Deep-Learning-Based High-Precision Localization With Massive MIMO

GUODA TIAN (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)

Department of Electrical and Information Technology, Lund University, 221 00 Lund, Sweden

CORRESPONDING AUTHORS: G. TIAN (guoda.tian@eit.lth.se) AND X. CAI (xuesong.cai@eit.lth.se)

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], [2], [3], [4], [5], [6], [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], [11], [12], [13], [14], [15], [16]. For example, the work in [10], [15], and [16] proposed novel estimators to jointly estimate angles and positions with large-scale arrays. The authors in [11], [12], and [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], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [32]. It is essential to appropriately select both fingerprints and algorithms. One can choose either the raw transfer function [17], [18], [19] or various channel fingerprints such as RSS, power delay profile (PDP), angular spectrum, correlation function, etc., [20], [21], [22], [23], [24], [25], [26], [28], and [27] 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], [22], [23], [32], and the deep learning family [17], [18], [20], [24], [25], [26], [27], [28], [29], [30], [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]

- We apply a localization framework that blends channel fingerprints that contain information from the delay and angular domains, respectively. It is not necessary to calibrate the whole BS array to obtain those channel fingerprints.
- 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 $\mathbf{p}_i(x_0, y_0) \in \mathbb{R}^2$, 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 subcarriers. 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 $\mathcal{T}$ snapshots are recorded and $\mathcal{T}$ 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.

---

[1] A 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.

A. SPATIAL COVARIANCE MATRIX

It is sometimes challenging to extract calibrated fingerprints such as AoA due to the presence of the RF chain matrix Γ, see (1). Therefore, we consider using the covariance matrix $\mathbf{C}_i = \mathbb{E}\{\mathbf{y}_{p_i}\mathbf{y}_{p_i}^H\} \in \mathbb{C}^{M \times M}$ 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 $\mathbf{p}_i$, there exist in total $\mathcal{N}_{p_i}$ positions in the neighborhood region of $\mathbf{p}_i$, whose channel responses are accessible. Those $\mathcal{N}_{p_i}$ samples are inside a circular area, with $\mathbf{p}_i$ as the center and d as the diameter, i.e., $||\mathbf{p}_j - \mathbf{p}_i||^2 \le \frac{d}{2}, j = 1, 2, \ldots, \mathcal{N}_{p_i}$. We then define the sample covariance matrix $\tilde{\mathbf{C}}_{i,\mathcal{N}_{p_i}} \in \mathbb{C}^{M \times M}$ to estimate $\mathbf{C}_i$. Specifically,

$$\tilde{\mathbf{C}}_{i,\mathcal{N}_{p_i}} = \frac{1}{\mathcal{N}_{p_i}} \sum_{j=1}^{\mathcal{N}_{p_i}} \mathbf{Y}_{p_j}\mathbf{Y}_{p_j}^H. \tag{2}$$

As shown in (2), $\tilde{\mathbf{C}}_{i,\mathcal{N}_{p_i}}$ depends on $\mathcal{N}_{p_i}$ and thus d. A special case is that if $d = 0$ and $\mathcal{N}_{p_i} = 1$, $\mathbf{C}_i$ is estimated by only correlating across all subcarriers of $\mathbf{Y}_{p_i}$ at a fixed position $\mathbf{p}_i$. 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 larger than half a wavelength, a major difference in the propagation channel can be observed and $\tilde{\mathbf{C}}_{i,\mathcal{N}_{p_i}}$ can therefore better approach $\mathbf{C}_i$. If d is large enough, the fingerprint $\tilde{\mathbf{C}}_{i,\mathcal{N}_{p_i}}$ 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 $\tilde{\mathbf{C}}_{i,\mathcal{N}_{p_i}}$ is a Hermitian matrix, i.e., $\tilde{\mathbf{C}}_{i,\mathcal{N}_{p_i}} = \tilde{\mathbf{C}}_{i,\mathcal{N}_{p_i}}^H$, the upper minor diagonal elements contain the same information as the lower. To decrease the computation complexity, we introduce another matrix $\check{\mathbf{C}}_{i,\mathcal{N}_{p_i}} \in \mathbb{R}^{M \times M}$ and a vector $\tilde{\mathbf{c}}_{i,\mathcal{N}_{p_i}} \in \mathbb{R}^{M^2}$ as

$$\check{\mathbf{C}}_{i,\mathcal{N}_{p_i}} = \text{ltril}\{\Re[\tilde{\mathbf{C}}_{i,\mathcal{N}_{p_i}}]\} + \text{sltril}\{\Im[\tilde{\mathbf{C}}_{i,\mathcal{N}_{p_i}}]\}$$
$$\tilde{\mathbf{c}}_{i,\mathcal{N}_{p_i}} = \text{vec}\{\check{\mathbf{C}}_{i,\mathcal{N}_{p_i}}\}, \tag{3}$$

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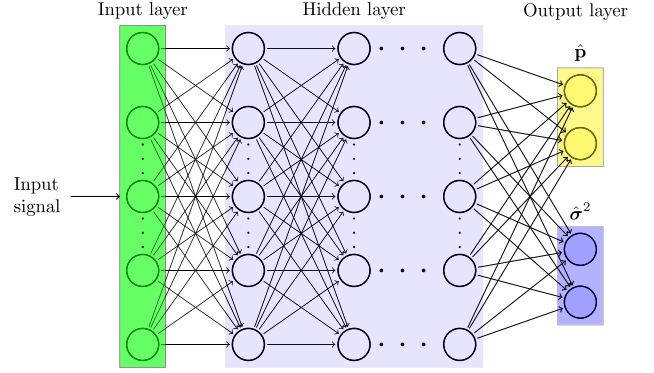

FIGURE 1. A typical structure of an FCNN.

B. TRUNCATED CHANNEL IMPULSE RESPONSE

The fingerprint $\mathbf{C}_i$ 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 $\Xi \in \mathbb{C}^{M \times L}$ 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 Ξ. Specifically, $\xi = [\text{vec}\{\Re(\Xi)\}^T, \text{vec}\{\Im(\Xi)\}^T]^T$.

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 $f : \mathbb{R}^{V_1} \to \mathbb{R}^{V_2}$, 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 UE ($\hat{\mathbf{p}} = [\hat{p}_x, \hat{p}_y]^T \in \mathbb{R}^2$) and the variance ($\hat{\sigma}^2 = [\hat{\sigma}_x^2, \hat{\sigma}_y^2]^T \in \mathbb{R}^2$).

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 γ_i nodes in the i-th layer, the values of which are collected by a signal vector $\mathbf{x}^i = [x_1^i, \ldots, x_{\gamma_i}^i] \in \mathbb{R}^{\gamma_i}$. The value of the k-th node is calculated by applying a weight vector $\mathbf{w} = [w_1^{i-1}, \ldots, w_{\gamma_{i-1}}^{i-1}] \in \mathbb{R}^{\gamma_{i-1}}$ to the signal

vector $\mathbf{x}^{i-1}$ in the previous layer. Specifically, x_k^i is computed as

$$x_k^i = g_i \left(\sum_{j=1}^{\gamma_{i-1}} x_j^{i-1} w_j^{i-1} \right) + b_i, \tag{4}$$

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 $\boldsymbol{\nu} = [\hat{\mathbf{p}}, \hat{\boldsymbol{\sigma}}^2]$.

