# Beam SNR Prediction Using Channel Charting

Parham Kazemi, Hanan Al-Tous, Tushara Ponnada, Christoph Studer, and Olav Tirkkonen

***Abstract*—We consider a machine learning approach for beam handover in mmWave 5G New Radio systems, in which User Equipments (UEs) perform autonomous beam selection, conditioned on a used Base Station (BS) beam. We develop a network-centric approach for predicting beam Signal-to-Noise Ratio (SNR) from Channel State Information (CSI) features measured at the BS, which consists of two phases; offline and online. In the offline training phase, we construct CSI features and dimensionality-reduced Channel Charts (CC). We annotate the CCs with per-beam SNRs for different combinations of a BS beam and the corresponding best UE beam, and train models to predict SNR from CSI features for different BS/UE beam combinations. In the online phase, we predict SNRs of beam combinations not being used at the moment. We develop a low complexity out-of-sample algorithm for dimensionality reduction in the online phase. We consider $k$-nearest neighbors, Gaussian process regression, and neural network-based predictions. To evaluate the efficacy of the proposed framework, we perform simulations for a street segment with synthetically generated CSI. We investigate the complexity-accuracy trade-off for different dimensionality reduction techniques and different predictors. Our results reveal that nonlinear dimensionality reduction of CSI features with neural network prediction shows the best performance, and the performance of the best CSI-based prediction method is comparable to prediction based on using known physical location.**

***Index Terms*—Beam-management, beam SNR prediction, complexity analysis, CSI feature, dimensionality reduction techniques, Neural Network, SNR prediction.**

## I. Introduction

MILLIMETER Wave (mmWave) communication with large bandwidths and multiple antennas at the Base Station (BS) and the User Equipment (UE) are key enablers for high data rate in fifth generation (5G) and beyond mobile systems. The high path loss at mmWave bands renders beamforming a critical technology to overcome the severe attenuation and to ensure connectivity [2], [3]. Using a large number of antennas is thus a necessity in mmWave systems. Due to the short wavelength at mmWave bands, large antenna arrays that have small physical size can be implemented and narrow beams with high gains can be formed. As both the BS and the UE are capable of highly directional beamforming, a beam management procedure is of paramount importance in order to keep the beams aligned [3], [4].

User mobility in cellular networks is introduced by handover procedures. Due to high path loss in mmWave systems, frequent handover between nearby BSs as well as between beams of one BS is likely. As a result, both intra-cell beam handovers [5], among beams of one BS, and inter-cell handovers have to be considered. In 5G networks, autonomous UE beamforming is adopted, where the UE selects a beam direction both for uplink transmission and for downlink reception. The BS has no control over the UE beam, and without an exhaustive search of multiple BS-UE beam pairs, the consequences of change of a BS beam would not be clear. As a result, the BS cannot simply determine the Signal-to-Noise Ratio (SNR) of the best BS beam by measuring the SNR of different BS beams from a UE transmission on a single beam. Hence, the problem of determining the SNR of the best BS-UE beam combination is crucial for mobile 5G systems.

P. Kazemi, H. Al-Tous, T. Ponnada, and O. Tirkkonen are with Department of Information and Communications Engineering, Aalto University, Espoo, Finland. Email: parham.kazemi@aalto.fi, hanan.al-tous@aalto.fi, tushara.ponnada@aalto.fi, and olav.tirkkonen@aalto.fi

C. Studer is with the Department of Information Technology and Electrical Engineering, ETH Zurich, Switzerland. Email: studer@ethz.ch

Early results of this work appeared in [1].

### A. Beam Management

Beam management is a critical aspect of ensuring reliable communication in mmWave systems. It comprises two key procedures: initial access, which establishes a new connection between the BS and UE through beam sweeping, and beam tracking, which involves updating beam alignment for mobile UEs. Conventional approaches for beam management include exhaustive and partial beam sweeping [6]. In exhaustive search, all possible beamforming directions within a predefined angular range are considered, utilizing a predetermined codebook [3] . This requires pilot transmissions from the BS to all potential beamforming directions, resulting in significant overhead [7].

Spatial scanning for beam alignment in 5G new radio involves an exhaustive search using narrow beams at both the BS and UEs to cover the entire angular space, see [3]. Beam sweeping is accomplished through regular transmissions of synchronization signal blocks, with a 20 ms interval and a 5 ms time window for measuring the reference signal of all beams at the UE [8]. The BS transmits a reference signal over different beams, and the UE measures the signal quality and reports the measurement result and the ID of the best BS beam to the BS. When the UE has multiple antennas, an additional beam sweep over UE beams is required. Following the 5G new radio approach, the UE determines the best UE beam autonomously, and the corresponding best BS beam to the BS.

Advanced methods with low latency and reduced signaling overhead need to be developed for scenarios with high mobility. Various methods ranging from hierarchical search, extended Kalman filtering, and Machine Learning (ML) approaches have been investigated [9], [10]. To reduce the overhead of beam sweeping, hierarchical codebook designs have been proposed, which are based on a few low-resolution wide beams. Although the total sweeping time can be reduced, such methods remain time-consuming, as pilot transmissions to the low-resolution beams are required and, once the wide beam is found, the corresponding narrow beams are searched over. While such approaches significantly reduce the beam-search complexity,

the performance of beam scanning is greatly affected by the beamforming codebook. A hierarchical beam search may also be affected by noise, which leads to a trade-off between beam-search delay and accuracy [6].

### B. Data Driven Beam Management

ML algorithms have shown great potential for handling Radio Resource Management (RRM) problems [11]–[15]. Corresponding studies are ranging from channel prediction [11], [12], where Channel State Information (CSI) is predicted from past CSI to reduce pilot overhead, to directly predicting the mmWave system beams [16]. Data-driven beamforming in mmWave systems was investigated in [13], where beam directions and beamwidths as well as transmit power are simultaneously optimized. In [14], power allocation and uplink beamformer prediction is performed by a Neural Network (NN), utilizing channel reciprocity where the input feature is channel matrix. In [15], a beam alignment method is proposed where received signals are mapped to the index of the best beamformer without using any prior knowledge. This aims to enhance spectral efficiency compared to hierarchical beam search where the input feature is obtained from channel coefficients between the BS and the UE.

Useful information can be extracted from side information (out-of-band, physical location, and sensor information) for beam management purposes. In [17], a deep learning framework was proposed where sub-6-GHz channel measurements are used to train a NN for predicting blockages and the best beam in the mmWave band. In [18], [19], ML models were proposed to predict the optimal beam and cell using only the physical location information. Physical location information is used to minimize the beam search space, hence avoiding the high overhead of real-time channel feedback and decreasing the delay and consumed power. Neighbor-assisted beam search has been proposed in [20], where both the location information and the nearest neighbor beam information are used for fast beam sweeping. Sensors on vehicles, such as LiDAR, can also capture more information from the surrounding environment and assist beam management [21].

Channel Charting (CC) [22], [23] is recent machine learning framework for pseudo localization, where a Dimensionality Reduction (DR) technique is applied to the collections of massive Multiple-Input Multiple-Output (MIMO) CSI at cellular BSs. In CC, self-supervised ML techniques are used to create a radio map of the cell which preserves the neighborhood relations of UEs. Similar to the location-based beam management methods, the channel chart can be used for beam management without the need to know the physical location. In [24], we discussed the use of CC for handover prediction where the SNR of a UE from a neighboring BS is predicted based on the channel chart locations.

### C. Contribution

In this paper, we concentrate on beam SNR prediction without any side information. We focus on handover between beams of BSs in a mmWave network and predict beam SNRs based on CSI features. In an offline phase, beam-specific CSI features are extracted. We consider both dimensionality reduced and raw features. These features are annotated with SNRs of target beams, and SNR predictors are trained. In the online phase, target beam SNRs are predicted, so that a handover decision could be made. The contributions of our paper are summarized as follows:

- We consider a network-centric supervised learning framework for beam SNR prediction. Based on CSI measurements at the BS towards a given beam, we predict the SNR of using a different beam on the same BS or on a different BS.
- We devise a CSI feature for predicting the SNR at other beams assuming autonomous beamforming at the UE, without information about the UE beam at the BS.
- We remove small-scale fading from the beam prediction problem by using covariance matrices as channel features for SNR prediction. This choice leads to robust predictions, and low spatial sampling density in the training phase.
- We analyze and evaluate the spatial consistency of the CSI feature at the BS. This shows that this feature is not robust against UE beamformer changes and that a per BS beam feature is needed.
- We develop a low complexity Out-of-Sample (OoS) algorithm that requires lower computations compared to the conventional OoS algorithms to be used in the online phase.
- We analyze the complexity-accuracy trade-off of the proposed schemes. K-nearest Neighbor (KNN), Gaussian Process Regression (GPR), and NN predictors are considered, together with linear and non-linear dimensionality reduction, as well as the raw features.

### D. Notation

We adopt the following notation: matrices and vectors are set in upper and lower boldface, respectively. $(\cdot)^T$, $(\cdot)^H$, $|\cdot|$ denote the transpose, the Hermitian, the absolute value, respectively. $\mathrm{Tr}(\cdot)$ indicates the trace of a matrix. $\mathbb{E}\{\cdot\}$ denotes expectation and $\|\cdot\|_F$ is the Frobenius norm. $\mathbb{C}$ is the set of complex numbers and $\mathbb{C}^{N\times M}$ is the space of $N \times M$ matrices. $\mathbf{X}_{ij}$ is the element in the $i^{\text{th}}$ row and $j^{\text{th}}$ column of matrix $\mathbf{X}$.

### E. Paper Outline

The rest of this paper is organized as follows. In Section II, the system model, basic concepts and background are introduced. In Section III, network operation for beam SNR prediction is presented. In Section IV, OoS extension framework is discussed. In Section VI, simulation settings and numerical results are presented and discussed. Finally, conclusions are drawn in Section VII.

## II. System Model and Background

We consider a multi-BS MIMO system, in which UEs and BSs have multiple antennas following the 5G new radio standard. Beamformers are used at the BS and UE. The UE beamformer is autonomous so that the UE uses the best UE beam towards a BS beam. We consider a time division duplex system with $B$ MIMO BSs, in which each BS is equipped with

an antenna array with $M$ elements and each UE antenna array has $T$ elements. Hence, in order to find the best transmit/receive beam pair following the 5G New Radio exhaustive search principles [8], the BS has to transmit $M$ beams and the UE has to measure $M \times T$ beam pairs.

For simplicity, we assume that the number of radio frequency chains is equal to the number of antenna elements at the BSs and the UEs. We assume time division duplexing as it is the preferred mode of operation in massive MIMO systems, since it enables reciprocity between uplink and downlink channels [25].

### A. Channel and Beam Models

We consider a massive MIMO system, with channel matrix $\mathbf{H}_b^u \in \mathbb{C}^{M\times T}$ between BS $b$ and UE $u$ over a subcarrier. The channel matrix models the path loss as well as large-scale and small-scale multipath fading effects. Let $\mathbf{w}_m \in \mathbb{C}^M$ denote the beamforming vector at the BS with $m = 1, \ldots, M$ and $\mathbf{v}_t \in \mathbb{C}^T$ denote the beamformer at the UE with $t = 1, \ldots, T$. Assuming a Uniform Linear Array (ULA) at both BS and UE, a Discrete Fourier Transform (DFT)-based codebook $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_M]$ is used with beams

$$\mathbf{w}_m = \frac{1}{\sqrt{M}} \left[1, e^{-j2\pi \frac{(m-1)}{M}}, \ldots, e^{-j2\pi \frac{(M-1)(m-1)}{M}}\right]^T, \quad (1)$$

for $m = 1, \ldots, M$. A codebook containing a similar set of $T$ DFT codewords is used at the UE. Without loss of generality, the codebook can be generalized to a uniform planar array and for the case with polarization, since the uniform planer arrays allow 3D beamforming by adopting the beam in both vertical and horizontal directions [26]. We assume that wideband beams are used at both the BS and the UE, i.e., for all subcarriers, the same beam is employed.

