(c) Autoencoder



(d) Siamese network



(e) UMAP



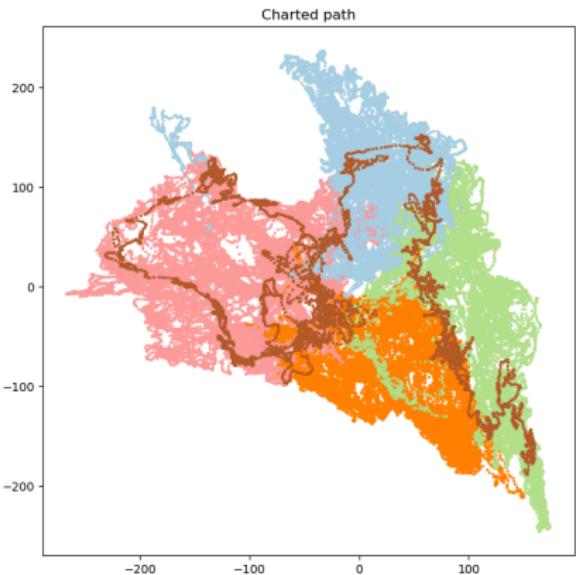
(f) Triplets network (margin cost)

# Performance summary of different CC methods

	TW	CT	KS
PCA ( $d = 2$ )	0.843	0.894	0.629
Siamese networks $d = 2$	0.846	0.910	0.494
Autoencoders $d = 2$	0.929	0.889	0.701
UMAP $d = 2$	0.951	0.923	0.526
Triplets (exp. loss) $d = 2$	0.957	0.975	0.238
Triplets (margin loss) $d = 2$	<b>0.967</b>	<b>0.977</b>	<b>0.206</b>
Siamese network $d = 5$	0.940	0.940	<b>0.496</b>
Autoencoders $d = 5$	0.972	0.968	0.877
Triplets (exp. loss) $d = 5$	0.969	0.974	0.767
Triplets (margin loss) $d = 5$	<b>0.973</b>	<b>0.976</b>	0.500
Siamese network $d = 10$	0.960	0.938	<b>0.500</b>
Autoencoders $d = 10$	0.975	0.967	0.694
Triplets (exp. loss) $d = 10$	0.978	<b>0.976</b>	0.850
Triplets (margin loss) $d = 10$	<b>0.979</b>	<b>0.976</b>	0.723

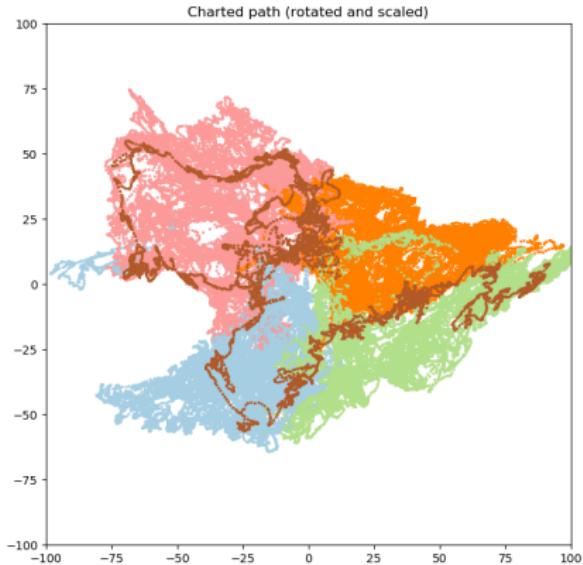
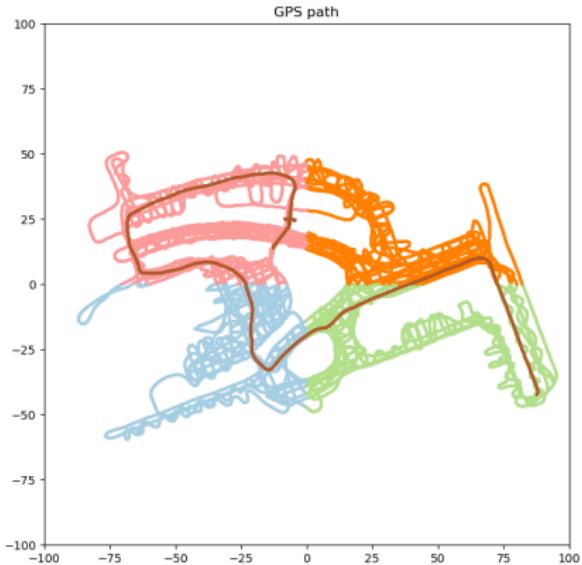
# Separate datasets for chart construction and exploitation

- NN trained with  $\approx 10^6$  CSI samples collected over 1.5 hours
- Brown dataset corresponds to a 5-minutes pedestrian trajectory recorded later



So what?