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 $\hat{\mathbf{p}}_i = [\hat{p}_{x_i}, \hat{p}_{y_i}]^T \in \mathbb{R}^2$ and the variance vector $\hat{\boldsymbol{\sigma}}_i^2 = [\hat{\sigma}_{x_i}^2, \hat{\sigma}_{y_i}^2]^T \in \mathbb{R}^2$, while the ground truth is $\mathbf{p}_i = [p_{x_i}, p_{y_i}]^T \in \mathbb{R}^2$. Taking into account all N_{tr} training samples, the loss function ψ is

$$\psi = \frac{1}{2N_{tr}} \sum_i \left(\frac{\log \hat{\sigma}_{x_i}^2 \hat{\sigma}_{y_i}^2}{2} + \frac{(p_{x_i} - \hat{p}_{x_i})^2}{2\hat{\sigma}_{x_i}^2} + \frac{(p_{y_i} - \hat{p}_{y_i})^2}{2\hat{\sigma}_{y_i}^2} \right). \tag{5}$$

Observe that ψ 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 ψ. 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 χ fingerprints and feed each fingerprint to an individual processing chain. Those fingerprints can be either the entire covariance matrix or sub-matrices (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
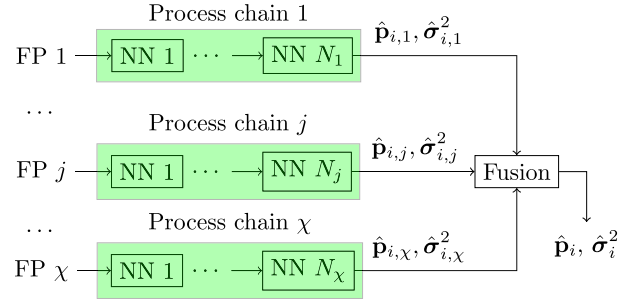

FIGURE 2. The positioning neural network structure.

j-th processing chain estimates the position of the UE and the variance as $\hat{\mathbf{p}}_{i,j} = [\hat{p}_{x_{i,j}}, \hat{p}_{y_{i,j}}] \in \mathbb{R}^2$ and $\hat{\boldsymbol{\sigma}}_{i,j}^2 = [\hat{\sigma}_{x_{i,j}}^2, \hat{\sigma}_{y_{i,j}}^2] \in \mathbb{R}^2$. By fusing all χ processing chains according to the maximum ratio combining (MRC) approach [37], $\hat{\mathbf{p}}_i$ and $\hat{\boldsymbol{\sigma}}_i^2$ are calculated as:

$$\hat{\sigma}_{x_i}^2 = \frac{1}{\sum_j 1/\hat{\sigma}_{x_{i,j}}^2}, \quad \hat{\sigma}_{y_i}^2 = \frac{1}{\sum_j 1/\hat{\sigma}_{y_{i,j}}^2} \tag{6}$$

$$\hat{p}_{x_i} = \hat{\sigma}_{x_i}^2 \left(\sum_j \frac{1}{\hat{\sigma}_{x_{i,j}}^2} \hat{p}_{x_{i,j}} \right), \quad \hat{p}_{y_i} = \hat{\sigma}_{y_i}^2 \left(\sum_j \frac{1}{\hat{\sigma}_{y_{i,j}}^2} \hat{p}_{y_{i,j}} \right). \tag{7}$$

However, the estimated variance by (6) may be overconfident, especially when the network is overfitted. According to (6), $\hat{\sigma}_{x_i}^2$ and $\hat{\sigma}_{y_i}^2$ 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 $\hat{\sigma}_{x_i}^2$ and $\hat{\sigma}_{y_i}^2$ act as the denominators of the second and third terms, respectively. To address this issue, we multiply a factor χ with $\hat{\boldsymbol{\sigma}}_i^2$ to get the modified vector $\hat{\boldsymbol{\sigma}}_{i,mod}^2 \in \mathbb{R}^2$, which is the harmonic averages of all estimated variances. Specifically,

$$\hat{\boldsymbol{\sigma}}_{i,mod}^2 = \chi \, \hat{\boldsymbol{\sigma}}_i^2. \tag{8}$$

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 $N_1 N_2$ 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,
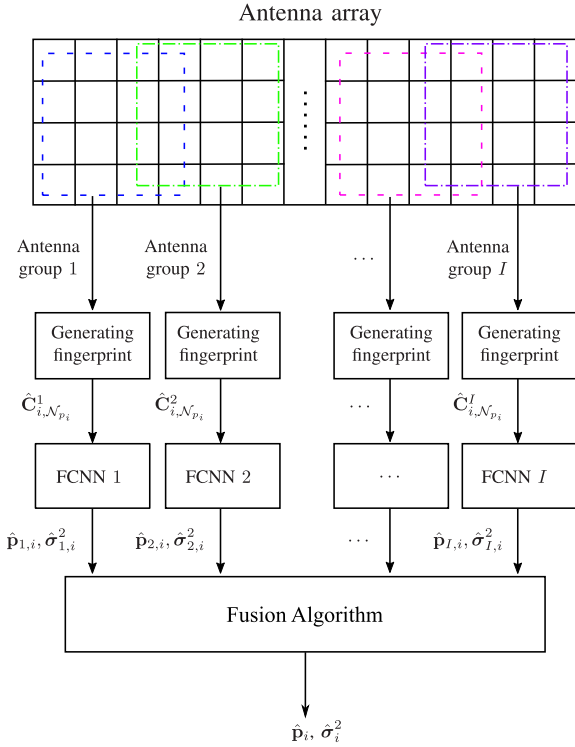
Antenna array

FIGURE 3. The subarray method.

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 $I = (\lfloor \frac{M1 - N1}{S1} \rfloor + 1)(\lfloor \frac{M2 - N2}{S2} \rfloor + 1)$ training groups are formulated, where $\lfloor . \rfloor$ denotes the floor function. We then formulate I sample covariance matrices that correspond to the UE position $\mathbf{p}_i$, which are denoted as $\hat{\mathbf{C}}_{i,\mathcal{N}_{p_i}}^1, \ldots, \hat{\mathbf{C}}_{i,\mathcal{N}_{p_i}}^I \in \mathbb{C}^{N_1 N_2 \times N_1 N_2}$. 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 $\tilde{\mathbf{c}}_{i,\mathcal{N}_{p_i}}$ 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 δ_d. We define a matrix