The received signal from UE $u$ at beam $m$ of BS $b$ when UE uses beam $t$ is given by

$$y_{bmt}^u = \mathbf{w}_m^H \mathbf{H}_b^u \mathbf{v}_t s + n = h_{bmt}^u s + n, \quad (2)$$

where $h_{bmt}^u = \mathbf{w}_m^H \mathbf{H}_b^u \mathbf{v}_t$ is the effective channel of UE $u$ using beam $t$ at BS $b$ beam $m$, $s$ represents the transmitted symbol with $\mathbb{E}\{|s|^2\} = 1$ and $n$ is additive white Gaussian noise. The UE determines its best beam for transmitting towards / receiving from beam $m$ of BS $b$ as:

$$t(m) = \arg\max_t \mathbb{E}\{|h_{bmt}^u|^2\}, \quad (3)$$

where the expectation is over frequency samples. This information is known at the UE, however, the BS does not know the UE's best beam. Thus, if the BS's best beam changes, the UE's best beam may change, and this needs to be considered in the beam prediction framework.

The average received SNR at beam $m$ of BS $b$ from a transmission of UE $u$ using its best beam is given by

$$\gamma_{bm}^u = \frac{1}{\sigma^2} \mathbb{E}\{|h_{bmt}^u|_{t=t(m)}^2\}, \quad (4)$$

where $\sigma^2$ is the noise power and the expectation is taken over frequency samples and temporal samples taken from the fast fading process within a short time interval.

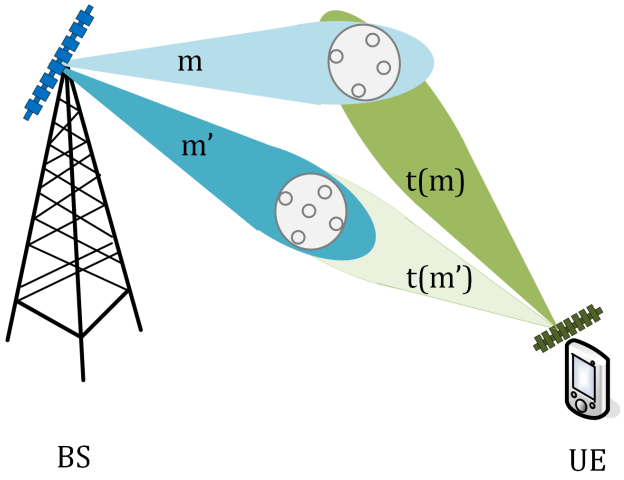



Fig. 1: The BS beams and the corresponding best UE beams.

The effective channels from UE $u$, transmitting towards BS $b$ beam $m$, measured by all BS antennas is

$$\mathbf{h}_{bt(m)}^u = \mathbf{W}^H \mathbf{H}_b^u \mathbf{v}_{t(m)}. \quad (5)$$

The measurements are conducted at the BS and UE transmissions are coordinated towards BS beam $m$. The effective channel for a transmission from a UE depends on which BS beam the UE is receiving the signal from.

Fig. 1 shows a BS and a UE, both with an antenna array. The channel between the BS and UE is created by two scattering clusters, which are represented by white circles. For each BS beam, illuminating a specific scattering cluster, there is a corresponding best UE beam. For clarity, we show only two beams. The UE beamformer is conditioned on the BS beam that the transmission is received from, as expressed in (5). If the UE autonomously selects its beamformer, the BS cannot measure and find the best beam towards the UE from an uplink transmission. If the BS were measuring the effective channel at beam $m$ when the UE uses $t(m)$, it cannot determine what the CSI would be if the UE was transmitting towards beam $\mathbf{w}_{m'}$ with $m' \neq m$. Therefore, from autonomous beamforming of the UE, a *beam mismatch* problem arises when the best UE beams $\mathbf{v}_{t(m')}$ and $\mathbf{v}_{t(m)}$ towards two BS beams may not be the same.

### B. CSI Feature and Distances

The CSI feature that we consider is a channel covariance feature. The covariance matrix is a large-scale feature that changes slowly compared to channel vectors. It is easier to estimate compared to the instantaneous CSI [27]. In [22], the authors analyzed the changes in the channel covariance based on scatterers in the radio environment and showed that it is a large-scale fading effect. We develop a feature based on the effective channel of UE towards BS $b$ beam $m$ as follows:

$$\mathbf{R}_{bm}^u = \mathbb{E}\{\mathbf{h}_{bt(m)}^u (\mathbf{h}_{bt(m)}^u)^H\}. \quad (6)$$

To compute the distance between two CSI features we use the Collinearity Matrix Distance (CMD) [28], which, for any two positive definite matrices $\mathbf{R}$ and $\mathbf{R}'$, is computed as follows:

$$d_{\mathrm{CMD}}(\mathbf{R},\mathbf{R}') = \frac{1}{2}\left\| \frac{\mathbf{R}}{\|\mathbf{R}\|_F} - \frac{\mathbf{R}'}{\|\mathbf{R}'\|_F} \right\|_F^2. \tag{7}$$

This measure is a normalized distance that reflects how similar the two matrices are. The distance ranges from zero (fully collinear, i.e., two matrices are scaled versions of each other) and becomes one (absolutely non-collinear) when two matrices are different to a large extent.

Another measure we use is the Log-Euclidean distance which is computed as follows [29]:

$$d_{\mathrm{L}}(\mathbf{R},\mathbf{R}') = \| \underbrace{\log \mathbf{R} - \log \mathbf{R}'}_{\mathbf{\Lambda}} \|_F = \sqrt{\mathrm{Tr}(\mathbf{\Lambda}\mathbf{\Lambda}^H)}. \tag{8}$$

The $\log$ indicates matrix logarithm. Generally, matrix logarithm of positive semidefinite matrix $\mathbf{A}$ is calculated by performing a Singular Value Decomposition (SVD), i.e., $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H$, $\mathbf{\Sigma}$ is a diagonal matrix and $\log(\mathbf{A}) = \mathbf{U}\,\mathrm{diag}\left([\log\sigma_1,\ldots,\log\sigma_{M'}]\right)\mathbf{V}^H$ and $\sigma_{m'} > 0$ for $m' = 1,\ldots,M'$.

### C. CC-Based on Linear Dimensionality Reduction

CC is the process of creating radio maps that preserve the neighborhood information of UEs. The basic idea of CC is that high dimensional CSI feature is heavily dependent on low-dimensional UE location. A channel chart is a map of the radio environment from collected CSI feature of massive MIMO BSs. The constructed channel chart has a neighborhood-preserving property so that nearby UEs in the physical domain are close to each other on the channel chart.

It has been shown in [22] that high dimensional CSI features depend heavily on the UE location which lies in low dimensional space. Thus, by reducing the dimensionality of the input feature, we can generate a mapping that reflects the neighborhood relation of the UEs. The CSI feature $\mathbf{x}$ can be extracted from the CSI covariance matrix which captures the large-scale properties of the wireless channel. The CC can be constructed for the CSI collected at a MIMO BS operating at any radio frequency band, both microwave and mmWave bands were considered [22], [23], [30]. In order to use off-the-shelf ML software, the complex-valued feature needs to be converted to a real-valued vector. We convert the CSI covariance matrix $\mathbf{R}$, to a vector consisting of real and imaginary values as:

$$\mathbf{x} = \left[\Re\{\mathrm{vec}(\widetilde{\mathbf{R}})\}^T, \Im\{\mathrm{vec}(\widetilde{\mathbf{R}})\}^T\right]^T, \tag{9}$$

where $\widetilde{\mathbf{R}}$ is a non-linearly transformed feature matrix. We shall consider non-linear transforms conforming to the CMD and Log-Euclidean distances, i.e., normalizing using the Frobenius norm, or taking the logarithm of the covariance matrix before vectorizing. Since the covariance matrix has a Hermitian property, lower diagonal elements are eliminated due to redundancy. Hence, the resulting CSI feature vector is an $M^2$-dimensional real-valued vector which would be referred as high dimensional due to the large number of antenna elements.

Principal Component Analysis (PCA), factor analysis, linear discriminant analysis, and Truncated SVD are examples of linear DR methods. Here, we focus on PCA. As a baseline, PCA is used as a simple method for DR [31]. In the extreme, reduction to a channel chart of 2/3 dimensions can be done. For RRM purposes, as opposed to localization, higher dimension channel charts can be considered.

PCA selects the most discriminative principal components so that the covariance of the low dimensional features is maximized. For a given CSI feature $\mathbf{x}$, the low dimensional channel chart point $\mathbf{z}$ is obtained using an optimal weighting matrix $\mathbf{G}$ as follows:

$$\mathbf{z} = \mathbf{G}^T\mathbf{x}. \tag{10}$$

Here, $\mathbf{G} = [\mathbf{g}_1,\ldots,\mathbf{g}_d]$ consists of orthonormal basis with $d$ as the dimension of the low dimensional space.

To obtain the weighting matrix $\mathbf{G}$, an optimization problem aiming to maximize the variance of the transformed data is formulated. For this purpose, we preprocess the set of CSI features $\mathbf{X} = [\mathbf{x}_1,\ldots,\mathbf{x}_U]$ by normalizing each row of $\mathbf{X}$ to have zero mean and call it $\bar{\mathbf{X}}$ and compute the empirical covariance matrix $\mathbf{\Xi} = \bar{\mathbf{X}}\bar{\mathbf{X}}^T$ of the centered features. The optimization problem associated with PCA is:

$$\arg\max_{\mathbf{G}} \ \mathrm{Tr}(\mathbf{G}^T\mathbf{\Xi}\mathbf{G}), \quad \text{subject to } \mathbf{G}\mathbf{G}^{\mathrm{T}} = \mathbf{I}_d. \tag{11}$$

The eigenvalue decomposition is applied to $\mathbf{\Xi}$ and principal components are the top-$d$ eigenvectors of $\mathbf{\Xi}$. By sorting the eigenvalues of $\mathbf{\Xi}$ in a descending order, the first $d$ eigenvectors form the transformation matrix $\mathbf{G}$. By reducing the dimensionality, a part of variability of the original high dimensional data is lost. However, it is not significant because only the small eigenvalues are discarded. The main assumption in PCA is that the high dimensional data lies on a linear embedding.

### D. CC-Based on Non-linear Dimensionality Reduction

Non-linear Dimensionality Reduction (NLDR) methods are more powerful at preserving the local neighborhood information of the data compared to linear methods such as PCA. NLDR methods aim to preserve the local structure of the data [32] in addition to global structure of the data. However, the number of selected neighbors has a crucial impact on the maintained structure and needs to be carefully selected. To construct a channel chart, a dissimilarity matrix $\mathbf{D}$ is first computed in some NLDR techniques. $\mathbf{D}$ is a square and symmetric matrix with pairwise CSI feature distances between the UEs. Then, with the help of DR techniques, such as Laplacian Eigenmaps (LE) [33] and t-Distributed Stochastic Neighbor Embedding (t-SNE) [34], a low dimensional representation of CSI features can be obtained by processing $\mathbf{D}$.

Laplacian Eigenmaps is a graph-based NLDR technique which is computationally efficient. First, a weighted graph is constructed using the neighborhood information of the dissimilarity matrix. Using the dissimilarity matrix, a weight matrix $\tilde{\mathbf{W}}$ is formed in which if node $i$ is in the $k$-nearest neighborhood of node $j$, they are assumed to be connected. The weight matrix can be constructed in two different ways.

