Deep-learning based high-precision localization with massive MIMO

Guoda Tian, Student Member, IEEE, Ilayda Yaman, Student Member, IEEE, Michiel Sandra, Student Member, IEEE, Xuesong Cai, Senior Member, IEEE, Liang Liu, Member, IEEE, and Fredrik Tufvesson, Fellow, IEEE

1Dept. of Electrical and Information Technology, Lund University, Sweden

Corresponding author: Guoda Tian (email: guoda.tian@eit.lth.se) and Xuesong Cai (email: xuesong.cai@eit.lth.se).

This work has been funded by Ericsson AB, the Horizon Europe Framework Programme under the Marie Skłodowska-Curie grant agreement No. 101059091, the Swedish Research Council (Grant No. 2022-04691), and the strategic research area ELLIIT, Excellence Center at Linköping-Lund in Information Technology.

ABSTRACT High-precision localization and machine learning (ML) are envisioned to be key technologies in future wireless systems. This paper presents an ML pipeline to solve localization tasks. It consists of multiple parallel processing chains, each trained using a different fingerprint to estimate the position of the user equipment. In this way, ensemble learning can be utilized to fuse all chains to improve localization performance. Nevertheless, a common problem of ML-based techniques is that network training and fine-tuning can be challenging due to the increase in network sizes when applied to (massive) multiple-input multiple-output (MIMO) systems. To address this issue, we utilize a subarray-based approach. We divide the large antenna array into several subarrays, feeding the fingerprints of the subarrays into the pipeline. In our case, such an approach eases the training process while maintaining or even enhancing the performance. We also use the Nyquist sampling theorem to gain insight on how to appropriately sample and average training data. Finally, an indoor measurement campaign is conducted at 3.7 GHz using the Lund University massive MIMO testbed to evaluate the approaches. Localization accuracy at a centimeter level has been reached in this particular measurement campaign.

INDEX TERMS Channel measurements, Deep learning, Localization, Massive MIMO

I. INTRODUCTION Cellular-based localization is expected to pave the way for various location-aware applications such as robotic navigation, emergency healthcare, and smart transportation [1]–[7]. The technology has undergone a significant improvement over the years, and high-precision wireless localization has currently been included as a key feature in the current fifth generation new radio (NR) standard, with strict requirements on localization accuracy [8].

Traditional localization approaches include proximity, triangulation (trilateration), fingerprint matching, and simultaneous localization and mapping [7]. Proximity approaches examine whether user equipment (UE) is close to pre-known locations by analyzing received wireless signal characteristics such as the received signal strength indicator (RSSI). Triangulation or trilateration technology is used to estimate UE locations from delays or angles according to geometry. The general concept of fingerprint-based localization is to establish a radio map for the area of interest by storing channel features or fingerprints. The UE coordinates are estimated by comparing the received fingerprints with the previously stored fingerprints. Furthermore, with the aid of ultra-wideband (UWB) [9] and/or massive multiple-input multiple-output (MIMO) systems, it is possible to improve positioning accuracy due to the high delay resolution in UWB and the high angular resolution in massive MIMO [10]–[16]. For example, the work in [10], [15], [16] proposed novel estimators to jointly estimate angles and positions with large-scale arrays. The authors in [11]–[13] provided solutions for localization by designing tracking filters to exploit and track important propagation channel characteristics, i.e. the autocorrelation function of the received signal and the phase of multipaths, respectively. Especially, [12] validated their methods via a real massive MIMO testbed and showed that localization accuracy can be significantly enhanced with the 40 MHz bandwidth. [14] presented a direct localization method considering localization as a joint optimization problem, which bypasses the channel estimation step and still achieves good positioning accuracy.
All of the aforementioned localization methods belong to the traditional signal processing family. The main challenges are high algorithm complexity and requirements of the base station (BS) array calibration [7]. On the other hand, machine learning (ML) based localization algorithms have gained significant interest [17]–[32]. It is essential to appropriately select both fingerprints and algorithms. One can choose either the raw transfer function [17]–[19] or various channel fingerprints such as RSS, power delay profile (PDP), angular spectrum, correlation function, etc, [20]–[28] as learning features. Moreover, a variety of ML algorithms have been investigated, which can be mainly classified into two categories, namely the traditional ML family such as the K-nearest neighbors (KNN), support vector machines, kernel methods, random forest, Gaussian process regression, [21]–[23], [32], and the deep learning family [17], [18], [20], [24]–[31]. Considering the features of massive MIMO systems, there is also great potential to apply ML techniques with massive MIMO systems to solve localization tasks. Early research [17], [18], [24] used convolutional neural networks (CNN) for localization. The work in [30] proposed an algorithm that trains an autoencoder to first calibrate the antenna array. Then, the angle spectrum is computed as a training feature. The work in [22] applied Gaussian regression to perform localization with distributed massive MIMO systems.

However, there are still some research gaps that need to be further addressed: i) Most of the existing ML-based localization algorithms directly output the position of the UE without considering the uncertainty of the estimation, thus lacking effective information fusion from different channel fingerprints. ii) The size of the neural networks increases significantly with the increasing number of antennas. This may hinder the training and fine-tuning of the network. Therefore, it is essential to develop efficient localization algorithms that are suitable for the massive MIMO system. iii) A theoretical analysis of the necessary training density is missing in the literature. It is important to investigate the density of the training sample under different channel conditions, as training data collection is a time-consuming task. To address those limitations, our main contributions are as follows.1

1A preliminary version of this work [33] has been presented at the 2023 IEEE International Conference on Communication. Unlike [33], this paper presents new material on the subarray method and a detailed analysis of the necessary training density. In addition, the pipeline in this paper is used to estimate both the UE position and error variances.

- By dividing the whole array into subarrays, the network size can be reduced, which facilitates the training process while improving the localization performance.
- We apply the Nyquist sampling theorem to analyze how to appropriately collect and average training data.
- Finally, an indoor measurement campaign with a massive MIMO testbed was conducted to evaluate our approach. The results show that our pipeline can reach centimeter-level positioning accuracy with only 20 MHz bandwidth for this measurement campaign.

The remainder of this paper is organized as follows. In Section II, we introduce the signal model and briefly discuss the selected fingerprints. In Section III, we present the localization algorithms. Section IV illustrates the measurement campaign, and Section V presents the results. Finally, conclusive remarks are included in Section VI.

