CHALMERS
UNIVERSITY OF TECHNOLOGY

Positioning via Digital-Twin-Aided Channel Charting with Large-Scale CSI Features

Citation for the original published paper (version of record):
Mateos Ramos, J., Zumegen, F., Wymeersch, H. et al (2026). Positioning via Digital-Twin-Aided Channel Charting with Large-Scale CSI Features. IEEE Transactions on Wireless Communications, 25: 22245-22260. http://dx.doi.org/10.1109/TWC.2026.3724634

(article starts on next page)

Positioning via Digital-Twin-Aided Channel Charting with Large-Scale CSI Features

José Miguel Mateos-Ramos, *Graduate Student Member, IEEE*, Frederik Zumegen, *Graduate Student Member, IEEE*, Henk Wymeersch, *Fellow, IEEE*, Christian Häger, *Member, IEEE*, Christoph Studer, *Senior Member, IEEE*,

***Abstract*—Channel charting (CC) is a self-supervised positioning technique whose main limitation is that the estimated positions lie in an arbitrary coordinate system that is not aligned with true spatial coordinates. In this work, we propose a novel method to produce CC locations in true spatial coordinates with the aid of a digital twin (DT). Our main contribution is a new framework that (i) extracts large-scale channel-state information (CSI) features from estimated CSI and the DT and (ii) matches these features with a cosine-similarity loss function. The DT-aided loss function is then combined with a conventional CC loss to learn a positioning function that provides true spatial coordinates without relying on labeled data. Our results for a simulated indoor scenario demonstrate that the proposed framework reduces the relative mean distance error by 29% compared to the state of the art. We also show that the proposed approach is robust to DT modeling mismatches and a distribution shift in the testing data.**

***Index Terms*—Channel charting, digital twin, machine learning, positioning, self-supervised learning.**

I. Introduction

Endowing communication networks with positioning capabilities is a key enabler in next-generation wireless systems for applications such as vehicle-to-everything communications, human activity sensing, or smart factories, among others [1], [2]. Conventional positioning methods based on time-of-arrival (ToA), angle-of-arrival (AoA), or received signal strength typically assume line-of-sight (LoS) propagation between transmitter and receiver [3], [4]. Under non-line-of-sight (NLoS) conditions, the performance of such methods significantly degrades. Alternative solutions that rely on large corpora of ground-truth labeled user equipment (UE) positions are robust to NLoS scenarios via supervised learning of neural networks (NNs) [5], [6]. The number of labeled positions can be reduced through self-supervised learning, where pseudo-labeled data is first generated to pre-train the NN, followed by a fine-tuning phase with few labeled positions [7], [8]. Similarly, semi-supervised learning integrates supervised objectives with additional loss terms that do not require labeled data [9]. Alternatively, labeled data demands can be reduced via crowdsourcing, where users report position updates using global navigation satellite systems or inertial measurement unit measurements to augment the corpus of labeled samples [10]–[13]. Nevertheless, ground-truth labeled data acquisition may imply a costly measurement campaign, even for a small dataset. Additionally, crowdsourcing positioning relies on the external sensors that often provide inaccurate positioning measurements that can propagate over time [14].

To overcome these limitations, channel charting (CC) formulates positioning as a self-supervised dimensionality-reduction problem, mapping high-dimensional channel-state information (CSI) to lower-dimensional position estimates [15], [16]. The key principle of CC is to preserve local geometry, ensuring that nearby estimated pseudo-positions correspond to nearby true positions. However, this approach yields estimated pseudo-positions in an arbitrary coordinate system, i.e., global geometry may not be preserved. Hence, there is a need to develop CC-based localization methods that provide coordinates in a global reference system, i.e., in true spatial coordinates.

As one possible approach to obtain coordinates in a global reference system, some prior works incorporate a small degree of ground-truth UE positions into the CC training objective. This can be done, for example, by adding the distance error between the pseudo-positions and the true positions to standard CC loss functions that preserve local geometry [17]–[22]. Another approach is to use the ground-truth positions as anchor points to compute an optimal affine transformation, with the goal of aligning the estimated positions with the real coordinate system [23]–[26]. However, labeled data collection still implies a measurement campaign. Moreover, new labeled data needs to be collected for every different environment. Another approach is to combine knowledge of access point (AP) locations with knowledge of LoS zones and path-loss models to embed pseudo-positions in real-world coordinates [27]. In contrast to such methods, we will focus on CC-based positioning methods that deliver true spatial coordinates without requiring any ground-truth position labels.

A. Relevant Prior Art

Existing "label-free" CC approaches that provide coordinates in a global reference system can be divided into the following five categories: (i) based on sensor fusion [28], (ii) based on digital twins (DTs) [29], (iii) based on affine transformations

The work of JMMR, HW, and CH was supported, in part, by a grant from the Chalmers AI Research Center Consortium (CHAIR), by the National Academic Infrastructure for Supercomputing in Sweden (NAISS), the Swedish Foundation for Strategic Research (SSF) (grant FUS21-0004, SAICOM), and Swedish Research Council (VR grant 2022-03007). The work of CH was also supported by the Swedish Research Council under grant no. 2020-04718. The work of FZ and CS was supported in part by the Swiss National Science Foundation (SNSF) grant 200021_207314 and by CHIST-ERA grant for the project CHASER (CHIST-ERA-22-WAI-01) through the SNSF grant 20CH21_218704.

José Miguel Mateos-Ramos, Christian Häger, and Henk Wymeersch are with the Department of Electrical Engineering, Chalmers University of Technology, Sweden (email: josemi@chalmers.se; christian.haeger@chalmers.se; henkw@chalmers.se).

Frederik Zumegen and Christoph Studer are with the Department of Information Technology and Electrical Engineering, ETH Zürich, 8092 Zürich, Switzerland (email: fzumegen@ethz.ch; cstuder@ethz.ch).

leveraging floor plans [30]–[33], (iv) based on conventional estimation methods [34], [35], and (v) based on path-loss models [27], [36].

The work in [28] introduces a sensor-fusion approach where the UEs are equipped with a two-dimensional laser scanner that collects depth measurements. The displacement of the UEs between time steps is estimated from the laser data and added to a tailored loss function during training to preserve global geometry. However, including a laser scanner in the UEs is costly and positioning requires tight synchronization between the UEs and the APs, which is impractical.

In [29], a digital representation of the physical environment, or *DT* [37], is used to preserve global geometry, avoiding the need for external sensors in the UEs. An outdoor urban environment is simulated with a DT to generate synthetic labeled positioning data that relates the CSI in the DT to ground-truth positions. DT simulations include detailed building and traffic information to obtain accurate synthetic data that is less costly to collect than real measurements. However, the fine-grained details in the DT drastically increase the modeling complexity of the scenario, the computational complexity of the simulations, and the likelihood of degrading positioning performance under model mismatches.

Other works leverage affine transformations without ground-truth positions to align the estimated positions with the real coordinate system [30]–[33]. These approaches typically do not require fine-grained details of the environment and are easier to implement in different scenarios. In [30], AP positions are estimated based on the measurements with the highest received power, which are then used as anchors to find the optimal transform. The method in [31] extends the optimal transport in [38], which aligns the estimated positions with the floor plan, to also learn the probabilities that positions fall within specific regions. This approach enables uncertainty quantification of the estimated position, e.g., for estimated positions belonging to unexplored areas. In [32], model-based and NN-based methods are combined, leveraging model-based estimates under LoS conditions and training a NN in a self-supervised manner to estimate NLoS samples. The self-generated labeled data for the NN consists of model-based estimations, where an optimal transport operation is also performed in the case of NLoS samples. In [33], the coverage area of each AP is known and the optimal transport operation maps the pseudo-positions estimated by each AP to their coverage area while preserving the overlap between the coverage areas of the APs. The first limitation of [30]–[33] is that they require a large number of estimated positions to apply the affine transform and obtain accurate performance. The approach in [30] needs to distinguish samples close to the APs, [31] constructs a probability map of the estimated positions, and [32] assumes that NLoS estimates follow a uniform distribution across the NLoS area. Under a limited number of estimated positions, e.g., if the UEs have not explored the entire environment, the performance of such methods likely degrades. Additionally, the considered transforms only apply affine operations, disregarding more complex non-affine mappings.

The work in [34] designs a loss function based on the root-multiple signal classification algorithm for AoA and ToA estimation. This loss function is added to a CC loss that preserves the local geometry. In [35], the Doppler effect is included in a tailored loss function assuming a free-space channel model and knowledge of the AP positions. The estimated positions of [34], [35] in global coordinates are directly obtained in real time without relying on affine transformations and conventional algorithms are outperformed, but the performance under NLoS conditions remains unclear.

Finally, works based on path-loss models [27], [36] assume that AP positions are known and enforce estimated positions to lie closer to the AP with the strongest received signal. For instance, [36] divides the floor plan into zones and leverages a small subset of zone-annotated CSI samples during training to predict the zone of new samples. Once a new sample is assigned to a zone, two weakly supervised losses enforce it to belong to the estimated zone and lie closer to access points (APs) with the strongest received power. However, the method in [36] requires zone-annotated data, which increases the complexity of the positioning algorithm and zone-division is based on a heuristic method. The recent work of [27] performs global positioning without dividing the floor plan into zones or annotated data. Instead, a power threshold based on received signals is computed to distinguish measurements in the LoS area of each AP and estimated positions are, using a bilateration loss, enforced to be closer to the AP that received the strongest signal. Additionally, if a bounding box describing the LoS area of each AP is available, LoS positions are further constrained to remain within these bounding boxes. Although the work in [27] provides an effective way of estimating positions in global coordinates, it presents several issues that limit its practical implementation: Firstly, distinguishing if a received signal belongs to the LoS area of an AP is based on thresholding the received power at each AP, which can fail in scenarios with heterogeneous devices that can select their transmission power. Secondly, the bilateration loss is based on path-loss models that fail under NLoS conditions. Therefore, there is a need to develop self-supervised positioning techniques that can work with different devices and under NLoS propagation conditions.

B. Contributions

In this work, we leverage a DT of an uplink distributed multiple input multiple output (D-MIMO) scenario with several UEs and APs in order to perform CC that preserves the global geometry of estimated positions. The key contributions of this work are as follows:

- *Efficient DT-aided self-supervised positioning:* We propose, for the first time, a computationally efficient self-supervised DT-aided positioning framework that can work with heterogeneous devices. Our method is based on large-scale features computed from the DT and the CSI. Recent work has shown that large-scale features can be computed in a DT for real-time applications [39]. As large-scale features, we consider the average received power, the power per angular direction and delay tap, the truncated delay profile, and the magnitude of the covariance matrix of the channel. Compared to [27], which is based on thresholding the power of the received signals at each AP,

our large-scale features do not depend on the transmitted power. This advantage enables our method to work with heterogeneous devices that freely select their transmit powers. Moreover, unlike [27], our method constrains the estimated positions, providing a bounded positioning error. Our results show that our approach outperforms the work in [27] for the same simulated scenario. At the same time, our proposed framework is based on large-scale features that drastically reduce the DT modeling and computational complexity compared to [29].