In the first, the weight for two connected nodes $i$ and $j$ is set to a constant, i.e., $\tilde{\mathbf{W}}_{ij} = \frac{1}{k}$, while for two unconnected nodes $i$ and $j$ is set to zero, i.e., $\tilde{\mathbf{W}}_{ij} = 0$. Alternatively, the weight matrix can also be constructed by the heat kernel with temperature parameter $\sigma_T$. If the nodes $i$ and $j$ are connected, $\tilde{\mathbf{W}}_{ij} = \exp(-\frac{\mathbf{D}_{ij}}{\sigma_T})/(\sqrt{E_i E_j})$, where $E_i$ and $E_j$ are the empirical expectations computed over $k$-nearest neighbors in the data, i.e., $E_i = \mathbb{E}_x[\exp(-\mathbf{D}_{ix})]$, where $x \in k$-nearest neighbors of $i$, otherwise $\tilde{\mathbf{W}}_{ij} = 0$. For Laplacian Eigenmaps DR we find a mapping which penalizes the neighbors in the original data being far in the mapped space by formulating the objective function as:

$$\arg\min_{\mathbf{Z}} \sum_{i,j} \|\mathbf{z}_i - \mathbf{z}_j\|_2^2 \tilde{\mathbf{W}}_{ij}. \tag{12}$$

To avoid the trivial solution of all zeros, a constraint is added and the optimization problem is written in the common form as follows [33]:

$$\arg\min_{\mathbf{Z}} \operatorname{Tr}(\mathbf{Z}^T \mathbf{L} \mathbf{Z}), \quad \text{subject to } \mathbf{Z}^T \mathbf{S} \mathbf{Z} = \mathbf{I}_d. \tag{13}$$

Here, $\mathbf{Z} = [\mathbf{z}_1^T, \ldots, \mathbf{z}_U^T]^T$ are the optimization variables (the low dimension points / channel chart locations) with $\mathbf{z}_i = [\mathbf{z}_i(1), \ldots, \mathbf{z}_i(d)]^T$, $\mathbf{L}$ is the graph Laplacian matrix, $\mathbf{S}$ is the degree matrix, and $\mathbf{I}_d$ is the identity matrix of order $d$. The diagonal degree matrix $\mathbf{S}$ is then formed as $\mathbf{S}_{ii} = \sum_{j=1}^{U} \tilde{\mathbf{W}}_{ij}$ and accordingly $\mathbf{L} = \mathbf{S} - \tilde{\mathbf{W}}$. The closed form solution of (13) can be obtained by solving a generalized Eigenvector problem based on KKT conditions, the eigenvectors corresponding to the second to $d+1$ smallest eigenvalues formulate the matrix $\mathbf{Z}$.

The DR technique t-SNE is widely used for data visualization and it has been shown to effectively convert high dimensional data into lower dimensions (particularly 2D/3D). In t-SNE, local structure of the data is favoured by weighting the pairwise distances in high dimensional space using Gaussian distribution and using Student-t distribution in the low dimensional space. Given two high dimensional points $\mathbf{x}_i$ and $\mathbf{x}_j$ and the corresponding pairwise distance $\mathbf{D}_{ij}$, the joint probability $p_{ij} = \frac{p_{i|j} + p_{j|i}}{2U}$ is calculated, where the conditional probability is defined as follows:

$$p_{i|j} = \frac{\exp\left(\frac{-\mathbf{D}_{ij}^2}{2\sigma_i^2}\right)}{\sum_{k \neq i} \exp\left(\frac{-\mathbf{D}_{ik}^2}{2\sigma_i^2}\right)}. \tag{14}$$

Here, $p_{i|j}$ is the probability that point $i$ is the neighbor of point $j$ and $p_{i|i} = 0$ by definition. The term $\sigma_i$ is the variance of Gaussian distribution which is centered on the high dimensional point $\mathbf{x}_i$. The value for $\sigma_i$ is determined based on the quantity called Perplexity. Perplexity is essentially the number of neighbors to the central point $\mathbf{x}_i$ of the distribution and it is an input parameter. An iterative algorithm is used to find $\sigma_i$ for a given value of the Perplexity [34].

The joint probability of the lower space embedding is defined as

$$q_{ij} = \frac{(1 + \|\mathbf{z}_i - \mathbf{z}_j\|_2^2)^{-1}}{\sum_{k \neq l} \left(1 + \|\mathbf{z}_k - \mathbf{z}_l\|_2^2\right)^{-1}}. \tag{15}$$

Then, the Kullback-Leiber divergence between the joint distributions $p$ and $q$ has to be minimized to obtain low dimensional embeddings. Thus, the optimization problem to be solved is given as:

$$\arg\min_{\mathbf{Z}} \sum_j \sum_i p_{ij} \log \frac{p_{ij}}{q_{ij}}. \tag{16}$$

The low dimensional points / channel chart locations are iteratively updated so that the cost function is minimized using gradient descent.

### *E. CC Evaluation*

Three metrics are used for channel chart quality measurement, namely Trustworthiness (TW), Continuity (CT), and Kruskal Stress (KS). TW measures how well false neighbors are avoided in the representation domain. CT indicates whether physical domain neighbor relations are preserved in the representation domain. KS shows to which extent the global structure is maintained in the representation domain. All three metrics are in the range $[0, 1]$ with optimal value 1 for TW and CT and 0 for KS. The neighborhood preservation metrics CT and TW can be computed by considering a neighborhood of $K$ points, denoted as $V_K(\mathbf{x}_i)$, around locations $\{\mathbf{x}_i\}_{i=1}^U$ in the original space, and the $K$-neighborhood denoted as $V'_K(\mathbf{z}_i)$, around the corresponding points $\{\mathbf{z}_i\}_{i=1}^U$ in the representation space. The equations to compute the average values are given as [35]:

$$\mathrm{CT}(K) = 1 - a \sum_i \sum_{\substack{j \in V_K(\mathbf{x}_i) \\ j \notin V'_K(\mathbf{z}_i)}} (r(i,j) - K), \tag{17}$$

$$\mathrm{TW}(K) = 1 - a \sum_i \sum_{\substack{j \notin V_K(\mathbf{x}_i) \\ j \in V'_K(\mathbf{z}_i)}} (r'(i,j) - K), \tag{18}$$

where $r(i,j)$ is the rank of a point $\mathbf{x}_i$ in terms of its distance from a point $\mathbf{x}_j$ in original space, $r'(i,j)$ is the rank of a point $\mathbf{z}_i$ in terms of its distance from a point $\mathbf{z}_j$ in representation space and $a = \frac{2}{UK(2U-3K-1)}$ is the normalization factor. In [36], the TW and CT are evaluated using $K$ equals the number of neighbours, which is used to create the graph in Laplacian Eigenmaps and t-SNE DR techniques. The idea to find how many of these neighbours are preserved/affected after applying the DR techniques. In [22], the TW and CT of the CC are reported using $K = 5\%$ of the total points.

KS is computed by comparing pairwise distance/dissimilarity matrix of the points in original space $\{\mathbf{x}_i\}_{i=1}^U$ with pairwise distance matrix of points in representation space $\{\mathbf{z}_i\}_{i=1}^U$ using a distance scaling factor $\lambda$ as:

$$\mathrm{KS} = \sqrt{\frac{\sum_{i,j} (\mathbf{D}_{ij} - \lambda \mathbf{\Delta}_{ij})^2}{\sum_{i,j} \mathbf{D}_{ij}^2}}, \tag{19}$$

where $\mathbf{\Delta}_{ij} = \|\mathbf{z}_i - \mathbf{z}_j\|_2$ and $\lambda = \sum_{i,j} \mathbf{D}_{ij} \mathbf{\Delta}_{ij} / \sum_{i,j} \mathbf{D}_{ij}^2$.

### *F. Regression Methods*

K Nearest Neighbors is the simplest ML algorithm for regression problems. To predict the mapping of a new sample using

KNN, first the distance between a new feature vector and all other feature vectors in the training set is calculated. Then the $k$-nearest neighbors are selected and the new sample's mapping is determined by averaging over the $k$-nearest neighbors' mapping value. This method can give a good approximation when the sampling density is high and there is a linear relation between mapping values, i.e., when the change of the predicted value is linear in the nearby area.

The Gaussian processes model is a popular non-parametric probabilistic ML framework for regression. Unfortunately, the non-parametric nature of this method causes computational problems for training over large data sets or high-dimensional input features [37]. A Gaussian Process Regression model makes predictions by incorporating prior knowledge and provides uncertainty measure over predictions. We aim to construct an approximation $\hat{f}(\mathbf{x})$ of the function $f(\mathbf{x})$ given a dataset $\mathcal{Q} = \{(\mathbf{x}_i, y_i), \mathbf{x}_i \in \mathbb{R}^d, y_i \in \mathbb{R}\}_{i=1}^{U}$. Assuming a simple regression problem, $y = f(\mathbf{x}) + \epsilon$ where $f(\mathbf{x})$ is a zero mean Gaussian process with corresponding covariance matrix, $\mathbf{K} = \{\kappa(\mathbf{x}_i, \mathbf{x}_{i'})\}_{i,i'=1}^{U}$, and $\epsilon$ is additive i.i.d Gaussian noise with variance $\sigma_0^2$.

Given the training data set $\mathcal{Q}$ and a new unseen data point $\mathbf{x}_j$ our task is to compute the posterior $p(f_j|\mathbf{x}_j, \mathcal{Q})$. The posterior density is defined as [38]:

$$\begin{aligned} &p(f_j|\mathbf{x}_j, \mathcal{Q}) \sim \mathcal{N}(\mathbb{E}[f_j], \mathbb{V}ar[f_j]), \\ &\mathbb{E}[f_j] = \mathbf{k}(\mathbf{x}_j)^T \mathbf{K}_y^{-1} \mathbf{y}, \\ &\mathbb{V}ar[f_j] = \kappa(\mathbf{x}_j, \mathbf{x}_j) - \mathbf{k}(\mathbf{x}_j)^T \mathbf{K}_y^{-1} \mathbf{k}(\mathbf{x}_j), \end{aligned} \tag{20}$$

where $\mathbf{k}(\mathbf{x}_j) = [\kappa(\mathbf{x}_1, \mathbf{x}_j), \ldots, \kappa(\mathbf{x}_U, \mathbf{x}_j)]^T$, $\mathbf{y} = [y_1, \ldots, y_U]^T$, and $\mathbf{K}_y = \mathbf{K} + \sigma_0^2 \mathbf{I}$. The covariance function reflects the prior information about the dataset. In the absence of any prior knowledge usually, a squared exponential covariance function is used

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \beta^2 \exp\left( -\sum_{\tau=1}^{d} \frac{(\mathbf{x}_i(\tau) - \mathbf{x}_j(\tau))^2}{l_\tau} \right), \tag{21}$$

where $\beta$ and $l_\tau$ are referred to as the hyperparameters which are optimized by considering a log likelihood function [38].

Neural Networks provide a parametric ML framework for regression problems. We consider multilayer feedforward NNs, consisting of an input layer, multiple hidden layers and an output layer. The deep architecture enables the network to extract appropriate information for regression. Hidden layers are fully connected to the adjacent layer and each link has its own weight and bias. Rectified Linear Unit (ReLU) activation function is used in all layers except the last layer to provide nonlinearity. The weights and biases (model parameters) are determined by training the network to minimize a loss function.