II. System Model and Fingerprint Generation

We consider the uplink of a single user massive MIMO system, which uses orthogonal frequency division multiplexing (OFDM) with \( F \) subcarriers. The UE has one antenna, while the BS is equipped with \( M \) antennas. Each antenna element is connected to an RF and a digital processing chain, which allows the BS to simultaneously process the received signals from all antennas. We assume that the UE moves at walking speed, and the 2-D position of this UE is given by \( \mathbf{p}_i(z_0, y_0) \in \mathbb{R}^2 \), shortened to \( \mathbf{p}_i \) in the following sections. Taking into account the propagation channel, the transfer function matrix \( \mathbf{Y}_{\mathbf{p}_i} = [\mathbf{Y}_{\mathbf{p}_i,1}, \ldots, \mathbf{Y}_{\mathbf{p}_i,F}] \in \mathbb{C}^{M \times F} \) for all subcarriers, corresponding to the position of the UE \( \mathbf{p}_i \), can be written as

\[
\mathbf{Y}_{\mathbf{p}_i} = \mathbf{H}_{\mathbf{p}_i} \odot \Gamma + \mathbf{N}, \tag{1}
\]

where \( \mathbf{H}_{\mathbf{p}_i} \in \mathbb{C}^{M \times F} \) represents the uplink wireless propagation channel, \( \Gamma \in \mathbb{C}^{M \times F} \) the complex coefficients (amplitude scaling and phase drift) of all the \( M \) RF chains and \( F \) sub-carriers. Additionally, \( \odot \) is the Hadamard product, and \( \mathbf{N} \in \mathbb{C}^{M \times F} \) denotes receiver noise at all \( M \) RF chains. When UE moves, a total of \( T \) snapshots are recorded and \( T \) different receive matrices \( \mathbf{Y}_{\mathbf{p}_i} \) are collected. Our aim is to find a functional relationship between \( \mathbf{Y}_{\mathbf{p}_i} \) and \( \mathbf{p}_i \), which falls into the category of a regression (estimation) task.

ML-based localization algorithms have the potential to achieve good performance if adequate channel fingerprints are selected as the input to the algorithms. Such fingerprints can be extracted from the raw received transfer function \( \mathbf{Y}_{\mathbf{p}_i} \). In this paper, we analyze two fingerprints, namely the spatial covariance matrix and the truncated channel impulse response (CIR), since they can be achieved with even an uncalibrated array.

A. Spatial Covariance matrix

It is sometimes challenging to extract calibrated fingerprints such as AoA due to the presence of the RF chain matrix \( \Gamma \), see (1). Therefore, we consider using the covariance matrix \( \mathbf{C}_i = \mathbb{E} \{ \mathbf{y}_{\mathbf{p}_i} \mathbf{y}_{\mathbf{p}_i}^H \} \in \mathbb{C}^{M \times M} \) as a fingerprint. The
main diagonal elements of $\mathbf{C}_i$ (auto-correlation) indicate
the received signal power for each antenna, whereas the
off-diagonal elements of $\mathbf{C}_i$ represent the cross-correlation
between different antennas. Note that typically one can only
estimate the covariance matrix in practice with a limited
number of samples to conduct the expectation operation.
Suppose that for each position $p_i$, there exist in total $N_{p_i}$
positions in the neighborhood region of $p_i$, whose channel
responses are accessible. Those $N_{p_i}$ samples are inside a
regular area, with $p_i$ as the center and $d$ as the diameter,
i.e., $|p_j - p_i|^2 \leq \frac{d^2}{2}$, $j = 1, 2, ..., N_{p_i}$. We then define the
sample covariance matrix $\tilde{\mathbf{C}}_{i,N_{p_i}} \in \mathbb{C}^{M \times M}$ to estimate $\mathbf{C}_i$.
Specifically,

$$
\tilde{\mathbf{C}}_{i,N_{p_i}} = \frac{1}{N_{p_i}} \sum_{j=1}^{N_{p_i}} \mathbf{Y}_{p_j} \mathbf{Y}_{p_j}^H.
$$

(2)

As shown in (2), $\tilde{\mathbf{C}}_{i,N_{p_i}}$ depends on $N_{p_i}$ and thus $d$.
A special case is that if $d = 0$ and $N_{p_i} = 1$, $\mathbf{C}_i$ is
estimated by only correlating across all subcarriers of $\mathbf{Y}_{p_i}$
at a fixed position $p_i$. We name this specific matrix the
one-sample covariance matrix. Note that it is challenging to
estimate $\mathbf{C}_i$ with this matrix for a narrowband system, since
channel responses with respect to different subcarriers are
strongly correlated. In contrast, when $d$ is large enough, the fingerprint $\tilde{\mathbf{C}}_{i,N_{p_i}}$ changes much more slowly than the one-sample covariance matrix with
the movement of the UE, since the influence of small-scale
fading is reduced due to the average operation. Due to this,
fewer training samples are needed.

Since the sample covariance matrix $\tilde{\mathbf{C}}_{i,N_{p_i}}$ is a Hermitian
matrix, i.e., $\tilde{\mathbf{C}}_{i,N_{p_i}} = \tilde{\mathbf{C}}_{i,N_{p_i}}^H$, the upper minor diagonal
elements contain the same information as the lower. To
decrease the computation complexity, we introduce another
matrix $\tilde{\mathbf{C}}_{i,N_{p_i}} \in \mathbb{R}^{M \times M}$ and a vector $\tilde{\mathbf{e}}_{i,N_{p_i}} \in \mathbb{R}^{M^2}$ as

$$
\tilde{\mathbf{C}}_{i,N_{p_i}} = \text{tril}\{\mathbb{R}\{\tilde{\mathbf{C}}_{i,N_{p_i}}\}\} + \text{sltri}\{\mathbb{I}\{\tilde{\mathbf{C}}_{i,N_{p_i}}\}\}
$$

(3)

where $\mathbb{R}\{\cdot\}$ and $\mathbb{I}\{\cdot\}$ denote the operation to take the real and
imaginary parts of a given matrix, respectively. $\text{tril}\{\cdot\}$ represents a matrix operation that replaces all values
above the diagonal as zero while maintaining the other
matrix elements. The operation $\text{sltri}\{\cdot\}$ keeps all elements
below the diagonal and substitutes all the remaining matrix
elements (including the diagonal elements) for zero. The
$\text{vec}\{\cdot\}$ operator denotes the operation of converting a matrix
to a vector.

B. Truncated channel impulse response

The fingerprint $\tilde{\mathbf{C}}_i$ does not contain channel information
from the delay domain; however, it is still important to
utilize the delay information to further improve the accuracy
of localization. To this end, the truncated CIR matrix

$$
\mathbf{\Xi} \in \mathbb{C}^{M \times L}
$$

is generated by calculating the inverse discrete Fourier transform (IDFT) along each row of $\mathbf{Y}_{p_i}$, followed
by choosing the first $L$ delay elements. We introduce a vector $\xi \in \mathbb{R}^{2ML}$, which includes all elements of $\mathbf{\Xi}$. Specifically, $\xi = \{\text{vec}(\mathbb{R}(\mathbf{\Xi}))^T, \text{vec}(\mathbb{I}(\mathbf{\Xi}))^T\}^T$.

III. ML-based localization approach

A. Neural Network Basics

Neural networks have been widely used to solve various
tasks such as channel estimation, wireless sensing, etc.,
owing to their excellent abilities to learn non-linear complex
models [34]. These models can generally be represented as a
multivariate function $f : \mathbb{R}^{V_1} \rightarrow \mathbb{R}^{V_2}$, where $V_1$ and
$V_2$ represent the dimensions of the learning characteristics
and the goals, respectively. An example of a typical fully
connected neural network (FCNN) is illustrated in Fig. 1,
consisting of an input layer, several hidden layers, and an
output layer. Regarding the input and output layers, the
number of their nodes is identical to $V_1$ and $V_2$, respectively.
Specific to this 2-D localization task, we view the output of
the neural network as a Gaussian distribution function, which
can be determined by the estimated position of the $\hat{p} = [\hat{p}_x, \hat{p}_y]^T \in \mathbb{R}^2$ and the variance ($\sigma^2 = [\sigma_x^2, \sigma_y^2]^T \in \mathbb{R}^2$).

