A CSI-Based Data-Driven Localization Framework Using Small-Scale Training Datasets in Single-Site MIMO Systems

Katarina Vuckovic, Saba Hosseini, Farzam Hejazi, and Nazanin Rahnavard, *Senior Member, IEEE*

***Abstract*— This paper presents a new method for user localization in single-site massive Multiple-Input-Multiple-Output (MIMO) systems, which circumvents the need for large labeled datasets typically required for training data-driven models. Instead, the proposed model utilizes a limited set of geo-tagged Channel State Information (CSI) samples for training. The approach combines a Fully-Connected Auto-Encoder (FC-AE) with a Gaussian Process Regression (GPR) model. The GPR model is efficient, as it requires only a minimal amount of labeled data for training, although it presents challenges in computational complexity. To address this complexity, the FC-AE is introduced, which encodes the Angle-Delay Profile (ADP) transformation of the CSI data. The training dataset for the FC-AE is crafted by employing data augmentation techniques on a small collection of unlabeled data. The simulation results demonstrate that FC-AE is scenario-independent and adaptable to new scenarios with similar ADP characteristics. Additionally, our FC-AE-GPR model surpasses the performance of the Convolutional Neural Network model and the non-parametric grid search method when provided with limited labeled data, applicable in both indoor and outdoor settings.**

***Index Terms*— Single-site MIMO, localization, Gaussian processes regression, autoencoders, data augmentation.**

I. Introduction

THE abundance of smartphone devices and wireless networks has ushered in a new era for location-based service (LBS) applications. LBSs now find applications across various domains, including navigation systems, emergency services, travel planning, asset management, recommendations, and geo-social networking [1]. While Global Positioning System (GPS) serves as the ubiquitous outdoor positioning system, its limited accuracy (around 5 meters for civilian use) and performance issues in rich-scattering environments like urban canyons and indoors [2] make it less suitable for applications requiring higher precision, such as autonomous driving, indoor asset and people localization, and augmented reality. Consequently, researchers are exploring the use of wireless communication systems for localization.

3GPP has been actively advancing 5G technology, with Release 18 emphasizing the evolution of Multiple-Input-Multiple-Output (MIMO) technology and exploring the use of Artificial Intelligence and Machine Learning (AI/ML) for MIMO [3], [4]. Traditional Channel State Information (CSI) feedback methods, like codebooks [5] and Compressive Sensing (CS) [6], often fall short of meeting the requirements for low complexity and high accuracy in MIMO systems. Additionally, geometry-based localization solutions using Angle-of-Arrival (AoA) and Time-of-Arrival (ToA) channel data have been proposed [7], [8]. However, they tend to be outperformed by Deep Learning (DL) Neural Network (NN) models [9].

FingerPrinting (FP) localization is a technique that maps each location in the environment to a unique wireless measurement during offline training. It then uses this geo-tagged map for location estimations based on measurements captured during the online phase. The core concept in FP localization is the uniqueness of the wireless channel between the user and the Base Station (BS), which is determined by the surrounding scattering environment at the user's location [10]. There are two types of FP models based on the type of wireless system architecture. The first type uses Receive Single Strength (RSS) measurements, which is suitable for systems with rich AP distributions like Wireless Sensor Networks [11], [12], WiFi networks [13], [14], [15], [16], and distributed MIMO systems [17], [18]. The second type is single-site MIMO localization. In single-site localization, FP parameters leverage multipath characteristics of the MIMO channel captured in the CSI. Unlike RSS parameters, which require multiple Access Points (APs), CSI fingerprints can be used for single-site localization but necessitate a MIMO antenna for generating a unique fingerprint. Some studies transform the CSI matrix into an Angle-Delay Profile (ADP) matrix using a linear transformation [19]. Several FP techniques based on CSI or ADP data have been proposed for massive MIMO (m-MIMO) systems [19], [20], [21], [22], [23], [24]. The highest performance is achieved by training a Convolutional Neural Network (CNN) with ADP fingerprints [19].

The main challenge of using CNNs in a data-driven approach is collecting labeled data for each location within

Manuscript received 22 April 2023; revised 14 November 2023 and 3 May 2024; accepted 23 July 2024. Date of publication 14 August 2024; date of current version 13 November 2024. This work was supported in part by the National Science Foundation under Grant CCF-1718195 and Grant ECCS-1810256. The associate editor coordinating the review of this article and approving it for publication was L. Simic. *(Corresponding author: Katarina Vuckovic.)*

Katarina Vuckovic, Saba Hosseini, and Nazanin Rahnavard are with the Department of Electrical and Computer Engineering, University of Central Florida, Orlando, FL 32816 USA (e-mail: ka966104@ucf.edu).

Farzam Hejazi is with Qualcomm, Santa Clara, CA 95051 USA.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TWC.2024.3440275.

Digital Object Identifier 10.1109/TWC.2024.3440275

the environment, which is time-consuming and impractical. Precise position labels matching the measured CSI data are hard to obtain, and scenario changes require retraining the model with an updated dataset, impeding real-world deployment. Some studies explore transfer learning techniques to adapt models to new environments [25], [26], [27], [28]. However, this still requires substantial labeled data for initial training, while our approach uses a much smaller labeled dataset during initial training.

Channel Charting (CC) is an unsupervised method for mapping CSI data to a channel chart space that captures the local radio geometry of transmitting User Equipment (UE) [29], [30], [31], [32]. Feature vectors from UEs' CSI data are used to construct low-dimensional charts using dissimilarity measures [29], [33]. Traditional CC represents UEs' positions in terms of proximity and spatial arrangement without precise physical locations [34]. Conversely, FP-based localization aims to pinpoint a device's global coordinates in a known area, relevant for indoor navigation, virtual reality, and augmented reality [35]. The CC methodology now blends supervised and unsupervised approaches using partially-annotated datasets [36], [37], [38], [39], [40]. A recent study [39], building on [38], explores SNR prediction with BS beam-specific CSI features. They develop SNR predictors, both FP and CC-based, for various target beams during offline training, applying them in real-time to deduce a user's SNR mapping function. For CC-based predictors, a low-dimensional CC is generated from unlabeled data, with chart locations marked by beam-wise SNR labels. GPR slightly outperformed an NN model as a predictor, though both yielded comparable results. The researchers note GPR's potential with other input features but exclude high-dimensional CSI fingerprint due to complexity, which our research aims to address. Another work [40] shows a semi-supervised CC-based framework outperforming conventional methods using 85% of 72,000 unlabeled samples to build the chart and an AE with 85% of 30,000 labeled CSI samples for accuracy. Our fully-supervised approach achieves favorable results with a significantly smaller dataset.

Due to the small number of hyper-parameters, GPR models have shown the ability to be trained with small-scale datasets. They have been primarily used in RSS-based FP for Indoor Positioning Systems (IPS) [13], [15], [16], [41], [42], [43], [44], [45]. DeepMap [45] proposes using Deep Gaussian Processes Regression (DGPR) [46] to construct an indoor radio map and performs RSS-based FP with commodity WiFi devices. The authors design a two-layer DGPR and compare it to a single-layer GPR. While their simulation results show that DeepMap improves the performance of GPR for reduced training datasets, the challenge with DeepMap is its computational complexity. As the size of the fingerprint increases, the computational complexity can become infeasible quickly [47]. Since the DGPR is even more complex than a GPR model, this approach is not practical for CSI measurements where the input sample can be considerably large. Furthermore, it is crucial to highlight that all the aforementioned research has exclusively focused on RSS FPs and omni antennas with multiple APs. None of these studies have explored the applicability of GPR models to single-site MIMO systems using CSI fingerprints, which is the primary focus of our paper. The only work that investigates single-site outdoor localization using GPR models is in [48] where the authors propose a DCGP regression model [49] for m-MIMO outdoor localization. The simulation results show that the DCGP regression outperforms CNN regression FP using ADPs. Even though DCGP demonstrates improved accuracy performance over CNN, the dataset size required to train the CNN and DCGP is the same, and authors do not discuss training with datasets of different sizes. The "deep convolution" part of the DCGP still requires a larger training dataset and the true benefit of GPR models (ability to train on small-scale datasets) is not exploited.

The most prominent weakness of GPR models is that they suffers from high computational and memory complexity. The complexity of the model often poses a limitation in many applications. Specifically, the GPR has a training computational complexity of $O(n^3)$ and a memory complexity of $O(n^2)$, where n is the number of training points in the dataset [50]. The computational complexity is also impacted by the dimension (d) of the input data sample vector. Although some kernels can be more computationally efficient than others, the complexity quickly increases as the size of the input data vector grows. Therefore, the training time can be reduced by either using a small training dataset or by reducing the dimension of the input vector. To reduce the computational complexity of the GPR model, we employ a compression model to decrease the size of the ADP fingerprint sample.

Compression of CSI data has been extensively explored in literature [4]. Inspired by the pioneering CsiNet research [51], the user-end encoder compresses downlink CSI, which is sent to the BS for reconstruction. CsiNet employs CNN for both the encoder and decoder and it demonstrates its superior CSI reconstruction quality compared to traditional CS-based methods. Additionally, preprocessing is vital for DL-based CSI feedback. Studies suggest transforming CSI from the spatial frequency domain to the angular-delay domain using 2D Discrete Fourier Transform (DFT) to reduce feedback overhead while preserving sparsity [51], [52], [53], [54], [55], [56].

Building upon the groundwork laid by CsiNet, subsequent developments in multi-user MIMO CSI feedback using NNs can be categorized into two groups. The first group adopts more complex network architectures, like CsiNet+ [57], JC-ResNet [58], and CRNet [59], albeit with higher computational demands. The second group leverages inherent correlation characteristics within channel responses [53], [54]. Enet [54] leverages inherent channel response correlations in the angular-delay domain, employing domain-specific compression strategies. This approach enhances CSI compression and feedback, significantly reducing the network size compared to existing NN-based solutions. In our proposed model, we convert the CSI to the sparse ADP and employ a Fully-Connected Auto-Encoder (FC-AE) model to compress the dimension of the ADP fingerprint. While other compression techniques may be explored, our simulations demonstrates that FC-AE encodes the samples sufficiently well.