# Several datasets



The output of channel charting shows **striking similarity to the GNSS position!**

# A simple cyclic dataset

ground truth



channel chart



Triplets (margin cost)

Chart topology shows strong similarity with the ground track!

# **Part 7:**

## Applications

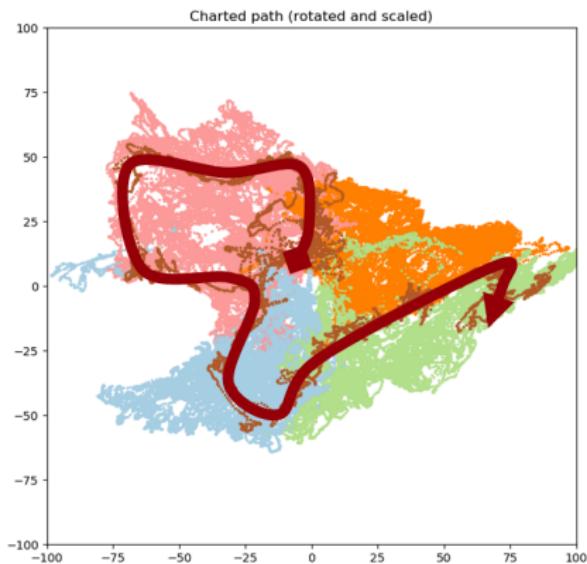
# CC as a pseudo-positioning tool

## Two-Step CC Processing

- **Long term:** Establish the DR mapping (for a given cell and scenario), based on a long-term dataset (many users/positions)
- **Real-time:** Out-of-sample processing to generate an instantaneous pseudo-position

Remember: this can all be accomplished without any ground-truth position (e.g., GNSS) information!

# Pseudo-position as a surrogate location



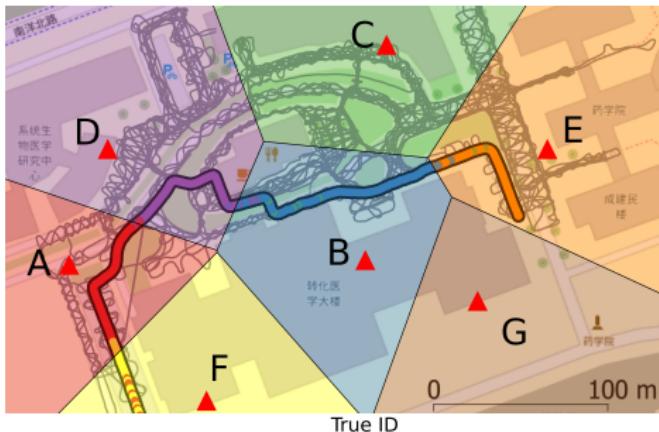
- CC-based pseudo-position as a replacement for (network-side) user position information, **without GNSS**

# Applications that benefit from channel charting

- SNR prediction (anticipate poor channel conditions) [1,4]
- Pilot assignment and reuse (determine which users get same or nonorthogonal pilots) [2]
- Beam management (beam finding and tracking for mmWave applications) [3,5,6,7,8]
- (Pseudo-)location-based beamforming [10]
- BS/cell association (find best BS to connect to) [9]

- [1] Kazemi et al., "SNR Prediction in Cellular Systems based on Channel Charting," IEEE 8rh Intl. Conf. on Commun. and Net. (ComNet), Oct. 2020
- [2] Ribeiro et al., "Channel Charting for Pilot Reuse in mMTC with Spatially Correlated MIMO Channels," GLOBECOM, Dec. 2020
- [3] Ponnada et al., "Location-Free Beam Prediction in mmWave Systems," IEEE 93rd Vehicular Technology Conference, Apr. 2021
- [4] Kazemi et al., "Channel Charting Based Beam SNR Prediction," Joint Europ. Conf. on Networks and Commun. & 6G Summit, June 2021
- [5] Ponnada et al., "Best Beam Prediction in Non-Standalone mm Wave Systems," Joint Europ. Conf. on Networks and Commun. & 6G Summit, June 2021
- [6] Kazemi et al., "Channel Charting Assisted Beam Tracking," IEEE 95th Vehicular Technology Conf., June 2022
- [7] Yassine et al., "Model-based Deep Learning for Beam Prediction based on a Channel Chart," Asilomar Conf. Nov. 2023
- [8] Agostini et al., "Channel Charting for Beam Management in Sub-THz Systems," Asilomar Conf. Nov. 2023
- [9] Ferrand et al., "Wireless Channel Charting: Theory, Practice, and Applications," IEEE Comm. Mag., 2024
- [10] Le Magoarou et al., "Channel charting based beamforming," Proc. Asilomar Conference, 2022