- *Robustness under modeling mismatches:* We evaluate the performance of our approach under mismatches in the modeling of the DT compared to the real scenario, in contrast to [29], in which the real and simulated data are drawn from the same DT simulation. We consider mismatches in the modeling of the position of the APs during training. We show that even under such mismatches, the proposed framework remains robust and obtains better performance compared to the state of the art.
- *Generalization capability:* We consider a distribution shift between the training and the testing data. Although the proposed framework is continuously trained, this test evaluates the performance when the scenario changes before the system is fine-tuned with new observed data, which is rarely studied in CC works. In our simulations, we decrease the height of the UEs during inference and show that the proposed approach only slightly degrades under such mismatches when large-scale features are considered.

C. Outline

The rest of the paper is organized as follows. Sec. II introduces the system model, including the DT description. Sec. III presents a background on CC and proposes our framework. Sec. IV compares the proposed approach with state-of-the-art baselines and Sec. V concludes.

D. Notation

Vectors and matrices are denoted as boldface lowercase and uppercase letters, respectively. Sets are denoted as uppercase calligraphic letters and are defined by curly brackets; the cardinality of a set $\mathcal{A}$ is denoted as $|\mathcal{A}|$. The transpose and conjugate transpose of a matrix $\mathbf{A}$ are denoted by $\mathbf{A}^\top$ and $\mathbf{A}^\mathsf{H}$, respectively. The ith row and jth column of a matrix $\mathbf{A}$ are denoted as $[\mathbf{A}]_{i,:}$ and $[\mathbf{A}]_{:,j}$, respectively. The upper-left submatrix containing the first M rows and N columns of $\mathbf{A}$ is denoted as $[\mathbf{A}]_{1:M,1:N}$. The matrix vectorization operation is denoted as $\mathrm{vec}(\mathbf{A})$. We denote the aggregation of two matrices $\mathbf{A} \in \mathbb{R}^{M\times N}$ and $\mathbf{B} \in \mathbb{R}^{M\times N}$ along the first dimension as $[\mathbf{A};\mathbf{B}] \in \mathbb{R}^{2M\times N}$ and along the second dimension as $[\mathbf{A},\mathbf{B}] \in \mathbb{R}^{M\times 2N}$. The positive part of a real number x is denoted as $(x)^+ = \max\{0, x\}$. The N-point unitary discrete Fourier transform (DFT) matrix is denoted as $\mathbf{F}_N$. The ℓ^2-norm of a vector and the Frobenius norm of a matrix are denoted as $\|\mathbf{a}\|$ and $\|\mathbf{A}\|_\mathrm{F}$, respectively; the all-ones vector is denoted as $\mathbf{1}$ and the indicator function is denoted as $\mathbb{1}\{\cdot\}$.

II. System Model

We consider an uplink D-MIMO scenario with multiple UEs and A APs. The UEs transmit orthogonal frequency-division multiplexing (OFDM) signals with S subcarriers at timestamp $t^{(n)}$ from the position $\mathbf{x}^{(n)}$, for $n \in \mathcal{N} = \{1, \ldots, N\}$. Each AP and UE is equipped with an antenna array of K and L antenna elements, respectively. We assume a multiple-access protocol that schedules the UE transmissions, ensuring that the received signal at each AP and timestamp corresponds to a single UE. The CSI matrix estimated by AP a at the nth timestamp $t^{(n)}$ is $\mathbf{H}^{(a,n)} \in \mathbb{C}^{K\times S}$, for $a = 1, 2, \ldots, A$. The estimated CSI matrix includes the effect of additive white Gaussian noise (AWGN) of variance σ^2. The CSI matrix at timestamp $t^{(n)}$ is $\mathbf{H}^{(n)} = [\mathbf{H}^{(1,n)}; \cdots; \mathbf{H}^{(A,n)}] \in \mathbb{C}^{M\times S}$, with $M = AK$. The collection $\{\mathbf{H}^{(n)}, t^{(n)}\}_{n=1}^N$ constitutes the database of data collected from the real environment.

We also consider a DT that models the uplink D-MIMO scenario. The DT is defined by the position and materials of reflective surfaces (walls, floor, ceiling, or other objects), the position of the APs, and radio-frequency parameters (carrier frequency, bandwidth, antennas, and transmitted signal). We predefine a set of UE positions in the DT $\{\tilde{\mathbf{x}}^{(p)}\}_{p=1}^P$, for $p = 1, \ldots, P$, where $\tilde{\mathbf{x}}^{(p)} \in \mathbb{R}^{D_\mathrm{x}}$ and D_x is either 2 or 3. The CSI matrix synthesized from the DT at AP a corresponding to the pth position is denoted by $\tilde{\mathbf{H}}^{(a,p)} \in \mathbb{C}^{K\times S}$ and the CSI matrix corresponding to the pth position is $\tilde{\mathbf{H}}^{(p)} = [\tilde{\mathbf{H}}^{(1,p)}; \ldots; \tilde{\mathbf{H}}^{(A,p)}] \in \mathbb{C}^{M\times S}$. The collection $\{\tilde{\mathbf{H}}^{(p)}, \tilde{\mathbf{x}}^{(p)}\}_{p=1}^P$ constitutes the database of simulated CSI and predefined positions from the DT. As mentioned in Sec. I-B, our proposed framework is based on large-scale features from the DT and the observed CSI matrix. Generally, DTs can simulate the CSI matrix observed by each AP and we can then compute large-scale features from the CSI matrix, but efficient DTs can directly provide large-scale features; see, e.g., [39]. We based the description of the DT and the proposed method of Sec. III on the general case that the DT simulates the CSI matrix, but in Sec. IV we also consider large-scale features directly computed from the DT.

The objective of positioning is to find a function that maps the estimated CSI from the measured data $\mathbf{H}^{(n)}$ to the true UE position $\mathbf{x}^{(n)}$, given some side information about the environment through the DT model. To find such a function, we first design a training phase that uses the unlabeled dataset $\{\mathbf{H}^{(n)}, t^{(n)}\}_{n=1}^N$ and the synthesized DT data $\{\tilde{\mathbf{H}}^{(p)}, \tilde{\mathbf{x}}^{(p)}\}_{p=1}^P$ to produce estimated positions close to the true positions.

III. Digital-Twin-Aided Channel Charting

We now describe the operating principles of conventional CC, which serves as a foundation for our proposed method. We then describe the proposed framework and formulate the training procedure as an optimization problem. We conclude with a detailed description of the considered large-scale features.

A. Background on Channel Charting

CC is composed of two operations [15]: (i) feature extraction and (ii) positioning CC function. In (i), large-scale feature

vectors are computed from the CSI since small-scale fading can produce large variations in the CSI for close positions in space [40]. We compute the large-scale CSI feature vector $\mathbf{f}^{(n)}$ at time instant $t^{(n)}$ as in [27] as follows:

$$\mathbf{f}^{(n)} = f(\mathbf{H}^{(n)}) = \frac{|\mathbf{h}^{(n)}|}{\|\mathbf{h}^{(n)}\|}. \tag{1}$$

Here, $\mathbf{h}^{(n)} = \mathrm{vec}([\mathbf{H}^{(n)}\mathbf{F}^{\mathsf{H}}_{S]:,1:C}) \in \mathbb{C}^{D_\mathrm{h}}$ and $D_\mathrm{h} = MC$. An inverse DFT of the CSI matrix across subcarriers is performed in [27] because most of the received power is expected to be on the first C time-domain taps and the dimensionality of $\mathbf{H}^{(n)}$ can be reduced with minimal information loss. In (ii), the positioning CC function is expressed as $\hat{\mathbf{x}}^{(n)}_{\boldsymbol{\theta}} = g_{\boldsymbol{\theta}}(\mathbf{f}^{(n)}) \in \mathbb{R}^{D_\mathrm{x}}$ with learnable parameters $\boldsymbol{\theta}$. This function can then be used to extract the estimated positions of the UEs.

The goal of CC is to optimize the parameters $\boldsymbol{\theta}$ so that training samples with similar feature vectors $\mathbf{f}^{(n)}$ yield estimated positions $\hat{\mathbf{x}}^{(n)}_{\boldsymbol{\theta}}$ that are close in $\mathbb{R}^{D_\mathrm{x}}$. Different loss functions have been proposed to optimize the parameters $\boldsymbol{\theta}$ [15], [17], [18], [41]–[43]. In this work, we utilize the timestamp-based triplet-loss approach from [41] to enable a fair comparison with [27] (see Sec. IV for the details), which we describe in the following. First, we define the set of triplets

$$\begin{aligned}\mathcal{T} = \{(n,c,f) \in \mathcal{N}^3 :\\ 0 < |t^{(n)} - t^{(c)}| \le T_\mathrm{c} \le T_\mathrm{f} < |t^{(n)} - t^{(f)}|\},\end{aligned} \tag{2}$$

where $T_\mathrm{c}, T_\mathrm{f} > 0$ are coherence-time parameters that distinguish if CSI measurements are close (T_c) or far (T_f) in time to a CSI measurement at a timestamp $t^{(n)}$. If $|t^{(n)} - t^{(c)}| < |t^{(n)} - t^{(f)}|$, we expect that $\hat{\mathbf{x}}^{(n)}_{\boldsymbol{\theta}}$ is closer to $\hat{\mathbf{x}}^{(c)}_{\boldsymbol{\theta}}$ than to $\hat{\mathbf{x}}^{(f)}_{\boldsymbol{\theta}}$. Second, the timestamp-based triplet loss is defined as

$$\begin{aligned}\mathfrak{L}_\mathrm{CC}(\boldsymbol{\theta}) = \frac{1}{|\mathcal{T}|}\sum_{(n,c,f)\in\mathcal{T}} (\|\hat{\mathbf{x}}^{(n)}_{\boldsymbol{\theta}} - \hat{\mathbf{x}}^{(c)}_{\boldsymbol{\theta}}\| \\ - \|\hat{\mathbf{x}}^{(n)}_{\boldsymbol{\theta}} - \hat{\mathbf{x}}^{(f)}_{\boldsymbol{\theta}}\| + M_\mathrm{t})^+,\end{aligned} \tag{3}$$

where $M_\mathrm{t} \ge 0$ is a hyper-parameter that enforces $\hat{\mathbf{x}}^{(n)}_{\boldsymbol{\theta}}$ to be at least M_t closer to $\hat{\mathbf{x}}^{(c)}_{\boldsymbol{\theta}}$ than to $\hat{\mathbf{x}}^{(f)}_{\boldsymbol{\theta}}$. Conventional CC aims to solve the optimization problem

$$\boldsymbol{\theta}^\star = \arg\min_{\boldsymbol{\theta}} \mathfrak{L}_\mathrm{CC}(\boldsymbol{\theta}), \tag{4}$$

where $\boldsymbol{\theta}^\star$ are the optimal parameters of the positioning CC function $g_{\boldsymbol{\theta}}(\cdot)$. The problem in (4) aims to preserve local geometry, but there is no guarantee that the learned CC positioning function reduces the error between the true and estimated positions. To resolve this limitation, we now introduce DT-aided CC .