We propose to combine the FC-AE encoder with the GPR model into a novel FC-AE-GPR framework. GPR, on its own, demonstrates superior performance over CNN when trained with a small dataset. However, in m-MIMO systems, increasing the number of array antennas and subcarriers directly raises ADP size and, consequently, increasing GPR training complexity. Therefore, we introduce an FC-AE to reduce the ADP matrix size before training the GPR model, thus reducing the GPR training complexity. The sparse ADP transformation of CSI enables efficient ADP compression [60], [61]. To train the FC-AE, we utilize a small set of unlabeled ADPs and then expand the training dataset through data augmentation techniques. The paper contributions are summarized as follows:

- It is asserted that state-of-the-art DL models pose significant challenges for real-world deployment, primarily due to the extensive labeled data required.
- A sophisticated GPR optimization model is introduced, which aims to identify the optimal kernel type and corresponding hyperparameters. This optimization tunes hyperparameters for five distinct kernel types, to select the most effective kernel and hyperparameter pairing.
- The issue of memory requirements and computational complexity, which escalates with increasing input sample size in GPR models, is addressed by introducing a FC-AE to compress the ADP prior to GPR training.
- The FC-AE's scenario-independence is argued and substantiated by tests conducted on the FC-AE in various representative environments. The FC-AE requires only a single training phase and can then be applied to new scenarios with similar ADP patterns. Data augmentation on a limited set of unlabeled data is employed to develop the FC-AE's training dataset.
- A novel FC-AE-GPR architecture is presented that integrates the FC-AE and GPR. Its performance is compared with that of standalone GPR, CNN, and a non-parametric grid search method, showcasing the superior performance of both GPR and FC-AE-GPR with small training datasets in different outdoor and indoor settings. Additionally, it is shown that while FC-AE-GPR can boost GPR's performance, its main benefit is the reduction in computational complexity.

The rest of the paper is organized as follows. Section II defines the channel model and the ADP matrix. Section III provides an in-depth GPR model discussion, while Section IV introduces the localization framework. Section V covers the simulation dataset and training model results, and Section VI assesses the localization performance.

II. Channel Model and Angle-Delay Profile

A. Channel Model

Consider a typical MIMO Orthogonal Frequency-Division Multiplexing (OFDM) wireless network with a single BS. Similar to [62], assume the BS is equipped with a Uniform Linear Array (ULA) antenna with N_t antenna elements and it uses OFDM signaling with N_c subcarriers. Furthermore, the UE has a single omni-directional antenna. The channel between the BS and the user is modeled using COST 2100 [63] with C distinguishable clusters. Moreover, each cluster constitutes R_C distinguishable paths. Each path can be characterized by a delay $\tau_m^{(k)}, k \in \{1,\dots,C\}, m \in \{1,\dots,R_C\}$, an AoA to the BS's antenna $\theta_m^{(k)}$ and a complex gain $\alpha_m^{(k)}$ [62]. Given a wide-band OFDM system, $\tau_m^{(k)} = n_m^{(k)} T_s$, where T_s and $n_m^{(k)}$ denote the sampling duration and the sampled delay belonging to the path m of the cluster k, respectively [21]. Then the bandwidth of each subcarrier is $f = 1/(N_c T_s)$ and $f_l = lf$ is the l^{th} subcarrrier. Channel Frequency Response (CFR) for each subcarrier l can be written as [64]

$$\boldsymbol{h}[l] = \sum_{k=1}^{C} \sum_{m=1}^{R_C} \alpha_m^{(k)} \boldsymbol{e}(\theta_m^{(k)}) e^{-j2\pi \frac{l\, n_m^{(k)}}{N_c}} \tag{1}$$

where $\boldsymbol{e}(\theta) = [1, e^{-j2\pi \frac{d cos(\theta)}{\lambda}}, \dots, e^{-j2\pi \frac{(N_t-1) d cos(\theta)}{\lambda}}]^T$ denotes the array response vector of the ULA and d is the gap between two adjacent antennas The overall CFR matrix of the channel between the BS and the user can be expressed as $\boldsymbol{H} = [\boldsymbol{h}[1], \dots, \boldsymbol{h}[N_c]]$. This matrix is known as CSI.

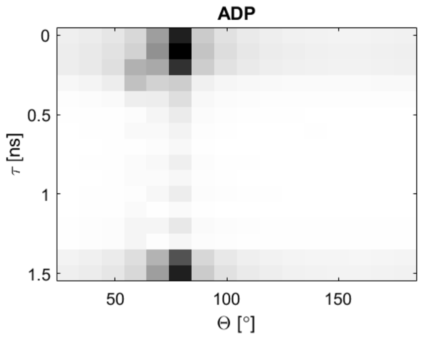

Fig. 1. Example of an $N_t \times N_c = 16 \times 16$ ADP matrix. The x-axis and y-axis show the angle θ and delay τ, respectively.

B. Angle-Delay Profile

The ADP matrix is a linear transformation of the CSI matrix computed by multiplying the CSI matrix with two DFT matrices [7]. The transformation maps the space frequency domain CSI to the angle and delay domain [21]. Referring to [19], the DFT matrix $\boldsymbol{V} \in \mathbb{C}^{N_t \times N_t}$ is defined as

$$[\boldsymbol{V}]_{z,q} \triangleq \frac{1}{\sqrt{N_t}} e^{-j2\pi \frac{\left(z(q - \frac{N_t}{2})\right)}{N_t}}, \tag{2}$$

and DFT matrix $\boldsymbol{F} \in \mathbb{C}^{N_c \times N_c}$ as

$$[\boldsymbol{F}]_{z,q} \triangleq \frac{1}{\sqrt{N_c}} e^{-j2\pi \frac{zq}{N_c}}. \tag{3}$$

Then, the ADP matrix $\boldsymbol{A} \in \mathbb{R}^{N_t \times N_c}$ is defined as

$$\boldsymbol{A} = |\boldsymbol{V}^H \boldsymbol{H} \boldsymbol{F}|, \tag{4}$$

where $|\cdot|$ is the absolute value.

The transformation in (4) converts the dense CSI matrix into a sparse ADP matrix. Sparse matrices are easier to compress using encoding methods, reducing GPR optimization complexity. Additionally, ADP fingerprints are highly correlated in the location domain, and similarities between ADPs decrease smoothly with physical distance [20]. This characteristic is crucial for the GPR model, forming the prior assumption in the Bayesian model.

III. Gaussian Process Regression

A. Motivation for Using Gaussian Processes

GPR is a versatile non-parametric supervised ML model rooted in Bayesian statistics [65]. Unlike parametric models, GPR does not assume a specific function form, enabling it to estimate various functions. While CNNs can also model complex functions, they typically require large datasets, whereas GPR generalizes better with smaller datasets [45]. Additionally, GPR incorporates prior parameter assumptions from a Bayesian standpoint, and by selecting appropriate kernels, it encodes spatial relationships among input points [66].

B. Gaussian Processes

A *Gaussian Process* (GP) is a collection of random variable functions indexed by time or space. The key property of a GP is that any finite subset of the random variables is jointly Gaussian distributed. For any finite set of vector elements $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathcal{X}$, the associated set of random variables $f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n)$ follows a joint Gaussian distribution. This joint distribution is fully determined by the mean vector $\boldsymbol{\mu}$ and covariance matrix $\mathbf{K}$, where $\boldsymbol{\mu} = (m(\mathbf{x}_1), \ldots, m(\mathbf{x}_n))^T$ and $[\mathbf{K}]ij = k(\mathbf{x}_i, \mathbf{x}_j)$, such that

$$\begin{bmatrix} f(\mathbf{x}_1) \\ \vdots \\ f(\mathbf{x}_n) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(\mathbf{x}_1) \\ \vdots \\ m(\mathbf{x}_n) \end{bmatrix}, \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & \ldots & k(\mathbf{x}_1, \mathbf{x}_n) \\ \vdots & \ddots & \vdots \\ k(\mathbf{x}_n, \mathbf{x}_1) & \ldots & k(\mathbf{x}_n, \mathbf{x}_n) \end{bmatrix} \right). \tag{5}$$

The following notation is commonly used in literature to represent the GP

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')), \tag{6}$$

where for any $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$ the mean and covariance functions are defined as [42]

$$m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})], \tag{7}$$

$$k(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}')]. \tag{8}$$

C. Covariance Functions

In statistics, covariance measures the correlation between two random variables $\mathbf{x}_i$ and $\mathbf{x}_j$ [67], while the covariance function (or *kernel*) describes the spatial or temporal correlation of a random process. The kernel defines the relationship between points, thus encoding the assumptions about the function that we want to learn [66]. It assumes strong correlation among nearby points, so the prediction is informed by nearby training points. In our model, ADP fingerprints represent input points, so a kernel should establish spatial correlation. Nearby locations should have similar fingerprints, with similarity decreasing monotonically as the distance increases. Kernels that meet this criteria are listed in Table I, where σ is the signal's standard deviation, l is the characteristic kernel scale, and α is the positive scale-mixture parameter, and $\mathbf{r}$ is the Euclidean distance between $\mathbf{x}_i$ and $\mathbf{x}_j$.