# Example: CC-based mmWave cell association



Estimated ID	Cell A	Cell B	Cell C	Cell D	Cell E	Cell F	Cell G
Cell A	.68	.00	.00	.03	.00	.30	.00
Cell B	.00	.92	.00	.01	.07	.00	.00
Cell C	.00	.71	.00	.00	.29	.00	.00
Cell D	.02	.09	.00	.89	.00	.00	.00
Cell E	.00	.00	.00	.00	1.00	.00	.00
Cell F	.17	.00	.00	.00	.00	.83	.00
Cell G	.00	.00	.00	.00	.00	.00	.00

Simulated dual-band network [1] with 7 mmWave cell. (Top) test user trajectory colored according to the estimated cell ID. The channel chart (not shown) is constructed from the outdoor CSI dataset shown in gray. (Bottom) confusion matrix showing the probability of the true mmWave cell ID, for each estimated cell ID

[1] P. Ferrand, M. Guillaud, C. Studer, O. Tirkkonen, "Wireless Channel Charting: Theory, Practice, and Applications," IEEE Communications Magazine, June 2023.

## Applications that *could* benefit from channel charting

- User grouping in device-to-device networks (identify nearby users)
- Handover prediction (predict user approaching cell edge)
- Rate adaptation and predictive buffering (predict UE congestion)
- Proximity detection (proximity tracing and advertisement)
- Assisting GNSS (initialize GNSS after “abduction”)
- Event labeling (e.g., link failures or strong interference/jamming)
- Context-aware communications (e.g., is the user a pedestrian or vehicle)

# Channel charting as a multi-purpose tool

- **3GPP RAN1 Study Item on AI/ML for NR Air Interface**  
(Dec. 2023) suggests to develop benchmarks for the following prototype applications:
  - Beam management
  - Positioning
  - CSI compression and prediction

Channel charting can be leveraged for all three applications!

- CC is neither the only nor the optimal learning-based approach to every CSI-related problem
- However, we see it as a **multi-purpose tool** with potential reuse across numerous applications (kill several birds with one stone!)



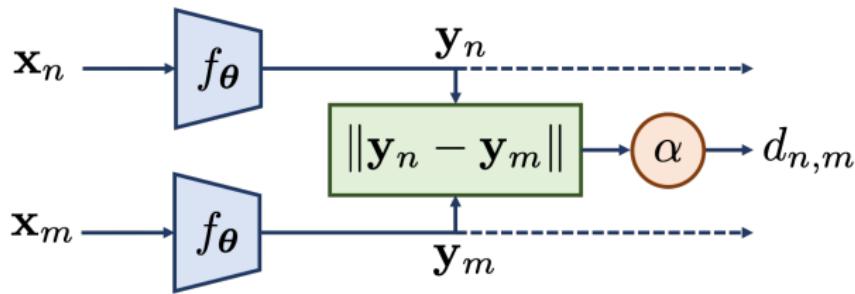
# **Part 8:**

## Extensions

# Extension I: Semi-supervised channel charting

# Semi-supervised channel charting [1]

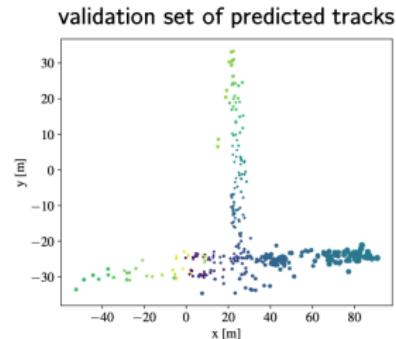
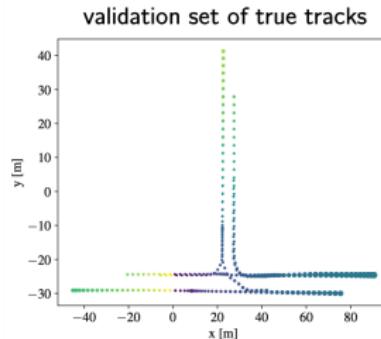
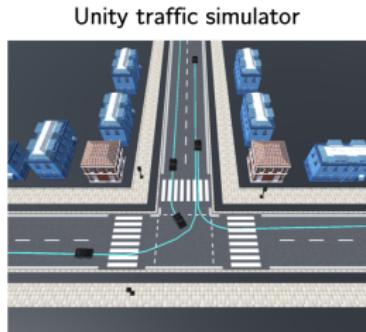
Idea: Measure a small number of anchor positions  $\mathbf{p}_n$ ,  
e.g., with GNNS information, and associated CSI features  $\mathbf{x}_n$