B. Proposed Method

The proposed method is based on conventional CC and the timestamp-based triplet-loss of (3). We include the DT in CC in two stages: (i) in conventional CC to confine the estimated positions to the convex hull of the predefined DT positions and (ii) adding a new loss function to the loss of (3) that aims to preserve global geometry. The proposed framework is represented in Fig. 1 and described in the following.

1) Modified Channel Charting: Firstly, we turn the conventional CC regression problem into a classification problem as shown in the green box of Fig. 1. Instead of estimating the positions as $\hat{\mathbf{x}}^{(n)}_{\boldsymbol{\theta}} = g_{\boldsymbol{\theta}}(\mathbf{f}^{(n)})$, we increase the dimensionality of the output of the learnable function as in [44] and estimate the probability mass function $\mathbf{p}^{(n)}_{\boldsymbol{\theta}} = g_{\boldsymbol{\theta}}(\mathbf{f}^{(n)}) \in [0,1]^P$ of the true position at $t^{(n)}$, i.e., $[\mathbf{p}^{(n)}_{\boldsymbol{\theta}}]_i$ is the probability that the ith predefined position $\tilde{\mathbf{x}}^{(i)}$ is the true position at $t^{(n)}$ and $\mathbf{1}^\top\mathbf{p}^{(n)}_{\boldsymbol{\theta}} = 1$. Treating the localization problem as a classification problem was shown in [44] to result in better positioning performance with supervised learning. We then compute the expected position as $\hat{\mathbf{x}}^{(n)}_{\boldsymbol{\theta}} = q(\mathbf{p}^{(n)}_{\boldsymbol{\theta}}) = \tilde{\mathbf{X}}\mathbf{p}^{(n)}_{\boldsymbol{\theta}}$, with $\tilde{\mathbf{X}} = [\tilde{\mathbf{x}}^{(1)}, \ldots, \tilde{\mathbf{x}}^{(P)}] \in \mathbb{R}^{D_\mathrm{x}\times P}$. This mapping allows us to use the triplet loss in (3) to preserve local geometry and the estimated position is confined to the convex hull of the predefined positions $\tilde{\mathbf{X}}$.

2) Digital-Twin-Aided Loss Function: Secondly, we introduce a novel DT-aided loss function to preserve global geometry, corresponding to the orange box in Fig. 1. The proposed loss function is based on large-scale features computed from the observed CSI and the DT. Large-scale features are expected to vary only slowly with the position of the UEs, while the CSI rapidly varies over space in the order of the wavelength. Moreover, learnable functions like the NN considered in this work (see Sec. IV-B) suffer from spectral bias for lower frequencies [45]–[47], which correspond to slowly varying features in our setup. From the DT-based CSI $\tilde{\mathbf{H}}^{(a,p)}$, the feature vector estimated by AP a corresponding to the pth position in the DT is $\bar{\mathbf{v}}^{(a,p)} = h(\tilde{\mathbf{H}}^{(a,p)}) \in \mathbb{C}^D$, for $p = 1, 2, \ldots, P$, where D is the dimension of the feature vector to consider (see Sec. III-C). We denote the function that computes the large-scale features generally as $h(\cdot)$ since it can be different from $f(\cdot)$. The feature vector for the p-th predefined position is $\bar{\mathbf{v}}^{(p)} = [\bar{\mathbf{v}}^{(1,p)}; \cdots; \bar{\mathbf{v}}^{(A,p)}] \in \mathbb{C}^{D_\mathrm{v}}$, where $D_\mathrm{v} = AD$. We compute the expected value of the large-scale feature vectors $\{\bar{\mathbf{v}}^{(p)}\}_{p=1}^P$ at timestamp $t^{(n)}$ using the estimated probabilities $\mathbf{p}^{(n)}_{\boldsymbol{\theta}}$ as $\tilde{\mathbf{v}}^{(n)}_{\boldsymbol{\theta}} = u(\mathbf{p}^{(n)}_{\boldsymbol{\theta}}) = \bar{\mathbf{V}}\mathbf{p}^{(n)}_{\boldsymbol{\theta}} \in \mathbb{C}^{D_\mathrm{v}}$, where $\bar{\mathbf{V}} = [\bar{\mathbf{v}}^{(1)}, \ldots, \bar{\mathbf{v}}^{(P)}] \in \mathbb{C}^{D_\mathrm{v}\times P}$. Similarly, from the measured CSI we compute a large-scale feature vector per AP at $t^{(n)}$ as $\mathbf{v}^{(a,n)} = h(\mathbf{H}^{(a,n)}) \in \mathbb{C}^D$. Aggregating across APs, we obtain another feature vector at $t^{(n)}$ as $\mathbf{v}^{(n)} = [\mathbf{v}^{(1,n)}; \ldots; \mathbf{v}^{(A,n)}] \in \mathbb{C}^{D_\mathrm{v}}$. We combine the feature vectors from the DT and the measured CSI in the loss function

$$\mathfrak{L}_\mathrm{DT}(\boldsymbol{\theta}) = \frac{1}{N}\sum_{n\in\mathcal{N}} \exp\left(-\frac{|(\mathbf{v}^{(n)})^\mathsf{H}\tilde{\mathbf{v}}^{(n)}_{\boldsymbol{\theta}}|^2}{\|\mathbf{v}^{(n)}\|^2\|\tilde{\mathbf{v}}^{(n)}_{\boldsymbol{\theta}}\|^2}\right). \tag{5}$$

The loss function in (5) involves the cosine similarity between the vectors $\mathbf{v}^{(n)}$ and $\tilde{\mathbf{v}}^{(n)}_{\boldsymbol{\theta}}$, which is used in other deep learning tasks [48], [49], and a nonlinear exponential term that empirically provides better results in our scenario. Moreover, the vector normalization in (5) accounts for mismatched real and simulated transmitted powers of the UEs. Note that the feature vector $\mathbf{v}^{(n)}$ from the estimated CSI does not depend on the learnable parameters as the function $h(\cdot)$ is not learnable. The loss in (5) provides a theoretical guarantee of the positioning performance when the synthesized DT CSI $\tilde{\mathbf{H}}^{(p)}$

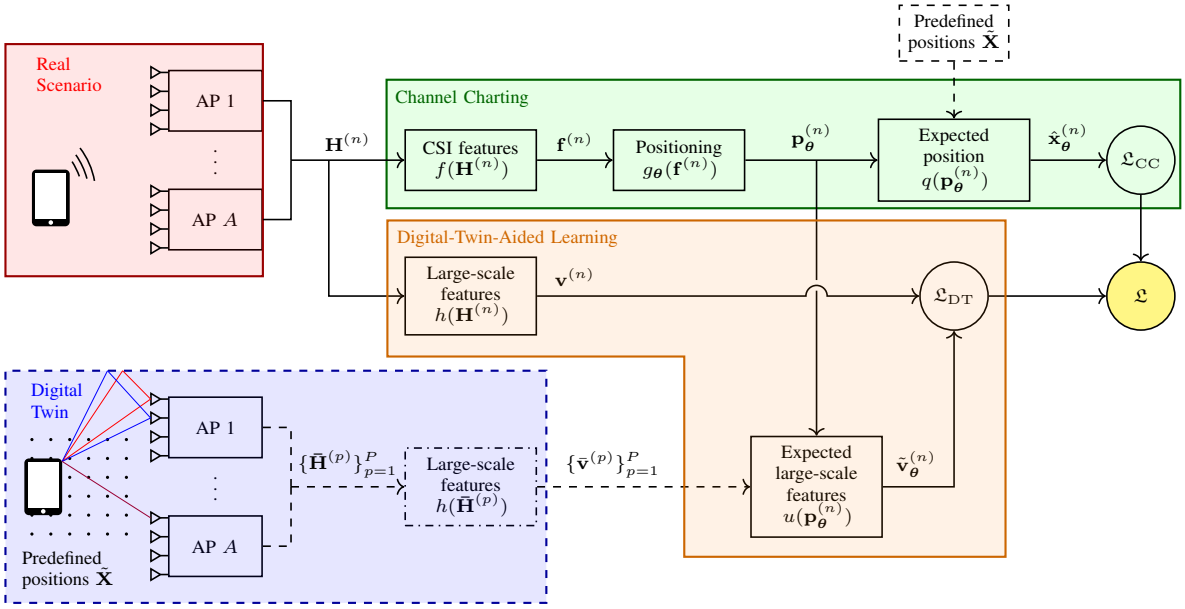


Fig. 1: Block diagram of the proposed approach. Dashed elements are predefined and kept fixed during training. The large-scale features in the DT can be computed directly without first obtaining the CSI $\tilde{\mathbf{H}}^{(p)}$. During inference, the DT and the DT-aided learning blocks are not used.

matches the estimated CSI $\mathbf{H}^{(n)}$.

Lemma 1. *The loss function in* (5) *is minimized if and only if* $\mathbf{v}^{(n)}$ *is linearly dependent with* $\tilde{\mathbf{v}}_{\boldsymbol{\theta}}^{(n)}$.