Two processes are usually involved when training a neural
network, namely the forward and backward propagation
processes. In the forward propagation process, the input
signals enter the neural network through the input layer. Then
it propagates through multiple hidden layers and ultimately
reaches the output layer. At each layer, the output of a node
is determined by the inputs from the previous layers, the
respective weights and biases, and a non-linear activation
function that is specific to that node. For example, we
suppose that an FCNN has $\gamma_i$ nodes in the $i$-th layer, the
values of which are collected by a signal vector $x_i =
[x_{i,1}^1, ..., x_{i,\gamma_i}^1] \in \mathbb{R}^\gamma_i$. The value of the $k$-th node is calculated
by applying a weight vector $w = [w_{i,1}^{-1}, ..., w_{i,\gamma_i-1}^{-1}] \in \mathbb{R}^{\gamma_i-1}$
to the signal vector $x_{i-1}$ in the previous layer. Specifically,

$$
x_{i,k}^1 = g_i\left(\sum_{j=1}^{\gamma_i-1} x_{i,j}^1 w_{i,j}^{-1}\right) + b_i,
$$

(4)
where $b_i$ represents an optional bias term and $g_i(.)$ the activation function. The same propagation pattern is followed for each layer, generating an output vector $\nu = [\hat{p}_i, \hat{\sigma}_i^2]$. To train the network, it is important to select an appropriate training criterion, or the so-called loss function. A popular criterion is the mean-square error (MSE), which measures the differences between estimated localization coordinates and the ground truth labels. However, the uncertainty of the predictions is not evaluated by MSE and therefore we consider the negative log-likelihood (NLL) loss function instead [35]. Suppose that the entire training dataset contains in total $N_{tr}$ training samples. For the $i$-th sample, the network outputs estimate the UE coordinate $\hat{p}_i = [\hat{p}_{xi}, \hat{p}_{yi}]^T \in \mathbb{R}^2$ and the variance vector $\hat{\sigma}_i^2 = [\hat{\sigma}_{xi}^2, \hat{\sigma}_{yi}^2]^T \in \mathbb{R}^2$, while the ground truth is $p_i = [p_{xi}, p_{yi}]^T \in \mathbb{R}^2$. Taking into account all $N_{tr}$ training samples, the loss function $\psi$ is

$$\psi = \frac{1}{2N_{tr}} \sum_i \left( \frac{\log \hat{\sigma}_{xi}^2 \hat{\sigma}_{yi}^2}{2} + \frac{(p_{xi} - \hat{p}_{xi})^2}{2\hat{\sigma}_{xi}^2} + \frac{(p_{yi} - \hat{p}_{yi})^2}{2\hat{\sigma}_{yi}^2} \right).$$

(5)

Observe that $\psi$ can be negative owing to the log term. After selecting the training criterion, all hyperparameters, namely all weights and bias terms in (4) in each layer, need to be fine-tuned to minimize $\psi$. This optimization procedure can be carried out by backward propagation, which propagates the error signal back to each neural network layer to update the weights. Due to page limitations, we avoid presenting the mathematical derivations; however, the relevant material can be found in [34].

As an evaluation procedure, we collect the test datasets and select the NLL loss as the evaluation criterion [35]. As indicated in (5), an under-confident variance estimate results in the increase of the first term, while an over-confident variance results in the increase of the second and third terms.

**B. ML-based Localization Pipeline**

We apply the idea of ensemble learning to the localization task. As a popular ML approach, ensemble learning targets performance improvements by training multiple base learners and then fusing their outputs [36]. Each learner itself should individually deliver decent results, and it is also important to embed enough diversity when selecting those base learners. Based on this insight, we apply the processing pipeline illustrated in Fig. 2. We select $\chi$ fingerprints and feed each fingerprint to an individual processing chain. Those fingerprints can be either the entire covariance matrix or submatrices (see the subarray method in the following section) or the truncated CIR. Each processing chain estimates 2-D UE coordinates as well as the variances. Suppose that the $j$-th processing chain estimates the position of the UE and the variance as $\hat{p}_{i,j} = [\hat{p}_{x_{i,j}}, \hat{p}_{y_{i,j}}] \in \mathbb{R}^2$ and $\hat{\sigma}_{i,j}^2 = [\hat{\sigma}_{x_{i,j}}^2, \hat{\sigma}_{y_{i,j}}^2] \in \mathbb{R}^2$. By fusing all $\chi$ processing chains according to the maximum ratio combining (MRC) approach [37], $\hat{p}_i$ and $\hat{\sigma}_i^2$ are calculated as:

$$\hat{\sigma}_{xi}^2 = \sum_j \frac{1}{\hat{\sigma}_{x_{i,j}}^2}, \quad \hat{\sigma}_{yi}^2 = \sum_j \frac{1}{\hat{\sigma}_{y_{i,j}}^2},$$

(6)

$$\hat{p}_{xi} = \hat{\sigma}_{xi}^2 (\sum_j \frac{1}{\hat{\sigma}_{x_{i,j}}^2} \hat{p}_{x_{i,j}}), \quad \hat{p}_{yi} = \hat{\sigma}_{yi}^2 (\sum_j \frac{1}{\hat{\sigma}_{y_{i,j}}^2} \hat{p}_{y_{i,j}}).$$

(7)

However, the estimated variance by (6) may be overconfident, especially when the network is overfitted. According to (6), $\hat{\sigma}_{xi}^2$ and $\hat{\sigma}_{yi}^2$ are less than each individual $\hat{\sigma}_{x_{i,j}}^2$ and $\hat{\sigma}_{y_{i,j}}^2$, respectively. This may increase the NLL, since $\hat{\sigma}_{x_{i,j}}^2$ and $\hat{\sigma}_{y_{i,j}}^2$ act as the denominators of the second and third terms, respectively. To address this issue, we multiply a factor $\chi$ with $\hat{\sigma}_i^2$ to get the modified vector $\hat{\sigma}_{i,\text{mod}}^2 \in \mathbb{R}^2$, which is the harmonic averages of all estimated variances. Specifically,

$$\hat{\sigma}_{i,\text{mod}}^2 = \chi \hat{\sigma}_i^2. \quad \text{(8)}$$

**C. Training on the subarrays**

The size of the neural network increases significantly with the number of antennas, which leads to a risk of over-fitting problems. To address this problem, subarray methods can be considered. In this paper, we use the covariance matrix as an example, however, this method can be generalized to other fingerprints. We assume that an $M_1 \times M_2$ rectangular antenna array is equipped at the BS side. The spatial correlation between channel responses of two antennas is reduced to a large extent if they are separated larger than the coherence distance. Enlightened by this fact, we divide the whole antenna into $I$ subarrays and train $I$ neural networks instead of feeding the whole covariance matrix into the processing chain. The subarrays are selected as follows.