TABLE I

Different Monotonically Decreasing Kernel

Name	**Kernel** $k(\mathbf{x}_i, \mathbf{x}_j)$	Θ
Squared Exponential	$\sigma^2 exp(-\frac{\mathbf{r}^2}{2l^2})$	(σ, l)
Exponential	$\sigma^2 exp(-\frac{\mathbf{r}}{2l^2})$	(σ, l)
Rational Quadratic	$\sigma^2(1 + \frac{\mathbf{r}}{2\alpha l^2}){-\alpha}$	(σ, α, l)
Matern32	$\sigma^2(1 + \frac{\sqrt{3}\mathbf{r}}{l}) exp(-\frac{\sqrt{3}\mathbf{r}}{l})$	(σ, l)
Matern52	$\sigma^2(1 + \frac{\sqrt{5}\mathbf{r}}{l} + \frac{5\mathbf{r}^2}{3l^2}) exp(-\frac{\sqrt{5}\mathbf{r}}{l})$	(σ, l)

D. Gaussian Process Regression Model

The GPR model employs GPs to represent a continuous function, assuming that any point in the function's domain follows a multivariate normal distribution [66]. This model assumes that a latent function $f(\cdot)$ generates observed outputs y_i from input vectors $\mathbf{x}_i$ as:

$$y_i = f(\mathbf{x}_i) + \epsilon_i, \tag{9}$$

where $f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$, and $\epsilon \sim \mathcal{N}(0, \sigma^2\mathbf{I})$ is the noise of the system that has an independent, identically distributed Gaussian distribution with zero mean and variance σ^2. The index i refers to the i^{th} observation. For a given training and testing dataset, $\mathcal{D}_{train} \triangleq (\mathbf{X}, \mathbf{y}) \triangleq \{\mathbf{x}_i, y_i\}_{i=1}^n, \mathbf{x}_i \in \mathbb{R}^d, y_i \in \mathbb{R}$ and $\mathcal{D}_{test} \triangleq (\mathbf{X}_*, \mathbf{y}_*) \triangleq \{\mathbf{x}_{*i}, y_{*i}\}_{i=1}^n, \mathbf{x}_{*i} \in \mathbb{R}^d, y_{*i} \in \mathbb{R}$, the posterior predictive distribution is obtained from the joint Gaussian distribution in (5). Therefore, the joint distribution of the observed values $\mathbf{y}$ and the predicted function $\mathbf{y}_*$ at the new testing sample $\mathbf{X}_*$ is defined as

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{y}_* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} \mathbf{K} + \sigma_n^2\mathbf{I} & \mathbf{K}_* \\ \mathbf{K}_*^{\mathbf{T}} & \mathbf{K}_{**} \end{bmatrix}\right), \tag{10}$$

where $\mathbf{K} = K(\mathbf{X}, \mathbf{X})$, $\mathbf{K}_* = K(\mathbf{X}, \mathbf{X}_*)$, and $\mathbf{K}_{**} = K(\mathbf{X}_*, \mathbf{X}_*)$ [65]. From conditional distribution, it follows that

$$\mathbf{y}_* | (\mathbf{y}, \mathbf{X}, \mathbf{X}_*) \sim \mathcal{N}(\mu_*, \Sigma_*) \tag{11}$$

where the mean $\bar{\mathbf{y}}_*$ and variance $\mathbb{V}[\bar{\mathbf{y}}_*]$ of the unknown function $\mathbf{y}_*$ are computed as [65]

$$\mu_* = \bar{\mathbf{y}}_* = \mathbf{K}_*^{\mathbf{T}}((\mathbf{K} + \sigma_n^2\mathbf{I})^{-1}\mathbf{y}, \tag{12}$$

$$\Sigma_* = \mathbb{V}[\bar{\mathbf{y}}_*] = \mathbf{K}_{**} - \mathbf{K}_*^{\mathbf{T}}(\mathbf{K} + \sigma_n^2\mathbf{I})^{-1}\mathbf{K}_*. \tag{13}$$

E. Hyper-Parameter Optimization

The hyper-parameters describe properties of the kernel and the noise in the GP. Bayesian optimization is used to find the optimal hyper-parameters given by the vector $\boldsymbol{\Theta}$ as defined in Table I. The hyper-parameters can be estimated by minimizing the negative log marginal likelihood (NLML) function [66]

$$\log(P(\mathbf{y}|\mathbf{X}, \boldsymbol{\Theta})) = -\underbrace{\frac{\mathbf{y}^T(\mathbf{K} + \sigma_n^2\mathbf{I})^{-1}\mathbf{y}}{2}}_{\text{model-fit}} - \underbrace{\frac{\log|\mathbf{K} + \sigma_n^2\mathbf{I}|}{2}}_{\text{complexity penalty}} - \underbrace{\frac{n\log(2\pi)}{2}}_{\text{normalization constant}}. \tag{14}$$

The NLML in (14) has three distinctive terms that can be readily interpreted. The first is the *model-fit* term, which is the only term that includes the observed targets. The second

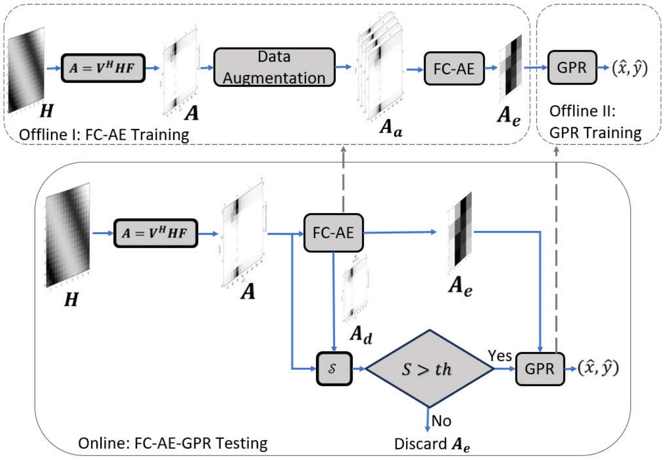

Fig. 2. The FC-AE is trained in the offline phase with the augmented ADP dataset. Afterwards, the GPR is trained offline with small number of location labeled ADP samples. During online testing, the framework predicts the user location using the measured CSI sample.

is the *complexity penalty* term, which only depends on the covariance function and the inputs. Finally, the last term is the *normalization constant*. The hyper-parameters are optimized by computing the partial derivative of the NLML with respect to $\boldsymbol{\Theta}$ and then using a gradient descent method to update the hyper-parameters in the GPR model at every iteration.

IV. Localization Framework

Fig. 2 illustrates the proposed localization framework, consisting of two main components: the FC-AE model and the GPR model, hence the name FC-AE-GPR. Each model is trained separately during the offline phase and then combined for testing. The FC-AE can be used across different scenarios, while the GPR needs retraining for each new location. The next section discusses each component in detail.

A. Offline Phase I: Data Augmentation and FC-AE Training

The ADP size in (4) depends on the number of antenna array elements (N_t) and subcarriers (N_c), i.e., $\boldsymbol{A} \in \mathbb{R}^{N_t \times N_c}$. GPR training complexity is manageable for small N_t and N_c. However, as N_t and N_c increase, the ADP size becomes computationally burdensome. To reduce GPR's training complexity, the FC-AE compresses the ADP before GPR training.

The FC-AE structure first reduces the number of perceptrons at each layer to a bottleneck (encoder) and then gradually expands the number of perceptrons in each subsequent layer (decoder) [68]. The encoder compresses input data into a latent representation, and the decoder reconstructs the original input from this representation. Training an FC-AE involves minimizing a reconstruction loss, penalizing differences between the decoded output and the input. Similarity $\mathscr{S}$ is the normalized correlation between input $\boldsymbol{A_a}$ and decoded output $\boldsymbol{A_d}$

$$\mathscr{S}(\boldsymbol{A_a}, \boldsymbol{A_d}) = \frac{\text{vec}(\boldsymbol{A_a}).\text{vec}(\boldsymbol{A_d})}{||\boldsymbol{A_a}||_F ||\boldsymbol{A_d}||_F}, \tag{15}$$

where vec(.) is an operator that concatenates columns of a matrix into a vector, operation . denotes inner product, and $||.||_F$ signifies the Frobenius norm. The FC-AE demands a substantial dataset for training. However, the dataset does not need to be labeled. Furthermore, the need for an extensive dataset is mitigated by data augmentation, where a small ADP dataset is expanded by applying various transformations such as rotation, translation, and random flipping on the ADP samples. This augmented dataset is diverse enough, enabling the FC-AE to handle both the training scenario and new scenarios. The adaptability of the FC-AE depends on its architecture as well as the quality and quantity of the augmented dataset.

B. Offline Phase II: GPR Training

GPR training occurs in the second offline phase. Initially, the ADPs are encoded using the pre-trained FC-AE. These labeled $\boldsymbol{A_e}$ samples are then employed to train the GPR model. As defined in (9), the GPR model is trained to predict user's location given a $\boldsymbol{A_e}$ input. Since the GPR model in (9) is a single-output model, two parallel models are trained to estimate both the $\hat{x}$ and $\hat{y}$ coordinates. To reduce computational complexity, two separate single-output GPR models are used for predicting $\hat{x}$ and $\hat{y}$ instead of one multi-output GPR. The training complexity of a multiple output GPR is $O(T^3 n^3)$ where T is the number of output parameters and n is the number of samples [50]. Given the emphasis is on the computational training efficiency of the GPR model, implementing two single-output model is a more efficient choice. Moreover, single GPR models may be trained in parallel, further reducing the overall training time.

C. Online Phase: Testing Framework

Finally, the online testing phase combines the offline trained models. As shown in Fig. 2, the framework receives a CSI measurement and converts it to ADP (4). The FC-AE then encodes the ADP. To test the validity of $\boldsymbol{A_e}$, the similarity in (15) compares $\boldsymbol{A}$ and $\boldsymbol{A_d}$. If the similarity exceeds a predefined threshold (th), the valid $\boldsymbol{A_e}$ is presented to the GPR models to estimate the user's coordinates $\hat{x}$ and $\hat{y}$ using (12). If the similarity falls below the threshold, $\boldsymbol{A_e}$ is discarded, and a new CSI measurement is procured.