Proof.

$$\mathfrak{L}_{\mathrm{DT}}(\boldsymbol{\theta}) = \frac{1}{N}\sum_{n\in\mathcal{N}} \exp\left(-\frac{|(\mathbf{v}^{(n)})^{\mathsf{H}}\tilde{\mathbf{v}}_{\boldsymbol{\theta}}^{(n)}|^2}{\|\mathbf{v}^{(n)}\|^2\|\tilde{\mathbf{v}}_{\boldsymbol{\theta}}^{(n)}\|^2}\right) \tag{6}$$

$$\geq \exp\left(-\frac{1}{N}\sum_{n\in\mathcal{N}} \frac{|(\mathbf{v}^{(n)})^{\mathsf{H}}\tilde{\mathbf{v}}_{\boldsymbol{\theta}}^{(n)}|^2}{\|\mathbf{v}^{(n)}\|^2\|\tilde{\mathbf{v}}_{\boldsymbol{\theta}}^{(n)}\|^2}\right) \tag{7}$$

$$\geq \exp(-1), \tag{8}$$

where in (7) we used Jensen's inequality [50] and in (8) we used Cauchy–Schwarz inequality [51]. The inequalities in (7) and (8) hold with equality if and only if $\mathbf{v}^{(n)}$ is linearly dependent with $\tilde{\mathbf{v}}_{\boldsymbol{\theta}}^{(n)}$. □

Proposition 1. *Let*

$$\mathcal{L}^{(P)} = \{\mathbf{p}\in[0,1]^P : \mathbf{1}^\top\mathbf{p} = 1\} \tag{9}$$

be the set of all possible probability vectors of dimension P. *Assume a matched DT such that*

$$\mathbf{x}^{(n)} = \tilde{\mathbf{X}}\mathbf{p}^\star \tag{10}$$

$$\mathbf{v}^{(n)} = \bar{\mathbf{V}}\mathbf{p}^\star \neq 0, \tag{11}$$

for some $\mathbf{p}^\star \in \mathcal{L}^{(P)}$ *and a fixed* $n \in \mathcal{N}$. *Suppose* $\bar{\mathbf{V}} \in \mathbb{C}^{D_{\mathrm{v}}\times P}$ *is tall and has full rank* ($P \leq D_{\mathrm{v}}$). *Then, any global minimizer* $\boldsymbol{\theta}^\star$ *satisfies*

$$\mathbf{p}_{\boldsymbol{\theta}^\star}^{(n)} = \mathbf{p}^\star \quad \textit{and} \quad \hat{\mathbf{x}}_{\boldsymbol{\theta}^\star}^{(n)} = \tilde{\mathbf{X}}\mathbf{p}_{\boldsymbol{\theta}^\star}^{(n)} = \mathbf{x}^{(n)}. \tag{12}$$

Proof. By Lemma 1, at any global minimizer $\boldsymbol{\theta}^\star$ we must have

$$\tilde{\mathbf{v}}_{\boldsymbol{\theta}^\star}^{(n)} = \bar{\mathbf{V}}\mathbf{p}_{\boldsymbol{\theta}^\star}^{(n)} = \alpha\mathbf{v}^{(n)} = \alpha\bar{\mathbf{V}}\mathbf{p}^\star, \tag{13}$$

for some $\alpha \in \mathbb{C}\backslash\{0\}$. Because $\bar{\mathbf{V}}$ is full rank, the mapping $\mathbf{p}\mapsto\bar{\mathbf{V}}\mathbf{p}$ is injective, implying that

$$\mathbf{p}_{\boldsymbol{\theta}^\star}^{(n)} = \alpha\mathbf{p}^\star. \tag{14}$$

Since both $\mathbf{p}_{\boldsymbol{\theta}^\star}^{(n)}$ and $\mathbf{p}^\star$ belong to $\mathcal{L}^{(P)}$, it follows that $\alpha = 1$ and hence, $\mathbf{p}_{\boldsymbol{\theta}^\star}^{(n)} = \mathbf{p}^\star$. Substituting into $\hat{\mathbf{x}}_{\boldsymbol{\theta}^\star}^{(n)} = \tilde{\mathbf{X}}\mathbf{p}_{\boldsymbol{\theta}^\star}^{(n)}$ yields $\hat{\mathbf{x}}_{\boldsymbol{\theta}^\star}^{(n)} = \mathbf{x}^{(n)}$. □

Remark 1. *Because* $\tilde{\mathbf{X}}$ *is typically a wide matrix* ($P > D_{\mathrm{x}}$), *the mapping* $\mathbf{p}\mapsto\tilde{\mathbf{X}}\mathbf{p}$ *is not injective—in fact, many* $\mathbf{p}\in\mathcal{L}^{(P)}$ *can represent the same position* $\mathbf{x}^{(n)} = \tilde{\mathbf{X}}\mathbf{p}$. *Hence, barycentric coordinates in position space are generally non-unique. However, since* $\bar{\mathbf{V}}$ *is tall and full column rank, the mapping* $\mathbf{p}\mapsto\bar{\mathbf{V}}\mathbf{p}$ *is injective. Consequently, even if several* $\mathbf{p}$ *vectors lead to the same* $\mathbf{x}^{(n)}$, *only one of them yields the correct feature vector* $\mathbf{v}^{(n)} = \bar{\mathbf{V}}\mathbf{p}^\star$. *The optimization in feature space therefore selects the unique* $\mathbf{p}^\star$ *consistent with the observed* $\mathbf{v}^{(n)}$, *leading to* $\hat{\mathbf{x}}_{\boldsymbol{\theta}^\star}^{(n)} = \tilde{\mathbf{X}}\mathbf{p}^\star = \mathbf{x}^{(n)}$. *In this sense, the tall matrix* $\bar{\mathbf{V}}$ *disambiguates the non-uniqueness introduced by the wide matrix* $\tilde{\mathbf{X}}$.

Although Proposition 1 guarantees that a global optimizer that minimizes (5) can yield a perfect estimation under a matched DT, fulfilling condition (11) for all $n\in\mathcal{N}$ while $\bar{\mathbf{V}}$ is full rank is not straightforward, as it depends on the environment, the AP positions, and the large-scale features to consider. Given that our goal is to propose a general framework that can work for any given environment, we propose to combine the loss functions of (3) and (5) as

$$\mathfrak{L}(\boldsymbol{\theta}) = \lambda_{\mathrm{CC}}\mathfrak{L}_{\mathrm{CC}}(\boldsymbol{\theta}) + \lambda_{\mathrm{DT}}\mathfrak{L}_{\mathrm{DT}}(\boldsymbol{\theta}), \tag{15}$$

where $\lambda_{\mathrm{CC}}, \lambda_{\mathrm{DT}} \geq 0$ are tunable weights. Based on $\mathfrak{L}(\boldsymbol{\theta})$, we aim to solve

$$\boldsymbol{\theta}^\star = \arg\min_{\boldsymbol{\theta}} \mathfrak{L}(\boldsymbol{\theta}) \tag{16}$$

Algorithm 1 Training algorithm of the proposed approach.

Input: Data set of estimated CSI $\{\mathbf{H}^{(n)}\}_{n=1}^{N}$, grid of points in the DT $\tilde{\mathbf{X}}$, large-scale features from the DT $\bar{\mathbf{V}}$.
Output: Optimized NN parameters $\boldsymbol{\theta}^{(N_{\text{iter}})}$.
Initialization: NN Parameters $\boldsymbol{\theta}^{(0)}$.
for $i = 0, \ldots, N_{\text{iter}} - 1$ **do**
  Compute large-scale feature $\mathbf{v}^{(n)} = h(\mathbf{H}^{(n)})$.
  Compute the CSI features $\mathbf{f}^{(n)}$ following (1).
  Estimation with NN: $\mathbf{p}_{\boldsymbol{\theta}^{(i)}}^{(n)} = g_{\boldsymbol{\theta}^{(i)}}(\mathbf{f}^{(n)})$.
  Compute $\hat{\mathbf{x}}_{\boldsymbol{\theta}^{(i)}} = \tilde{\mathbf{X}}\mathbf{p}_{\boldsymbol{\theta}^{(i)}}^{(n)}$.
  Average large-scale features $\tilde{\mathbf{v}}_{\boldsymbol{\theta}^{(i)}}^{(n)} = \bar{\mathbf{V}}\mathbf{p}_{\boldsymbol{\theta}^{(i)}}^{(n)}$.
  Calculate $\mathfrak{L}_{\text{CC}}(\boldsymbol{\theta}^{(i)})$ following (3).
  Compute DT loss $\mathfrak{L}_{\text{DT}}(\boldsymbol{\theta}^{(i)})$ in (5).
  Compute $\mathfrak{L}(\boldsymbol{\theta}^{(i)})$ according to (15).
  Update $\boldsymbol{\theta}^{(i)}$ based on gradient descent.
end for

to learn a positioning function $g_{\boldsymbol{\theta}}(\cdot)$ that maps CSI features to positions close to the true positions. The proposed approach is summarized in Algorithm 1 for a fixed data set. If new samples of CSI are estimated, the same algorithm can be continuously applied with an extended data set including the new data points.

C. Large-Scale Features

Here we propose different large-scale feature processing functions $h(\cdot)$. These functions determine the large-scale features to compute from the estimated CSI and by the DT.

1) Power: We consider that the power received by AP a at timestamp $t^{(n)}$ is $h(\mathbf{H}^{(a,n)}) = \|\mathbf{h}^{(a,n)}\|^2$, where $\mathbf{h}^{(a,n)} = \text{vec}([\mathbf{H}^{(a,n)}\mathbf{F}_S^{\mathsf{H}}]_{:,1:C})$, which means that $D = 1$. This is the most straightforward large-scale feature vector and the easiest to acquire. The advantage is that this feature vector also works in a simpler scenario where the APs have a single antenna element. However, the *Power* feature presents a disadvantage as proven in the following proposition.

Proposition 2. *Let*

$$\mathcal{P}^{(a)} = \{p \in \{1, \ldots, P\} : \|\tilde{\mathbf{H}}^{(a',p)}\|_F^2 = 0 \text{ for } a' \neq a\} \quad (17)$$

be the set of points in the DT whose transmitted signal is at most received by AP a. If $[\mathbf{p}_{\boldsymbol{\theta}}^{(n)}]_p = 0$ for $p \notin \mathcal{P}^{(a)}$, then $\mathfrak{L}_{\text{DT}}$ does not depend on $\boldsymbol{\theta}$.

Proof. The claim follows from the fact that for points $p \in \mathcal{P}^{(a)}$, $\bar{\mathbf{v}}^{(p)}$ is nonzero in the ath position, and the normalized vector $\tilde{\mathbf{v}}_{\boldsymbol{\theta}}^{(n)}/\|\tilde{\mathbf{v}}_{\boldsymbol{\theta}}^{(n)}\|$ in (5) does not depend on $\boldsymbol{\theta}$. □

Proposition 2 implies that if there is a subset of points in the DT that are only served by one AP, the learnable function $g_{\boldsymbol{\theta}}(\cdot)$ stops updating the parameters $\boldsymbol{\theta}$ when the estimated position $\tilde{\mathbf{x}}^{(n)}$ is a combination of that subset of points. In areas covered by just one AP, the positioning error is therefore likely to be large. This means that the signal from the UEs must be received in the DT by at least two APs to reduce the positioning error.[1]

[1]Note that this does not mean that the UEs must be in LoS with the APs, but the signal transmitted by the UEs must be received by at least two APs.

2) Angle-power profile (APP): In order to attempt to resolve the issue of the *Power* feature, we define the *APP* of AP a at $t^{(n)}$ as $h(\mathbf{H}^{(a,n)}) = |\mathbf{H}_{\text{F}}^{(a,n)}|^2\mathbf{1}$, with $\mathbf{H}_{\text{F}}^{(a,n)} = \mathbf{F}_K\mathbf{H}^{(a,n)}$, which means that $D = K$. The APP aims to compute the power of the CSI matrix for different angles. This approach is much less likely to suffer the same problem as the *Power* feature if $K > 1$ since the dimensionality of $\mathbf{v}^{(n)}$ and $\tilde{\mathbf{v}}_{\boldsymbol{\theta}}^{(n)}$ increases. In the case of single-antenna APs, the APP is equivalent to the *Power* feature.