$\check{\mathbf{C}} = [\tilde{\mathbf{c}}_{1,\mathcal{N}_{p_i}}, \ldots, \tilde{\mathbf{c}}_{Q,\mathcal{N}_{p_i}}] \in \mathbb{R}^{M^2 \times Q}$ to collect all those Q channel response vectors. By performing the 1-D discrete Fourier transform of $\check{\mathbf{C}}$ along the horizontal axis, we can formulate a matrix $\Psi \in \mathbb{C}^{M^2 \times Q}$ that characterizes the channel variations along those Q positions. Specifically,

$$\Psi = \check{\mathbf{C}}\Lambda, \qquad (9)$$

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

$$\Delta_d = Q\delta_d/L. \qquad (10)$$

We then form those L columns into a new matrix $\Psi^L \in \mathbb{C}^{M^2 \times L}$ and define η as the ratio between the Frobenius norm of Ψ^L and Ψ, that is $\eta = \frac{||\Psi^L||_F^2}{||\Psi||_F^2}$. Here, η shows the extent of aliasing of different sampling intervals. In the following sections, we will analyze the influence of η on localization accuracy and discuss the choice of Δ_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 1st, 13th, 25th, ..., 1188th 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 4×25 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. $\mathcal{T}_i$ 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 $\mathcal{A}_i \in \mathbb{C}^{\mathcal{T}_i \times M \times F}$ was then formulated to collect all snapshots.
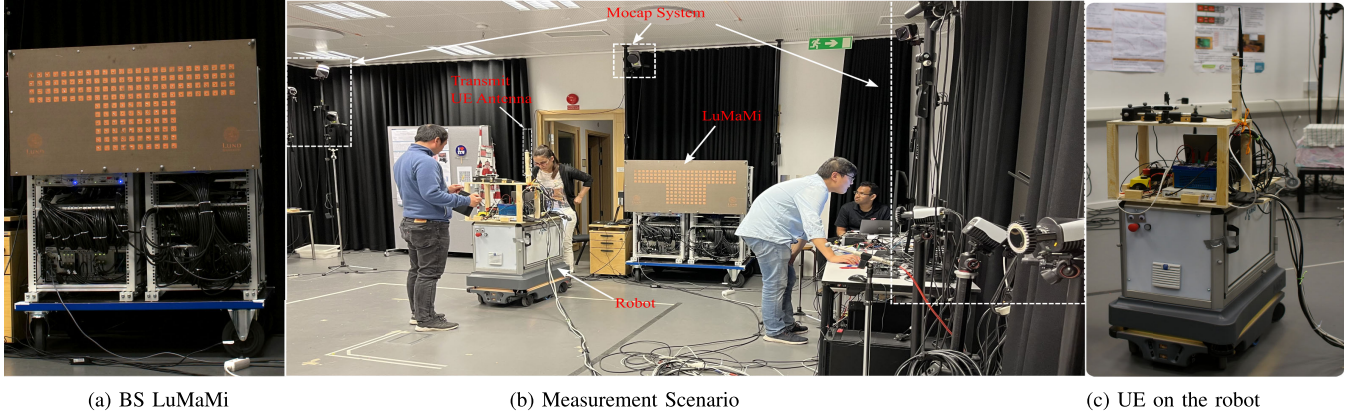
(a) BS LuMaMi   (b) Measurement Scenario   (c) UE on the robot

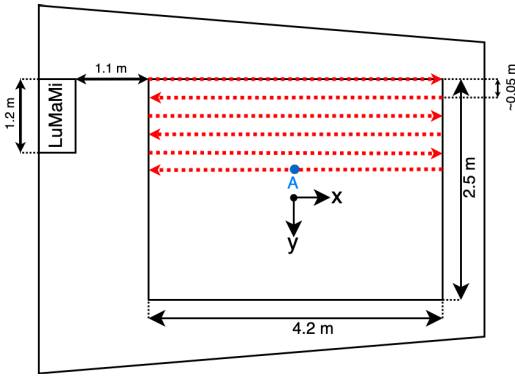**FIGURE 4. The indoor measurement campaign.**

FIGURE 5. Measurement arrangement in the mocap studio, the red dotted arrow shows the trajectory of the antenna.

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 $4.2 \times 2.5 \, \text{m}^2$ 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 $\mathcal{T} = \sum_i \mathcal{T}_i = 302500$ channel snapshots. We define a tensor $\mathcal{A}' \in \mathbb{C}^{\mathcal{T} \times M \times N}$ that combines all $\mathcal{A}_i$. $\mathcal{A}'$ is then normalized by multiplying itself with a scalar so that the Euclidean norm of $\mathcal{A}'$ is equal to $\mathcal{T} M N$. All $\mathcal{T}$ collected samples are divided into two datasets, namely the training dataset with $\mathcal{X}$ samples and the testing dataset with $\mathcal{T} - \mathcal{X}$ samples. Training samples are evenly distributed with a distance along the x-axis as Δ_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

$$\rho(\Delta_d) = \frac{1}{\mathcal{P}'} \sum_{\mathbf{p}_x} \left\{ \frac{\tilde{\mathbf{y}}_{\mathbf{p}_x}^H \tilde{\mathbf{y}}_{\mathbf{p}_{x+\Delta_d}}}{\sqrt{||\tilde{\mathbf{y}}_{\mathbf{p}_x}||^2 ||\tilde{\mathbf{y}}_{\mathbf{p}_{x+\Delta_d}}||^2}} \right\}, \quad (11)$$

where $\tilde{\mathbf{y}}_{\mathbf{p}_x} \in \mathbb{C}^{MN}$ is achieved by reorganizing the received channel matrix $\mathbf{Y}_{\mathbf{p}_i}$ as a vector. $\mathcal{P}'$ denotes the total number of UE positions, while Δ_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 Δ_d ranges from 0 to 2λ. As shown, a strong spatial correlation can be expected when $\Delta_d \leq \frac{1}{8}\lambda$, 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,
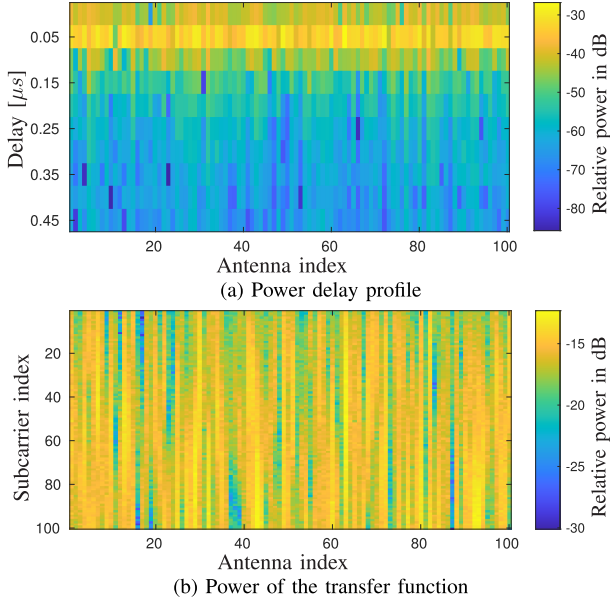
(a) Power delay profile

(b) Power of the transfer function

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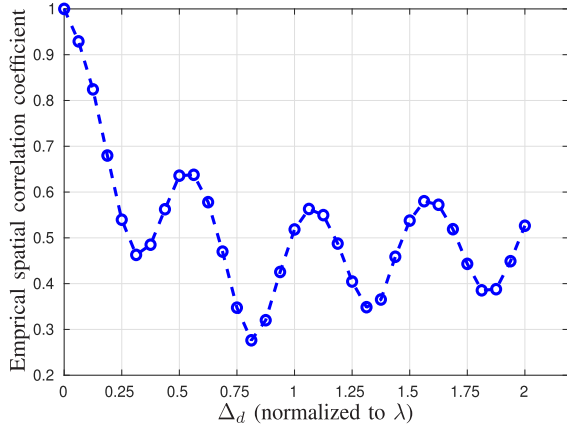

FIGURE 7. Empirical spatial correlation function w.r.t one UE moving trajectory.

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 $\mathbf{Y}_{\mathbf{p}_i}$ 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

TABLE 1. Parameter settings of the neural network.