Algorithm 1 Online Phase: User Localization Testing

```
Require: measured CSI at time $t$ ($\boldsymbol{H}$); similarity threshold $th$
Convert $\boldsymbol{H}$ to ADP $\boldsymbol{A}$ using (4)
Reshape ADP into vector $\boldsymbol{A}_v \in \mathbb{R}^{n^2 \times 1} \leftarrow \text{vec}(\boldsymbol{A} \in \mathbb{R}^{n \times n})$
$\boldsymbol{A_e} \in \mathbb{R}^{m^2 \times 1} \leftarrow \text{encoder}(\boldsymbol{A}_v \in \mathbb{R}^{n^2 \times 1})$ , $m < n$
$\boldsymbol{A_d} \in \mathbb{R}^{n^2 \times 1} \leftarrow \text{decoder}(\boldsymbol{A_e} \in \mathbb{R}^{m^2 \times 1})$
$s \leftarrow \mathscr{S}(\boldsymbol{A_v}, \boldsymbol{A_d})$
if $s > thresh$ then
    $\hat{x} \leftarrow \text{GPR}_\text{x}(\boldsymbol{A_e})$
    $\hat{y} \leftarrow \text{GPR}_\text{y}(\boldsymbol{A_e})$
else
    Encoded ADP ($\boldsymbol{A_e}$) is not valid
end if
```

TABLE II

CHANNEL MODEL PARAMETERS REQUIRED TO GENERATE THE DATASETS FROM THE DEEPMIMO DATASET GENERATION FRAMEWORK

Environment	$O1$	$I3$	$O2_{dyn}$
Frequency Band	3.5 GHz	2.4 GHz	3.5 GHz
Antenna Elements (N_t)	4,8,16,32,64	4,8,16	16
Subcarrier Number (N_c)	4,8,16,32,64	4,8,16	16

TABLE III

SCENARIO SPECIFIC PARAMETERS REQUIRED TO GENERATE THE DATASETS FROM THE DEEPMIMO DATASET GENERATION FRAMEWORK

Scenario	Rows	Area
$O1$-BS1 / BS4	300-700 / 800-1200	36 m × 80 m
$O1$-BS5 / B10	1300-1700 / 2000-2400	36 m × 80 m
$I3$-BS1 / BS2	1-551	1.2 m × 5.5 m
$O2_{dyn}$-BS1 / BS2	1-31	280 m × 6 m

V. SIMULATION

A. Simulation Dataset

The CSI-location pair dataset is generated using the DeepMIMO dataset framework [69], a publicly available tool that allows users to select environments and configure channel model and BS antenna parameters. DeepMIMO channels are based on accurate ray-tracing data. We consider three different environments in our simulations. The parameters for these environments are summarized in Table II. To be consistent with the single-site channel model, only one BS with a ULA antenna with N_t elements aligned along the y-axis with half-wavelength spacing is active. There is one user with an omni-directional antenna. The bandwidth is set to 100 MHz, and OFDM signaling with N_c subcarriers is employed.

The outdoor ($O1$) environment is an urban setting with two streets and one intersection. We generate datasets for 4 different location, each surrounding a different active BS. Each dataset contains 72400 samples and the separation between adjacent user locations is 20 cm. Note that only one BS is active at a given time. The indoor ($I3$) environment is a conference room with two BSs at different locations. Each indoor dataset contains 60500 samples and the separation between adjacent user locations is 4 cm. Finally, the dynamic outdoor ($O2_{dyn}$) environment contains many candidate users that are moving through an urban setting. These users change their positions with every scene. There are a total of 1000 captured scenes. Table III provides information on the area surrounding the active BS. The "Rows" column corresponds to parameters within the DeepMIMO framework, which must be configured to generate the datasets. Details on the DeepMIMO dataset generation may be found in [69].[1]

B. FC-AE Training

The number of neurons in each layer is shown in Table IV. For the 16×16 ADP input and larger sizes, the FC-AE has three encoder layers, while for the 8×8 ADP two layers suffice. The models are trained to encode the ADP to a 4×4 matrix. The table displays the neuron counts at each layer, followed by a Leaky ReLU activation function. Training uses an MSE loss function with an Adam optimizer at a learning rate of 0.0005. Training spans 1000 epochs with a batch size of 32.

TABLE IV

DETAILS OF THE FC-AE ARCHITECTURE SHOWING THE TOTAL NEURON COUNT FOR EACH LAYER

ADP Size	64×64	32×32	16×16	8×8
Encoded Layer 1	1024	256	144	36
Encoded Layer 2	256	64	64	16
Encoded Layer 3	16	16	16	-
Decoded Layer 1	256	64	64	36
Decoded Layer 2	1024	256	144	64
Decoded Layer 3	4096	1024	256	-

To create the training dataset for the $O1$ environment, we initially gather a small dataset of 362 ADP samples from various locations around active BS1. Subsequently, we employ data augmentation techniques to expand the ADP dataset size. For instance, the dataset expands to 1,049,070 samples for 16×16 ADP and to 294,030 samples for 8×8 ADP. Our augmentation methods include rotation ($-90°$, $+90°$, and $+180°$), horizontal and vertical flipping, and translation. This augmented dataset trains the $O1$ FC-AE model. Training is executed once for $O1$-BS1, and subsequent testing occurs at different scenarios within $O1$. The FC-AE's performance is assessed by the similarity between the input ADP sample and the corresponding decoded ADP output sample.

The FC-AE's performance results are presented in Table V, showcasing the mean and standard deviation of the average similarity between the original and decoded ADP. Notably, BS1 demonstrates the best performance with similarity of 0.91, as the training samples were collected in its vicinity. For the 16×16 ADP FC-AE, there is a 2-3% decrease in performance when applied to new, previously unobserved BS scenarios. The 8×8 ADP FC-AE maintains consistent performance of 0.91 across all BS scenarios. This shows how data augmentation allows training the FC-AE on limited number of samples from one area and then applying it to new scenarios. The higher average similarity for the 8×8 ADP dataset is expected due to its lower compression ratio. The average similarity for the 64×64 and 32×32 ADP FC-AE for $O1$-BS1 are 0.84 and 0.86, respectively. Although there is a small decline in the FC-AE performance with these larger ADP inputs, it does not affect the localization performance. In fact, as shown in the subsequent section, the FC-AE-GPR performs even better than the models with smaller ADP inputs.

A similar approach is applied to train the FC-AE for the indoor $I3$ environment. 31 samples collected around active BS1 are augmented to generate the training FC-AE dataset. We assess the FC-AE's performance within the $I3$ environment for both BS1 and BS2 scenarios. For the 16×16 ADP, the similarity scores are slightly lower than those observed in outdoor settings, registering at 0.84 for BS1 and 0.83 for BS2. In contrast, the 8×8 ADP consistently yields higher similarity values, maintaining 0.92 for BS1 and 0.91 for BS2, respectively. Fig. 3 shows an example of the original ADP

[1] https://www.deepmimo.net/

TABLE V

FC-AE Performance Results for Scenarios in the $O1$ and $I3$ Environments Defined by the Similarity. The Similarity Mean and Standard Deviation Are Presented for Each Active BS Dataset

Scenario	$\mathscr{S}$ mean±stdev Encoding 16x16 ADP	$\mathscr{S}$ mean±stdev Encoding 8x8 ADP
$O1$-BS1	0.91±0.06	0.91±0.05
$O1$-BS4	0.88±0.06	0.91±0.05
$O1$-BS5	0.89±0.05	0.91±0.05
$O1$-BS10	0.88±0.07	0.91±0.05
$I3$-BS1	0.84±0.05	0.92±0.04
$I3$-BS2	0.83±0.08	0.91±0.05

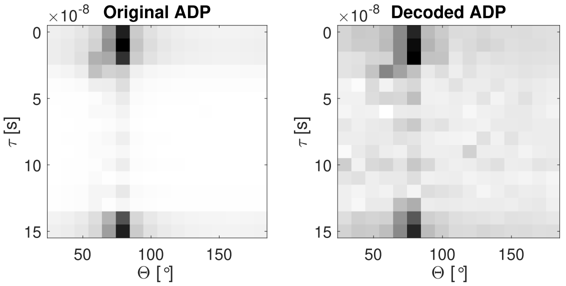

Fig. 3. Comparison between original input ADP and the decoded output ADP with a similarity of 0.95.

TABLE VI

Results for Minimum OF for Each Kernel Type

Kernel	Minimum OF	Kernel Scale l	σ
Squared Exponential	**3.9637**	$2.4 * 10^{-3}$	5.4
Exponential	4.0049	$1.7 * 10^{-3}$	$1.1 * 10^{-4}$
Rational Quadratic	4.0627	$2.5 * 10^{-3}$	7.6
Matern32	4.1482	$1.8 * 10^{-4}$	$1.3 * 10^{-3}$
Matern52	4.0166	$2.6 * 10^{-3}$	5.3

(left) and the decoded ADP (right) that have similarity of 0.95. The FC-AE is not completely lossless, which means that there will be noise in the decoded ADP. However, the defining features of the ADP which are the strong multiplath components are preserved. Therefore, the information about the channel contained in the ADP as well as the uniqueness of the ADP remain preserved.

C. GPR Training

The GPR model is implemented in MATLAB 2019b using the *fitgpr* function [70]. This function automatically tunes the parameters and selects the best kernel function. The model employs *bayesopt*, a Bayesian optimizer, and utilizes five-fold cross-validation for loss evaluation, aiming to minimize the Objective Function (OF) [71]

$$\mathrm{OF} = \log(1 + \text{cross-validation loss}). \quad (16)$$

Each kernel is optimized with respect to the kernel hyperparameters $\boldsymbol{\Theta}$ which include the kernel scale l and standard noise deviation σ. In this section, we demonstrate the GPR optimization for a 4×4 ADP dataset. While this section specifically addresses GPR optimization for 4×4 ADP dataset, the same optimization procedure is repeated for other ADP sizes and for encoded ADPs.

Fig. 4 illustrates the optimization for the squared exponential kernel. The OF on the z-axis is optimized with respect to l and σ. The model iterates over various l and σ combinations until the OF converges or the maximum 20 iterations is reached. Each iteration is represented by a blue dot (observed point), and the green dot represents the iteration with the smallest OF value found. The mean model surface is generated by interpolating between the blue dots. This process is repeated for the remaining four kernel types in Table I, creating an OF model for each. The minimum OF for each kernel type is summarized in Table VI, listing the kernel types, minimum estimated OF value, and associated hyperparameters. The results show the minimum OF occurs for the squared exponential kernel with $l = 2.4 * 10^{-3}$ and $\sigma = 5.4$.[2]

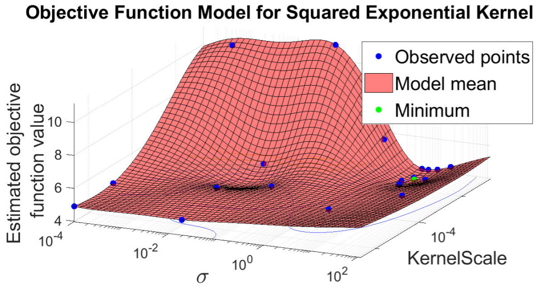

Fig. 4. The OF model for the squared exponential kernel. The OF on the z-axis is optimized with respect to σ and l. The OF is estimated at each l and σ combination depicted with a blue dot. The green dot represents the location of the minimum estimated OF.