3) Delay-power profile (DPP): Similarly to the *APP*, we define the *DPP* of AP a at $t^{(n)}$ as $h(\mathbf{H}^{(a,n)}) = \mathbf{1}^\top|\mathbf{H}^{(a,n)}\mathbf{F}_S^{\mathsf{H}}|^2$, which aims to compute the power of the CSI matrix for different delay taps. Note that here we consider all S delay taps compared to the C taps in (1) because the large-scale features are only used to compute the loss in (5), whereas the CSI features $\mathbf{f}^{(n)}$ are processed by the positioning function $g_{\boldsymbol{\theta}}(\cdot)$, which is more computationally demanding. We include this feature to compare the effect of using angular and delay information in the proposed framework. In this case $D = S$. Assuming $S > K$ (which is typically the case in practice), this feature increases the dimensionality of $\mathbf{v}^{(n)}$ compared to the *APP*. For single-antenna APs, the *DPP* amounts to the *Power* feature.

4) Covariance: We define the covariance vector of AP a at $t^{(n)}$ as $h(\mathbf{H}^{(a,n)}) = \text{vec}(|\mathbf{H}_{\text{F}}^{(a,n)}(\mathbf{H}_{\text{F}}^{(a,n)})^{\mathsf{H}}|)$, which means that $D = K^2$. We introduce this feature since second-order moments have previously been used for conventional CC [15]. Note that the *APP* values are contained in $\mathbf{v}^{(n)}$. Assuming $K > 1$, this feature increases the dimensionality of $\mathbf{v}^{(n)}$ compared to the *APP*.

5) Truncated delay profile (TDP): We follow (1) for the CSI of each AP, i.e., $h(\mathbf{H}^{(a,n)}) = |\mathbf{h}^{(a,n)}|/\|\mathbf{h}^{(a,n)}\|$, where $\mathbf{h}^{(a,n)} = \text{vec}([\mathbf{H}^{(a,n)}\mathbf{F}_S^{\mathsf{H}}]_{:,1:C})$. This considers the large-scale feature of [27] both as input to the positioning function and for the DT-aided loss function. In this case, $D = KC$.

D. Discussion on Digital Twin Implementation

Our proposed approach relies on a DT to perform positioning. The DT comprises three components: the real space, the virtual space, and a communication link between them [37], [52]. The virtual space consists of a three-dimensional representation of the real environment to acquire simulated data, which in turn includes geometrical and material information. Geometrical modeling of the environment can be constructed via manual measurements, camera information (see [53] and references therein), or laser scanners [54]. Material properties of objects can be estimated through additional measurements [55], [56], however, they imply extra costs and may not scale well to large environments. We instead consider that materials are available a priori (e.g., from building specifications) and minor mismatches in material properties can be mitigated by our proposed approach, as shown in Sec. IV. After its initial construction, the DT is updated based on information collected from external sensors [57]. The communication link between the DT and the real environment ensures that the DT, and consequently the large-scale features, are updated and that the predicted positions of our proposed approach are transmitted to the APs.

IV. Results

We now present the results to assess the positioning performance of the proposed DT-aided CC approach.[2] We first describe the state-of-the-art baselines and the considered evaluation metrics. We then assess the achieved positioning results, the effect of the density of points in the DT, and the robustness under modeling mismatches in the DT.

A. Baselines

We consider three baselines to compare with our proposed approach. The first baseline is the best-performing CSI-based positioning method that does not rely on labeled data, synchronized APs, or affine transformations. The goal of this baseline is to frame the proposed approach with respect to state-of-the-art CC. The second baseline is a semi-supervised approach that uses labeled data after a first CC learning phase to compute the optimal affine transformation that obtains global coordinates. The last baseline is a supervised CSI fingerprinting approach that serves as a lower bound on the positioning error. Although the last two baselines use additional information in terms of labeled data compared to the proposed approach, we include them to assess how close the proposed approach is compared to methods that use labeled data.

1) Bilateration and Bounding-Box (BBB) Losses [27]: This approach also enhances conventional CC to improve positioning performance without the use of labeled data and it is based on two assumptions. The first assumption is the knowledge of the AP positions. From the CSI matrix $\mathbf{H}^{(a,n)}$, the power of AP a at $t^{(n)}$ is computed as $P^{(a,n)} = 20\log_{10}(\|\mathbf{H}^{(a,n)}\|_{\mathrm{F}})$. This defines the set of estimated LoS APs for each UE position as

$$\mathcal{A}^{(n)} = \{a \in \mathcal{A} : P^{(a,n)} > P_{\mathrm{thr}}\}, \tag{18}$$

where P_{thr} is a threshold set after observing the values of $P^{(a,n)}$ to ensure an effective distinction of LoS and NLoS samples. From $\mathcal{A}^{(n)}$, they define the set of AP pairs

$$\mathcal{P}^{(n)} = \{(a_{\mathrm{c}}, a_{\mathrm{f}}) \in (\mathcal{A}^{(n)})^2 : P^{(a_{\mathrm{c}},n)} > P^{(a_{\mathrm{f}},n)} + M_{\mathrm{p}}\}, \tag{19}$$

where $P^{(a_{\mathrm{c}},n)}$ and $P^{(a_{\mathrm{f}},n)}$ are the power received by a close and far AP, respectively, and M_{p} ensures that only pairs of APs whose powers differ by at least M_{p} are included. The set $\mathcal{P}^{(n)}$ is the foundation for the *bilateration loss*, defined as

$$\mathfrak{L}_{\mathrm{bi}}(\boldsymbol{\theta}) = \frac{1}{\sum_{n\in\mathcal{N}} |\mathcal{P}^{(n)}|} \sum_{n\in\mathcal{N}} \sum_{(a_{\mathrm{c}},a_{\mathrm{f}})\in\mathcal{P}^{(n)}} (\|\hat{\mathbf{x}}_{\boldsymbol{\theta}}^{(n)} - \underline{\mathbf{x}}^{(a_{\mathrm{c}})}\| - \|\hat{\mathbf{x}}_{\boldsymbol{\theta}}^{(n)} - \underline{\mathbf{x}}^{(a_{\mathrm{f}})}\| + M_{\mathrm{b}})^{+}, \tag{20}$$

where $\underline{\mathbf{x}}^{(a)}$ is the location of AP a and $M_{\mathrm{b}} \geq 0$ is a hyper-parameter that ensures that $\hat{\mathbf{x}}_{\boldsymbol{\theta}}^{(n)}$ is at least M_{b} closer to $\underline{\mathbf{x}}^{(a_{\mathrm{c}})}$ than to $\underline{\mathbf{x}}^{(a_{\mathrm{f}})}$. The second assumption is the knowledge of a bounding box corresponding to the LoS area of each AP. The bounding box for AP a is defined as

$$\mathcal{B}^{(a)} = \{(x,y) \in \mathbb{R}^2 : x \in [x_{\min}^{(a)}, x_{\max}^{(a)}], y \in [y_{\min}^{(a)}, y_{\max}^{(a)}]\}, \tag{21}$$

where the values $x_{\min}^{(a)}, x_{\max}^{(a)}, y_{\min}^{(a)}, y_{\max}^{(a)}$ are set based on geometrical information of the environment. This assumption enables the definition of a second loss function as

$$\mathfrak{L}_{\mathrm{box}}(\boldsymbol{\theta}) = \frac{1}{\sum_{n\in\mathcal{N}} |\mathcal{A}^{(n)}|} \sum_{n\in\mathcal{N}} \sum_{a\in\mathcal{A}^{(n)}} l_a(\hat{\mathbf{x}}_{\boldsymbol{\theta}}^{(n)}), \tag{22}$$

with

$$l_a(\mathbf{x}) = \mathbb{1}\{\mathbf{x} \notin \mathcal{B}^{(a)}\} \min_{\mathbf{x}'\in\mathcal{B}^{(a)}} \|\mathbf{x} - \mathbf{x}'\|^2. \tag{23}$$

This loss enforces estimated positions $\hat{\mathbf{x}}_{\boldsymbol{\theta}}^{(n)}$ in the LoS area of AP a to belong to the bounding box of that AP. Note that (22) still relies on the set $\mathcal{A}^{(n)}$ since the estimated positions can lie in arbitrary coordinates, and the received signals must be associated with the corresponding LoS APs. The losses in (20) and (22) are combined with the timestamp-based triplet-loss in (3) as

$$\mathfrak{L}_{\mathrm{BBB}}(\boldsymbol{\theta}) = \lambda_{\mathrm{CC}}\mathfrak{L}_{\mathrm{CC}}(\boldsymbol{\theta}) + \lambda_{\mathrm{bi}}\mathfrak{L}_{\mathrm{bi}}(\boldsymbol{\theta}) + \lambda_{\mathrm{box}}\mathfrak{L}_{\mathrm{box}}(\boldsymbol{\theta}). \tag{24}$$

Compared to [27], we assume that a DT provides large-scale features and we do not require thresholding the power received by the APs to discern if the UE is in the LoS area.[3]

2) Affine Transformation with Ground-truth Positions: This baseline considers a two-step approach for positioning. First, the loss in (15) with $\lambda_{\mathrm{CC}} = 1, \lambda_{\mathrm{DT}} = 0$ is used to train the NN, i.e., we use the timestamp-based triplet loss and the predefined positions to obtain a first estimate $\bar{\mathbf{x}}^{(n)}$ of the positions. Then, we assume that ground-truth positions are available for the entire training dataset and we solve the following least-squares problem

$$\{\hat{\mathbf{A}}, \hat{\mathbf{b}}\} = \arg \min_{\substack{\mathbf{A}\in\mathbb{R}^{D_{\mathrm{x}}\times D_{\mathrm{x}}} \\ \mathbf{b}\in\mathbb{R}^{D_{\mathrm{x}}}}} \sum_{n\in\mathcal{N}} \|(\mathbf{A}\bar{\mathbf{x}}^{(n)} + \mathbf{b}) - \mathbf{x}^{(n)}\|^2, \tag{25}$$

with $\mathbf{x}^{(n)}$ the true UE position at time $t^{(n)}$. The estimated position is computed using the affine transform $\hat{\mathbf{x}}_{\boldsymbol{\theta}}^{(n)} = \hat{\mathbf{A}}\bar{\mathbf{x}}^{(n)} + \hat{\mathbf{b}}$. This method assumes a labeled data collection step prior to training the NN. This baseline represents an extreme case of methods that use affine transformations, e.g., [24], [30], where all the samples are used to compute the optimal affine transformation.