The learning process consists of a forward and backward propagation phase. In the forward phase, the input is propagated across the hidden layers until the output layer. The difference between the predicted output and the given output is minimized using a loss function. In the back propagation phase, model parameters will be updated. Levenberg-Marquardt (LM) [39] and Adam [40] algorithms are used to minimize the loss function. The Levenberg-Marquardt algorithm has been shown to be efficient for moderate sized NNs and converges rapidly. However, the computational cost is higher as compared to Adam algorithm.

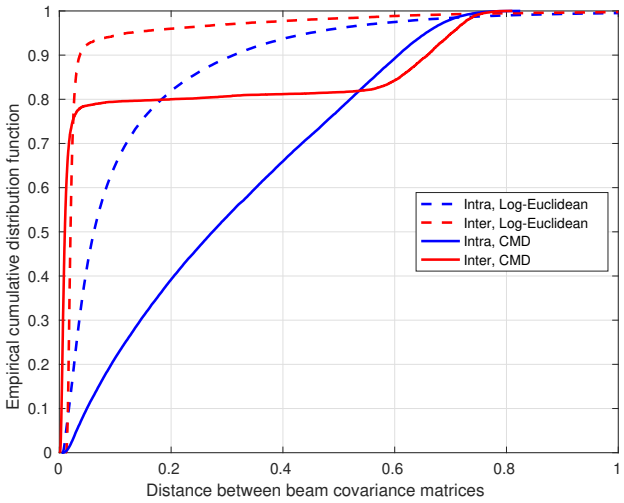



Fig. 2: The empirical CDF of CMD and Log-Euclidean measures for inter and intra location distances.

## III. Beam-SNR Prediction

The best combination of UE and BS beams using (4) is given by $(m^*, t^*) = \arg\max_{m,\ t} \gamma_{mt}$. Assuming autonomous UE beamforming, where the UE selects the best UE beam towards a BS beam, the BS best beam is:

$$m^* = \arg\max_{m} \gamma_{m\, t(m)} \tag{22}$$

A naive solution is based on *beam sweep*, where the BS transmits reference signals from each of its beams $m$ in sequence, and the UE selects its best beam $t(m)$ for each $m$ and reports the measured channel qualities to the BS.

Here, we develop an alternative method where the BS predicts beam-SNRs from measured uplink CSI, and a number of annotated channel charts. Before we dive into the beam SNR prediction, we explain the necessity of using beam-based CSI features.

### A. Spatial Consistency at BS

We should highlight the effect that changing the UE direction of transmission has on the effective channel measurements at the BS. The UE measures the BS beams, selects its best beam and transmits towards the BS. At each BS beam, we calculate the covariance matrix of the UEs' transmission. To gain insight, we compare the dissimilarity of the covariance matrices using different measures. We investigate the effect on the defined CSI features, measured at the BS, if the UE changes its beamformer.

We consider a BS and a set of 3600 UEs in a street segment of 10 m $\times$10 m. The BS has a ULA with 32 antennas. The UE has an 8-element ULA antenna. The channel coefficients are generated considering 3GPP 38.901 Urban Macro cellular Non-Line of Sight (UMa-NLOS) specification [41] where the distance between the street and BS is 100 m, and the carrier frequency is 28 GHz.

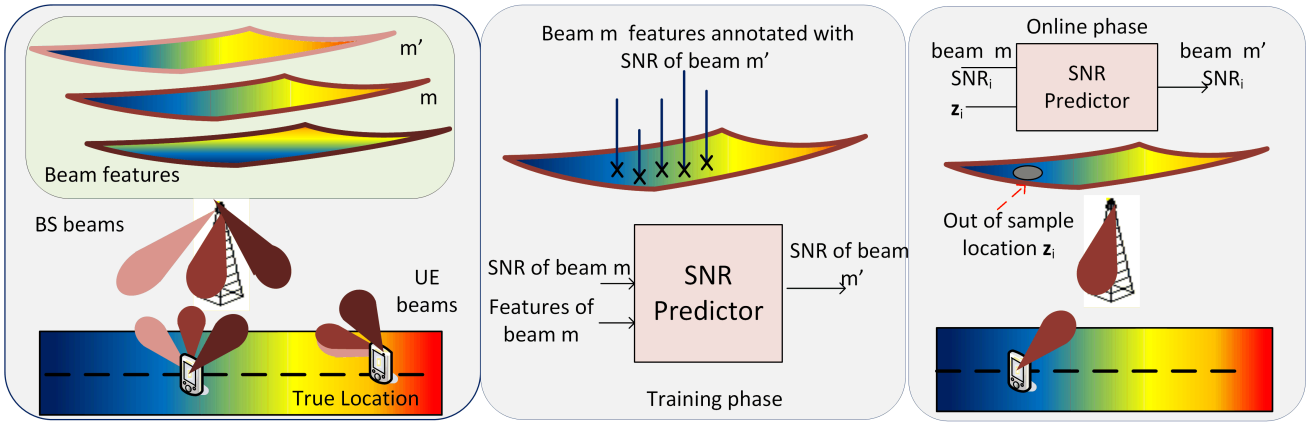



Fig. 3: Beam SNR prediction based on beam CSI feature: (Left); Construction of beam based features and their dimensionality reduction. (Middle); Beam feature annotation and SNR prediction (the offline phase). (Right); Beam SNR prediction based on beam feature (online phase).

For UE $u$ transmitting towards beam $\mathbf{w}_m$, we calculate the dissimilarity between the best beam covariance matrix and other beams for that UE. The dissimilarity of the transmission towards beams $m$ and $m'$ with covariance $\mathbf{R}^u_m$ and $\mathbf{R}^u_{m'}$ is called *intra-location* distance.

We contrast this to the dissimilarity arising from the UE moving in space, keeping its best beam. The distance between covariance matrix of the best beam of a UE and the covariance matrix of the best beam of any of its 10 nearest neighbors in the created sampling is called *inter-location* distance. The empirical Cumulative Distribution Function (CDF)s of *inter-* and *intra-location* distances for CMD and Log-Euclidean distance of covariance matrices are shown in Fig. 2. The Log-Euclidean distance is not a normalized measure. In order to obtain a normalized measure with a similar range as CMD, as shown in Figure. 2, the Log-Euclidean distance is first normalized with the largest value in the data set and then exponentiated.

In the spatial neighborhood of a UE, we expect that the user's best beam covariance matrix will not change rapidly. Thus, if we use the best beam for each location, we will not see a large variation in the measured effective channel covariance matrix. The *inter-location* curves in Fig. 2 confirm this. Changing the UE beam for the same location, however, is similar to transmitting towards a random direction. The figure shows that changing the UE beam has a larger effect on feature distance than moving in the environment.

This investigation implies that we cannot use a single covariance matrix at the BS as a feature, such a feature would not be robust against UE beamformer changes. For prediction, we need to have a set of covariance matrices conditioned on the transmission of the UE being towards *a specific BS beam*, assuming the UE selects its best beam.

### B. Problem Formulation

Our objective is to avoid BSs transmitting pilots on multiple beams, and the related UE reporting, thereby reducing both signaling overhead and the time spent on best beam pair search.

The problem addressed here thus is: *assuming that the BS knows that the UE uses a beam targeted towards BS beam $m$, what are the SNRs of the different BS beams $m' \neq m$?*

From this information, the best BS beam can be obtained, and as a result, the best beam can be selected. Note that the BS does not know the full channel matrix $\mathbf{H}^u_b$ nor which specific beam $t = t(m)$ the UE uses. The BS only knows the effective channel $\mathbf{h}^u_{bt(m)}$ of (5), from which it can directly compute the SNR $\gamma^u_{bm}$. It has, however, no means for directly computing $\gamma^u_{bm'}$ for $m' \neq m$.

For this, we shall consider methods to *learn to predict the beam SNRs* for $m' \neq m$, using an ML framework. We devise a CSI feature (fingerprint) towards a target BS beam assuming that the UE uses an autonomous beam. The CSI feature can be the raw feature covariance CSI or dimensionality reduced feature (linear and non-linear DRs are considered). We train a predictor to estimate the SNR of a UE at other beams given one beam CSI. Our goal is to approximate an unknown mapping function $\Gamma$:

$$\Gamma : \mathbf{x} \mapsto y, \tag{23}$$

between CSI features and beam SNRs. The SNR prediction is formulated as a regression problem, modeling the input-output variables relationship.

### C. Solution Approaches

We consider a network-centric beam based handover for data collection, model training, and SNR prediction as illustrated in Fig. 3. The beam SNR prediction framework is applicable to any MIMO system, where both the UE and the BS use beamformers. The basic idea is that for each beam, a set of features is extracted from the available CSI and the features are annotated with SNRs of neighboring beams of the current BS or the neighboring BSs. We predict the SNR of a UE at other beams given the extracted features, then based on the SNR difference of the serving beam and other beams a handover decision is made. The operation is divided into two parts: offline and online phases.

During the offline phase, a data set of beam CSI features and SNRs at other beams is collected for each beam. Dimensionality reduction is performed on the CSI features, giving rise to a CC. The CSI features and thus the samples constituting the CC are annotated with information about SNRs of beams. The UEs at

the sample locations find their best beam for all BS beams and report the corresponding SNRs to the BS. For this, we only assume measurements at UEs required by 3GPP standards for beam management, see e.g. [3].

Given annotated CSI feature of each beam, a SNR predictor is trained to predict the SNR of UEs at other beams. The channel feature $\mathbf{x}_{bm}^{u}$ is defined as the CSI of UE $u$ transmitting towards beam $m$ of BS $b$ and it is annotated with $y_{b'm'}^{u}$, where $y_{b'm'}^{u}$ is the SNR information at beam $m'$ of BS $b'$. This information is collected for a set of sample UEs. Given the CSI feature of a transmission towards $bm$, the SNR of a transmission towards $b'm'$ is predicted. Here, the subscript "$bm$" indicates the serving BS and beam and "$b'm'$" indicates the target beam with different BS and or beam.

We extract a high dimensional CSI features in form of the covariance matrix of the UE transmission towards a specific beam. Inspired by fingerprinting approaches, we leverage different predictors (i.e., KNN, GRP, and NN) so that the SNR of the UE at other beams can be predicted.

In the online phase, the serving beam CSI feature is used to predict the SNR at other beams. Raw CSI feature (i.e., the beam covariance matrix) is processed further either by normalizing it or by taking a logarithm of the covariance matrix. Then, depending on the DR method that is chosen, two paths are available. Either the feature is used directly, or adaptive sampling set reduction is applied. If PCA is chosen as the DR method, the feature is directly mapped to a lower dimension. Otherwise, by using an operation called adaptive sampling set reduction, the most similar features from the training set are chosen. Depending on the form of CSI feature we use for prediction, the nearest neighbors on the training set are found. If the KNN predictor is chosen, the SNR is predicted by averaging over the nearest neighbors' SNR value. If the prediction is based on NLDR, after finding the nearest neighbors, the CC location of the served UE is computed and then fed to the SNR predictors for prediction. Fig. 4 shows the details of the online phase.

In the online phase, the UE solely transmits sounding reference signals using a beam $\hat{t}(m)$ selected based on the current BS beam $m$ used to serve the UE. CSI measurements and computations related to feature extraction and dimensionality reduction are performed at the BS.

Our solution to beam SNR prediction is based on the served beam CSI feature. We use different forms of the feature, either the raw CSI feature vector or the dimensionality reduced (linear or non-linear) version of the feature for SNR prediction. This feature is processed and prepared during both offline and online phases.

## IV. Out-of-Sample Extension

In a realistic scenario, after training the SNR predictors, we need to predict the probable target beams for the handover of a new UE that establishes a connection with one beam during the online phase. Other than the raw feature-based SNR prediction model, an OoS extension is needed to locate the new UE in the feature space for DR method.