	Network 1	Network 2	Network 3
Input Features	Transfer function	Cov-matrix	Trunc. CIR
Input layer	$2MN \times MN$	$M^2 \times 20M$	$2ML \times ML$
Hidden layer 1	$MN \times MN/2$	$20M \times 20M$	$ML \times ML$
Hidden layer 2	$MN/2 \times MN/4$	$20M \times 20M$	$ML \times ML$
Hidden layer 3	$MN/4 \times MN/4$	$20M \times 20M$	$ML \times 512$
Hidden layer 4	$MN/4 \times 1024$	$20M \times 1024$	512×512
Hidden layer 5	1024×512	1024×512	512×256
Hidden layer 6	512×128	512×128	256×128
Hidden layer 7	128×32	128×32	128×32
Hidden layer 8	32×4	32×4	32×4
Output Layer	4×4	4×4	4×4
Batch Size	64	64	64
Epoch	200	200	200
Complexity	$O(M^2N^2)$	$O(M^3)$	$O(M^2L^2)$

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 10^{-5} while the second and third as 10^{-4} and all the learning rates are reduced 20% every 10 epochs. Compared to our previous work [33], we reduced the time complexity of Network 2 from $O(M^4)$ to $O(M^3)$.

Fig. 8 shows the localization accuracy of applying three networks individually, as well as the accuracy when fusing networks 2 and 3 according to (7). In Fig. 8 (a), Δ_d equals to $\frac{1}{8}\lambda$ 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 in this fingerprint. There are two potential explanations: *i)* It is challenging to resolve multipath components due to limited bandwidth and the propagation channel has a strong LoS property. *ii)* Owing to the pre-processing, the angular information can be better exploited by the neural network. The fusion algorithm results in a slight improvement in localization accuracy in comparison to using the pure one-sampled covariance matrix, since the system bandwidth is limited to 20 MHz and it is challenging to provide a good delay resolution. However, the CSI in the delay domain is still beneficial for localization tasks even with limited bandwidth.
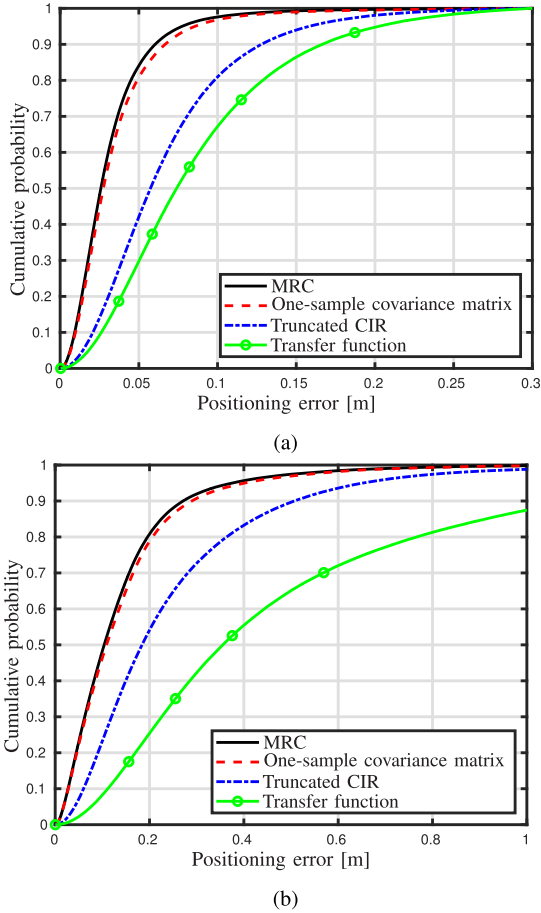
FIGURE 8. Positioning error cumulative density function with respect to different training densities: (a) $\Delta_d = \frac{1}{8}\lambda$, (b) $\Delta_d = \frac{3}{4}\lambda$.

TABLE 2. The NLL loss evaluated on the testing dataset.

	NLL ($\frac{1}{8}\lambda$)	NLL ($\frac{3}{4}\lambda$)
Network 1	-1.95	-0.13
Network 2	-2.13	-0.43
Network 3	3.09	12.26
Fuse networks 2 and 3, using (6)	3.23	12.66
Fuse networks 2 and 3, using (8)	-0.07	4.95

TABLE 3. Antenna indexes in 5 groups.

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

TABLE 4. Network structure for each subarray.

Layer Number	FCNN size	Layer Number	FCNN size
Input Layer 1	1024×1024	Hidden Layer 5	1024×512
Hidden Layer 2	$1024 \times 1024,$	Hidden Layer 6	512×128
Hidden Layer 3	$1024 \times 1024,$	Hidden Layer 7	128×32
Hidden Layer 4	$1024 \times 1024,$	Output Layer 8	32×4

Thus, we believe that the delay domain information can contribute more, under scenarios with rich multipath or for a system occupying wider bandwidth. Compared to Fig. 8 (a), the localization accuracy shown in Fig. 8 (b) significantly decreases. We postulate that when $\Delta_d = \frac{3}{4}\lambda$, the training density is not sufficient to represent the instantaneous channel properties.

We then calculate the NLL loss of all the aforementioned localization algorithms in the training and test dataset, and 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.

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 the training 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 2×10^{-4} and reduced 20 % every 10 epoch.
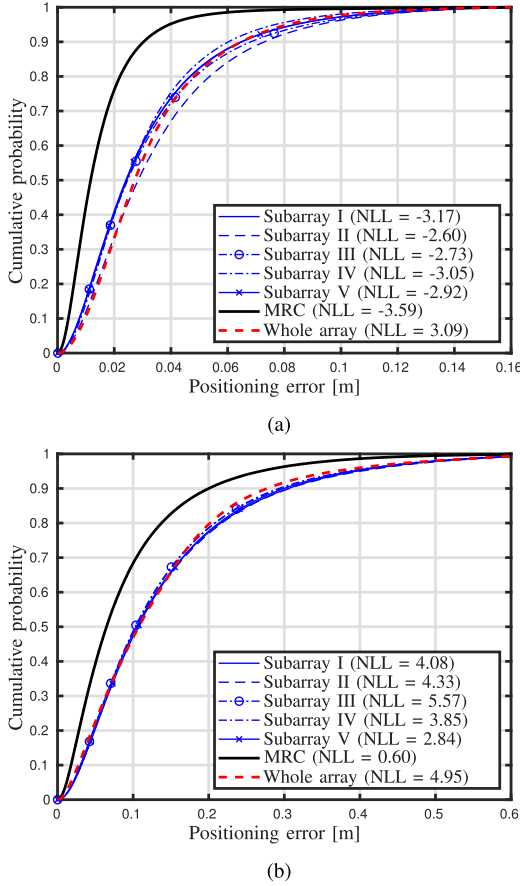
(a)

(b)

FIGURE 9. Positioning errors of using subarrays and the whole-array w.r.t. different training densities: (a) $\Delta_d = \frac{1}{8}\lambda$, (b) $\Delta_d = \frac{3}{4}\lambda$.