3) CSI Fingerprinting [5], [6]: In CSI fingerprinting, labeled data of the true position of the UE is also available and it consists of two phases. In the first phase, known as the offline training phase, the labeled dataset $\{\mathbf{H}^{(n)}, \mathbf{x}^{(n)}\}_{n=1}^{N}$ relating CSI and UE positions is used to train the positioning function $g_{\boldsymbol{\theta}}(\cdot)$. To arrive at a fair comparison with our method, the position is estimated following the modified CC pipeline of our proposed method, i.e., $\hat{\mathbf{x}}_{\boldsymbol{\theta}}^{(n)} = q(\mathbf{p}_{\boldsymbol{\theta}}^{(n)})$. The loss used in fingerprinting during training is

$$\mathfrak{L}_{\mathrm{SL}}(\boldsymbol{\theta}) = \frac{1}{N} \sum_{n\in\mathcal{N}} \|\mathbf{x}^{(n)} - \hat{\mathbf{x}}_{\boldsymbol{\theta}}^{(n)}\|^2. \tag{26}$$

In the second phase, known as the online test phase, the learned positioning function is used to estimate the positions based on

[2]Source code to reproduce the simulation results in this paper is available at https://github.com/josemateosramos/DT_CC.

[3]One can also leverage a DT to compute the set $\mathcal{A}^{(n)}$ in (18) based on synthetic data. However, the DT and the real scenario may be mismatched. Thus, we follow the thresholding of [27] based on real data.

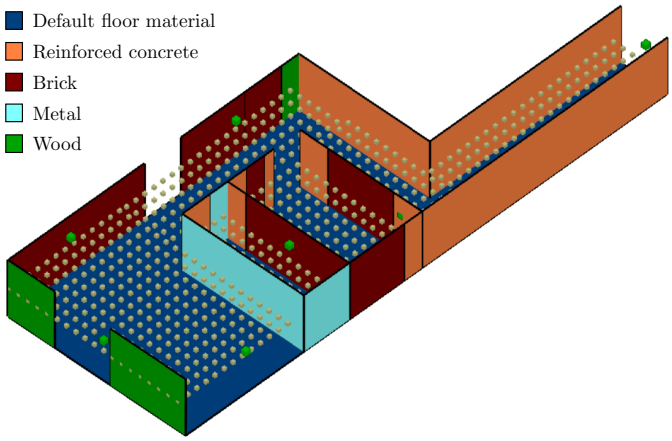


Fig. 2: Simulated indoor scenario. The light green points represent the AP positions and the beige points represent the predefined points in the DT. The colors of the walls represent different materials, where the default floor material is provided by Wireless InSite.

TABLE I: Parameters of the simulated indoor scenario.

Parameter	Value
Number of APs	$A = 8$
Number of antennas per AP	$K = 4$
Number of antennas of the UE	$L = 1$
AP, UE and DT antenna type	Half-wavelength dipole
ULA spacing	Half-wavelength
Number of UE positions	$N = 14606$
Number of DT positions	$P = 713$
Spacing between UE positions	2.5 cm
Spacing between DT positions	$\Delta = 0.5$ m
AP antenna height	2 m
UE antenna height	1.5 m
DT positions height	1.5 m
Carrier frequency	2.4 GHz
Bandwidth	20 MHz
Number of subcarriers	$S = 52$
Number of truncated taps	$C = 13$
Maximum SNR per AP	25 dB
Learnable function $g_{\boldsymbol{\theta}}(\cdot)$	Multi-layer perceptron
Number of hidden layers	4
Neurons per hidden layer	256
Hidden layer activation function	Rectified linear unit (ReLU)
Optimizer	Adam
Learning rate	$5 \cdot 10^{-3}$, *APP*, *TDP* $1 \cdot 10^{-3}$, otherwise
Training iterations	5000
Dropout (hidden layers)	0.15
Coherence time in (2)	$T_{\mathrm{c}} = T_{\mathrm{f}} = 2$ s
Distance threshold in (3)	$M_{\mathrm{t}} = 0.9$ m
CC weight in (15)	$\lambda_{\mathrm{CC}} = 10$, *Power* and *APP* $\lambda_{\mathrm{CC}} = 5$, *DPP* and *Covariance* $\lambda_{\mathrm{CC}} = 1$, *TDP*

new observed CSI data. CSI fingerprinting constitutes a lower bound in terms of positioning error since it uses ground-truth positions and unlike the affine transformation, the true positions are included in the loss function to train the NN, providing a more general mapping between the features $\mathbf{f}^{(n)}$ and the estimated positions $\hat{\mathbf{x}}_{\boldsymbol{\theta}}^{(n)}$.

B. Scenario

We simulate the uplink D-MIMO indoor scenario of [27], illustrated in Fig. 2. In our simulations, we particularize the proposed framework in Sec. III-B for a two-dimensional position estimation problem, i.e., $D_{\mathrm{x}} = 2$ and a single UE in the environment. The UE moves in this scenario (see Fig. 3a for a view of the trajectory of the UE), covering all rooms, which defines the training and testing sets. We consider a grid of points as the predefined positions in the DT, represented in Fig. 2.[4] The simulations use Remcom's Wireless InSite ray-tracing software to obtain the CSI matrices $\{\mathbf{H}^{(n)}\}_{n=1}^{N}$ and the simulated CSI $\{\tilde{\mathbf{H}}^{(p)}\}_{p=1}^{P}$. Initially, there is a perfect match between the DT and the real scenario (see Sec. IV-D for the mismatched case). The *Power* feature is directly available from the software, whereas the *APP, DPP*, and *Covariance* features are derived by first simulating the CSI matrices $\{\tilde{\mathbf{H}}^{(p)}\}_{p=1}^{P}$ and applying the functions described in Sec. III-C to compute the DT features $\bar{\mathbf{v}}^{(p)}$. Since no real measurements are available, we contaminate the CSI matrices $\mathbf{H}^{(n)}$ with AWGN. We define the SNR for AP a and timestamp $t^{(n)}$ as $\mathrm{SNR}^{(a,n)} = \|\mathbf{h}^{(a,n)}\|^2/\sigma^2$, with $\mathbf{h}^{(a,n)} = \mathrm{vec}([\mathbf{H}^{(a,n)}\mathbf{F}_S^{\mathsf{H}}]_{:,1:C})$ and σ^2 the variance of the AWGN. We add AWGN to $\mathbf{H}^{(n)}$ such that $\max_{a,n}\{\mathrm{SNR}^{(a,n)}\} = 25$ dB across training and testing sets. To compare with the results of [27], we do not add AWGN to samples where $\mathbf{H}^{(a,n)} = 0$, but these samples are still used for the computation of the CSI features $\mathbf{f}^{(n)}$ and the large-scale features $\mathbf{v}^{(n)}$. Note that we do not add AWGN to the DT CSI matrices $\tilde{\mathbf{H}}^{(p)}$. The dataset $\{\mathbf{H}^{(n)}\}_{n=1}^{N}$ is randomly split into training and test sets, with 80% of samples used for training.[5] As learnable function for positioning $g_{\boldsymbol{\theta}}(\mathbf{f}^{(n)})$, we consider a multi-layer perceptron with input and output dimensions determined by D_{h} in (1) and P, respectively. The learnable parameters $\boldsymbol{\theta}$ correspond to the weights and biases of the MLP layers. The simulation parameters are described in Table I. The dataset of collected CSI from the real scenario is based on standard 5G requirements. For example, a recent 5G testbed captured CSI with a target SNR of 28 dB every 10–20 ms [59]. A spatial resolution of 2.5 cm would correspond to an average UE velocity of 1.25–2.5 m/s, which lies within the typical walking speed range of healthy individuals [60]. Moreover, the considered unlabeled dataset would be collected in less than five minutes and the proposed approach can be executed for prolonged durations.

C. Positioning Results

Table II presents the inference results for the indoor scenario in Fig. 2. The metrics are described in Appendix A and the arrows next to the evaluation metrics in Table II indicate whether a higher or lower value is better. For practical applications, the mean distance error (MDE) is the foremost metric to evaluate, but other metrics are included for completeness. The results are averaged over 10 random NN initializations and expressed as $\mu \pm \sigma$, where μ and σ are the mean and standard deviation, respectively. The standard deviations for trustworthiness (TW),

[4] In this work, we assume that a single DT covers the entire scenario. Very large environments require division into smaller zones and subsequent zone stitching or identification, which is left as a future work.

[5] In hardware-limited systems, the number of training samples can be reduced, as shown in [58]. However, this study focuses on the best attainable positioning performance for the given training dataset.

TABLE II: Positioning results for the simulated indoor scenario.

	Method	Figure	TW ↑	CT ↑	KS ↓	RD ↓	MDE ↓	95th PDE ↓
Baselines	BBB [27]	3b	0.979	0.976	0.144	0.710	1.15 ± 0.04	2.84 ± 0.44
	Affine transformation	3c	0.954	0.957	0.375	0.879	3.20 ± 0.64	6.53 ± 1.43
	CSI fingerprinting	3d	0.993	0.993	0.060	0.479	0.41 ± 0.02	1.01 ± 0.08
Proposed	*Power*	3e	0.990	**0.989**	**0.098**	**0.603**	$\mathbf{0.81 \pm 0.05}$	$\mathbf{1.78 \pm 0.13}$
	Angle-power profile (APP)	3f	**0.991**	0.988	0.111	0.631	1.06 ± 0.07	2.67 ± 0.34
	Delay-power profile (DPP)	3g	0.990	0.987	0.105	0.620	0.91 ± 0.07	2.27 ± 0.36
	Covariance	3h	**0.991**	0.985	0.126	0.658	1.15 ± 0.14	3.11 ± 0.45
	Truncated delay profile (TDP)	3i	0.989	0.982	0.143	0.665	1.24 ± 0.46	3.63 ± 2.47

continuity (CT), Kruskal stress (KS), and Rajski distance (RD) are omitted due to their negligible values (similar to [27]). The results in Table II show that the proposed *Power*, *APP*, and *DPP* features outperform the state-of-the-art method in [27] and the affine transformation. This indicates that the use of a DT improves positioning performance compared to (i) the knowledge of the AP positions and their LoS area and (ii) methods that rely on the timestamp-based triplet loss and affine transformations. As expected, fingerprinting achieves superior performance, but it requires labeled UE position data.

Among the proposed features, *Power* performs best for most evaluation metrics, which suggests that relying on smooth large-scale features over space provides better results. The performance degrades when small-scale features are introduced in the *APP*, *DPP*, *TDP*, and *Covariance* features, confirming the relevance of large-scale features for CC [15].

Fig. 3 shows a top-down view of the true and estimated UE trajectories according to the proposed features and the baselines listed in Table II. As shown in Proposition 2, the *Power* feature performs poorly in the area enclosed by four walls, where only a single AP is available. The *APP* and *DPP* features provide an alternative solution that improves the performance in such areas at the cost of a slightly worse overall performance. The results in Table II and Fig. 3 indicate that the proposed approach is an effective CSI-based positioning method without requiring labeled data. Given that the *TDP* and *Covariance* features yield the poorest performances, they are not further considered in the remainder of the paper.