TABLE VII

The First Two Values Represent the Size of the 2D Filter and the Third Value Denotes the Number of Filters in the CNN Model

Layer	kernel x filter	Layer	kernel x filter
1	$32 \times 32 \times 2$	4	$7 \times 7 \times 16$
2	$16 \times 16 \times 4$	5	$5 \times 5 \times 32$
3	$8 \times 8 \times 8$	6	$3 \times 3 \times 64$

D. CNN Training

The CNN is designed as an L layer regression network. The kernel size and number of filters for each layer are listed in Table VII. We developed five CNN models, each tailored to a different input size, ranging from 4×4 to 64×64 ADP input. The model for the smallest 4×4 ADP input comprises two layers, as indicated by layers 5 and 6. For each increase in input size, we add an additional layer to the model, resulting in a progressive enhancement of the model's depth and complexity. Furthermore, the CNN uses ReLU for the activation function and each convolutions layer is followed by a maxpool layer with kernel size 2×2. There is also a fully connected layer at the output. The CNN is trained over 300 epoch with a batch size of 32.

VI. Results and Discussion

A. Outdoor Scenario O1-BS1

We compare the proposed FC-AE-GPR to the CNN FP from [20], and to the GPR without FC-AE compression. For benchmarking purposes, we also compare our results against a standard non-parametric FP approach [72]. This approach

[2] Code released: https://github.com/katarinavuckovic/FC-AE-GPR-Localization-

TABLE VIII
LOCALIZATION MEAN RMSE ERROR IN [M] OVER 50 TRAINING/TESTING TRIALS FOR $O1$-BS1 SCENARIO. THE PROPOSED FC-AE-GPR MODEL IS COMPARED TO THE BENCHMARK MODELS. THE RESULTS ARE ASSESSED FOR DIFFERENT ADP RESOLUTIONS AND VARYING NUMBER OF TRAINING DATASET SAMPLES

Training samples		57920	36200	7240	3620	724	362	181	72
64×64 ADP	CNN	0.38	0.39	0.59	0.86	68.74	103.79	178.86	304.68
	Correlation-based			0.45	0.59	1.65	**2.11**	**3.08**	**4.87**
	FC-AE-GPR	-		**0.35**	**0.54**	**1.50**	2.26	3.16	4.96
32×32 ADP	CNN	0.66	0.91	1.29	2.01	2.82	16.14	87.25	196.39
	Correlation-based			0.48	0.62	1.51	**1.83**	**2.81**	**4.81**
	FC-AE-GPR			**0.33**	**0.51**	**1.44**	2.20	3.15	4.78
16×16 ADP	CNN	0.56	0.57	0.56	0.98	15.89	15.33	16.84	57.36
	Correlation-based			0.64	1.22	2.85	3.90	5.15	7.17
	GPR			0.60	0.98	2.56	3.91	5.43	7.52
	FC-AE-GPR			**0.48**	**0.74**	**2.03**	**2.99**	**4.16**	**6.14**
8×8 ADP	CNN	1.76	1.76	1.79	1.99	4.59	6.36	7.83	42.17
	Correlation-based			6.28	6.74	7.87	8.51	9.24	10.63
	GPR			**0.89**	**1.76**	**4.20**	5.77	7.16	10.56
	FC-AE-GPR			1.22	1.89	4.25	**5.57**	**7.00**	**9.05**
4×4 ADP	CNN	6.37	6.37	15.07	16.03	16.55	17.29	19.29	75.51
	Correlation-based	-	-	7.46	7.82	8.96	9.49	10.49	12.20
	GPR			**4.27**	**5.05**	**7.50**	**8.81**	**10.80**	**13.40**

treats a training fingerprint $\mathbf{A}(\mathbf{x_i})$ as a function of its position. The approach computes the position from a new fingerprint $\mathbf{A}_{\text{new}}$ through a grid-search over normalized correlations as

$$\hat{\mathbf{x}} = \underset{\mathbf{x}_i \in \{\mathbf{x}_1, \dots, \mathbf{x}_{N_{train}}\}}{\arg\max} \left| \frac{\text{Tr}\{\mathbf{A}(\mathbf{x}_i)^H \mathbf{A}_{\text{new}}\}}{\sqrt{\text{Tr}\{\mathbf{A}(\mathbf{x}_i)^H \mathbf{A}(\mathbf{x}_i)\} \text{Tr}\{\mathbf{A}_{\text{new}}^H \mathbf{A}_{\text{new}}\}}} \right|. \tag{17}$$

Table VIII presents the localization error in the outdoor scenario $O1$ near active BS1. The results highlight the impact of the number of training samples on the estimation error. The size of the training dataset ranges from 0.1% to 10% of the total dataset (equivalent to 72 to 7240 samples). Furthermore, since the training dataset is small, model performance strongly depends on the selected training samples. Therefore, we conducted 50 trials with different training samples, and averaged the results to report the mean RMSE.

1) Training Dataset Size: First, we compare the results with respect to the varying number of samples in the training dataset. The CNN can achieve high accuracy but it requires a much larger dataset to train. For instance, with 57920 training samples, the 64×64 ADP CNN achieves a performance of 0.38 m. However, the FC-AE-GPR still outperforms it achieving 0.35 m with only 7240 training samples. Furthermore, as the number of samples decreases, the performance of CNN quickly deteriorates as there is an insufficient number of training samples for the CNN model to learn the correlation.

The results demonstrate that the CNN trained with the small datasets completely fails to predict the user's location, while the performance of GPR and FC-AE-GPR gradually decreases as the training dataset is reduced. On the other hand, the non-parametric correlation-based approach is much more robust to the training dataset reduction. However, it is still outperformed by FC-AE-GPR in most cases. When the ADP becomes really large 32×32 or larger and the training dataset becomes 362 samples or less, the correlation based model does perform better than FC-AE-GPR. Furthermore, in a few cases, GPR may outperform FC-AE-GPR by a small margin. This is observed in 8×8 ADP model evaluation. However, the main reason for incorporating the FC-AE component is its substantial reduction in computational complexity. Therefore, we sacrifice some performance to gain on complexity reduction. Additionally, in these cases, the FC-AE-GPR model still outperforms both the correlation-based and CNN results.

Finally, increasing the training dataset to 50% and 80% does not yield a significant improvement in CNN performance. Furthermore, the performance of the CNN model for 16×16 and 8×8 ADP does not exhibit continued improvement as the dataset size increases. In the remaining cases, the improvements are observed as the dataset grows from 10% to 50%, but further dataset increase does not improve results.

2) ADP Resolution: With respect to the resolution of ADP, all models perform better when the ADP size is larger. The larger the ADP resolution, the more details it captures in the fingerprint which enables for a more accurate prediction. However, as the ADP resolution decreases, the CNN and the correlation based approach performance declines at a much higher rate compared to the FC-AE-GPR. The degraded performance is the most notable in the correlation based approach for 8×8 and 4×4 ADPs.

3) Complexity: CNN and FC-AE-GPR require offline training, but their estimation complexity is relatively low. In contrast, the correlation-based approach does not require training, but its complexity is high during the prediction phase. Therefore, the CNN and FC-AE-GPR trade offline training complexity for lower complexity at the prediction phase. When compared to using CNNs and FC-AE-GPR, the correlation-based grid-search approach has significant drawbacks in terms of prediction estimation complexity. Specifically, the computational complexity of the normalized correlation estimation is on the order of $O(nm_1^3)$, where n represents the number of training samples and m_1 is the length of the vectorized

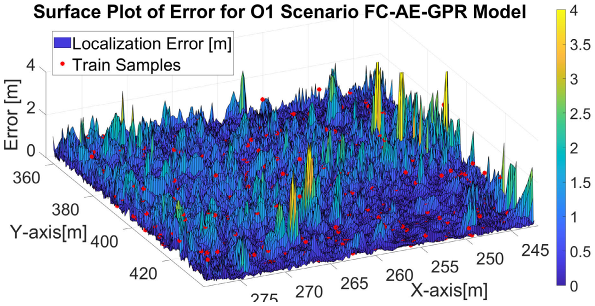

Fig. 5. Scenario O1 surface error plot for FC-AE-GPR model trained on 724 32 × 32 ADP samples.

ADP matrix. In practical terms, m ranges from 4 to 64, and n varies between 72 and 7240. Conversely, the estimation complexity of GPR is $O(nm_2 + n^2)$, but since FC-AE-GPR utilizes encoded ADPs, m_2 is consistently equal to 4. The cubic dependence on m_1 results in the grid search complexity becoming prohibitively large, often surpassing the GPR complexity, especially for larger ADPs. As a result, the prediction process may become overly delayed, compromising its timeliness and relevance. For the same reason, we chose not to train correlation-based model, GPR and FC-AE-GPR on larger datasets, as our primary goal was to demonstrate performance with a small-scale training dataset. In conclusion, FC-AE-GPR emerges as the optimal model, demonstrating robust performance across all ADP sizes, while other models face challenges with smaller ADPs or a limited number of training samples.

4) Error Distribution: While the RMSE provides an overall accuracy measure, it does not describe the error variation. The surface plot in Fig. 5 maps the error distribution, showing regions with different localization accuracies and revealing the model's performance beyond aggregate RMSE. This plot visualizes prediction errors across the environment for the 32× 32 ADP O1 scenario using the FC-AE-GPR model. The z-axis shows error magnitude in meters, while the x and y axes represent spatial dimensions. Red markers indicate training sample positions, demonstrating the relationship between training sample density and accuracy. Regions with denser/sparser training samples have lower/higher errors, as expected in a GPR model, visually and quantitatively captured in the plot.