- Pass low-dimensional vectors  $\mathbf{y}_n$  and  $\mathbf{y}_m$  as secondary outputs
- By adding  $\sum_{n \in \mathcal{N}} \|\mathbf{y}_n - \mathbf{p}_n\|^2$  to loss  $L(\theta)$ , we can enforce a-priori measured position information  $\mathbf{p}_n$  from  $n \in \mathcal{N}$

[1] E. Lei, O. Castañeda, O. Tirkkonen, T. Goldstein, and C. Studer, "Siamese Neural Networks for Wireless Positioning and Channel Charting," 57th Annual Allerton Conf., Sep. 2019

# T-intersection simulation with very small dataset



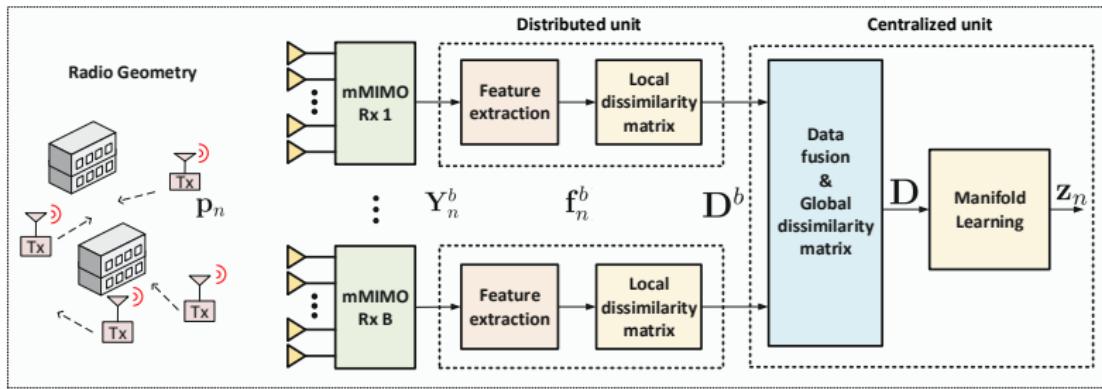
- Siamese network trained from only 20 UE traces at a T-intersection with 10% anchor point data:

	Q-LoS	Q-NLoS
MDE [m]	2.95	5.32
KS	0.188	0.471
TW ( $K = 80$ )	0.755	0.729
CT ( $K = 80$ )	0.974	0.944

- Performs well even with very small datasets
- Absolute positioning possible

# Extension II: Multi-point channel charting

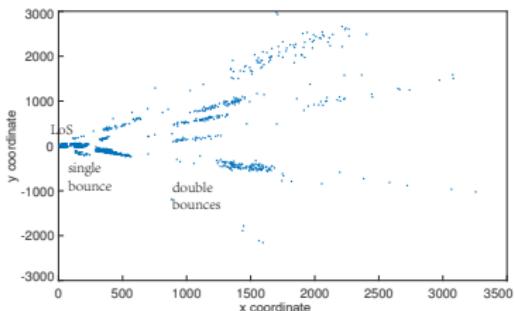
# Multi-point channel charting [1]



- 1 Several multi-antenna BSs serve all UEs distributed in area
- 2 Each BS estimates the CSI for a given UE
- 3 CSI is summarized into **channel features** at each BS
- 4 Perform data fusion and manifold learning for channel charting

[1] J. Deng, S. Medjkouh, N. Malm, O. Tirkkonen, and C. Studer, "Multipoint channel charting for wireless networks," 52nd Asilomar Conf. on Signals, Systems, and Computers, Oct. 2018