We define a rectangular sliding kernel with a size of $N_1$ rows and $N_2$ columns, which captures in total $N_1N_2$ antennas. We first place the kernel in the upper left corner of the whole array, to select antennas that belong to the first $N_1$ rows and $N_2$ columns. The sliding kernel then moves $S_2$ columns to the right and assigns its antennas to a new group. When the sliding kernel reaches the last column, it moves $S_1$ rows downward, followed by moving $S_2$ columns to the left until the kernel hits the first column. This procedure is repeated until the entire array is scanned by the kernel and $I = ([M_1 - N_1]/S_1) + 1) ([M_2 - N_2]/S_2) + 1$.
training groups are formulated, where \( \lfloor . \rfloor \) denotes the floor function. We then formulate \( I \) sample covariance matrices that correspond to the UE position \( p_i \), which are denoted as \( \mathbf{C}^{1}_{i,N_{P_{i}}}, \ldots, \mathbf{C}^{I}_{i,N_{P_{i}}} \in \mathbb{C}^{N_{1}N_{2} \times N_{1}N_{2}} \). These covariance matrices are fed into the pipeline shown in Fig. 3, to obtain the estimated UE positions and variances.

**D. Training Density**

A fundamental question of ML-based localization is to determine the number of necessary training samples. According to the Nyquist sampling theorem, insufficient numbers of training samples result in aliasing, which has a detrimental effect on system performance. To figure out the necessary training density, we apply this theorem to investigate maximum separation distances between two adjacent training samples during the training process. Some degree of aliasing is allowed since our task is to estimate the UE localization, rather than perfectly reconstructing the propagation channels. For convenience, we confine the scope of our approach to uniform sampling.

We consider the vector \( \mathbf{c}_{i,N_{P_{i}}} \) generated by (3), which varies when UE moves to \( Q \) different positions. To simplify this analysis, the UE position labels are assumed to be evenly distributed along a straight line, and the geographical distances between these \( Q \) positions are \( \delta_{d} \). We define a matrix \( \mathbf{C} = [\mathbf{c}_{1,N_{P_{1}}}, \ldots, \mathbf{c}_{Q,N_{P_{Q}}}] \in \mathbb{R}^{M^{2} \times Q} \) to collect all those \( Q \) channel response vectors. By performing the 1-D discrete Fourier transform of \( \mathbf{C} \) along the horizontal axis, we can formulate a matrix \( \mathbf{Ψ} \in \mathbb{C}^{M^{2} \times Q} \) that characterizes the channel variations along those \( Q \) positions. Specifically,

\[
\mathbf{Ψ} = \mathbf{CΛ},
\]

where \( \mathbf{Λ} \in \mathbb{C}^{Q \times Q} \) is the DFT matrix. We then define a spectrum window \( L \), which covers consecutive \( L \) columns w.r.t. the lower frequency components of \( \mathbf{Ψ} \). Once \( L \) is selected, the corresponding sampling distance \( \Delta_{d} \) between two adjacent samples can be calculated as

\[
\Delta_{d} = Q \delta_{d} / L.
\]

We then form those \( L \) columns into a new matrix \( \mathbf{Ψ}^{L} \in \mathbb{C}^{M^{2} \times L} \) and define \( \eta \) as the ratio between the Frobenius norm of \( \mathbf{Ψ}^{L} \) and \( \mathbf{Ψ} \), that is \( \eta = \|\mathbf{Ψ}^{L}\|_{F} / \|\mathbf{Ψ}\|_{F} \). Here, \( \eta \) shows the extent of aliasing of different sampling intervals. In the following sections, we will analyze the influence of \( \eta \) on localization accuracy and discuss the choice of \( \Delta_{d} \).

**IV. Massive MIMO Measurement Campaign**

To validate our approach, an indoor measurement campaign was carried out in the Lund University Humanities Lab motion capture studio. Photos of the mocap studio are shown in Fig. 4. We give a brief introduction to the measurement campaign, while more details can be found in [38].

**A. Introduction to the measurement campaign**

In this measurement, we use a robot to carry the UE with a single dipole antenna that is placed at a height of 1.73 m. The parameter settings of our measurement system are similar to those of the LTE system. Specifically, our system occupies 20 MHz bandwidth which consists of 100 physical resource blocks (PRBs), and each PRB has 12 subcarriers. The subcarrier resource is allocated to multiple users in such a way that each UE occupies every 12 subcarrier and in total 100 subcarriers. Specific to this measurement, the UE transmits uplink pilots on the 1\textsuperscript{st}, 13\textsuperscript{th}, 25\textsuperscript{th}, ..., 1188\textsuperscript{th} subcarriers to estimate the uplink channel and the estimated channel responses are recorded every 10 ms. Those pilots are received by the Lund University massive MIMO testbed (LuMaMi) [39], with 100 active patch antennas operating at a center frequency of 3.7 GHz (wavelength \( \lambda \approx 0.081 \) m). The antennas are separated by a distance of around 4 cm (half wavelength at 3.7 GHz) in both the vertical and horizontal directions. Since our objective is to exploit more information from the azimuth compared to the elevation domain, a wide 4 \( \times \) 25 antenna configuration is selected.

We analyzed 75 pre-defined robot trajectories, where the robot was the only moving object and all other objects were static. \( T_{i} \) channel snapshots have been recorded on the \( i \)-th trajectory, and each snapshot was represented by a matrix with dimension \( M \times F \) \( (M = F = 100) \). A complex tensor \( \mathbf{A}_{i} \in \mathbb{C}^{T_{i} \times M \times F} \) was then formulated to collect all snapshots.

While the robot was moving, the position of the antenna was continuously recorded every 10 ms by the Mocap system. The measurements began with locating the robot at the edge of the predefined 4.2 \( \times \) 2.5 m\(^2\) measurement area. The robot
moved at a speed 0.1 m/s straight along the x direction; see Fig. 5. Between different measurements, the robot was moved approximately 5 cm along the y direction while maintaining its orientation. This procedure was repeated 75 times to densely scan the entire measurement area with approximately a resolution of 5 cm in the y direction and 1 mm in the x direction. When scanning the whole measurement area, we collect \( T = 302500 \) channel snapshots.

We define a tensor \( A' \in \mathbb{C}^{T \times M \times N} \) that combines all \( A_i \). \( A' \) is then normalized by multiplying itself with a scalar so that the Euclidean norm of \( A' \) is equal to \( T MN \). All \( T \) collected samples are divided into two datasets, namely the training dataset with \( \mathcal{T} \) samples and the testing dataset with \( T - \mathcal{T} \) samples. Training samples are evenly distributed with a distance along the x-axis as \( \Delta_d \). If channel samples are not selected for training purposes, they are used as testing data unless otherwise noted.

**B. Measured propagation channel characteristics**

One UE position is selected (position A, see Fig. 5) to illustrate the measured indoor channel properties. We present the power delay profile and the power of the transfer functions for all 100 antennas in Fig. 6. The power delay profile shows a typical indoor short-range channel characteristic: the first few delay bins contain the majority of the power in the delay domain. Such characteristics are also revealed in the transfer functions in Fig. 6, showing significant variations in channel responses among different antennas, and the frequency correlation is rather high. In contrast, the channel responses vary much smoother between different sub-carriers for every single antenna.