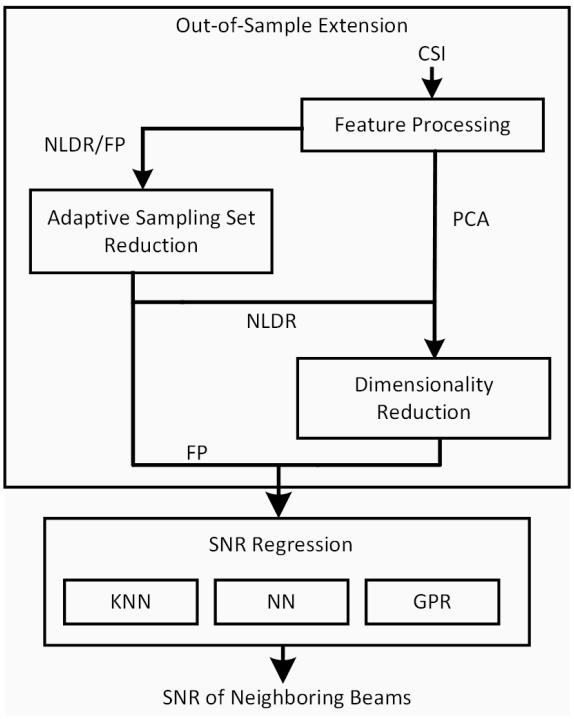



Fig. 4: The online phase: CSI feature of the serving beam is processed and fed to the predictor.

### A. Full Dataset Out-of-Sample Extension

Out-of-sample extension is essentially computationally complex and expensive. This is due to the computation of dissimilarity of the OoS point with respect to all other training points. As long as the training dataset is of a small size, the computation burden is tolerable. However, when the size of the training dataset increases it becomes problematic. Even though several OoS extension algorithms have been proposed, they are computationally complex and time-consuming [42].

For PCA, a parametric OoS exists.

For Laplacian Eigenmaps, however, a non-parametric approximation method is used. The OoS is framed as an optimization of the objective which finds the normalized kernel function that minimizes the mean square error. To obtain the embedding for a new point, the normalized kernel matrix $\tilde{\mathbf{W}}$ is used as weights. Using the obtained mappings for the original dataset points $\mathbf{z}_i$ for $i = 1, \ldots, U$, the embedding for the new OoS point, $\mathbf{z}_{\mathrm{OoS}_{LE}}$ is computed as:

$$\mathbf{z}_{\mathrm{OoS}_{LE}} = \sum_{i=1}^{U} \tilde{\mathbf{W}}_{ij} \mathbf{z}_i, \qquad j \notin \{1, ...U\}, \tag{24}$$

where $\tilde{\mathbf{W}}_{ij}$ is the weight of the $j^{\mathrm{th}}$ OoS point with respect to $i^{\mathrm{th}}$ point in the original dataset. We note that the OoS location will only be affected by the weight of $k$-nearest points in the data set as the weights of other points are set to zero.

However, t-SNE technique has limitations on OoS as it is a non parametric DR. The problem of t-SNE is that with every new data point, a new mapping is obtained, and extending the mapping to new points is difficult.

A non-parametric out-of-sample extension that can be applied to all nonlinear dimensionality reduction techniques is based

on finding the nearest neighbor of the new point in the high-dimensional representation and computing the linear mapping from the nearest neighbor to its corresponding low-dimensional representation. The low-dimensional representation of the new data point is found by applying the same linear mapping to this data point. This approach entails the computational complexity of finding the nearest neighbors in the high space.

Except PCA, the main step in the OoS extension algorithms is that for the new data point, dissimilarity to all training points is needed to be calculated. With a large training dataset, both memory and computation issues arise.

Given dimensionality reduced input CSI features, we compute the location of the new point on the embedding during the online phase, which is fed to the SNR predictor. With full knowledge of dissimilarity matrix between the new point and the training dataset, we find the $k$-nearest points to the new point. Then, a weighted average of the the $k$-nearest locations is used to approximate the OoS location as:

$$\mathbf{z}_{\text{OoS}} = \sum_{i=1}^{k} \mathbf{z}_i \mathbf{\Omega}_{ij}, \tag{25}$$

where the exponential weighting for $i^{\text{th}}$ neighbor is:

$$\mathbf{\Omega}_{ij} = \frac{\exp(-\mathbf{D}_{ij})}{\sum_i \exp(-\mathbf{D}_{ij})}, \tag{26}$$

and $\mathbf{D}_{ij}$ is the dissimilarity of the new point to the $i^{\text{th}}$ point in the $k$-nearest neighbors and $\mathbf{z}_i$ is the location of corresponding training point.

### *B. Adaptive Sample Set Reduction*

Computing the dissimilarity between a new point to all training points is the most computationally heavy part for a large dataset[1]. We propose a heuristic method where fewer dissimilarity computations are used. We refer to this approach as hierarchical method, we find the OoS point through an iterative process by zooming into the neighbourhood of the nearest neighbors found in the previous step. The goal is to find an estimation of the OoS point with fewer number of dissimilarity calculations. The process is summarized in Algorithm 1.

First, $L_1$ training points are chosen as landmarks. In order to uniformly choose the landmark points, a simple K-means clustering algorithm is employed to find $L_1$ clusters in the entire area.

We choose the closest point to the cluster center and assign it as a landmark. Then, the dissimilarity of the new point to landmarks is calculated and the $S_1$ closest ones are chosen. Here, an iterative process starts and the step index $i$ is set to 1. At the $i^{\text{th}}$ step, a region between the $S_i$ closest landmarks is formed. The region is formed based on the center of mass rule where the center is calculated as the average of $S_i$ landmark location in each dimension and the radius is calculated as the average distance of the center to $S_i$ landmarks. Let $|\mathcal{S}_i|$ be the number of points are in the region. Among these points, $\min\{L_{i+1} - S_i, |\mathcal{S}_i|\}$ are picked as the second stage landmarks and $S_{i+1}$ closest points are chosen. The iteration ends here and with new $S_{i+1}$ samples a new region is formed. The convergence criterion is maximum number of steps or $|\mathcal{S}_i| = 0$. If any of them is met, the OoS is computed with the last closest landmark points as in (25).

[1]The computational complexity is discussed in detail in Section V.

**Algorithm 1** Adaptive Sample Set Reduction

**Given**: The channel covariance matrices $\tilde{\mathbf{R}}_u$ and $\mathbf{z}_u$ for $u = 1, \ldots, U$, covariance of the new point $\tilde{\mathbf{R}}_j$, parameters $L_i, S_i,\ i = 1, \ldots, I_{\max}$.
**Initialize**: $i \leftarrow 1$, a set $\mathcal{S}$ of candidates (pre-selected) from $\tilde{\mathbf{R}}_u$.
$\mathcal{L} \leftarrow$ Select $L_i$ landmark from $\mathcal{S}$.
$\mathcal{L} \rightarrow$ compute dissimilarity to $\tilde{\mathbf{R}}_j$.
Select the $S_i$ closest landmarks.
**while** $i < I_{\max}$ AND $\mathcal{S} \neq \emptyset$ **do**
    Construct a region from $S_i$ landmarks.
    Find a set of new landmarks $\mathcal{S}$ in the region.
    $\mathcal{S} \leftarrow$ Select $\min(L_{i+1} - S_i, |\mathcal{S}|)$ sample points from $\mathcal{S}$ at random.
    Compute the dissimilarities of $\mathcal{L} \cup \mathcal{S}$ to $\tilde{\mathbf{R}}_j$.
    Select $S_{i+1}$ closest landmarks.
    $i \leftarrow i + 1$.
**end while**
**Output**: The set of $S_i$ closest landmarks.
**Compute**: $\mathbf{z}_{\text{OoS}}$ using (25).

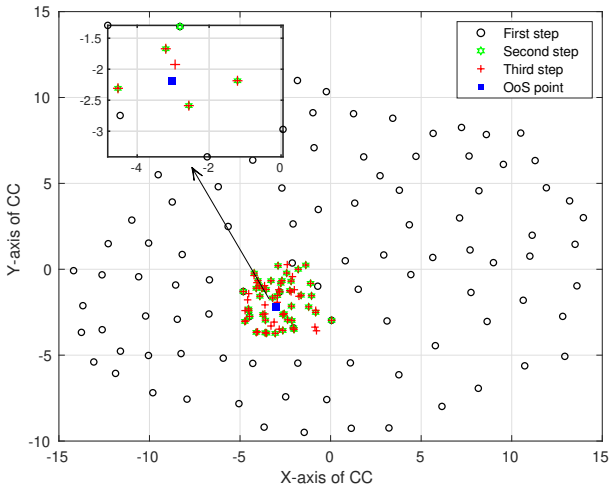



Fig. 5: Hierarchical OoS where the first set of landmarks are shown by the black circles, second step landmarks are shown by green asterisks and final step landmarks are shown by red crosses.

Fig. 5 shows an example of the hierarchical OoS approach. In the first step, a set of landmarks is chosen and the closest ones to the new OoS point are chosen (we assumed $S_i = 5$ for all iterations). Then, in the next step, the region between the 5-nearest neighbors is constructed and new landmarks are sampled from this region. Here, after three steps the algorithm converges, and as can be seen in the last two steps the difference between the chosen landmarks is small. The zoomed area shows the five closest landmarks in each step. Now, the OoS point location is computed as an average of the last step nearest neighbors and is shown by the light blue square.

## V. Complexity Analysis

We now examine the computation complexity of the online phase in terms of real multiplication, which includes the complexity of the OoS extension function and the predictors (i.e., KNN, NN, and GPR). The number of additions is neglected because in a typical algorithm for multiplying two $n$ digit numbers, the computational complexity is $\mathcal{O}(n^2)$, whereas adding the same numbers has a complexity of $\mathcal{O}(n)$. Multiplication is, therefore, the most time-consuming part of the implementation procedure. For comparison of the computation complexity of different procedures, we consider float values of 16 digits. The training phase complexity, which is carried out offline, is not considered.

Here, we point out the complexity of some elementary functions that are used in the online phase. According to [43], the complexity of elementary functions differs from the complexity of multiplication only by some multiplicative constants. Assuming multiplication complexity as a constant time as $\mathcal{O}(1)$, the complexity of division is $\mathcal{O}(M_{\text{div}})$. Analysis shows that $M_{\text{div}} = 4$. For the $\exp(\cdot)$ and $\log(\cdot)$ functions, the complexity is the same, $\mathcal{O}(M_{\text{exp}}) = \mathcal{O}(M_{\log})$ where the multiplicative constant $M_{\text{exp}} = 52$ and for the square root function the complexity is $\mathcal{O}(M_{\text{sq}})$ with the constant $M_{\text{sq}} = \frac{11}{2}$.

### A. Out-of-Sample Complexity

The main complexity of OoS extension function is due to dissimilarity computation. For a new UE, dissimilarity is computed via either (7) or (8). For (7), the normalized feature requires $\mathcal{O}(M^2)$ multiplications to be calculated. For (8), the CSI feature is processed, and then the distance is computed where for the matrix log processing, a Singular Value Decomposition (SVD) is needed. The SVD complexity is $\mathcal{O}(M^3)$ considering $M \times M$ covariance matrix. Secondly, for all training points, the $\text{Tr}(.)$ function is needed to be computed. For the trace function, we need to compute only the diagonal elements of the matrix. Each diagonal element is obtained by $M$ complex multiplications. Thus, $4M^2$ real multiplications are needed.