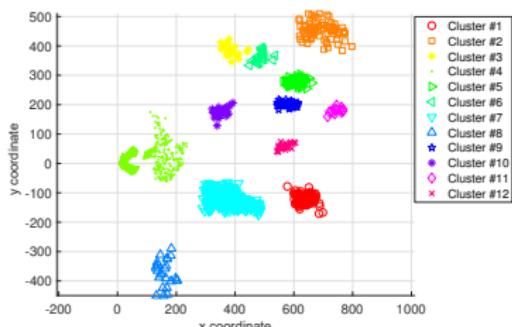
# CSI features suitable for mmWave channels



- Project direction-of-arrival (DoA) and associated power into **virtual sources**
- Nearby UE locations would have similar virtual sources

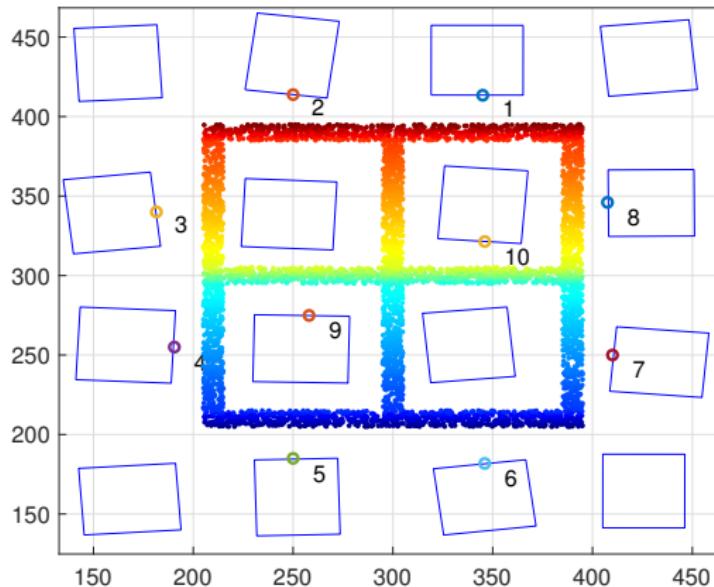
MUSIC\*: directions-of-arrival  $\{\phi_\ell\}_{\ell=1}^L$  and receive powers  $\{|\beta_\ell|^2\}_{\ell=1}^L$  of the multi-path components → distance between virtual sources

\*multiple signal classification



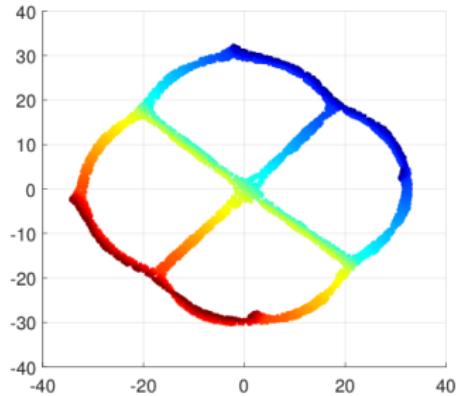
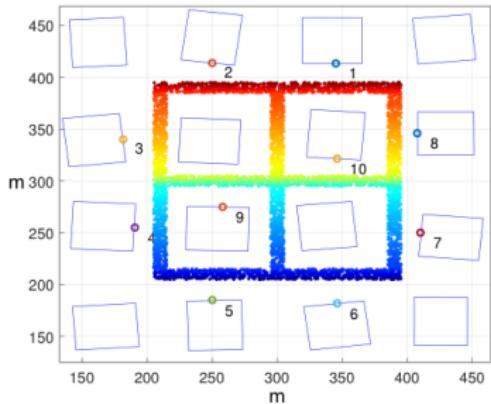
- Cluster the virtual sources after appropriate scaling
- **Dissimilarity is distance of virtual sources in clusters**

# Scenario: Manhattan grid via ray-tracing



- 10 BSs samples UE Tx from positions marked by colors
- Each BS has 64 antennas with ULA  $\lambda/2$  spacing
- 28 GHz carrier; 256 MHz bandwidth; 23 dBm UE-Tx power