Fig. 9 compares the localization performances of the subarray method with the whole array method. The localization accuracy and the NLL loss, when Δ_d is $\frac{1}{8}\lambda$ and $\frac{3}{4}\lambda$, 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.
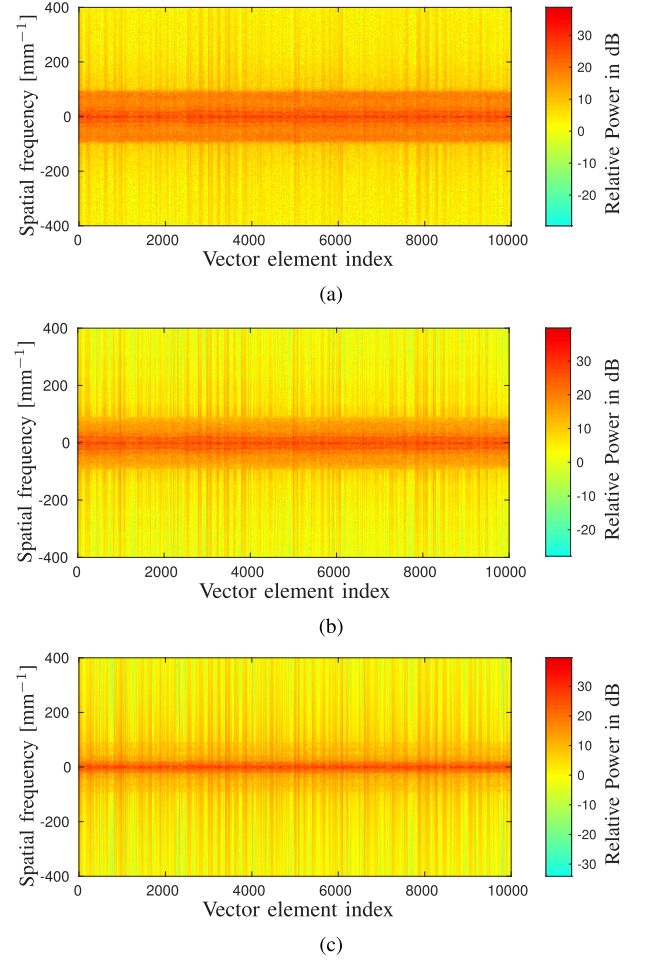

(a)

(b)

(c)

FIGURE 10. Spatial spectra of covariance matrices w.r.t. different average distances: (a) $d = 0$ (one-sample covariance matrix), (b) $d = \lambda/2$ (c) $d = 2\lambda$.

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 $\Psi_{i,d=0}$, $\Psi_{i,d=\frac{\lambda}{2}}$, $\Psi_{i,d=2\lambda} \in \mathbb{C}^{M^2 \times \mathcal{T}_i}$. 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 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.
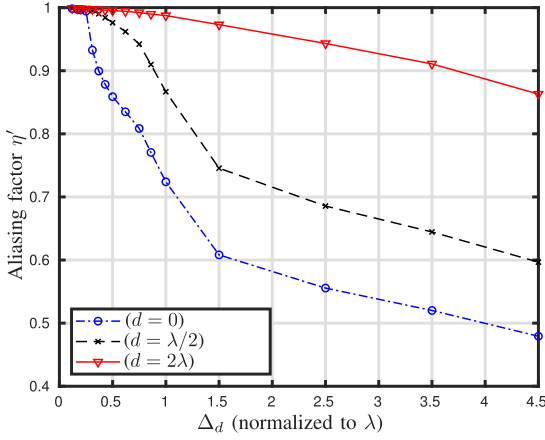
FIGURE 11. η' with respect to sampling distance Δ_d.

We further investigate the relationship between the Δ_d along the x-axis (see Fig. 5) and the level of aliasing noise introduced to the system. Once Δ_d increases, the captured spectrum window $L_i = \frac{T_i}{\Delta_d}\delta_d$ 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, Δ_d is selected to be the same for all 75 measurement trajectories during the training phase. Based on this, we define a parameter $\eta' = \frac{\sum_i ||\Psi_{i,d}^{L_i}||}{\sum_i ||\Psi_{i,d}||}$, to characterize the extent of frequency aliasing. If η' is closer to 1, the aliasing noise is weak. We plot η' with regard to three covariance matrices in Fig. 11. If $d = 0$ and Δ_d exceeds the Nyquist distance $\Delta_{nqt} = \frac{1}{4}\lambda$, η' drops apparently, and the influence of aliasing noise is not trivial. Compared to $d = \frac{1}{2}\lambda$, η' drops more smoothly after $\Delta_{nqt} = 0.5\lambda$. If $d = 2\lambda$, Δ_{nqt} increases to λ. Even if Δ_d exceeds the Nyquist distance, η' 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 Δ_d further exceeds Δ_{nqt}, positioning accuracy decreases more because the negative effect of the aliasing noise cannot be ignored. The Nyquist distance Δ_{nqt} can be extended by increasing the average distance d to cover more samples when formulating the covariance matrix. By comparing Fig. 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, $\mathbf{C}_i = \mathbb{E}\{\mathbf{y}_{p_i}\mathbf{y}_{p_i}^H\}$ 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 $\tilde{\mathbf{C}}_{i,\mathcal{N}_{p_i}}$ approaches better $\mathbf{C}_i$. (3) When d is large, $\tilde{\mathbf{C}}_{i,\mathcal{N}_{p_i}}$ changes much more smoothly with different positions due to the absence of small-scale fading and η' 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 requirement, 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 $\hat{\mathbf{C}}_{Tr,i}$ as the i-th training fingerprint stored in the database and $\hat{\mathbf{C}}_{Te}$ as the fingerprint with respect to a testing sample. We then define a scalar $l_i = ||\hat{\mathbf{C}}_{Tr,i} - \hat{\mathbf{C}}_{Te}||_F^2$. After calculating all N_{tr} distances l_i, we select the first $k = 4$ lowest l_i and denote their coordinates as $\tilde{\mathbf{p}}_i$. Applying the weighted KNN algorithm [42], the final estimated position $\check{\mathbf{p}} \in \mathbb{R}^2$ is calculated as

$$\check{\mathbf{p}} = \sum_{i=1}^{4} w_i \tilde{\mathbf{p}}_i, \quad (12)$$

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 Δ_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
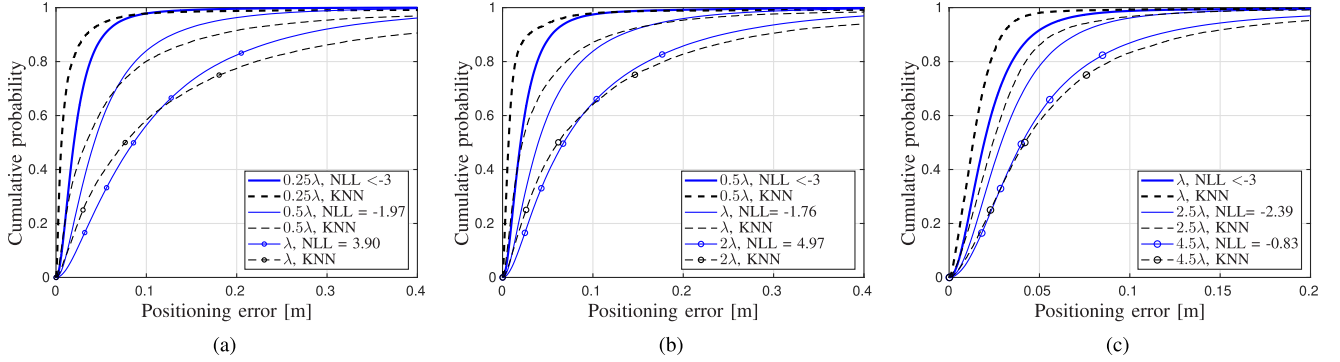
FIGURE 12. Comparison between positioning error of the pipeline and kNN method when using covariance matrices w.r.t. different average distances as training fingerprints. (a) $d = 0$, one-sample covariance matrix, (b) $d = \lambda/2$ (c) $d = 2\lambda$.