1) Effect of the Grid Spacing in the DT: In this test, we study the performance of the proposed approach as a function of the spacing Δ in the predefined DT positions. The spacing in Table II was $\Delta = 0.5$ m, corresponding to the grid shown in Fig. 2. In Table III, we show the MDE and 95th percentile distance error (PDE) results for different spacings in the predefined DT positions. The NN architecture described in Table I remains unchanged across Δ values except in the last layer, which depends on the number of predefined points P.

The results show that a finer spacing of $\Delta = 0.25$ m leads to a significant average performance drop and a larger standard deviation across NN initializations. However, the best-performing random initializations for $\Delta = 0.25\,\text{m}$ yield MDE $= 0.96\,\text{m}$ and 95th PDE $= 2.52\,\text{m}$ for *Power*, MDE $= 1.58\,\text{m}$ and 95th PDE $= 4.66\,\text{m}$ for *APP*, and MDE $= 0.79\,\text{m}$ and 95th PDE $= 1.74\,\text{m}$ for *DPP* which are comparable to the average performance obtained for $\Delta > 0.25\,\text{m}$. Given that the NN architecture and the learning hyper-parameters are optimized for $\Delta = 0.5\,\text{m}$, changing the grid spacing Δ reduces the convergence stability of the optimization problem in (16) across different random initializations.

TABLE III: Testing results for different position spacings Δ in the DT. The MDE (top) and 95th PDE (bottom) are shown in the form of $\mu \pm \sigma$, with μ and σ the mean and standard deviation over 10 random NN realizations.

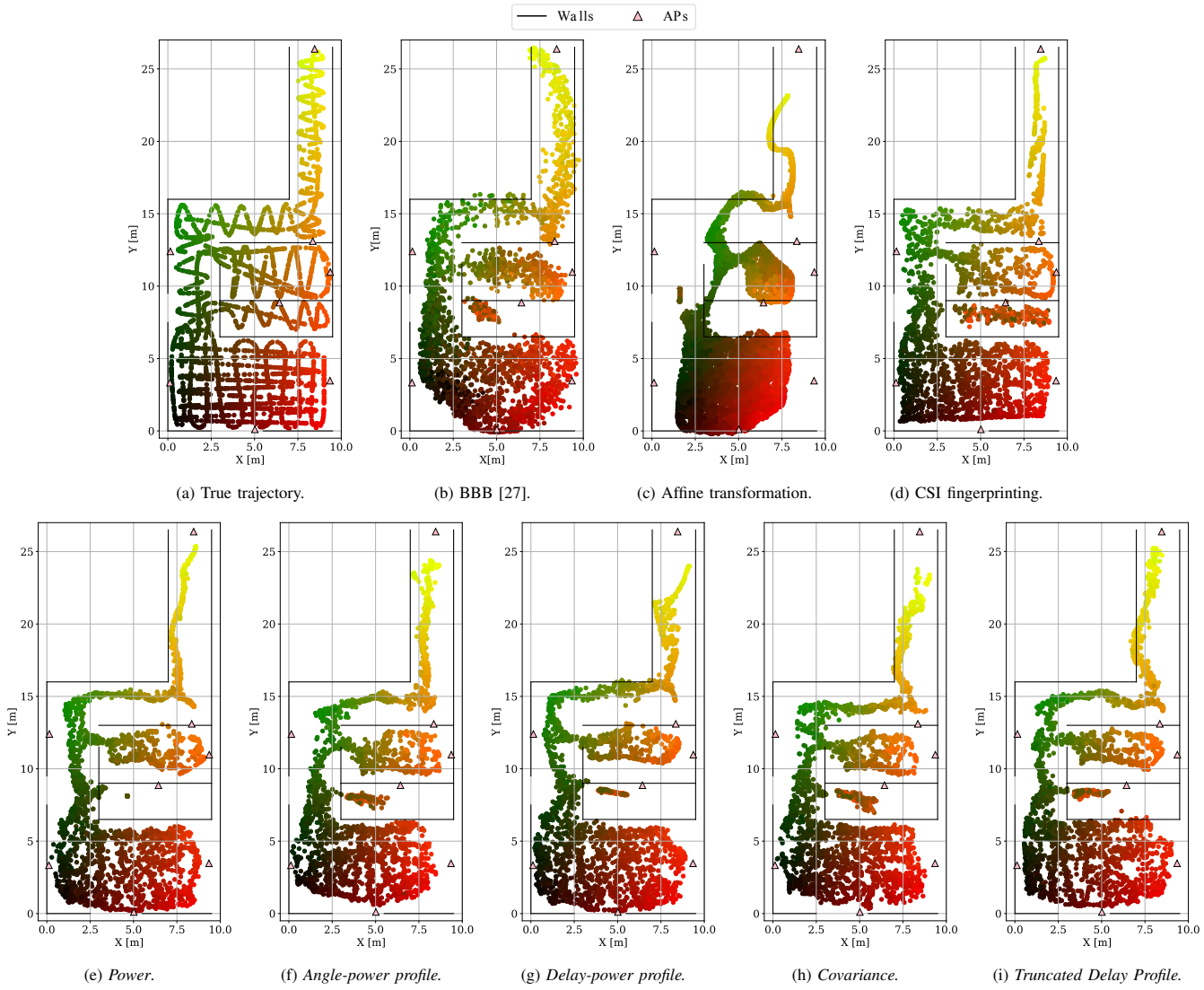
Method	$\Delta = 0.25$ m	$\Delta = 0.5$ m	$\Delta = 1$ m	$\Delta = 1.5$ m
Power	4.21 ± 3.74 9.05 ± 6.59	0.81 ± 0.05 1.78 ± 0.13	0.91 ± 0.02 1.90 ± 0.08	0.83 ± 0.03 1.80 ± 0.11
APP	5.32 ± 2.99 11.02 ± 4.67	1.06 ± 0.07 2.67 ± 0.34	1.26 ± 0.05 2.64 ± 0.28	1.66 ± 0.25 3.38 ± 0.84
DPP	3.74 ± 3.33 7.73 ± 5.88	0.91 ± 0.07 2.27 ± 0.36	1.18 ± 0.18 2.67 ± 0.93	1.72 ± 1.72 4.00 ± 3.25

The *Power* feature obtains similar results for $\Delta > 0.5$ m, while the *APP* and *DPP* features degrade in performance with a larger spacing. This difference arises because the *Power* is a large-scale feature that fluctuates more slowly over space compared to the *APP*, allowing the MLP to effectively interpolate feature vectors for off-grid positions. In contrast, the faster spatial variations of the *APP* and *DPP* hinder interpolation when the grid spacing is large. From the results in Table III, we conclude that the *Power* feature is robust under a sparser grid of predefined points in the DT, enabling reduced DT complexity without substantial performance loss.

2) Effect of Labeled Data on Fingerprinting: The results in Table II show a substantial performance difference between the proposed approach and fingerprinting. However, fingerprinting leverages the entire database as labeled samples, which is typically costly to obtain. To reduce the costs of fingerprinting, we combine the CC loss of (3) and the supervised loss of (26) as

$$\mathfrak{L}(\boldsymbol{\theta}) = \lambda_{\text{CC}}\mathfrak{L}_{\text{CC}} + \lambda_{\text{SL}}\mathfrak{L}_{\text{SL}}, \tag{27}$$

where $\lambda_{\text{SL}} > 0$ is a hyper-parameter that balances the contribution of the CC loss and the supervised loss. Moreover, we reduce the number of labeled training samples used for fingerprinting and compare the resulting performance with the proposed approach. In Fig. 4, we report the MDE of fingerprinting following (27), together with the proposed approach based on the *Power* feature. The training samples for fingerprinting are uniformly distributed in space, and we set $\lambda_{\text{CC}} = 1$ and $\lambda_{\text{SL}} = 10$. Our proposed approach is label-agnostic, resulting in the flat behavior observed in Fig. 4. The performance of fingerprinting degrades with fewer labeled samples since their spatial coverage becomes sparser and extrapolation to unseen locations becomes more challenging.



(a) True trajectory. (b) BBB [27]. (c) Affine transformation. (d) CSI fingerprinting.

(e) *Power*. (f) *Angle-power profile*. (g) *Delay-power profile*. (h) *Covariance*. (i) *Truncated Delay Profile*.

Fig. 3: True and estimated trajectories for the proposed approaches and the baselines in the simulated indoor scenario. For each estimated trajectory, we represent the best of the 10 realizations of the NN.

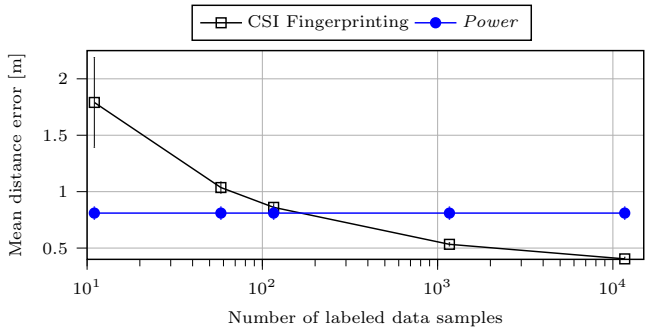


Fig. 4: Positioning performance as a function of the number of labeled data samples during training. The curves indicate the mean and the error bars the standard deviation across random NN initializations.

Additionally, under 100 labeled samples the proposed approach using the *Power* feature outperforms fingerprinting. This result underscores the dependence of fingerprinting on dense labeled datasets and its higher data acquisition cost compared to the proposed approach.

D. Robustness Tests

In this section, we evaluate the robustness of the proposed approach against mismatches at training and inference time. The former affects the assumed model of the DT while the latter studies the effect of unexpected elements in the real environment once the proposed approach has been trained.

1) Mismatched Digital Twin: So far, we have considered a perfect match between the DT and the considered scenario. In practice, information about the environment may be incomplete or inaccurate. We therefore evaluate the performance of the proposed approach under DT modeling mismatches during training. Particularly, we consider the following increasingly severe mismatches: (i) the APs in the DT are shifted by half a wavelength (6.25 cm) in the positive direction of the X-axis, (ii) the walls and floor materials in the DT are changed to wood, (iii) an additional wooden wall is introduced in the DT, and (iv) four additional walls are introduced in the DT. The two last cases

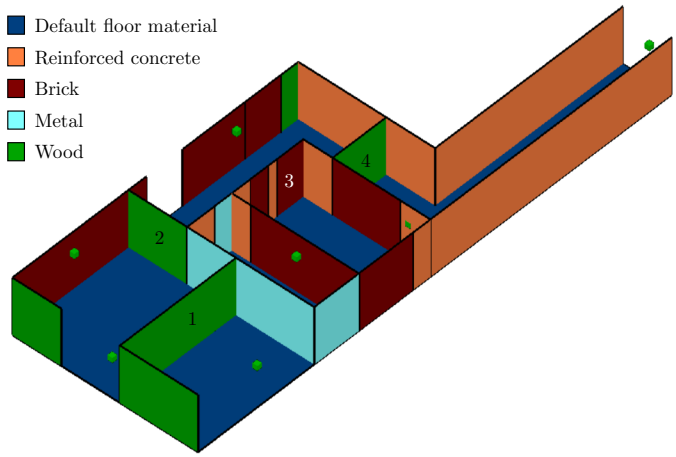


Fig. 5: Mismatched digital twin in the case of four additional walls. The additional walls are indicated with numbers. The case with *one additional wooden wall* only considers wall number 4.