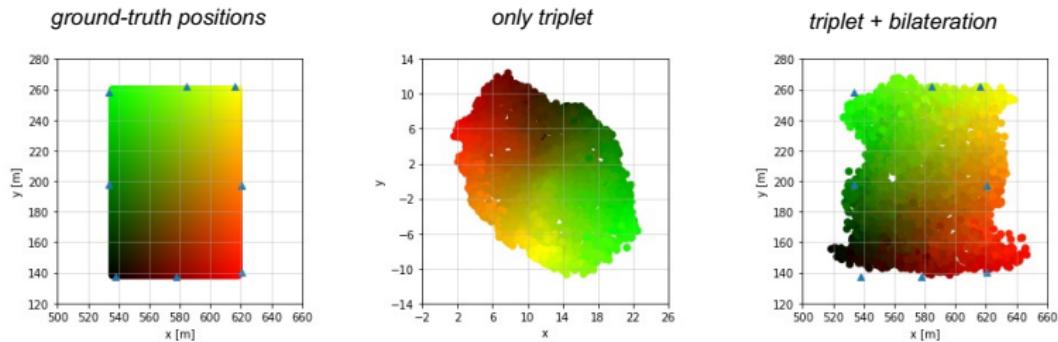
# Multipoint channel charting



Multipoint channel charting helps to unwrap the manifold  
→ **geometry is restored almost perfectly!**

Channel chart is still in **arbitrary coordinate system...**

# Absolute positioning with D-MIMO channel charting [1]



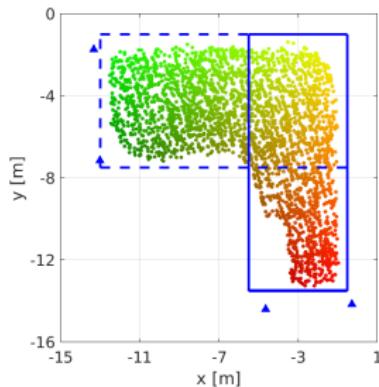
- Assume: known access point (AP) positions  $\mathbf{p}_a$ ,  $a = 1, \dots, A$
- **Bilateration loss:** if Rx power at AP  $a$  is larger than at AP  $a'$ , then user should be closer to AP  $a$

Embeds channel chart in real-world coordinates while being weakly supervised (only AP locations need to be known)

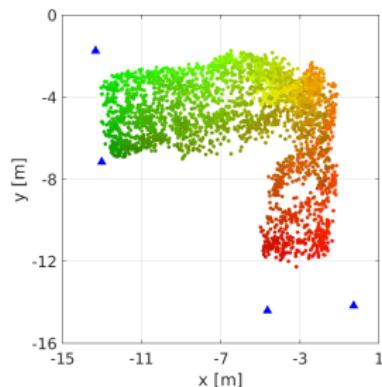
# Also works under non-LoS conditions!

- LoS zone and typical Rx power for being in that zone must be known for every AP (weakly supervised)
- Add regularizer that penalizes LoS UEs to be outside LoS zone

ground-truth positions



channel chart (test set)

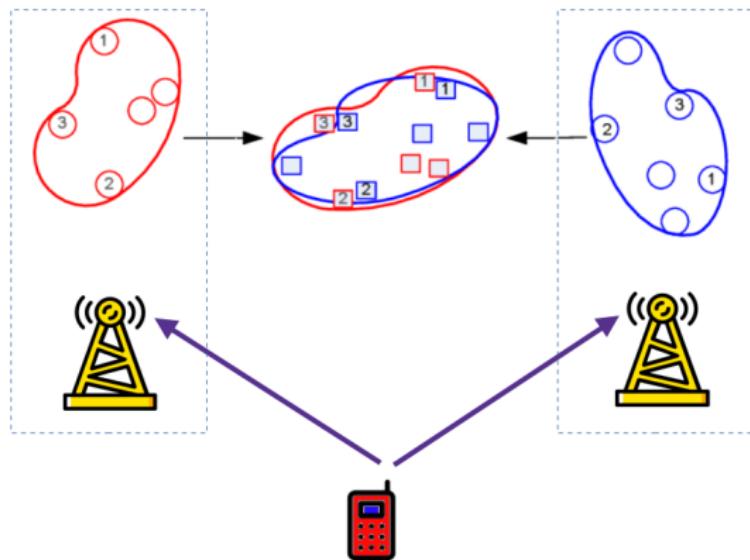


Test-set performance on DICHASUS dataset:  $TW = 0.972$ ,  $CT = 0.980$ ,  $KS = 0.192$ ,  $RD = 0.799$ , and  $MAE = 1.33$  m