Considering the OoS algorithm, the complexity of computing the OoS point in the last step of algorithm as in (25) compared to the Singular Value Decomposition (SVD) computation is negligible. Also, the complexity of partial sorting in the algorithm is not considered since its complexity is comparable to a set of additions. As a result the OoS extension complexity, using Log-Euclidean feature is

$$C_{\text{OoS}} = \mathcal{O}\big(M^3 + 4M^2 \sum_{i=1}^{I_{\max}} L_i\big). \tag{27}$$

The complexity of OoS for PCA method can be divided into calculating the logarithm of the covariance feature and then using (10), a matrix multiplication for a $d$ dimensional final feature as

$$C_{\text{PCA}} = \mathcal{O}\big(M^3 + dM^2\big). \tag{28}$$

### B. Predictor Complexity

The computation complexity of KNN method, is mainly due to feature dissimilarity computation. We consider KNN when CSI Fingerprint (CSI FP) and PCA features are used. If CSI fingerprint features are used the complexity of KNN method is derived from (27) since the dissimilarity to all training points is needed to be calculated ($L_i = U$). If PCA features are used, after the OoS operation, Euclidean distance between the dimensionality reduced training points and the OoS point is calculated. The added complexity from KNN method is $\mathcal{O}(Ud)$ which is for calculating the dissimilarity of the OoS point to the dimensionality reduced training points. The total complexity of beam SNR prediction is the OoS and regression complexities.

TABLE I: Computational Complexity

| Operation | Complexity |
|---|---|
| OoS PCA | $\mathcal{O}\big(M^3 + dM^2\big)$ |
| OoS | $\mathcal{O}\big(M^3 + 4M^2 \sum_{i=1}^{I_{\max}} L_i\big)$ |
| K-NN | $\mathcal{O}(Ud)$ |
| GPR | $\mathcal{O}(U(N_{\text{in}} + M_{\text{exp}} + 2) + U^2)$ |
| NN | $\mathcal{O}(N_{\text{in}}N_h + 2N_h^2 + N_h N_o)$ |

The computation complexity of a GPR model is mainly a function of the number of training points, on the other hand. The main complexity arises from kernel computations. Computation of kernel function between a new point and the training set points is the major computation burden in GPR model. Considering Gaussian covariance kernel function, computation of kernel function for a new point needs $C_{\text{kern}}$ multiplications as

$$C_{\text{kern}} = \mathcal{O}\big(N_{\text{in}} + M_{\text{exp}} + 2\big), \tag{29}$$

where $N_{\text{in}}$ is the number of input features. The complete GPR model takes $C_{GPR}$ multiplications as

$$C_{\text{GPR}} = \mathcal{O}(UC_{\text{kern}} + U^2). \tag{30}$$

When $N_{\text{in}}$ is small compared to the number of training points $U$, the kernel function computation complexity is negligible. However, when $N_{\text{in}}$ increases, a significant computational load is incurred by the kernel computation. According to [37], distance calculations between points for the kernel function will be problematic for high-dimensional input features. This is due to the fact that the points are relatively farther away in high dimensions, hence it becomes harder to learn the mapping between input and output pairs.

Computational complexity associated with NN is mainly a function of the number of neurons and layers. Considering, a NN with three dense layers and each layer with $N_{\text{h}}$ neurons, the resulting model requires $C_{\text{NN}}$ multiplications as

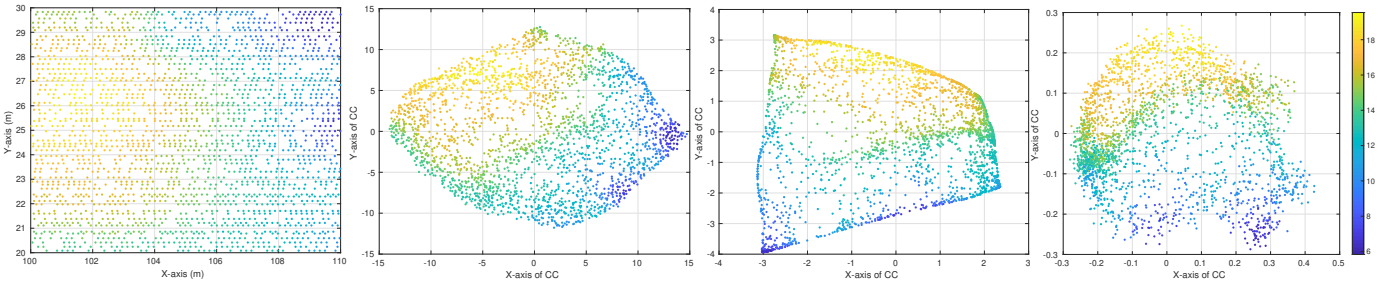
$$C_{\text{NN}} = \mathcal{O}(N_{\text{in}}N_{\text{h}} + 2N_h^2 + N_{\text{h}}N_{\text{o}}), \tag{31}$$

where $N_{\text{in}}$ is the number of input layer features and $N_{\text{o}}$ is the number of elements in the output layer.

Table I summarizes the computational complexity of different OoS and predictors.

## VI. Simulations

To evaluate the performance of the SNR prediction and OoS extension methods, we conducted simulations using synthetic




(a) True Physical Location (b) t-SNE (c) LE (d) PCA

Fig. 6: (a) The physical location annotated with the SNR of one beam. (b)-(d) The CC feature (t-SNE, LE and PCA of one beam annotated with the SNR of another beam).

TABLE II: Simulation Parameters

| Parameter | Value | Parameter | Value |
|---|---|---|---|
| Center Freq. | 28 GHz | Subcarriers | 256 |
| Scenario | 3GPP 38.901 UMa-NLOS | Subcarrier BW | 240 KHz |
| BS Array | 32 ULA | UE Array | 8 ULA |
| BS Height | 25 m | UE Height | 1.5 m |

TABLE III: Average Performance Measures of Different Features

| | $\mathbf{R}/\|\mathbf{R}\|_F$ | | | $\log(\mathbf{R})$ | | |
|---|---|---|---|---|---|---|
| Scheme | TW | CT | KS | TW | CT | KS |
| CSI FP | 0.89 | 0.94 | 0.50 | 0.99 | 0.99 | 0.28 |
| PCA (2D) | 0.81 | 0.91 | 0.51 | 0.92 | 0.96 | 0.29 |
| PCA (30D) | 0.90 | 0.93 | 0.45 | 0.97 | 0.98 | 0.28 |
| LE (2D) | 0.92 | 0.95 | 0.41 | 0.99 | 0.99 | 0.18 |
| t-SNE (2D) | 0.93 | 0.96 | 0.33 | 0.99 | 0.99 | 0.16 |

data generated by the Quasi Deterministic Radio Channel Generator (QuaDRiGa) simulator [44]. In order to model a realistic environment, the adjustable parameters of the simulator are based on the 3GPP channel model [41]. The simulation is carried out in a street segment of 10 m$\times$ 10 m, where 2800 UEs are scattered randomly. The layout includes two BSs located at xy-coordinate $[0,\ 0]$ m and $[210,\ 50]$m, respectively. We consider a 3D UMa-NLOS scenario that takes into account both large-scale and small-scale effects, including multi-path fading. The standard values for delay spread, AoA and AoD distributions are from [41]. The simulation parameters used in our study are summarized in Table II.

### A. Annotated Beam CC

After channel generation, for calculating the covariance matrix of a UE moving with speed of 50 Km/h, we generate 100 small scale fading temporal samples within a 100 ms time frame. The CSI covariance features of each beam are created according to (6). Sample covariances are created by averaging over frequency and spatial samples. The Log-Euclidean distance and CMD are used to calculate the dissimilarity of UE CSI at each beam. Linear and non-linear dimensionality reduced CSI features of each beam are created with different dimensions.

We evaluate the quality both of the raw CSI covariance features, referred to as CSI fingerprints, and the dimensionally reduced features resulting from CC. To benchmark the capability of the CSI features to capture characteristics of ground truth physical locations, we compute the CT, TW, and KS performance measures.

Table III shows the corresponding values of performance measures for CMD (7) and Log-Euclidean (8) distance. Specifically, covariance features are processed to the form of the column headers, corresponding to the two different metrics of interest, after which feature dissimilarity of UEs is computed using the Frobenius norm.

We set 5% of total points as the number of neighbors for calculating the performance measures, and the performance values are averaged values. For the data set used this represents the neighbors within a circular disc of radius $\sim$1.25 m based on the physical locations, on average. With the chosen measure we are thus evaluating the neighborhood preservation within this disc.

The results summarized in the table show first that linear dimensionality reduction may *destroy some of the underlying ground truth spatial geometry information hidden in the CSI FPs*. E.g. PCA to two dimensions performs worse than the CSI FPs in terms of all metrics, for both dissimilarities considered. However, non-linear dimensionality reduction, exemplified here by LE and t-SNE, is able to extract more information about spatial geometry from the FPs than using them directly. This is the principle underlying the efficacy of CC [22].

Furthermore, the table shows that $\log(\mathbf{R})$ based dissimilarity outperforms CMD, i.e., the normalized CSI feature dissimilarity. In particular, global geometry is better preserved by $\log(\mathbf{R})$ dissimilarity. Therefore, we focus on Log-Euclidean distance from now on. The performance of non-linear DR techniques (i.e., Laplacian Eigenmaps and t-SNE) are better compared to PCA. For Laplacian Eigenmaps and t-SNE, the neighbourhood preservation is better than for PCA. Also, the global structure of UEs is relatively well maintained.

Note that the construction of the CC, producing the results above, is fully self-supervised, no information about the spatial geometry is needed.

Continuing with the offline phase of Section III-C, we annotate the constructed CSI features / channel charts with the SNR values of all BS beams, measured by the UEs at the

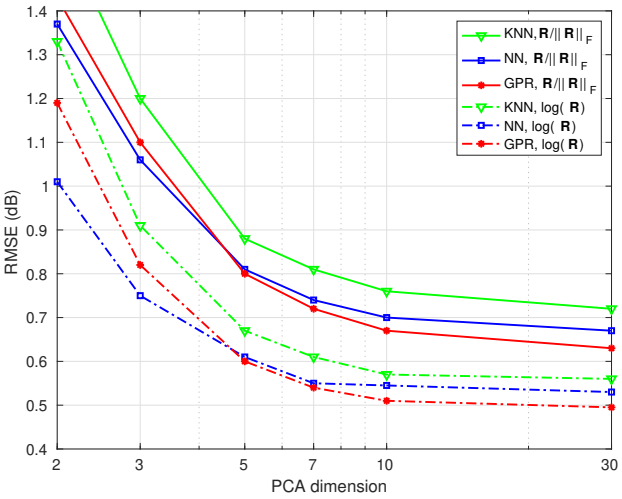



Fig. 7: Average RMSE of GPR and NN predictors as a function of dimensionality reduced beam CSI feature using PCA.

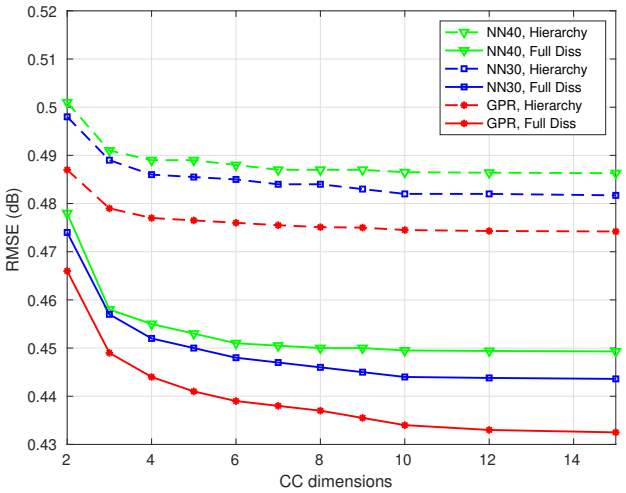



Fig. 8: Average RMSE of different predictors as function of beam CC dimension for t-SNE. The CC dimension is limited to 15, since most of the RMSE gain is achieved within five dimensions.

sample locations. For these measurements, each sample UE autonomously selects its best beam towards each of the BS beams. It should be noted that in the considered street segment, 15 beams from 2 BSs are dominant. The annotated physical location and CCs are depicted in Fig. 6. The left part of the figure shows the physical location of UEs annotated with SNR of one beam. These SNR values are shown at the channel chart locations of another beam in the right part of the figure.