accuracy. However, when the training density is reduced, it cannot be guaranteed that 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 $\mathcal{N}_r$ prestored training fingerprints in total. This leads to a high complexity in time, which is $o(M^2 \mathcal{N}_r)$ 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 $\mathcal{N}_r$ 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 $\mathcal{Y}_{p_i} \in \mathbb{R}^{M \times F \times 2}$ by collecting the real and imaginary part of the raw received complex transfer function Y_{p_i}. Then Y_{p_i} 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 $\tilde{\mathcal{Y}}_{p_i} \in \mathbb{R}^{M \times F \times 2}$. The inverse Fourier transform was also performed to obtain the CIR matrix
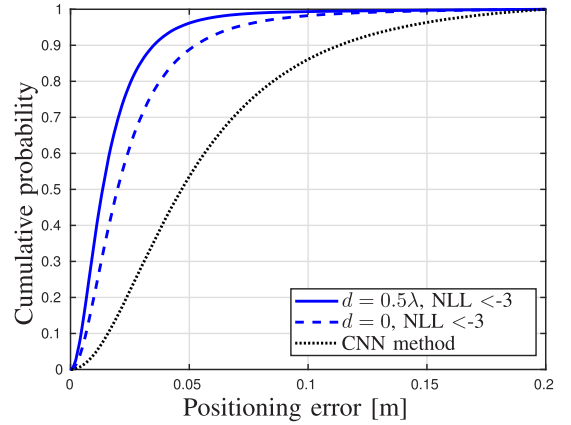

FIGURE 13. Positioning errors of our pipeline and localization approach in [40] under LoS scenario.

$\tilde{\Xi} \in \mathbb{C}^{M \times F}$ and the corresponding tensor $\hat{\mathcal{Y}}_{p_i} \in \mathbb{R}^{M \times F \times 2}$. In the next step, the author formulated a tensor $\mathcal{I} \in \mathbb{R}^{M \times F \times 6}$ by concatenating $\mathcal{Y}_{p_i}$, $\tilde{\mathcal{Y}}_{p_i}$, and $\hat{\mathcal{Y}}_{p_i}$ 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 $\Delta_d = \frac{1}{4}\lambda$ 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 pre-processing step as well as the subarray method. We have also performed a time complexity analysis of the convolutional neural network, which is $o(M F l_1 l_2 C_{in} C_{out})$, 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 $M^2 \leq F l_1 l_2 C_{in} C_{out}$ 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.
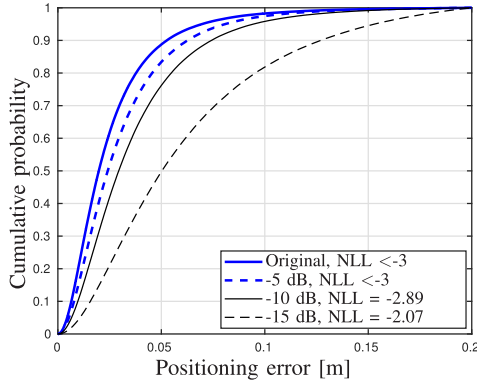
FIGURE 14. Positioning errors of our pipeline under different SNR.

F. INVESTIGATING DIFFERENT CASES

1) THE INFLUENCE OF SNR ON POSITIONING ACCURACY

As stated in Section IV, the SNRs at the subcarriers range between 1 and 11 dB in our measurement scenario. To further test the performance of our localization pipeline, especially under low SNR, synthetic white Gaussian noise is added to our measurement data to emulate measurement environments with mean SNR -5 dB, -10 dB, and -15 dB. Fig. 14 illustrates the localization accuracy of our algorithm that trains on the covariance matrix ($d = \frac{1}{2}\lambda$) in different SNR scenarios. The percentage of training is around 2.5% and $\Delta_d = \frac{1}{2}\lambda$. As illustrated, our algorithm still delivers good localization performance under the -10 dB SNR scenario. Even when the SNR drops to -15 dB, the localization accuracy is still acceptable for applications such as indoor navigation. This is because our processing pipeline can harvest the SNR gain from correlating over other frequencies and averaging over other one-sample covariance matrices in the neighborhood region according to (2).

2) INVESTIGATING NLoS MEASUREMENT SCENARIOS

We investigate the localization performance of our proposed pipeline in none-line-of-sight (NLoS) measurement scenarios. To this end, our localization is applied to an open source indoor measurement dataset [43]. We provide a brief introduction to the NLoS measurement campaign and parameter settings, while more details can be found in [40]. Fig. 16 illustrates the arrangement of indoor measurement, where 4 UEs, which occupy different subcarriers, move within the four gray squares, each with a size 1.2×1.2 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 0.56×0.56 m. The index of the antenna on the i-th row and the j-th column
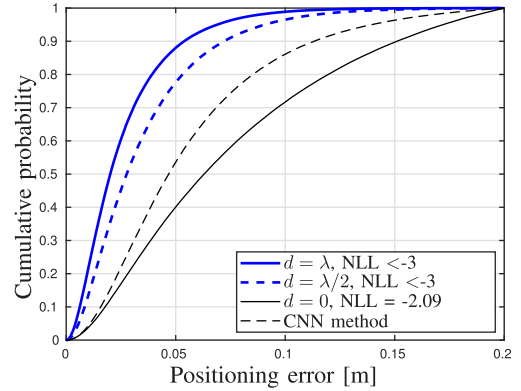

FIGURE 15. Positioning errors of our pipeline and localization approach in [40] under NLoS scenario.

TABLE 5. Antenna indexes in 4 groups.

Subarray group number	Antenna index
I	1-6, 9-14, 17-22, 25-30, 33-38,41-46
II	2-8, 11-16, 19-24, 27-32, 35-40, 43-48
III	17-22, 25-30, 33-38, 41-46, 49-54, 57-62
IV	19-24, 27-32, 35-40, 43-48, 51-56, 59-64

is $8(i-1)+j$. A metal blocker with size 1.6×1.3 m is placed between the base station and the UE, blocking the LoS component. Each UE sounds 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 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 1296×1024. 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 $\tilde{C}_{i,\mathcal{N}_{p_i}}$ 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
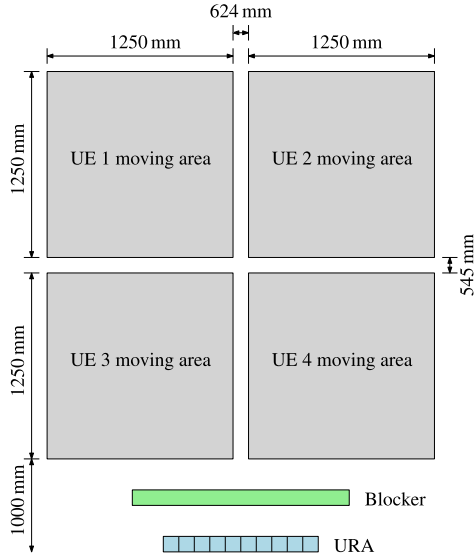
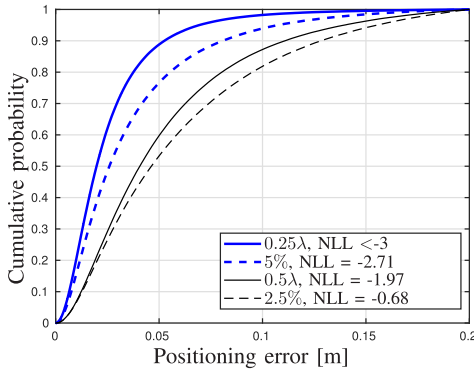**FIGURE 16. A demonstration of the indoor NLoS environment.**

FIGURE 17. Comparison between different ways of constructing the dataset.