We evaluate the spatial correlation of the channel at different UE positions by computing the correlation coefficient \( \rho(\Delta_d) \) as

\[
\rho(\Delta_d) = \frac{1}{\mathcal{P}_s} \sum_{p_s} \left\{ \frac{\tilde{\mathbf{y}}_{p_s}^H \tilde{\mathbf{y}}_{p_s+\Delta_d}}{\sqrt{||\tilde{\mathbf{y}}_{p_s}||^2 ||\tilde{\mathbf{y}}_{p_s+\Delta_d}||^2}} \right\},
\]

where \( \tilde{\mathbf{y}}_{p_s} \in \mathbb{C}^{M \times N} \) is achieved by reorganizing the received channel matrix \( \mathbf{Y}_{p_s} \) as a vector. \( \mathcal{P}_s \) denotes the total number of UE positions, while \( \Delta_d \) denotes the distances between two adjacent UE positions. To visualize the spatial correlation, the absolute value of \( \rho(\Delta_d) \) with respect to the first UE trajectory is plotted according to (11) in Fig. 7. The separation distance \( \Delta_d \) ranges from 0 to 2λ. As shown, a strong spatial correlation can be expected when \( \Delta_d \lesssim \frac{1}{8} \lambda \), however, it decreases significantly for larger separations.

For all measurement data, the signal frequency point SNR ranges from 1dB-11dB, which depends on the distance between UE and BS and the constructive or destructive influence of small-scale fading.

**V. Results and Discussion**

In this section, we evaluate our localization pipeline using the measurement data set. We first investigate various channel fingerprints and then demonstrate the localization accuracy gain achieved by the subarray method. Spatial spectra of estimated covariance matrices are generated, in order to further evaluate the impact of training density on the localization accuracy leveraging the Nyquist sampling theorem. Finally, we compare our approach with a classic K-nearest neighbors (KNN) based and a CNN-based algorithm [40].
A. Investigation on channel fingerprints

We first investigate two commonly used channel fingerprints, namely, the truncated CIR and the one-sample covariance matrix, which respectively capture the delay and spatial domain CSI. It is a straightforward process to generate these fingerprints. Their localization performances are compared to the case when only using the channel transfer function. To this end, we train 3 neural networks: network 1 trains on the raw received transfer function $Y_{p_i}$ itself; network 2 the one-sample covariance matrix of the whole array with $M = 100$ antennas; network 3 trains on the truncated CIR in the first $L_w = 10$ delay bins, which considers the limited system bandwidth (20 MHz) and the typical indoor measurement scenario with a strong line of sight (LoS) component; The frameworks of the three FCNNs are programmed based on Fig.1, which are illustrated in Table 1. Since it is important to avoid the problem of vanishing gradient [41], we apply a leaky rectified linear unit (LReLU) as the nonlinear activation function at the input layer and all hidden layers as well. At the output layer, softmax is applied as the activation function to estimate the variances of the position, while LReLU is applied to estimate UE positions. We initially set the learning rates for the first FCNN as 0.07 4.95 while the second and third as -0.07 4.95. As presented in Fig. 8, training in truncated robot trajectories, compared to Fig. 8 (b) equals $\frac{1}{4}\lambda$ along all 75 robot trajectories, compared to Fig. 8 (b) where $\Delta_d = \frac{3}{4}\lambda$. As presented in Fig. 8, training in truncated CIR outperforms the raw transfer function, although they embed the same CSI. We postulate that when training on truncated CIR, the reduction in network size facilitates the training process. The signal-to-noise ratio (SNR) is also enhanced if the tail part of the CIRs is truncated since this part includes only noise instead of useful CSI. Localization accuracy when training on the one-sample covariance matrix significantly outperforms the raw transfer function and the CIR, although the delay domain information is not embedded.

![Power delay profile and power of the transfer function at position A.](image)

**FIGURE 6:** Power delay profile and power of the transfer function at position A.

![Empirical spatial correlation function w.r.t one UE moving trajectory.](image)

**FIGURE 7:** Empirical spatial correlation function w.r.t one UE moving trajectory.

<table>
<thead>
<tr>
<th>TABLE 1: Parameter settings of the neural network</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network 1</td>
</tr>
<tr>
<td>Input Features</td>
</tr>
<tr>
<td>Input layer</td>
</tr>
<tr>
<td>Hidden layer 1</td>
</tr>
<tr>
<td>Hidden layer 2</td>
</tr>
<tr>
<td>Hidden layer 3</td>
</tr>
<tr>
<td>Hidden layer 4</td>
</tr>
<tr>
<td>Hidden layer 5</td>
</tr>
<tr>
<td>Hidden layer 6</td>
</tr>
<tr>
<td>Hidden layer 7</td>
</tr>
<tr>
<td>Hidden layer 8</td>
</tr>
<tr>
<td>Output Layer</td>
</tr>
<tr>
<td>Batch Size</td>
</tr>
<tr>
<td>Epoch</td>
</tr>
<tr>
<td>Complexity</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLE 2: The NLL loss evaluated on the testing dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>NLL ($\frac{1}{L}\lambda$)</td>
</tr>
<tr>
<td>Network 1</td>
</tr>
<tr>
<td>Network 2</td>
</tr>
<tr>
<td>Network 3</td>
</tr>
<tr>
<td>Fuse networks 2 and 3, using (6)</td>
</tr>
<tr>
<td>Fuse networks 2 and 3, using (8)</td>
</tr>
</tbody>
</table>
The results of the test data set are illustrated in Table 2. As mentioned in Section III, this loss function considers the localization accuracy and the estimated variance jointly. The NLL loss of Network 3 is higher than that of the other two networks, even though it delivers better localization accuracy. We observe that the standard deviation predicted by network 3 is much smaller than the position error, which results in a significant increase of the second and third terms of (5). Based on this observation, network 3 is overconfident. This problem is even more severe if we fuse the outputs of Network 2 and Network 3 according to (6), because (6) produces a fused variance that is smaller than that of all individuals. In contrast, this problem can be alleviated by calculating the harmonic averages of the estimated variances of Network 2 and Network 3 according to (8). In the following section, we apply the subarray method to further address this overconfidence problem and focus merely on spatial channel fingerprints, considering the limited system bandwidth.