### *B. Beam SNR Prediction*

The beam SNR mapping functions using KNN, GPR, and NN predictors are created. The mean squared error loss function is used in both the NN and the GPR. The NN comprise of three hidden layers with $N_h = 30$ neurons in each layer and ReLu activation function. The Root Mean Squared Error (RMSE) is used as the performance measure for the SNR predictors in dB scale. The data set is divided into 3 sets, 1200 for training, 1200 for testing and 400 for validation.

The input dimension for the predictor is a concatenated vector of the current beam SNR and dimensionaly reduced feature vector. As for the Laplacian Eigenmaps and t-SNE based predictions, 2D to 10D CC dimensions are used. For PCA, 2D - 30D are used. Raw CSI fingerprint with 1024 dimensions is used. The OoS point is obtained by either computing the dissimilarity to all points in the dataset called full dissimilarity or to fewer points using the hierarchical method. The hierarchical OoS is assumed to be performed in 3 steps with $L_1 = 100, L_2 = 45$, and $L_3 = 45$. In each step $S_i = 5$ neighbors for $i = 1, 2, 3$ are considered. Thus, we have at most 180 dissimilarity calculations which compared to 1200 required calculations in the full dissimilarity OoS approach, is relatively small. For PCA the OoS is obtained directly from (10).

In the NN predictors, the output layer has 14 dimensions. Thus for each input beam, the SNR at other beams can be predicted concurrently. However, with GPR and KNN predictors, a separate predictor is trained for each pair of beams. The reported RMSE is averaged over all beams' predicted SNR.

Fig. 7 depicts the average RMSE of target beam SNR prediction as a function of dimensionality reduced beam features using PCA. Increasing the feature dimension is beneficial for more accurate predictions, but at some point (around 30 dimensions) no more gain could be obtained. Results for both $\mathbf{R}/\|\mathbf{R}\|_F$ and $\log(\mathbf{R})$ features show a similar trend, whereas for the $\log(\mathbf{R})$ feature superior performance is obtained. That is the reason for choosing $\log(\mathbf{R})$ as the feature vector for the rest of the simulations.

Fig. 8 shows the RMSE performance result for NN and GPR predictors based on t-SNE DR. Various CC dimensions are considered and as expected, increasing the dimension of input CC improves the SNR prediction performance. GPR predictor has shown better performance than NN predictors with all features. There is $0.04$ dB gap between the proposed hierarchical OoS and the full dissimilarity OoS. Simulation results show that most of the gain comes from increasing CC dimension up to $15$, and beyond this the gain is marginal. Based on this, we limit the CC dimension to $15$.

The beam SNR prediction results for different predictors, various forms of CSI feature, and two OoS methods are summarized in Table IV. For Laplacian Eigenmaps and t-SNE CCs, RMSE values corresponding to OoS using full dissimilarity and the hierarchical method are reported. In the hierarchical approach, dissimilarity computations to $15\%$ of the training set is used. By comparing the results of full dissimilarity OoS and proposed hierarchical OoS, small degradation of $\sim 10\%$ in prediction performance is observed. CSI features based on t-SNE, outperform other DR-based inputs because the t-SNE CC has preserved the neighborhood information very well.

High dimensional input is a critical challenge for GPR [37], and the learning task becomes difficult as the number of input variables impacts the search space. Therefore we eliminate the prediction result of this scheme for CSI fingerprint. A different NN consisting of 5 hidden layers and 300 neurons in each layer is considered for the SNR prediction using CSI fingerprint. In this network, Adam optimizer is used for training the model parameters. Since we are using the CSI fingerprint feature, no OoS is needed during the online phase and this approach can

TABLE IV: Performance of Different Predictors and Beam CSI Features

| | GPR | | NN | | KNN | |
|---|---|---|---|---|---|---|
| Feature | Full | Hier. | Full | Hier. | Full | Hier. |
| LE | 0.46 | 0.50 | 0.47 | 0.50 | | |
| t-SNE | 0.43 | 0.47 | 0.44 | 0.48 | | |
| CSI FP | | | 0.57 | | 0.50 | 0.67 |
| PCA 2D | 1.19 | | 1.01 | | 1.33 | |
| PCA 30D | 0.50 | | 0.53 | | 0.56 | |
| GNSS (15 cm) | 0.42 | | 0.42 | | 0.46 | |
| GNSS (2 m) | 1.44 | | 1.46 | | 1.8 | |

TABLE V: Complexity (M)

| **Regression** | 2D | 30D | 1024D |
|---|---|---|---|
| NN (LM), $N_h$=30 | 0.002 | 0.003 | |
| NN (Adam), $N_h$=100 | | | 0.17 |
| NN (Adam), $N_h$=300 | | | 0.70 |
| GPR | 21.00 | 21.56 | |
| KNN | 0.002 | 0.036 | |
| **OoS** | | | |
| NLDR / FP, Full | 4.91 | | |
| NLDR / FP, Hier. | 0.77 | | |
| PCA | 0.035 | 0.063 | |

be called end-to-end learning. In the case of KNN predictor, if OoS with full dissimilarity knowledge is chosen, comparable performance is obtained. However, the hierarchical OoS can not provide the same performance.

For PCA features, the OoS extension has a closed form expression (10). We have reported the RMSE of beam SNR prediction for the case of 2D and 30D. Comparing the result of PCA with non-linear DR techniques, we see that it has the worst RMSE performance.

We predict the beam SNR based on the physical location as a benchmark. For the physical location-based model, in the offline phase, the network has to collect data (i.e., physical locations and SNR of all BS-UE beam pairs) and then train the ML model, so we are examining all beam pairs and measuring the SNR at all BS-UE beam pairs. We annotate the physical location map, with the SNR of different beams and train the predictors.

With 15 cm accuracy of Global Navigation Satellite System (GNSS) [45], ideal true location-based prediction has a comparable RMSE to the CC-based prediction. Predictions for higher values of inaccuracy (i.e, 2 m for a typical system) is considered. Results show inaccurate SNR prediction when the input location is not precisely known.

### *C. Computational Complexity*

We compare the computational complexity of different predictors and OoS methods. The total complexity of the beam SNR prediction is the complexity of OoS operation and the beam SNR predictor. The corresponding complexities are shown in Table V.

The complexity of different predictors can be calculated according to (27)-(30) at the online phase. Table V shows the complexity of different regression and OoS methods, where the complexity is reported in number of multiplications in million (M). Regression complexity of the NN predictor (with Levenberg-Marquardt optimizer) for a network with $N_{\text{in}} = 2, 30$ and three hidden layers for dimensionality reduced features is computed. Also, for the network with $N_{\text{in}} = 1024$ and 5 hidden layers (i.e., the NN with Adam optimizer), which is applied to the CSI fingerprint, the equivalent number of multiplications are 0.17 M to 0.7 M for the network with 100 and 300 neurons in hidden layers, respectively.

On the other hand, the GPR predictor itself requires 1.5 M multiplications for predicting one target beam SNR which is larger compared to NN-based prediction. For a complete beam SNR prediction, 14 GPR predictors are needed, and hence 21 M multiplications for 2D input and 21.56 M for 30D input vector are needed. When using the KNN predictor, the main complexity comes from dissimilarity calculation. Thus for CSI fingerprint-based prediction, the complexity of KNN can be omitted, since in the OoS operation, nearest neighbors are found and then can be used. In the case of PCA, after the OoS operation as discussed in Section V-A, the Euclidean distances between the OoS point and training set points have to be calculated. The Euclidean distance calculation takes 0.002 M and 0.036 M multiplications for 2D and 30D input features, respectively. It should be noted that for PCA, the complexity of Log-Euclidean feature generation (i.e, 0.032 M) is added.

The full dissimilarity OoS (NLDR/ fingerprint, Full) complexity is obtained by (27) when all 1200 training points are used as landmarks (i.e, $L_1 = U, L_2 = L_3 = 0$). The hierarchical OoS (NLDR/ fingerprint, Hier.) is assumed to use 180 dissimilarity calculations within 3 steps. As mentioned earlier, PCA has low complexity and the corresponding values for computing 2D and 30D features are listed in the PCA row.

The complexity-accuracy trade-off of all OoS methods and predictors in terms of millions of multiplication and RMSE is illustrated in Fig. 9. On the rightmost part, we have the GPR predictor for different DR input features. Even though the best RMSE is obtained by GPR predictor, the complexity is the highest. Using non-linear DR and NN predictors gives the best performance in terms of both complexity and accuracy. The first two points (from the left) on the t-SNE and Laplacian Eigenmaps curves correspond to NN predictor with hierarchical and full dissimilarity OoS. PCA has shown the lowest complexity. However, its performance is worse compared to non-linear DR techniques. The points on the PCA curve are showing the performance of 2D, 3D, and 30D using NN predictor and the last one corresponds to GPR with 30D input. The first 3 points on CSI fingerprint curve show different NN structures as predictors which have a low complexity but higher RMSE value. Since Levenberg-Marquardt algorithm is not able to process a large input vector feature and is efficient for moderate sized NNs, for the raw CSI fingerprint, we have used Adam optimizer in training. For this raw feature, a deep NN with 5 layers has been used with 100, 200, and 300 neurons in each layer. By increasing the complexity of the NN, the RMSE is reduced but it is still higher than any CC-based prediction. The last point on the CSI fingerprint curve is full dissimilarity OoS using KNN and shows a comparable performance.

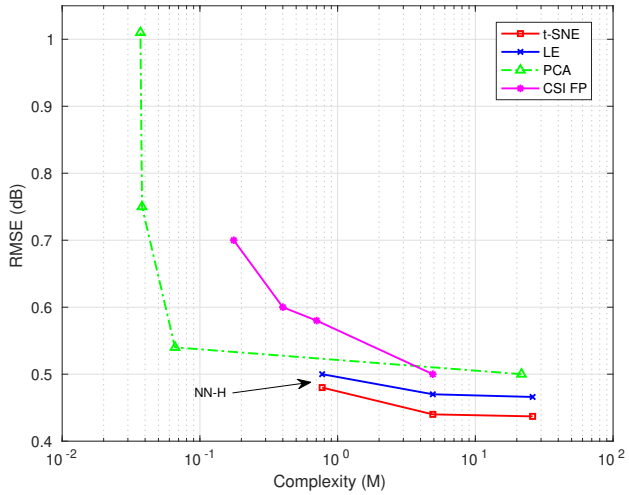



Fig. 9: Complexity vs. RMSE accuracy trade-off for different beam SNR prediction algorithms and beam CSI features. The average accuracy is in terms of RMSE and the complexity is in million of multiplication operation.

For the current scenario, the maximum ML inference complexity is for GPR, requiring 21 M multiplications. Since all the computation is performed on the BS side, this is in scope of contemporary BS hardware. As the computational capability of modern commercial processors is measured in Giga Floating Point Operations per Second (GFLOPS), the inference time is small even for the most complex inference method. For instance, a commercial FPGA board (Xilinx Alveo U50) has been shown to provide 200 GFLOPS, for a general-purpose ML accelerator [46]. The 21 M operations required by GPR would then take $\sim \frac{1}{10}$ ms; the considered ML method can be used in real time.