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 $\Delta_d = \frac{1}{4}\lambda$. If $\Delta_d = \frac{1}{2}\lambda$, 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.

REFERENCES

[1] X. Cai, X. Cheng, and F. Tufvesson, "Toward 6G with terahertz communications: Understanding the propagation channels," 2022, *arXiv:2209.07864*.

[2] R. Whiton, "Cellular localization for autonomous driving: A function pull approach to safety-critical wireless localization," *IEEE Veh. Technol. Mag.*, vol. 17, no. 4, pp. 28–37, Dec. 2022.

[3] A. A. Abdallah, C.-S. Jao, Z. M. Kassas, and A. M. Shkel, "A pedestrian indoor navigation system using deep-learning-aided cellular signals and ZUPT-aided foot-mounted IMUs," *IEEE Sensors J.*, vol. 22, no. 6, pp. 5188–5198, Mar. 2022.
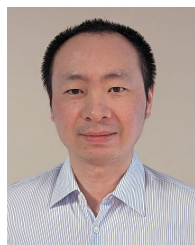
[4] M. Maaref and Z. M. Kassas, "Ground vehicle navigation in GNSS-challenged environments using signals of opportunity and a closed-loop map-matching approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 7, pp. 2723–2738, Jul. 2020.

[5] X. Mu, Y. Liu, L. Guo, J. Lin, and R. Schober, "Intelligent reflecting surface enhanced indoor robot path planning: A radio map-based approach," *IEEE Trans. Wireless Commun.*, vol. 20, no. 7, pp. 4732–4747, Jul. 2021.

[6] R. Whiton, J. Chen, T. Johansson, and F. Tufvesson, "Urban navigation with LTE using a large antenna array and machine learning," in *Proc. IEEE 95th Veh. Technol. Conf. (VTC-Spring)*, Jun. 2022, pp. 1–5.

[7] J. A. D. Peral-Rosado, R. Raulefs, J. A. López-Salcedo, and G. Seco-Granados, "Survey of cellular mobile radio localization methods: From 1G to 5G," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 2, pp. 1124–1148, 2nd Quart., 2018.

[8] S. Dwivedi et al., "Positioning in 5G networks," *IEEE Commun. Mag.*, vol. 59, no. 11, pp. 38–44, Nov. 2021.

[9] E. Leitinger, P. Meissner, C. Rüdisser, G. Dumphart, and K. Witrisal, "Evaluation of position-related information in multipath components for indoor positioning," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 11, pp. 2313–2328, Nov. 2015.

[10] A. Hu, T. Lv, H. Gao, Z. Zhang, and S. Yang, "An ESPRIT-based approach for 2-D localization of incoherently distributed sources in massive MIMO systems," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 5, pp. 996–1011, Oct. 2014.

[11] X. Zeng, F. Zhang, B. Wang, and K. J. R. Liu, "Massive MIMO for high-accuracy target localization and tracking," *IEEE Internet Things J.*, vol. 8, no. 12, pp. 10131–10145, Jun. 2021.

[12] X. Li, E. Leitinger, M. Oskarsson, K. Åström, and F. Tufvesson, "Massive MIMO-based localization and mapping exploiting phase information of multipath components," *IEEE Trans. Wireless Commun.*, vol. 18, no. 9, pp. 4254–4267, Sep. 2019.

[13] L. Lian, A. Liu, and V. K. N. Lau, "User location tracking in massive MIMO systems via dynamic variational Bayesian inference," *IEEE Trans. Signal Process.*, vol. 67, no. 21, pp. 5628–5642, Nov. 2019.

[14] N. Garcia, H. Wymeersch, E. G. Larsson, A. M. Haimovich, and M. Coulon, "Direct localization for massive MIMO," *IEEE Trans. Signal Process.*, vol. 65, no. 10, pp. 2475–2487, May 2017.

[15] X. Cai, W. Fan, X. Yin, and G. F. Pedersen, "Trajectory-aided maximum-likelihood algorithm for channel parameter estimation in ultrawideband large-scale arrays," *IEEE Trans. Antennas Propag.*, vol. 68, no. 10, pp. 7131–7143, Oct. 2020.

[16] X. Cai and W. Fan, "A complexity-efficient high resolution propagation parameter estimation algorithm for ultra-wideband large-scale uniform circular array," *IEEE Trans. Commun.*, vol. 67, no. 8, pp. 5862–5874, Aug. 2019.

[17] S. D. Bast, A. P. Guevara, and S. Pollin, "CSI-based positioning in massive MIMO systems using convolutional neural networks," in *Proc. IEEE 91st Veh. Technol. Conf. (VTC-Spring)*, May 2020, pp. 1–5.

[18] S. De Bast and S. Pollin, "MaMIMO CSI-based positioning using CNNs: Peeking inside the black box," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, Jun. 2020, pp. 1–6.

[19] J. R. Sánchez, O. Edfors, and L. Liu, "Positioning for distributed large intelligent surfaces using neural network with probabilistic layer," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2021, pp. 1–6.

[20] E. Gönültaş, E. Lei, J. Langerman, H. Huang, and C. Studer, "CSI-based multi-antenna and multi-point indoor positioning using probability fusion," *IEEE Trans. Wireless Commun.*, vol. 21, no. 4, pp. 2162–2176, Apr. 2022.

[21] K. N. R. S. V. Prasad, E. Hossain, and V. K. Bhargava, "Low-dimensionality of noise-free RSS and its application in distributed massive MIMO," *IEEE Wireless Commun. Lett.*, vol. 7, no. 4, pp. 486–489, Aug. 2018.

[22] K. N. R. S. V. Prasad, E. Hossain, and V. K. Bhargava, "Machine learning methods for RSS-based user positioning in distributed massive MIMO," *IEEE Trans. Wireless Commun.*, vol. 17, no. 12, pp. 8402–8417, Dec. 2018.

[23] X. Guo and N. Ansari, "Localization by fusing a group of fingerprints via multiple antennas in indoor environment," *IEEE Trans. Veh. Technol.*, vol. 66, no. 11, pp. 9904–9915, Nov. 2017.

[24] J. Vieira, E. Leitinger, M. Sarajlic, X. Li, and F. Tufvesson, "Deep convolutional neural networks for massive MIMO fingerprint-based positioning," in *Proc. IEEE 28th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Oct. 2017, pp. 1–6.

[25] X. Sun, C. Wu, X. Gao, and G. Y. Li, "Fingerprint-based localization for massive MIMO-OFDM system with deep convolutional neural networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 11, pp. 10846–10857, Nov. 2019.