### TABLE 3: Antenna indexes in 5 groups

<table>
<thead>
<tr>
<th>Subarray group number</th>
<th>Antenna index</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>1-8, 26-33, 51-58, 76-83</td>
</tr>
<tr>
<td>II</td>
<td>5-12, 30-37, 55-62, 80-87</td>
</tr>
<tr>
<td>III</td>
<td>11-18, 36-43, 61-68, 86-93</td>
</tr>
<tr>
<td>IV</td>
<td>15-22, 40-47, 65-72, 90-97</td>
</tr>
<tr>
<td>V</td>
<td>18-25, 43-50, 68-75, 93-100</td>
</tr>
</tbody>
</table>

### TABLE 4: Network Structure for each subarray

<table>
<thead>
<tr>
<th>Layer Number</th>
<th>FCNN size</th>
<th>Layer Number</th>
<th>FCNN size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Layer 1</td>
<td>1024 x 1024</td>
<td>Hidden Layer 5</td>
<td>1024 x 512</td>
</tr>
<tr>
<td>Hidden Layer 2</td>
<td>1024 x 1024,</td>
<td>Hidden Layer 6</td>
<td>512 x 128</td>
</tr>
<tr>
<td>Hidden Layer 3</td>
<td>1024 x 1024,</td>
<td>Hidden Layer 7</td>
<td>128 x 32</td>
</tr>
<tr>
<td>Hidden Layer 4</td>
<td>1024 x 1024,</td>
<td>Output Layer 8</td>
<td>32 x 4</td>
</tr>
</tbody>
</table>

B. Enhancement by subarray method

We apply the subarray method in order to address the overconfidence problem and further enhance localization performance. Specific to this measurement setup, we consider the trade-off between complexity and localization accuracy and formulate in total 5 subarrays, and each subarray has 32 antennas ($N_1 = 4$ and $N_2 = 8$). We present the antenna indexes for each subarray in Table 3. The antenna indexes are grouped in such a way that the physical distances between each antenna are close to each other; therefore, the signals captured by those antennas are strongly correlated. Note that a few antennas belong to multiple groups, and thus the spatial correlation information among antennas from different subarray groups is included as well. These subarrays are fed into 5 subnetworks that have identical network structures, which are presented in Table 4. Compared to Network 2, the size of each subnetwork is significantly reduced, which facilitates thetraining process since it is easier to avoid overfitting. For all those 5 networks, the activation functions and total
training epochs are the same as in Network 2. The initial learning rates for all 5 networks are set at $2 \times 10^{-4}$ and reduced 20% every 10 epoch.

Fig. 9 compares the localization performances of the subarray method with the whole-array method. The localization accuracy and the NLL loss, when $\Delta_d = \frac{1}{2} \lambda$ and $\Delta_d = \frac{3}{4} \lambda$, are shown in Fig. 9 (a) and (b), respectively. Fig. 9 shows that the localization errors of all 5 groups are close to each other, which is comparable to using the one-sample covariance matrix of the entire array. The NLL loss w.r.t. the subarray is much lower than the whole array. This indicates that the subarray method better estimates the uncertainty.

Localization performance, in terms of both accuracy and NLL loss, can be further improved by applying the MRC method to fuse the outputs of all 5 subarrays. This result illustrates the importance of selecting a proper training input, since the performance gain can be clearly seen, even if the covariance matrices of the entire array contain the same necessary information as all subarrays altogether. However, we still observe that if the training density is decreased, the localization accuracy will be degraded. Therefore, we address this problem in the following sections by first investigating the influence of training density on localization performance. At the next step, more accurate estimated covariance matrices are calculated by averaging more samples at different positions, and those matrices are applied as the training fingerprints.

**C. Training density analysis using Nyquist Theorem**

We apply the Nyquist Theorem to the measurement dataset to analyze the influence of training densities. This paper focuses on the covariance matrix as an example, but our method can be generally applied to other channel fingerprints. The spatial spectra of the covariance matrices with respect to different average distances ($d = 0, \frac{\lambda}{2}, 2 \lambda$) for all 75 measurements are computed, according to (2)-(3) and (9). The spatial spectra of the $i$-th measurement w.r.t. three distances are denoted as $\Psi_{i,d=0}, \Psi_{i,d=\frac{\lambda}{2}}, \Psi_{i,d=2\lambda} \in \mathbb{C}^{M \times T}$. To visualize the spectra, we then select $i = 75$ and plot those three spatial spectra in Fig. 10. As shown, when $d$ increases, most of the

![Spatial spectra of covariance matrices](image-url)
spectral energy is concentrated in the low-frequency region, indicating that the channel changes more slowly when the UE moves to different positions. This phenomenon can be explained from a channel propagation perspective: small-scale fading is smoothed out by the averaging operation so that the swift change of channel responses cannot be observed. This allows us to further reduce the necessary training samples.

We further investigate the relationship between the $\Delta_d$ along the x-axis (see Fig. 5) and the level of aliasing noise introduced to the system. Once $\Delta_d$ increases, the captured spectrum window $L_i = \frac{T_i}{\sum \delta_d}$ in the $i$-th measurement decreases, see (10), and more aliasing noise will be introduced. To simplify the evaluation of the effect of aliasing noise, $\Delta_d$ is selected to be the same for all 75 measurement trajectories during the training phase. Based on this, we define a parameter $\eta' = \frac{\sum_i ||\Psi_{\alpha,d}||}{\sum_i ||\Psi_{\alpha,d}||}$, to characterize the extent of frequency aliasing. If $\eta'$ is closer to 1, the aliasing noise is weak. We plot $\eta'$ with regard to three covariance matrices in Fig. 11. If $d = 0$ and $\Delta_d$ exceeds the Nyquist distance $\Delta_{d_{\text{NQT}}} = \frac{1}{\lambda}$, $\eta'$ drops apparently, and the influence of aliasing noise is not trivial. Compared to $d = \frac{1}{2}\lambda$, $\eta'$ drops more smoothly after $\Delta_{d_{\text{NQT}}} = 0.5\lambda$. If $d = 2\lambda$, $\Delta_{d_{\text{NQT}}}$ increases to $\lambda$. Even if $\Delta_d$ exceeds the Nyquist distance, $\eta'$ drops very slowly, compared to $d = \frac{1}{2}\lambda$. This indicates that the influence of small-scale fading is rather weak.

**D. Comparison between different covariance matrices**

Fig. 12 illustrates the localization performances of three aforementioned covariance matrices with different training densities. To fairly compare performance, subarray methods are applied and antennas are grouped in the same way as in Table 3. All three networks are programmed according to Table 4. It can be observed from Fig. 12 that when $\Delta_d$ further exceeds $\Delta_{d_{\text{NQT}}}$, positioning accuracy decreases more because the negative effect of the aliasing noise cannot be ignored. The Nyquist distance $\Delta_{d_{\text{NQT}}}$ can be extended by increasing the average distance $d$ to cover more samples when formulating the covariance matrix. By comparingFig. 12 (a), (b), and (c), we see that when $d$ increases, both the localization accuracy and the NLL improve, especially under low training density. We postulate that three important factors contribute to this improvement: (1) by averaging more samples, the noise energy is reduced and the SNR is increased; (2) the system bandwidth is limited to 20 MHz, while the coherence bandwidth for the channel is around 10 MHz. Under this condition, $C_i = \mathbb{E}\{y_{p_i}^k y_{p_i}^k\}$ cannot be represented by the one-sample covariance matrix since many subcarriers are still strongly correlated. However, if we consider different positions far enough from each other but within the wide-sense stationary region, their corresponding channel responses are weakly correlated. The estimated covariance matrix $C_{i,N_{p_i}}$ approaches better $C_i$. (3) When $d$ is large, $C_{i,N_{p_i}}$ changes much more smoothly with different positions due to the absence of small-scale fading and $\eta'$ drops much more slowly. This guarantees that with the same training density, less aliasing noise is introduced to the system.