## VII. Conclusions

In this paper, we have considered SNR prediction for combinations of BS and UE beams in mmWave systems using a BS beam-specific CSI feature. We have devised a network centric beam handover mechanism in which all processing is performed at the BS. During the offline phase, beam CSI features and SNRs are collected and SNR predictors are trained based on the annotated CSI features. In the online phase, target beam's SNR of the new UE is predicted from the received CSI at the BS and serving beam SNR. A heuristic, less computationally complex OoS algorithm has been devised. K-nearest neighbors, Gaussian process regression and neural networks have been used for predicting the SNR mapping function. The SNR RMSE in dB has been used as the performance metric. Results have shown excellent beam SNR prediction accuracy using GPR, and NN. The performance loss of CC-based beam SNR prediction, as compared to prediction based on physical location, is negligible if the physical location is accurately known. When it comes to CC-dimensionality, there is little gain from using a CC with more than 3 dimensions. We have analyzed the complexity-accuracy trade-off for different predictors and DR methods. The Pareto front is given by PCA for different dimensions with NN predictor, and $t$-SNE dimensional reduction, with different OoS mechanisms and predictors. The DR-based methods outperform vanilla CSI fingerprinting with a wide margin in the complexity-accuracy domain. Simulation results demonstrate that the proposed CC-assisted approach can significantly reduce complexity and accurately enable beam management.

## Acknowledgments

This work was funded in part by the WINDMILL project funded by the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie (grant 813999), and by the Academy of Finland (grant 319484). We acknowledge the computational resources provided by the Aalto Science-IT project.

## References

[1] P. Kazemi, T. Ponnada, H. Al-Tous, Y.-C. Liang, and O. Tirkkonen, "Channel charting based beam SNR prediction," in *Proc. of Eur. Conf. on Networks and Commun. & 6G Summit (EuCNC/6G Summit)*, 2021, pp. 72–77.

[2] T. S. Rappaport, S. Sun, R. Mayzus, H. Zhao, Y. Azar, K. Wang, G. N. Wong, J. K. Schulz, M. Samimi, and F. Gutierrez, "Millimeter wave mobile communications for 5G cellular: It will work!" *IEEE Access*, vol. 1, pp. 335–349, May. 2013.

[3] M. Giordani, M. Polese, A. Roy, D. Castor, and M. Zorzi, "A tutorial on beam management for 3GPP NR at mmWave frequencies," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 173–196, Sep. 2019.

[4] X. Gao, L. Dai, S. Han, C. I, and R. W. Heath, "Energy-efficient hybrid analog and digital precoding for mmWave MIMO systems with large antenna arrays," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 4, pp. 998–1009, Apr. 2016.

[5] N. W. Sung and Y. S. Choi, "Fast intra-beam switching scheme using common contention channels in millimeter-wave based cellular systems," in *Proc. of Int. Conf. Advanced Commun. Technol. (ICACT)*, 2016, pp. 760–765.

[6] M. Giordani, M. Mezzavilla, C. N. Barati, S. Rangan, and M. Zorzi, "Comparative analysis of initial access techniques in 5G mmWave cellular networks," in *Proc. Ann. Conf. Info. Sci. Syst. (CISS)*, 2016, pp. 268–273.

[7] P. Wang, Y. Li, L. Song, and B. Vucetic, "Multi-gigabit millimeter wave wireless communications for 5G: from fixed access to cellular networks," *IEEE Commun. Mag.*, vol. 53, no. 1, pp. 168–178, Jan. 2015.

[8] E. Dahlman, S. Parkvall, and J. Skold, *5G NR: The Next Generation Wireless Access Technology*. Elsevier Science, 2020.

[9] Z. Xiao, T. He, P. Xia, and X.-G. Xia, "Hierarchical codebook design for beamforming training in millimeter-wave communication," *IEEE Trans. Wireless Commun.*, vol. 15, no. 5, pp. 3380–3392, May. 2016.

[10] A. Alkhateeb, O. El Ayach, G. Leus, and R. W. Heath, "Channel estimation and hybrid precoding for millimeter wave cellular systems," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 5, pp. 831–846, Oct. 2014.

[11] Y. Guo, Z. Wang, M. Li, and Q. Liu, "Machine learning based mmWave channel tracking in vehicular scenario," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, 2019, pp. 1–6.

[12] H. Liu, H. Kim, S. Moon, and I. Hwang, "Deep learning for channel estimation and tracking in vehicular to infrastructure communications," in *Proc. Int. Conf. Inf. Commun. Technol. Convergence (ICTC)*, 2020, pp. 777–782.

[13] P. Zhou, X. Fang, X. Wang, Y. Long, R. He, and X. Han, "Deep learning-based beam management and interference coordination in dense mmWave networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 1, pp. 592–603, Jan. 2019.

[14] H. Huang, Y. Peng, J. Yang, W. Xia, and G. Gui, "Fast beamforming design via deep learning," *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 1065–1069, Jan. 2020.

[15] W. Ma, C. Qi, and G. Y. Li, "Machine learning for beam alignment in millimeter wave massive MIMO," *IEEE Wireless Commun. Lett.*, vol. 9, no. 6, pp. 875–878, Jun. 2020.

[16] H. Echigo, Y. Cao, M. Bouazizi, and T. Ohtsuki, "A deep learning-based low overhead beam selection in mmWave communications," *IEEE Trans. Veh. Technol.*, vol. 70, no. 1, pp. 682–691, Jan 2021.

[17] M. Alrabeiah and A. Alkhateeb, "Deep learning for mmWave beam and blockage prediction using Sub-6 GHz channels," *IEEE Trans. Commun.*, vol. 68, no. 9, pp. 5504–5518, Jun 2020.

[18] M. Arvinte, M. Tavares, and D. Samardzija, "Beam management in 5G NR using geolocation side information," in *Proc. of Ann. Conf. Inf. Sci. Syst. (CISS)*, 2019, pp. 1–6.

[19] Y. Heng and J. G. Andrews, "Machine learning-assisted beam alignment for mmWave systems," in *IEEE Global Commun. Conf. (GLOBECOM)*, 2019, pp. 1–6.

[20] S. Tomasin, C. Mazzucco, D. De Donno, and F. Cappellaro, "Beam-sweeping design based on nearest users position and beam in 5G mmWave networks," *IEEE Access*, vol. 8, pp. 124 402–124 413, Jun 2020.

[21] M. Dias, A. Klautau, N. González-Prelcic, and R. W. Heath, "Position and LIDAR-aided mmWave beam selection using deep learning," in *Proc. of IEEE Int. Workshop Signal Process. Advances Wireless Commun. (SPAWC)*, 2019, pp. 1–5.

[22] 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*, vol. 6, pp. 47 682–47 698, Aug 2018.

[23] J. Deng, S. Medjkouh, N. Malm, O. Tirkkonen, and C. Studer, "Multipoint channel charting for wireless networks," in *Proc. of Asilomar Conf. on Signals, Systems, and Computers*, 2018, pp. 286–290.

[24] P. Kazemi, H. Al-Tous, C. Studer, and O. Tirkkonen, "SNR prediction in cellular systems based on channel charting," in *Proc. of the IEEE Int. Conf. Commun. and Networking (ComNet)*, 2020, pp. 1–8.

[25] E. G. Larsson, O. Edfors, F. Tufvesson, and T. L. Marzetta, "Massive MIMO for next generation wireless systems," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 186–195, Feb 2014.

[26] M. Rebato, L. Resteghini, C. Mazzucco, and M. Zorzi, "Study of realistic antenna patterns in 5G mmWave cellular scenarios," in *Proc. of IEEE Int. Conf. Commun. (ICC)*, 2018, pp. 1–6.

[27] D. Neumann, M. Joham, and W. Utschick, "Covariance matrix estimation in massive MIMO," *IEEE Signal Process. Lett.*, vol. 25, no. 6, pp. 863–867, Apr 2018.

[28] A. Maatouk, S. E. Hajri, M. Assaad, H. Sari, and S. Sezginer, "Graph theory based approach to users grouping and downlink scheduling in FDD massive MIMO," in *Proc. of the IEEE Int. Conf. Commun. (ICC)*, 2018, pp. 1–7.

[29] S. Bonnabel and R. Sepulchre, "Riemannian metric and geometric mean for positive semidefinite matrices of fixed rank," *SIAM J. Matrix Analysis and Applications*, vol. 31, no. 3, pp. 1055–1070, 2010.

[30] T. Ponnada, P. Kazemi, H. Al-Tous, Y.-C. Liang, and O. Tirkkonen, "Best beam prediction in non-standalone mmWave systems," in *Proc. of Eur. Conf. on Networks and Commun. & 6G Summit (EuCNC/6G Summit)*, 2021, pp. 532–537.

[31] H. Hotelling, "Analysis of a complex of statistical variables into principal components." *J. Educational psychology*, vol. 24, no. 6, p. 417, 1933.

[32] Y. Ma and Y. Fu, *Manifold learning theory and applications*. CRC press, Boca Raton, FL, 2012, vol. 434.

[33] M. Belkin and P. Niyogi, "Laplacian Eigenmaps for dimensionality reduction and data representation," *Neural Computation*, vol. 15, p. 1373–1396, 2003.

[34] L. v. d. Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Machine Learning Research*, vol. 9, pp. 2579–2605, Nov 2008.

[35] J. Venna and S. Kaski, "Neighborhood preservation in nonlinear projection methods: An experimental study," in *Proc. of the Int. Conf. on Artif. Neural Networks (ICANN)*, 2001, pp. 485–491.

[36] L. Maaten, E. Postma, and H. Herik, "Dimensionality reduction: A comparative review," 2008.

[37] M. Binois and N. Wycoff, "A survey on high-dimensional gaussian process modeling with application to bayesian optimization," *ACM Trans. Evol. Learning and Optimization.*, vol. 2, no. 2, pp. 1–26, aug 2022.

[38] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

[39] M. T. Hagan and M. B. Menhaj, "Training feedforward networks with the marquardt algorithm," *IEEE Trans. Neural Netw.*, vol. 5, no. 6, pp. 989–993, Nov 1994.

[40] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Int. Conf. Learning Representations*, 2015, pp. 1–15.

[41] 3GPP, "Study on channel model for frequencies from 0.5 to 100 GHz," 3rd Generation Partnership Project (3GPP), Technical Specification TS 38.901, Jan. 2018, version 14.3.0.

[42] H. Zhang, P. Wang, X. Gao, Y. Qi, and H. Gao, "Out-of-sample data visualization using bi-kernel t-SNE," *Information Visualization*, vol. 20, no. 1, pp. 20–34, Jan 2021.

[43] R. P. Brent, "Multiple-precision zero-finding methods and the complexity of elementary function evaluation," in *Analytic computational complexity*. Elsevier, 1976, pp. 151–176.

[44] S. Jaeckel, L. Raschkowski, K. Borner, and L. Thiele, "QuaDRiGa: A 3-D multi-cell channel model with time evolution for enabling virtual field trials," *IEEE Trans. Antennas Propag.*, vol. 62, no. 6, pp. 3242–3256, Jun. 2014.

[45] M. R. Mosavi and I. EmamGholipour, "De-noising of GPS receivers positioning data using wavelet transform and bilateral filtering," *Wireless Pers. Commun.*, vol. 71, no. 3, pp. 2295–2312, Aug 2013.

[46] J.-H. Kim, S. Lee, S. Moon, S. Yoo, and J.-Y. Kim, "A 409.6 GOPS and 204.8 GFLOPS mixed-precision vector processor system for general-purpose machine learning acceleration," in *Poc. IEEE Asian Solid-State Circuits Conference*, 2022, pp. 1–3.