B. Outdoor Scenarios in O1 With New Base Stations

Next, we assess FC-AE-GPR's performance in areas around other BSs in the $O1$ environment. We use the FC-AE trained on samples from active BS1 and apply it to samples from new BS4, BS5, and BS10 areas. In Section V-B, we discussed the FC-AE's performance in new BS scenarios and showed that the similarity holds steady for different BS datasets in $O1$. This section focuses on the localization accuracy of FC-AE-GPR. While the FC-AE remains unchanged for new BS scenarios, the GPR model is retrained for each new BS to map ADPs to geo-locations. Table IX shows FC-AE-GPR's performance on 16×16 and 8×8 ADP inputs, comparing BS1 from the previous simulation to the new BS datasets. The models show similar performance across different BSs, indicating that FC-AE efficiently encodes ADP samples. Results among the different BSs vary by no more than 10%.

TABLE IX

LOCALIZATION ERROR OF THE FC-AE-GPR MODEL PRESENTED AS MEAN RMSE IN [M] OVER 50 TRAINING/TESTING TRIALS IN SCENARIO $O1$.THE RESULTS DEMONSTRATE PERFORMANCE IN NEW BSs (BS4, BS5, BS10) PREVIOUSLY UNSEEN IN THE TRAINING DATASET BY THE FC-AE MODEL

Training samples		7240	3620	724	362	181	72
16×16 ADP	BS1	0.57	0.96	2.99	4.46	6.26	8.72
	BS4	0.59	0.97	2.86	4.36	6.12	8.78
	BS5	0.60	0.98	3.03	4.50	6.35	8.82
	BS10	0.59	0.97	0.89	4.35	6.01	8.84
8×8 ADP	BS1	1.22	1.89	4.25	5.57	7.00	9.05
	BS4	1.29	1.98	4.28	5.57	6.98	8.94
	BS5	1.25	1.97	4.30	5.56	7.11	9.01
	BS10	1.35	2.05	4.52	5.85	7.21	9.19

TABLE X

LOCALIZATION ERROR REPORTED AS MEAN RMSE IN [M] OVER 50 TRAINING/TESTING TRIALS FOR $O1$-BS1 SCENARIO WITH DIFFERENT URA ANTENNA CONFIGURATIONS. THE FIRST TWO VALUES IN THE ADP DIMENSION REPRESENT THE NUMBER OF ANTENNAS ALONG THE X-AXIS AND Y-AXIS, RESPECTIVELY, AND THE THIRD DIMENSION IS Nc

Training samples		7240	3620	724	362	181	72
1×8×8 ADP	CNN	1.79	1.99	4.59	6.36	7.83	42.17
	FC-AE-GPR	**1.22**	**1.89**	**4.25**	**5.57**	**7.00**	**9.05**
2×4×8 ADP	CNN	1.52	1.99	4.01	7.85	61.9	121.3
	FC-AE-GPR	**1.25**	**1.88**	**3.35**	**5.75**	**7.33**	**9.61**
4×2×8 ADP	CNN	1.49	2.39	3.89	9.81	83.4	189.5
	FC-AE-GPR	**1.17**	**1.67**	**3.45**	**5.01**	**6.09**	**8.39**

C. Extension to URA Antenna Configurations

In this section, we explore the impact of different antenna configuration on the localization performance and demonstrate that the FC-AE-GPR model is also applicable to Uniform Rectangular Array (URA) antenna configurations. Specifically, we explore URA configurations aligned along the x and y axis. We consider two case $Nx_t = 2, Ny_t = 4$ and $Nx_t = 4, Ny_t = 2$, maintaining a total number of 8 antenna elements. We compare this to the performance of the $Nx_t = 1, Ny_t = 8$ configuration presented in the previous section. We generate the dataset around BS1 in the $O1$ environment and keep all the other channel parameters the same. We train two FC-AE models to encode a 2×4×8 and 4×2×8 ADP to a reduced 4×4 ADP. The similarity performance of the FC-AE models is 0.90 and 0.93, respectively. These similarities are comparable to the $Nx_t = 1, Ny_t = 8$ ADP FC-AE performance of 0.91 presented in Table V.

Table X presents a performance comparison between FC-AE-GPR and CNN for the different antenna configurations. Before model training, we project the 3D ADP into a 2D version. Notably, we observe similar performance across all three configurations. In all cases, the FC-AE-GPR outperforms the CNN. Additionally, we notice that the critical point for CNN in URA configurations occurs at 181 training samples whereas for ULA antennas, this spike occurs at 72 samples.

TABLE XI

LOCALIZATION ERROR FOR INDOOR SCENARIO $I3$-BS1 AND $I3$-BS2 PRESENTED AS MEAN RMSE IN [M] OVER 50 TRAINING/TESTING TRIALS

		Training samples	6050	3025	605	303	151	61
BS1	16×16 ADP	CNN	**0.05**	0.15	0.44	0.69	0.79	3.59
		FC-AE-GPR	**0.05**	**0.07**	**0.08**	**0.11**	**0.14**	**0.22**
	8×8 ADP	CNN	**0.04**	0.09	0.16	0.18	2.98	10.8
		FC-AE-GPR	**0.04**	**0.08**	**0.11**	**0.14**	**0.14**	**0.22**
	4×4 ADP	CNN	0.16	0.17	0.22	0.70	2.74	6.03
		GPR	**0.11**	**0.11**	**0.16**	**0.17**	**0.19**	**0.25**
BS2	16×16 ADP	CNN	0.05	0.13	0.42	0.77	0.85	6.50
		FC-AE-GPR	**0.04**	**0.05**	**0.07**	**0.09**	**0.14**	**0.21**
	8×8 ADP	CNN	0.05	0.12	0.16	0.21	4.60	17.21
		FC-AE-GPR	**0.04**	**0.04**	**0.08**	**0.11**	**0.14**	**0.19**
	4×4 ADP	CNN	0.16	0.18	0.23	0.77	2.50	4.50
		GPR	**0.10**	**0.10**	**0.15**	**0.18**	**0.21**	**0.25**

The spike in the error suggests that the training dataset size is insufficient to train the CNN model.

D. Indoor I3 Scenarios

We evaluate the FC-AE-GPR's performance against the CNN in the indoor environment labeled $I3$. The simulation results for BS1 and BS2 are shown in Table XI. The FC-AE is trained with samples from BS1 and tested on both BS1 and BS2. The training dataset size ranges from 61 to 6050 samples, representing 0.1% to 10% of the total BS1 dataset. With a 6050-sample training dataset, FC-AE-GPR matches or outperforms CNN for larger datasets. As the training dataset size decreases, FC-AE-GPR consistently surpasses CNN. With just 61 training samples, FC-AE-GPR achieves sub-0.25 m accuracy in all cases, while CNN's best performance is 3.59 m for 16×16 ADP at BS1. This highlights FC-AE-GPR's superiority with small labeled training datasets.

E. Similarity Threshold Analysis

Thus far, we compared our models' performance across the entire dataset to maintain consistency with other benchmarks. However, in the proposed algorithm in Fig. 2, we introduced a similarity threshold criterion to determine the reliability of encoded ADP samples. If the similarity is below the threshold, the ADP measurement is deemed unreliable, and the system should acquire another measurement for accurate location prediction. Other benchmark models lack this validity check and must trust all ADP measurements.

Table XII illustrates the impact of varying thresholds on the results for a 16 × 16 ADP model, including the number of samples that are discarded for each threshold. An optimal threshold minimizes the amount of discarded data while significantly improving localization error. The threshold may also depend on the accuracy requirement and the CSI estimation rate. If CSI are obtained at higher rates, we may be able to set the discard threshold higher. Our findings show that no samples have a similarity below 0.4. Adjusting the threshold to 0.5 enhances the results by 2.06% while discarding only 0.09% of the samples. While the discarded rate is very low, it does not substantially enhance performance. On the other hand, a threshold of 0.9 improves performance by 33.15% but discards 46.85% of the samples. Although the performance gain is significant, discarding nearly half of the samples may be impractical, as it can cause the location estimation to update at a very slow rate, rendering the estimation no longer relevant. Consequently, a threshold of 0.8 seems optimal, offering a 9.31% improvement in performance while discarding only 6.98% of the dataset.

TABLE XII

PERFORMANCE OF FC-AE-GPR MODEL FOR DIFFERENT THRESHOLD VALUES FOR THE 16 × 16 ADP WITH 7240 TRAINING SAMPLES

Threshold	**% mean RMSE [m]**	**% RMSE improvement**	**% of discarded samples**
0.4	0.480	0.00%	0.00%
0.5	0.470	2.06%	0.09%
0.6	0.464	3.33%	0.37%
0.7	0.453	5.73%	1.40%
0.8	0.435	9.31%	6.98%
0.9	0.321	33.15%	46.85%

F. Offline Training Time

One of the main concerns of GPR model design is the computational time involved in training. The computational complexity requirement for GPR optimization is $O(n^3 d)$, where n is the number of training samples and d is the length of the input vector $\mathbf{x}$ (vectorized ADP). The cubic complexity comes from the inversion and determinant of the kernel matrix $\mathbf{K}$ in (14). Furthermore, the computation of kernel functions $k_{ij}(\mathbf{x}_j, \mathbf{x}_j)$ that form the kernel matrix $\mathbf{K}$ depends on the size of $\mathbf{x}$ and has a complexity of $O(d^2)$. The squared complexity comes from the vector multiplication in the kernel.

The graph in Fig. 6 shows how the GPR optimization time relates to the number of training samples (n) and the dimension of the input (d). As the number of training samples increases ten times from 72 to 724 samples, the computation increases by approximately two orders of magnitude regardless of the ADP input size. In addition, as the size of the ADP increase from 4^2 to 8^2 and then to 16^2, the optimization time increases by one and two orders of magnitude, respectively. To mitigate this, we employ the FC-AE to compress the ADPs, thus ensuring that the GPR model receives consistently small-dimensional inputs. By reducing the original ADP to an encoded ADP with a fixed dimension of 4×4, we maintain constant training complexity for the GPR model, regardless of increases in the original ADP size. This encoding significantly reduces computational demands as the size of the original ADP grows. While there is an increase in the complexity of the FC-AE at higher compression ratios, the FC-AE is designed to be scenario-independent and requires only a single training session. Once trained, it can be reused in different scenarios. Since the FC-AE does not necessitate retraining, its complexity is not a major concern.