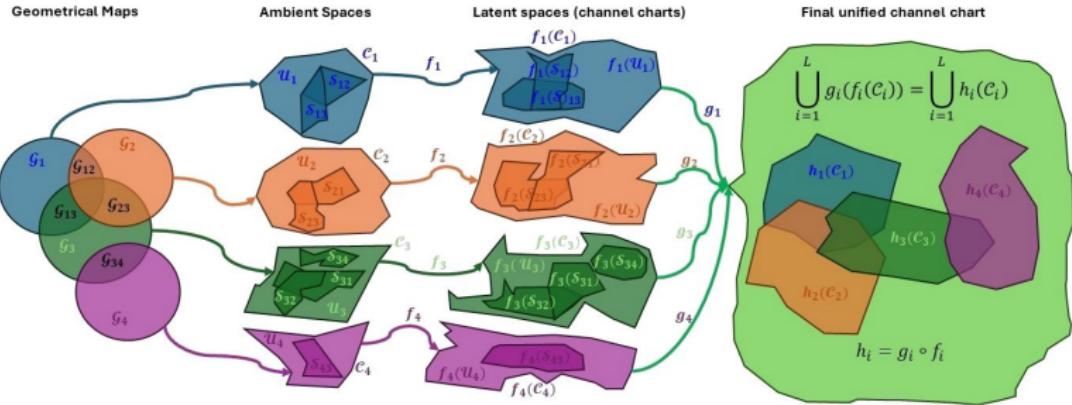
# Extension III: Multi-site CC

# Multi-Site DR

- Basestations have overlapping coverage areas
- CSI can be obtained for the same user equipment(UE) or instant at several locations on the network



# Multi-Site DR

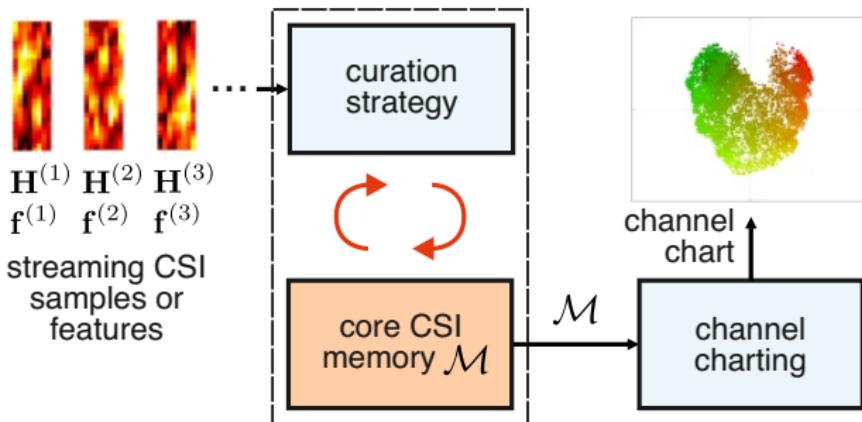


- Objective:
  - Ensure chart consistency across basestations
  - Enable distributed computation
  - Result in low communication overhead
- Leverage optimal transport [1]

[1] Y. Vindas and M. Guillaud, "Multi-Site Wireless Channel Charting Through Latent Space Alignment," IEEE SPAWC, 2024

# Extension IV: CC with streaming data

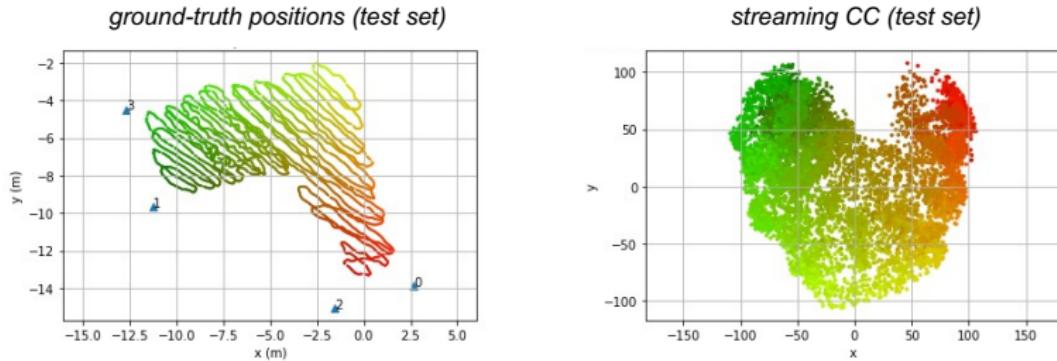
# CC with streaming data and constrained memory



- High-dimensional CSI acquired at fast rates ( $100\times$  per second)
- Extreme volumes of streaming CSI data, but limited storage and limited processing capabilities
- Architecture from [1] with core memory  $\mathcal{M}$  updated with curation strategy; CC learned from core memory

# Results for channel charting in streaming scenario

- Idea: only add new CSI feature (and replace one in the core memory) if it improves worst dissimilarity in core memory [1]
- We use DICHASUS dataset with a 1000-CSI-feature memory