[26] Y. Lin, K. Yu, L. Hao, J. Wang, and J. Bu, "An indoor Wi-Fi localization algorithm using ranging model constructed with transformed RSSI and BP neural network," *IEEE Trans. Commun.*, vol. 70, no. 3, pp. 2163–2177, Mar. 2022.

[27] Y. Yuan, X. Liu, Z. Liu, Z. He, and Z. Xu, "Indoor localization with wireless heterogeneous devices by composite fingerprint sets and hybrid classification," *IEEE Trans. Veh. Technol.*, vol. 71, no. 11, pp. 12117–12127, Nov. 2022.

[28] P. Ferrand, A. Decurninge, and M. Guillaud, "DNN-based localization from channel estimates: Feature design and experimental results," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2020, pp. 1–6.

[29] J. Fan, X. Dou, W. Zou, and S. Chen, "Localization based on improved sparse Bayesian learning in mmWave MIMO systems," *IEEE Trans. Veh. Technol.*, vol. 71, no. 1, pp. 354–361, Jan. 2022.

[30] S. De Bast, E. Vinogradov, and S. Pollin, "Expert-knowledge-based data-driven approach for distributed localization in cell-free massive MIMO networks," *IEEE Access*, vol. 10, pp. 56427–56439, 2022.

[31] X. Ye, X. Yin, X. Cai, A. P. Yuste, and H. Xu, "Neural-network-assisted UE localization using radio-channel fingerprints in LTE networks," *IEEE Access*, vol. 5, pp. 12071–12087, 2017.

[32] R. Bharadwaj, A. Alomainy, and S. K. Koul, "Experimental investigation of body-centric indoor localization using compact wearable antennas and machine learning algorithms," *IEEE Trans. Antennas Propag.*, vol. 70, no. 2, pp. 1344–1354, Feb. 2022.

[33] G. Tian, I. Yaman, M. Sandra, X. Cai, L. Liu, and F. Tufvesson, "High-precision machine-learning based indoor localization with massive MIMO system," 2023, *arXiv:2303.03743*.

[34] S. Haykin, *Neural Networks: A Comprehensive Foundation*. New York, NY, USA: Prentice-Hall, 1998.

[35] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Proc. Conf. Neural Inf. Process. Syst.* Red Hook, NY, USA: Curran Associates, 2017, pp. 6405–6416.

[36] Z. H. Zhou, *Ensemble Methods: Foundations and Algorithms*. London, U.K.: CRC Press, 2012.

[37] T. Hill, "Conflations of probability distributions," *Trans. Amer. Math. Soc.*, vol. 363, no. 6, pp. 3351–3372, Jun. 2011.

[38] I. Yaman et al., "The LuViRA dataset: Measurement description," 2023, *arXiv:2302.05309*.

[39] S. Malkowsky et al., "The world's first real-time testbed for massive MIMO: Design, implementation, and validation," *IEEE Access*, vol. 5, pp. 9073–9088, 2017.

[40] A. Colpaert, S. De Bast, R. Beerten, A. P. Guevara, Z. Cui, and S. Pollin, "Massive MIMO channel measurement data set for localization and communication," *IEEE Commun. Mag.*, vol. 61, no. 9, pp. 114–120, Sep. 2023.

[41] M. Khalid, J. Baber, M. K. Kasi, M. Bakhtyar, V. Devi, and N. Sheikh, "Empirical evaluation of activation functions in deep convolution neural network for facial expression recognition," in *Proc. 43rd Int. Conf. Telecommun. Signal Process. (TSP)*, Jul. 2020, pp. 204–207.

[42] P. Cunningham and S. J. Delany, "K-nearest neighbour classifiers—A tutorial," *ACM Comput. Surv.*, vol. 54, no. 6, pp. 1–25, Jul. 2022.

[43] S. De Bast and S. Pollin, "Ultra dense indoor MaMIMO CSI dataset," Katholieke Universiteit Leuven, Leuven, Belgium, Tech. Rep. nr6k-8r78-21, 2021. [Online]. Available: https://ieee-dataport.org/open-access/ultra-dense-indoor-mamimo-csi-dataset, doi: 10.21227/nr6k-8r78.

GUODA TIAN (Member, IEEE) received the bachelor's degree in automation and control engineering from Northeastern University in 2016 and the master's degree in wireless communication from Lund University in 2018, where he is currently pursuing the Ph.D. degree, under the supervision of Fredrik Tufvesson, Ove Edfors, Bo Bernhordsson, and Xuesong Cai. His current research interests include applying machine learning to wireless localization, sensing, and communication systems. During the master's degree, he received the Lund University Global Scholarship.

ILAYDA YAMAN (Student Member, IEEE) received the bachelor's degree from Istanbul Technical University in 2018, and the master's degree in embedded electronics engineering from Lund University, Sweden, in 2020, where she is currently pursuing the Ph.D. degree under the supervision of Liang Liu, Ove Edfors, Kalle Åström, and Steffen Malkowsky. Her current research interests include fusing vision and radio sensors to design low-power hardware using machine learning algorithms. During the master's degree, she received the Lund University Global Scholarship for her academic success.

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 in channel modeling for maritime communications with Lund University, 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.
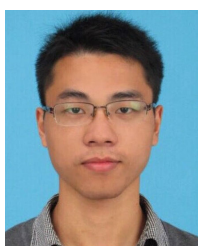
XUESONG CAI (Senior Member, IEEE) received the B.S. and Ph.D. (Hons.) degrees from Tongji University, Shanghai, China, in 2013 and 2018, respectively.

In 2015, he conducted a three-month internship with Huawei Technologies, Shanghai. He was a Visiting Scholar with Universidad Politcnica de Madrid, Madrid, Spain, in 2016. From 2018 to 2022, he conducted several post-doctoral stays at Aalborg University; Nokia Bell Labs, Denmark; and Lund University, Sweden. He is currently an Assistant Professor of communications engineering and a Marie Skłodowska-Curie Fellow with Lund University, closely cooperating with Ericsson and Sony. His current research interests include radio channel characterization, high-resolution parameter estimation, over-the-air testing, resource optimization, and radio-based localization for 5G/B5G wireless systems. He was a recipient of the China National Scholarship (the highest honor for Ph.D. candidates) in 2016, the Outstanding Doctorate Graduate awarded by the Shanghai Municipal Education Commission in 2018, the Marie Skłodowska-Curie Actions (MSCA) Seal of Excellence in 2019, the EU MSCA Fellowship (ranking top 1.2%, overall success rate 14%), and the Starting Grant (success rate 12%) funded by the Swedish Research Council in 2022. He was also selected by the ZTE Blue Sword-Future Leaders Plan in 2018 and the Huawei Genius Youth Program in 2021. He is an Associate Editor of IEEE Transactions on Vehicular Technology, *IET Communications*, and *Wireless Communications and Mobile Computing*.

LIANG LIU (Member, IEEE) received the 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-doctoral position. Since 2016, he has been an Associate Professor with Lund University. His current research interests include wireless systems and digital integrated circuit 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 current research interests include 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. His research has been awarded the Neal Shepherd Memorial Award for the Best Propagation Paper in IEEE Transactions on Vehicular Technology and the IEEE Communications Society Best Tutorial Paper Award.