**E. Comparison with other approaches**

We now compare our pipeline with other two representative approaches, namely, the traditional KNN localization (naive fingerprinting) approach and the deep-learning-based approach using CNNs [40].

1) **KNN approach**

This approach first establishes a database that stores all training fingerprints. When receiving a new localization request, the BS finds the first $K$ closest fingerprints from the database. In this paper, the estimated covariance matrix is selected as the fingerprint. We denote $C_{i,T_{p_i}}$ as the $i$-th training fingerprint stored in the database and $C_{T_{p_i}}$ as the fingerprint with respect to a testing sample. We then define a scalar $l_i = ||C_{i,T_{p_i}} - C_{T_{p_i}}||^2$. After calculating all $N_{tr}$ distances $l_i$, we select the first $k = 4$ lowest $l_i$ and denote their coordinates as $p_i$. Applying the weighted KNN algorithm [42], the final estimated position $\hat{p} \in \mathbb{R}^2$ is calculated as

$$\hat{p} = \sum_{i=1}^{4} w_i \hat{p}_i,$$

where the weight $w_i$ is defined as $w_i = \frac{1/l_i}{\sum_j 1/l_j}$.

Fig. 12 displays the localization accuracy of the KNN method with respect to different $\Delta_d$. For a fair comparison, the same training data are used here as our pipeline. As a concern of the complexity issue, we randomly select 20000 testing channel samples rather than using all available ones. We observe that the KNN methods perform better than the pipeline if the channel is densely sampled. However, when the training density is reduced, the neural network method approaches and outperforms the KNN. This can be explained as follows: when the channel is heavily oversampled, it is possible to find a few pre-stored channel fingerprints in
the database, which are very similar to the test channel fingerprint. The localization accuracy is already good by directly reading the coordinates of the closest fingerprints in the database, let alone the further improvement by the interpolation operation shown in (12). In comparison, the neural network mechanism estimates the UE position based on interpolating on the whole training datasets rather than the few closest fingerprints, which results in suboptimal localization accuracy. However, when the training density is reduced, it cannot be guaranteed that the the channel fingerprints in the database are close to the test channel fingerprint. Therefore, it only performs well if there exists such a fingerprint. In contrast, the neural network is likely more suitable for the localization task thanks to its nonlinear interpolation ability. Bear in mind that KNN methods generally require computing the Euclidean distance between the testing fingerprint and \(N_r\) prestored training fingerprints in total. This leads to a high complexity in time, which is \(o(M^2N_r)\) according to (12). In comparison, our pipeline has a better time complexity, that is \(o(M^3)\), since the antenna number \(M\) is much smaller than \(N_r\) for most commercial devices. Even when \(M\) becomes larger, one can use the subarray method to reduce the number of antennas in each group and to reduce running time in practice. On the other hand, if one wants to achieve a better localization result using the KNN method, it is necessary to pre-store sheer numbers of measurement samples in the database. However, it is a resource-intensive endeavor to construct such very densely sampled indoor measurement datasets both in terms of finance and time manners, if the distance between adjacent samples is smaller than the Nyquist distance (only a few centimeters at sub-6 GHz frequency). For most of the applications, one would spend less resources to collect data, and thus lower densities are expected. Therefore, the usage of the KNN method for a real-time operation scenario is rather limited. From this perspective, the processing pipeline still has its advantages even under the condition of a high training density.

2) CNN based approach

We then compare our localization approach with [40] that trained a deep residual CNN to perform the indoor localization tasks, where the open-source code is available. The author in [40] first formulated a tensor \(\tilde{\mathbf{p}}\) by collecting the real and imaginary part of the raw received complex transfer function \(\mathbf{Y}_{\mathbf{p}}\). Then \(\mathbf{Y}_{\mathbf{p}}\) was converted to the polar domain by calculating the amplitudes and phases of each entry of the received transfer function matrix. This formulated a tensor \(\tilde{\mathbf{y}}\) by concatenating \(\mathbf{y}\) and \(\tilde{\mathbf{y}}\) and sent this tensor to a residual CNN. The network structure is programmed according to [40], as well as the open source code. We modify the size of the input layer, since the antenna number in our case is 100 instead of 64. We plot the localization accuracy of this approach in Fig.13, where \(\Delta_d = \frac{1}{2}\lambda\) and the training percentage is around 5%. As illustrated, our pipeline has better localization accuracy compared to [40] even with the use of a one-sample covariance matrix for training. Localization accuracy can be slightly improved if we increase the average distance \(d\) to \(\frac{1}{2}\lambda\) under this training density. We postulate that our pipeline benefits from the preprocessing step as well as the subarray method. We have also performed a time complexity analysis of the convolutional neural network, which is \(o(MF_{l1}l_2C_{in}C_{out})\), where \(l_1\) and \(l_2\) represent the 2-D size of the convolutional kernel and \(C_{in}\) and \(C_{out}\) the numbers of input and output channels at each layer. In comparison, the time complexity of our pipeline is \(o(M^3)\). Consider that \(M^2 \leq F_{l1}l_2C_{in}C_{out}\) for most commercial systems, we also have advantages in terms of time complexity. Furthermore, our system has the ability to predict uncertainty, which is an additional advantage.
We investigate the localization performance of our proposed pipeline in none-line-of-sight (NLoS) measurement scenarios. Specifically, when we estimate the localization accuracy is still acceptable for applications such as indoor navigation. Even when the SNR drops to $-15$ dB, the localization accuracy is still acceptable for applications such as indoor navigation. We investigate the localization performance of our proposed pipeline in none-line-of-sight (NLoS) measurement scenarios, which occupy different subcarriers, move within the four gray squares, each with a size $1.2 \times 1.2$ m. Each UE is equipped with a dipole antenna that is placed at the height of $0.4$ m. The UE trajectories are densely sampled, resulting in up to $252004$ channel samples with geographical distance between each sample $5$ mm. The ground truth positions of UE are recorded by a mechanical device with an error of less than $1$ mm. The base station consists of $64$ patch antennas operating at the center frequency $2.6$ GHz, and all antennas formulate a uniform rectangular array with size $0.56 \times 0.56$ m. The index of the antenna on the $i$-th row and the $j$-th column is $8(i-1)+j$. A metal blocker with size $1.6 \times 1.3$ m is placed between the base station and the UE, blocking the LoS component. Each UE sound the OFDM signal from the uplink as a pilot for channel estimation, which has 100 subcarriers, occupying in total $20$ MHz.