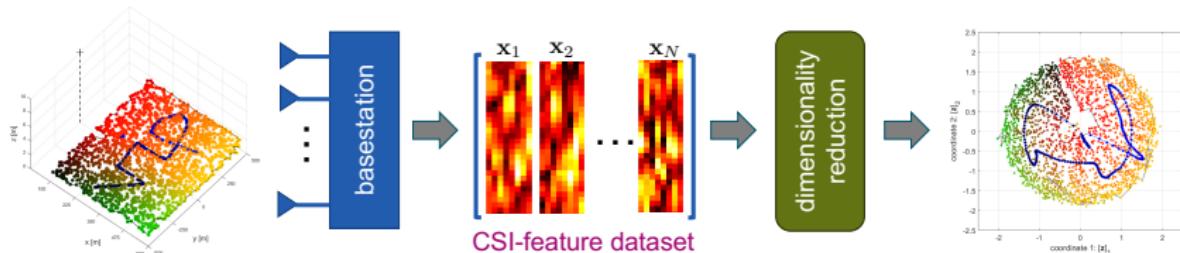
Avoids catastrophic forgetting and achieves similar channel-chart quality as learning from full CSI dataset

[1] S. Taner, M. Guillaud, O. Tirkkonen, C. Studer, "Channel Charting for Streaming CSI Data," Asilomar 2023.

## **Part 9:**

Conclusions and additional resources

# Conclusions



- CC is integrated sensing and communication since 2018! [1]
  - Probing signals are pilots; processed information is CSI
  - Self-supervised, data-driven, and scalable
- Channel charting enables
  - **self-supervised** pseudo positioning
  - **weakly-supervised absolute** positioning in D-MIMO

Channel charting can assist and improve numerous location-based tasks in wireless communication systems

[1] C. Studer et al., "Channel charting: Locating users within the radio environment using channel state information," IEEE Access, Aug. 2018

# How to get started?

## Channel Charting Resources

locating user equipment within the radio environment using channel state information

### What is Channel Charting?

Channel charting learns a mapping from channel state information (CSI) to a so-called **channel chart** in which nearby datapoints indicate nearness in real space. In other words, the learned channel chart captures the spatial geometry of the transmitting user equipments (UEs), effectively encoding relative (or logical) UE locations. Channel charting is self-supervised as the mapping from CSI to the channel chart is learned only using a database of passively collected CSI information. Such a data-driven localization approach has the advantages of being scalable and avoiding reference location information, e.g., from global navigation satellite systems (GNSS). The self-supervised nature of channel charting also avoids the need for line-of-sight (LoS) propagation conditions or (costly) measurement campaigns, while enabling the infrastructure basestations or access points to perform cognitive and predictive radio access network (RAN) tasks which are tied to UE location.

[channelcharting.github.io](https://channelcharting.github.io)

- Up-to-date and complete list of channel charting papers
- Channel charting source code
- Data sets and much more!

### Key papers to get started:

- [1] C. Studer, S. Medjkouh, E. Gönültaş, T. Goldstein, and O. Tirkkonen, "Channel Charting: Locating Users Within the Radio Environment Using Channel State Information," IEEE Access, Aug. 2018
- [2] P. Ferrand, A. Decurninge, L. Garcia Ordoñez, and M. Guillaud, "Triplet-Based Wireless Channel Charting: Architecture and Experiments," IEEE J. on Selected Areas in Communications, Aug. 2021.
- [3] P. Ferrand, M. Guillaud, C. Studer, and O. Tirkkonen, "Wireless Channel Charting: Theory, Practice, and Applications," IEEE Communications Magazine, June 2023.
- [4] L. Van Der Maaten, E. Postma, H. Van Den Herik, "Dimensionality reduction: A comparative review," Journal of Machine Learning Research, Oct. 2009

# The CHASER project

- CHASER = channel CHArting as a SERvice
- Goal: transform channel charting from research into a practical and universal service API
- Funded by CHIST-ERA
- Principal investigators:
  - O. Tirkkonen (Aalto, Finland)
  - M. Guillaud (Inria, France)
  - J. Torres-Sospedra (Uminho, Portugal)
  - A. Moreira (Uminho, Portugal)
  - C. Studer (ETH Zurich, Switzerland)
- Project website: [chaser-project.github.io/](https://chaser-project.github.io/)
-  [www.linkedin.com/company/chistera-chaser/](https://www.linkedin.com/company/chistera-chaser/)



Thank you!



[channelcharting.github.io](https://channelcharting.github.io)