G. Dynamic Scenarios and Future Work

The natural extension to the proposed FC-AE-GPR model is its applications in dynamic scenarios. In dynamic environments, wireless channel characteristics change often due to

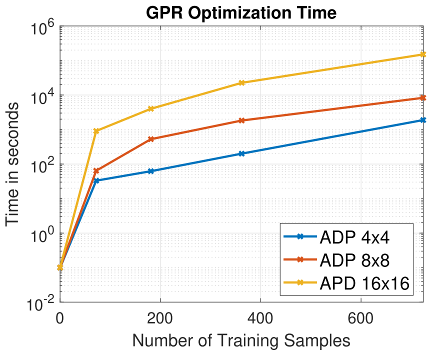

Fig. 6. GPR hyper-parameter training time for different ADP dimensions and different number of training samples.

TABLE XIII
THE FC-AE PERFORMANCE RESULTS FOR DYNAMIC OUTDOOR $O2_{dyn}$ ENVIRONMENT DEFINED BY THE SIMILARITY. THE SIMILARITY MEAN AND STANDARD DEVIATION ARE PRESENTED FOR EACH ACTIVE BS-SCENE DATASET

Scenario	Scene	$\mathscr{S}$ mean±stdev Encoding 16x16 ADP
$O2_{dyn}$-BS1 / BS2	1	0.83±0.09 / 0.81±0.11
$O2_{dyn}$-BS1 / BS2	15	0.83±0.10 / 0.81±0.10
$O2_{dyn}$-BS1 / BS2	505	0.85±0.10 / 0.85±0.10

moving objects, people, or obstacles. This variation affects path loss, shadowing, and multipath effects. Dynamic environments are typical in scenarios like vehicular communications, where vehicles change positions relative to each other and to wireless transmitters and receivers. Furthermore, the Doppler effect, which arises due to the relative motion between the transmitter and receiver, could also influence the signal characteristics in such dynamic settings. While not explicitly modeled in this study, its impact will be considered in future work. In this section, we demonstrate that the FC-AE model performs well in dynamic $O2_{dyn}$ environment. We test the FC-AE model trained on static $O1$-BS1 dataset on the $O2_{dyn}$ scenarios. We consider three different scenes on two different BSs in the $O2_{dyn}$ environment. The performance of the FC-AE in the new $O2_{dyn}$ environment, with no retraining, is showcased in Table XIII. It exhibits a mean similarity ranging from 0.83 to 0.85, with an average standard deviation of 0.1. The FC-AE model's similarity in the $O2_{dyn}$ environment falls within the same range as the performance in $I3$ (as shown in Table V), which we have demonstrated via the $I3$ simulations is sufficient to effectively encode the ADPs.

However, FP-based models like GPR and CNN are primarily designed for static environments, as we demonstrated in our previous work [19]. Nevertheless, our previous study also revealed that time-series analysis of ADP samples effectively addresses dynamic changes [19]. In future work, we will explore extending the FC-AE-GPR model to incorporate time-series analysis for detecting and correcting for dynamic changes that occur in the environment, including the Doppler effect.

VII. CONCLUSION

This paper introduces the FC-AE-GPR framework, a new approach for fingerprint-based localization in single-site massive MIMO systems. This innovative framework integrated a Fully-Connected Auto-Encoder (FC-AE) with Gaussian Process Regression (GPR) models, tailored for use with small-scale labeled training datasets. The GPR model enables training on limited data, while the FC-AE efficiently compresses the Angle-Delay Profile input, easing the computational load prior to training the GPR model. Data scarcity for FC-AE training is addressed through data augmentation techniques. Simulation results reveal that both GPR and FC-AE-GPR surpass the performance of the state-of-the-art Convolutional Neural Network regression model and the normalized-correlation model in both outdoor and indoor environments when trained on small datasets. Additionally, the FC-AE significantly enhances the GPR model by reducing computational time, thereby remedying one of GPR's primary limitations.

REFERENCES

[1] M. A. Cheema, "Indoor location-based services: Challenges and opportunities," *SIGSPATIAL Special*, vol. 10, no. 2, pp. 10–17, Nov. 2018, doi: 10.1145/3292390.3292394.

[2] J. A. del 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.

[3] X. Lin et al., "5G new radio: Unveiling the essentials of the next generation wireless access technology," *IEEE Commun. Standards Mag.*, vol. 3, no. 3, pp. 30–37, Sep. 2019.

[4] J. Guo, C.-K. Wen, S. Jin, and G. Y. Li, "Overview of deep learning-based CSI feedback in massive MIMO systems," *IEEE Trans. Commun.*, vol. 70, no. 12, pp. 8017–8045, Dec. 2022.

[5] D. Love, R. Heath, V. N. Lau, D. Gesbert, B. Rao, and M. Andrews, "An overview of limited feedback in wireless communication systems," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 8, pp. 1341–1365, Oct. 2008.

[6] Z. Qin, J. Fan, Y. Liu, Y. Gao, and G. Y. Li, "Sparse representation for wireless communications: A compressive sensing approach," *IEEE Signal Process. Mag.*, vol. 35, no. 3, pp. 40–58, May 2018.

[7] K. Vuckovic, F. Hejazi, and N. Rahnavard, "MAP-CSI: Single-site map-assisted localization using massive MIMO CSI," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2021, pp. 1–6.

[8] O. Kanhere, S. Ju, Y. Xing, and T. S. Rappaport, "Map-assisted millimeter wave localization for accurate position location," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2019, pp. 1–6.

[9] M. Comiter and H. T. Kung, "Localization convolutional neural networks using angle of arrival images," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–7.

[10] X. Sun, X. Gao, G. Y. Li, and W. Han, "Single-site localization based on a new type of fingerprint for massive MIMO-OFDM systems," *IEEE Trans. Veh. Technol.*, vol. 67, no. 7, pp. 6134–6145, Jul. 2018.

[11] A. Puckdeevongs, "Indoor localization using RSSI and artificial neural network," in *Proc. 9th Int. Electr. Eng. Congr. (iEECON)*, Mar. 2021, pp. 479–482.

[12] R. Niu, A. Vempaty, and P. K. Varshney, "Received-signal-strength-based localization in wireless sensor networks," *Proc. IEEE*, vol. 106, no. 7, pp. 1166–1182, Jul. 2018.

[13] S. Kumar, R. M. Hegde, and N. Trigoni, "Gaussian process regression for fingerprinting based localization," *Ad Hoc Netw.*, vol. 51, pp. 1–10, Nov. 2016.

[14] F. Lezama, G. G. González, F. Larroca, and G. Capdehourat, "Indoor localization using graph neural networks," in *Proc. IEEE URUCON*, Nov. 2021, pp. 51–54.

[15] G. Zhang, P. Wang, H. Chen, and L. Zhang, "Wireless indoor localization using convolutional neural network and Gaussian process regression," *Sensors*, vol. 19, no. 11, p. 2508, May 2019.

[16] Y. Xie, C. Zhu, W. Jiang, J. Bi, and Z. Zhu, "Analyzing machine learning models with Gaussian process for the indoor positioning system," *Math. Problems Eng.*, vol. 2020, no. 1, 2020, Art. no. 4696198.

[17] V. Savic and E. G. Larsson, "Fingerprinting-based positioning in distributed massive MIMO systems," in *Proc. IEEE 82nd Veh. Technol. Conf. (VTC-Fall)*, Sep. 2015, pp. 1–5.

[18] S. Samira Moosavi and P. Fortier, "Fingerprinting localization method based on clustering and Gaussian process regression in distributed massive MIMO systems," in *Proc. IEEE 31st Annu. Int. Symp. Pers., Indoor Mobile Radio Commun.*, Aug./Sep. 2020, pp. 1–7.

[19] F. Hejazi, K. Vuckovic, and N. Rahnavard, "DyLoc: Dynamic localization for massive MIMO using predictive recurrent neural networks," in *Proc. IEEE Conf. Comput. Commun.*, May 2021, pp. 1–9.

[20] 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.

[21] 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.

[22] M. M. Butt, A. Rao, and D. Yoon, "RF fingerprinting and deep learning assisted UE positioning in 5G," in *Proc. IEEE 91st Veh. Technol. Conf.*, May 2020, pp. 1–7.

[23] 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.

[24] 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.

[25] C. T. Nguyen et al., "Transfer learning for wireless networks: A comprehensive survey," *Proc. IEEE*, vol. 110, no. 8, pp. 1073–1115, Aug. 2022.

[26] 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.

[27] M. Stahlke, T. Feigl, M. H. C. Garcia, R. A. Stirling-Gallacher, J. Seitz, and C. Mutschler, "Transfer learning to adapt 5G AI-based fingerprint localization across environments," in *Proc. IEEE 95th Veh. Technol. Conf. (VTC-Spring)*, Jun. 2022, pp. 1–5.

[28] Z. Guo, K. Lin, X. Chen, and C.-Y. Chit, "Transfer learning for angle of arrivals estimation in massive MIMO system," in *Proc. IEEE/CIC Int. Conf. Commun. China (ICCC)*, Aug. 2022, pp. 506–511.

[29] C. Studer, S. Medjkouh, E. Gonultas, T. Goldstein, and O. Tirkkonen, "Channel charting: Locating users within the radio environment using channel state information," *IEEE Access*, vol. 6, pp. 47682–47698, 2018.

[30] M. Stahlke, G. Yammine, T. Feigl, B. M. Eskofier, and C. Mutschler, "Indoor localization with robust global channel charting: A time-distance-based approach," *IEEE Trans. Mach. Learn. Commun. Netw.*, vol. 1, pp. 3–17, 2023.

[31] P. Agostini, Z. Utkovski, S. Stańczak, A. A. Memon, B. Zafar, and M. Haardt, "Not-too-deep channel charting (N2D-CC)," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2022, pp. 2160–2165.

[32] H. Al-Tous, P. Kazemi, C. Studer, and O. Tirkkonen, "Channel charting with angle-delay-power-Profile features and Earth-mover distance," in *Proc. 56th Asilomar Conf. Signals, Syst., Comput.*, Oct. 2022, pp. 1195–1201.

[33] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006.