The subarray method is also applied and $4$ groups are formulated, each group contains $36$ antennas, and the antenna indices are shown in Table 5. The subarray method is also applied and $4$ groups are formulated, each group contains $36$ antennas, and the antenna indices are shown in Table 5. The covariance matrices of each subarray are formulated and sent to four individual FCNNs for training. Each FCNN has the same structure as Table 4, except the input layer size is $1296 \times 1024$. Figure 15 illustrates the localization performances of our pipeline when using the estimated covariance matrix, compared with the approach illustrated in [40]. Specifically, when we estimate the covariance matrix, three average distances are investigated.
namely, \( d = 0 \) (one-sample covariance matrix), \( d = 0.5\lambda \), and \( d = \lambda \). The distances of the training samples are \( \frac{1}{4}\lambda \) in both horizontal and vertical directions. As illustrated, the algorithm in [40] achieves better performance than training on the one-sample covariance matrix in the NLoS scenario. However, if the average distance \( d \) increases over \( \frac{1}{2}\lambda \), the localization error decreases, and they outperform [40]. We postulate that the one-sample covariance matrix is more unstable under the NLoS scenario because the absence of a prominent LoS component amplifies the effect of small-scale fading. Therefore, in this scenario, it is challenging to collect only one channel sample to estimate the covariance matrix \( C_i \) at the position \( p_i \). Compared to other studies, when the average distance \( d \) increases, the effect of small-scale fading becomes weak and \( C_{i,N_{d_i}} \), can better approach \( C_i \). This shows that if the covariance matrices are selected as a training fingerprint in the NLoS scenario, it is more important to cover more samples in the neighborhood region of \( p_i \) than in the LoS scenario. On the basis of the observation above, our pipeline is also suitable for the NLoS scenario and can still achieve better performances than the literature.

3) Random selection of the training samples

We now investigate the localization performance of our pipeline, when the training data set is constructed by randomly selecting training samples from the robot trajectory. To enable a fair comparison, the network structures and all other parameters, such as the training percentages of the two datasets (evenly and randomly sampled), are the same. Specific to our data set, the percentage of training is 5% when the distance between two adjacent samples is \( \Delta_d = \frac{1}{2}\lambda \). If \( \Delta_d = \frac{1}{4}\lambda \), the training percentage drops to 2.5%. Fig. 17 illustrates the localization accuracy of our pipeline when we train on the one-sample covariance matrix. As shown, the localization accuracy deteriorates when the training samples are randomly selected. Therefore, the channel property may not be well captured, which deteriorates the localization performance. This example illustrates the importance of appropriately selecting training samples when we construct the training datasets.

VI. Conclusion and future work

This paper investigated the potential to apply ML to a massive MIMO system for solving localization tasks. We analyzed a novel ML-based localization pipeline, which estimated UE positions and variances by using different channel fingerprints, including covariance matrices and truncated CIR. For a system with a massive number of antennas, a sub-array method was applied to facilitate the training process. Furthermore, we applied the Nyquist sampling theorem to investigate the effect of training density. An indoor massive MIMO measurement campaign was conducted at 3.7 GHz using 20 MHz bandwidth to evaluate our approaches, where centimeter-level localization accuracy has been achieved. Measurement results show that: 1) The information from both the delay and angle domains contributes to the localization performance, although in our case the delay domain CSI contributes less than the angle domain CSI due to the limited available bandwidth. 2) Compared to training on the whole antenna array, the subarray method can achieve significant enhancements in both positioning accuracy and better uncertainty prediction quality. 3) As expected, the localization accuracy decreases when the sampling interval is larger than the Nyquist sampling distances. It is worth mentioning that during the measurement campaign, the channel remained stationary and no individuals were present. In upcoming research, we will examine how the presence of people and other moving objects, as well as the difference in the properties of the UE antenna at the training and testing phase, affect the accuracy of the localization and apply transfer learning to address possible problems. We will also investigate localization pipelines that jointly process information from multiple snapshots.
ACKNOWLEDGMENT

The authors thank Lund University Humanities Lab and the LTH robotic lab for help with the equipment. Special thanks to Henrik Garde, Alexander Dürr, Steffen Malkowsky, Sara Willhammar, and Sirvan Abdullah Poor for their assistance throughout the measurements. The authors also appreciate the discussion with Ph.D candidate Ziliang Xiong regarding the uncertainty prediction.

REFERENCES


Guoada Tian (Student Member, IEEE) finished his bachelor’s degree in automation and control engineering at Northeastern University in 2016 and his master’s degree in wireless communication at Lund University in 2018. During his master’s degree, he received the Lund University Global Scholarship. Currently, he is pursuing Ph.D study at Lund University under the supervision of Fredrik Tufvesson, Ove Edfors, Bo Bernhardsson, and Xuesong Cai. His current research topic is applying machine learning to wireless localization, sensing, and communication systems.

Iluyda Yaman (Student Member, IEEE) completed her bachelor’s degree at Istanbul Technical University in 2018 and moved to Sweden to pursue her master’s degree in Embedded Electronics Engineering at Lund University. During her master’s degree, she received the Lund University Global Scholarship for her academic success, and completed her studies in 2020. She is currently a third-year Ph.D. student at Lund University under the supervision of Liang Liu, Ove Edfors, Kalle Åström, and Steffen Malkowsky. Her current research area is fusing vision and radio sensors to design low-power hardware using machine learning algorithms.

Michiel Sandra (Student Member, IEEE) received the M.Sc. degree (summa cum laude) in engineering technology from KU Leuven in 2020. He is currently pursuing the Ph.D. degree at Lund University in channel modeling for maritime communications under the supervision of Xuesong Cai and Anders J Johansson. His current research interests include channel sounding, channel modeling, signal processing, antenna design, and wireless system design.

Xuesong Cai (Senior Member, IEEE) received the B.S. degree and the Ph.D. degree (with distinction) from Tongji University, Shanghai, China, in 2013 and 2018, respectively. In 2015, he conducted a three-month internship with Huawei Technologies, Shanghai, China. He was also a Visiting Scholar with Universidad Politécnica de Madrid, Madrid, Spain in 2016. From 2018-2022, he conducted several postdoctoral stays at Aalborg University and Nokia Bell Labs, Denmark, and Lund University, Sweden. He is currently an Assistant Professor in Communications Engineering and a Marie Skłodowska-Curie Fellow at Lund University, closely cooperating with Ericsson and Sony. His research interests include radio channel characterisation, high-resolution parameter estimation, over-the-air testing, resource optimization, and radio-based localization for 5G/B5G wireless systems.

Liang Liu (Member, IEEE) received Ph.D. degree from Fudan University in 2010. He joined the Department of Electrical and Information Technology (EIT), Lund University, Sweden, where he held a post-doc position. Since 2016, he has been Associate Professor with Lund University. His research interests include wireless systems and digital integrated circuits design. He is a member of the Technical Committee of VLSI Systems and Applications and CAS for Communications of the IEEE Circuit and Systems Society.

Fredrik Tufvesson (Fellow, IEEE) received the Ph.D. degree from Lund University, Lund, Sweden, in 2000. After two years at a startup company, he joined the Department of Electrical and Information Technology, Lund University, where he is currently a Professor of radio systems. He has authored around 100 journal articles and 150 conference papers. His main research interest is the interplay between the radio channel and the rest of the communication system with various applications in 5G/B5G systems, such as massive multiple-input multiple-output (MIMO), mmWave communication, vehicular communication, and radio-based positioning. Dr. Tufvesson’s research has been awarded the Neal Shepherd Memorial Award for the Best Propagation Paper in the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY and the IEEE Communications Society Best Tutorial Paper Award.