[34] B. Rappaport, E. Gönültaş, J. Hoydis, M. Arnold, P. K. Srinath, and C. Studer, "Improving channel charting using a split triplet loss and an inertial regularizer," in *Proc. 17th Int. Symp. Wireless Commun. Syst. (ISWCS)*, Sep. 2021, pp. 1–6.

[35] J. M. Rocamora, I. Wang-Hei Ho, W.-M. Mak, and A. P.-T. Lau, "Survey of CSI fingerprinting-based indoor positioning and mobility tracking systems," *IET Signal Process.*, vol. 14, no. 7, pp. 407–419.

[36] E. Lei, O. Castañeda, O. Tirkkonen, T. Goldstein, and C. Studer, "Siamese neural networks for wireless positioning and channel charting," in *Proc. 57th Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Sep. 2019, pp. 200–207.

[37] P. Huang et al., "Improving channel charting with representation-constrained autoencoders," in *Proc. IEEE 20th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Jul. 2019, pp. 1–5.

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

[39] P. Kazemi, H. Al-Tous, T. Ponnada, C. Studer, and O. Tirkkonen, "Beam SNR prediction using channel charting," *IEEE Trans. Veh. Technol.*, vol. 72, no. 10, pp. 13130–13145, Oct. 2023.

[40] Q. Zhang and W. Saad, "Semi-supervised learning for channel charting-aided IoT localization in millimeter wave networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2021, pp. 1–6.

[41] A. Bekkali, T. Masuo, T. Tominaga, N. Nakamoto, and H. Ban, "Gaussian processes for learning-based indoor localization," in *Proc. IEEE Int. Conf. Signal Process., Commun. Comput. (ICSPCC)*, Sep. 2011, pp. 1–6.

[42] S. Yiu and K. Yang, "Gaussian process assisted fingerprinting localization," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 683–690, Oct. 2016.

[43] K. Liu, Z. Meng, and C.-M. Own, "Gaussian process regression plus method for localization reliability improvement," *Sensors*, vol. 16, no. 8, p. 1193, Jul. 2016.

[44] Y. Zhao, C. Liu, L. S. Mihaylova, and F. Gunnarsson, "Gaussian processes for RSS fingerprints construction in indoor localization," in *Proc. 21st Int. Conf. Inf. Fusion (FUSION)*, Jul. 2018, pp. 1377–1384.

[45] X. Wang, X. Wang, S. Mao, J. Zhang, S. C. G. Periaswamy, and J. Patton, "Indoor radio map construction and localization with deep Gaussian processes," *IEEE Internet Things J.*, vol. 7, no. 11, pp. 11238–11249, Nov. 2020.

[46] A. Damianou and N. D. Lawrence, "Deep Gaussian processes," in *Proc. 16th Int. Conf. Artif. Intell. Statist.*, 2013, pp. 207–215.

[47] K. Vafa, "Training and inference for deep Gaussian processes," Ph.D. dissertation, Harvard Univ., Cambridge, MA, USA, 2016.

[48] X. Wang, M. Patil, C. Yang, S. Mao, and P. A. Patel, "Deep convolutional Gaussian processes for mmWave outdoor localization," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 8323–8327.

[49] K. Blomqvist, S. Kaski, and M. Heinonen, "Deep convolutional Gaussian processes," in *Proc. Eur. Conf. Mach. Learn. Knowl. Discovery Databases (ECML PKDD)*. Würzburg, Germany: Springer, 2020, pp. 582–597.

[50] H. Liu, Y.-S. Ong, X. Shen, and J. Cai, "When Gaussian process meets big data: A review of scalable GPs," 2018, *arXiv:1807.01065*.

[51] C.-K. Wen, W.-T. Shih, and S. Jin, "Deep learning for massive MIMO CSI feedback," *IEEE Wireless Commun. Lett.*, vol. 7, no. 5, pp. 748–751, Oct. 2018.

[52] S. Jo, J. Lee, and J. So, "Deep learning-based massive multiple-input multiple-output channel state information feedback with data normalisation using clipping," *Electron. Lett.*, vol. 57, no. 3, pp. 151–154, Feb. 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:234126178

[53] S. Ji and M. Li, "CLNet: Complex input lightweight neural network designed for massive MIMO CSI feedback," *IEEE Wireless Commun. Lett.*, vol. 10, no. 10, pp. 2318–2322, Oct. 2021.

[54] Y. Sun, W. Xu, L. Liang, N. Wang, G. Y. Li, and X. You, "A lightweight deep network for efficient CSI feedback in massive MIMO systems," *IEEE Wireless Commun. Lett.*, vol. 10, no. 8, pp. 1840–1844, Aug. 2021.

[55] M. Gao, T. Liao, and Y. Lu, "Fully connected feedforward neural networks based CSI feedback algorithm," *China Commun.*, vol. 18, no. 1, pp. 43–48, Jan. 2021.

[56] X. Chen, C. Deng, B. Zhou, H. Zhang, G. Yang, and S. Ma, "High-accuracy CSI feedback with super-resolution network for massive MIMO systems," *IEEE Wireless Commun. Lett.*, vol. 11, no. 1, pp. 141–145, Jan. 2022.

[57] J. Guo, C.-K. Wen, S. Jin, and G. Y. Li, "Convolutional neural network-based multiple-rate compressive sensing for massive MIMO CSI feedback: Design, simulation, and analysis," *IEEE Trans. Wireless Commun.*, vol. 19, no. 4, pp. 2827–2840, Apr. 2020.

[58] C. Lu, W. Xu, S. Jin, and K. Wang, "Bit-level optimized neural network for multi-antenna channel quantization," *IEEE Wireless Commun. Lett.*, vol. 9, no. 1, pp. 87–90, Jan. 2020.

[59] Z. Lu, J. Wang, and J. Song, "Multi-resolution CSI feedback with deep learning in massive MIMO system," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2020, pp. 1–6.

[60] K. Vuckovic and N. Rahanvard, "Localization techniques in multiple-input multiple-output communication: Fundamental principles, challenges, and opportunities," in *MIMO Communications—Fundamental Theory, Propagation Channels, and Antenna Systems*, D. A. Kishk and D. X. Chen, Eds. Rijeka, Croatia: InTechOpen, 2023, ch. 15.

[61] L. Chu, A. Alghafis, and A. F. Molisch, "SA-loc: Scenario adaptive localization in highly dynamic environment using adversarial regressive domain adaptation," in *Proc. IEEE Int. Conf. RFID (RFID)*, May 2022, pp. 132–137.

[62] A. Ali, N. González-Prelcic, and R. W. Heath, "Millimeter wave beam-selection using out-of-band spatial information," *IEEE Trans. Wireless Commun.*, vol. 17, no. 2, pp. 1038–1052, Feb. 2018.

[63] L. Liu et al., "The COST 2100 MIMO channel model," *IEEE Wireless Commun.*, vol. 19, no. 6, pp. 92–99, Dec. 2012.

[64] A. Alkhateeb and R. W. Heath, "Frequency selective hybrid precoding for limited feedback millimeter wave systems," *IEEE Trans. Commun.*, vol. 64, no. 5, pp. 1801–1818, May 2016.

[65] J. Wang, "An intuitive tutorial to Gaussian processes regression," *Comput. Sci. Eng.*, 2023.

[66] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA, USA: MIT Press, 2005.

[67] J. A. Rice, *Mathematical Statistics and Data Analysis*. Boston, MA, USA: Cengage Learning, 2006.

[68] A. Geron, *Hands-on Machine Learning With Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. Sebastopol, CA, USA: O'Reilly Media, 2019.

[69] A. Alkhateeb, "DeepMIMO: A generic deep learning dataset for millimeter wave and massive MIMO applications," in *Proc. Inf. Theory Appl. Workshop (ITA)*, San Diego, CA, USA, Feb. 2019, pp. 1–8.

[70] *Fit a Gaussian Process Regression (GPR) Model—MATLAB*. [Online]. Available: https://www.mathworks.com/help/stats/fitrgp.html

[71] B. A. Khuwaileh and W. A. Metwally, "Gaussian process approach for dose mapping in radiation fields," *Nucl. Eng. Technol.*, vol. 52, no. 8, pp. 1807–1816, Aug. 2020.

[72] Z.-H. Wu, Y. Han, Y. Chen, and K. J. R. Liu, "A time-reversal paradigm for indoor positioning system," *IEEE Trans. Veh. Technol.*, vol. 64, no. 4, pp. 1331–1339, Apr. 2015.

Katarina Vuckovic received the B.S. degree in aerospace engineering and the B.S. and M.S. degrees in electrical engineering from Florida Institute of Technology, Melbourne, FL, USA, in 2017, 2017, and 2019, respectively. She is currently pursuing the Ph.D. degree in electrical engineering with the University of Central Florida. She maintains her steadfast position as an Engineer at Collins Aerospace. Over the past seven years at Collins Aerospace, she has worked on a wide range of applications including wireless communication systems, automation, and data analytic.

Saba Hosseini received the B.Sc. and M.Sc. degrees in electrical engineering from distinguished universities. She is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Central Florida. Her research interests include machine learning and its applications in wireless communications systems.

Farzam Hejazi received the B.S., M.S., and Ph.D. degrees in electrical engineering from the Sharif University of Technology, Tehran, Iran, in 2011, 2013, and 2018, respectively. From 2019 to 2021, he was a Post-Doctoral Researcher with the University of Central Florida, Orlando, FL, USA. He is currently a Senior Machine Learning Engineer at Qualcomm. His research interests include machine learning for wireless communication, initial access in 6G/5G THz and mmWave systems, and radio frequency localization.

Nazanin Rahnavard (Senior Member, IEEE) received the Ph.D. degree from the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA, in 2007. She is currently a Professor with the Department of Electrical and Computer Engineering, University of Central Florida, Orlando, FL, USA. Her research interests include deep learning, communications, networking, and signal processing areas. She was a recipient of the NSF CAREER Award in 2011, the UCF Faculty Excellence in Mentoring Doctoral Students Award in 2022, and the UCF Faculty Excellence in Mentoring Postdoctoral Scholars Award in 2023. She serves on the editorial board of the *Journal of Computer Networks* (Elsevier) and the Technical Program Committee of several prestigious international conferences.