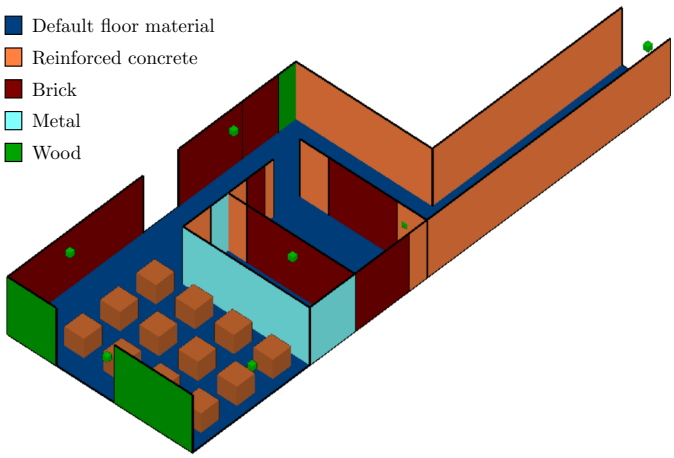


Fig. 6: Mismatched scenario with additional wooden boxes in the large room. The boxes have a size of 1 m in all axes.

are represented in Fig. 5. Furthermore, given access to a DT, we consider two practical DT-based fingerprinting baselines: (i) training a fingerprinting model using synthetic CSI generated by the DT [61], and (ii) performing fingerprinting using synthetic *Power* features from the DT. The latter corresponds to leveraging a computationally efficient DT [39] and the NN architecture is accordingly adjusted, reducing the input layer to a single neuron. For both baselines, we use the training data described in Sec. IV-B, i.e., the full training dataset. Table IV shows the inference results.

When the APs are shifted in the DT, the MDE of the proposed approach and DT-based *Power* fingerprinting are similar to the case of no mismatches in the DT, while DT-based CSI fingerprinting shows a slight performance degradation. This result indicates that a displacement of half a wavelength substantially alters the phase of the estimated CSI, thereby introducing a mismatch between the training and testing fingerprinting data. In contrast, the large-scale *Power* feature remains largely unchanged under the same displacement. As we introduce more severe impairments, all methods generally degrade in performance, reflecting the growing discrepancy between the DT and the real scenario. However, the degradation trends differ across methods, as discussed next.

Comparing DT-based CSI fingerprinting with the proposed approach, the former achieves better performance under no or slight mismatches in the DT as it exploits the full CSI information from the DT, while the proposed approach relies on lower-dimensional large-scale features such as the received power. Under severe impairments (e.g., four additional wooden walls), the proposed approach outperforms DT-based CSI fingerprinting. This indicates two relevant aspects. First, large-scale features are more robust to modeling mismatches as they vary more smoothly over space. This observation is consistent with the fact that DT-based *Power* fingerprinting outperforms DT-based CSI fingerprinting under severe mismatches. Second, incorporating CC in the proposed approach aligns the DT and the observed data, improving robustness. Conversely, DT-based CSI fingerprinting only harnesses synthetic fine-grained data, which limits the generalization of the learned positioning function to the real environment. This is further supported by comparing DT-based *Power* fingerprinting and the proposed approach, which use the same information from the DT, but the proposed framework additionally integrates observed data.

In summary, the results from Table IV indicate three regimes: (i) if a high-fidelity DT that computes the CSI is available, DT-based CSI fingerprinting yields the best results, at the cost of higher computational complexity; (ii) in the case of a low-fidelity (potentially mismatched) DT that generates CSI, DT-based CSI fingerprinting achieves better performance under slight mismatches, while the proposed approach remains robust under severe mismatches; and (iii) with a computationally efficient DT that only provides the received power, our approach achieves the best results.

2) Distribution Shift: Finally, we evaluate the positioning performance under mismatches between training and testing

TABLE IV: Mismatched DT test results. The MDE (top) and 95th PDE (bottom) are shown in the form of $\mu \pm \sigma$, with μ and σ the mean and standard deviation over 10 random NN realizations.

Method	No mismatch	Shifted AP positions	Material mismatches	One additional wooden wall	Four additional walls
Power	0.81 ± 0.05 1.78 ± 0.13	0.83 ± 0.06 2.02 ± 0.40	0.92 ± 0.06 2.33 ± 0.29	0.91 ± 0.04 2.20 ± 0.20	1.08 ± 0.21 2.49 ± 0.75
DT-based CSI fingerprinting	0.41 ± 0.02 1.01 ± 0.08	0.52 ± 0.02 1.26 ± 0.08	0.65 ± 0.02 1.65 ± 0.06	0.88 ± 0.06 2.44 ± 0.31	1.92 ± 0.18 4.56 ± 0.46
DT-based *Power* fingerprinting	0.90 ± 0.06 2.20 ± 0.15	0.99 ± 0.04 2.38 ± 0.06	1.04 ± 0.02 2.52 ± 0.04	1.15 ± 0.06 2.77 ± 0.17	1.54 ± 0.05 3.45 ± 0.10

TABLE V: Generalization test results. The MDE (top) and 95th PDE (bottom) are shown in the form of $\mu \pm \sigma$, with μ and σ the mean and standard deviation over 10 random NN realizations.

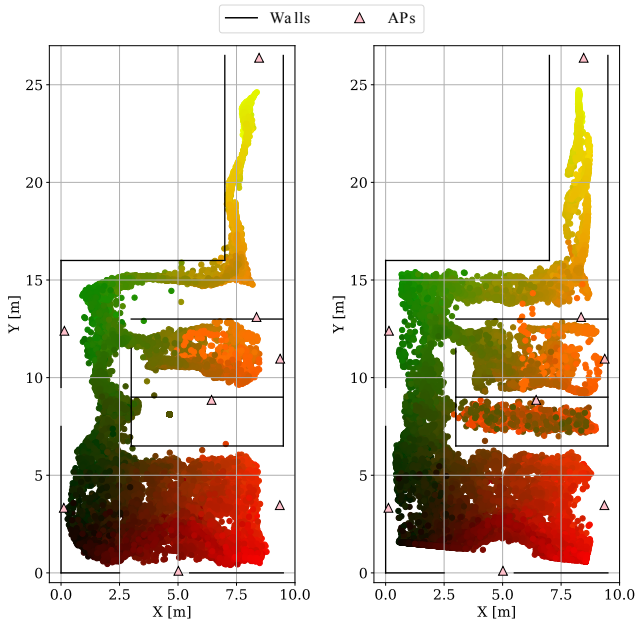
Method	No mismatch	Decreased UE height	Wooden boxes	Wooden wall
Power	0.81 ± 0.05 1.78 ± 0.13	0.97 ± 0.05 2.49 ± 0.17	0.82 ± 0.04 1.80 ± 0.14	1.10 ± 0.08 3.68 ± 0.21
APP	1.06 ± 0.07 2.67 ± 0.34	1.26 ± 0.06 3.18 ± 0.22	1.06 ± 0.06 2.69 ± 0.34	1.43 ± 0.05 4.15 ± 0.12
DPP	0.91 ± 0.07 2.27 ± 0.36	1.06 ± 0.08 2.70 ± 0.35	0.91 ± 0.07 2.25 ± 0.39	1.18 ± 0.05 3.82 ± 0.17
CSI fingerprinting	0.41 ± 0.02 1.01 ± 0.08	0.63 ± 0.01 1.79 ± 0.04	0.43 ± 0.01 1.02 ± 0.04	0.93 ± 0.03 3.79 ± 0.23

data. Two types of distribution shifts are considered: (i) changes in the UE position and (ii) changes in the environmental model. Note that in both cases, the DT is matched to the real scenario during training. For the first case, we consider that during training, the estimated CSI corresponds to a UE at 1.5 m height. At inference time, the estimated CSI corresponds to a UE at 0.8 m height. For the second case, we consider additional wooden boxes in the large room, as represented in Fig. 6, or the same additional wooden wall as in Fig. 5. This simulation reflects practical scenarios where environmental changes occur after deployment (e.g., a door being closed). The results are summarized in Table V, where we also included the results from Table II.

The performance slightly degrades under a UE height decrease compared to no mismatches in the DT. The degradation is mainly due to the pathloss effect, which we detail in the following. Fig. 7 represents the estimated positions when the UE height is changed. Compared with Fig. 3, Fig. 7 shows that no estimated positions appear near the APs (especially for the AP at the bottom center). This occurs because pathloss primarily depends on the distance between the AP and the UE. For example, the pathloss observed by a UE located at 1.5 m height and far from an AP in the horizontal plane is similar to that of a UE at 0.8 m height but closer in the horizontal plane. As a result, when inference is performed at the lower height, positions that are actually close to the AP are misinterpreted as points located farther away in the horizontal plane.

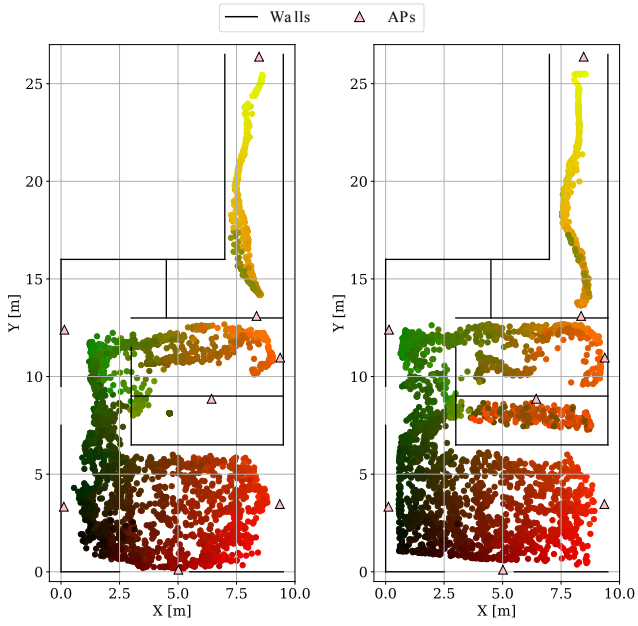
On the other hand, introducing wooden boxes yields similar performance to matched testing data, which suggests that the large-scale features remain mostly unaltered. Conversely, adding a wall in the upper corridor degrades the performance of the proposed approach as it partially blocks the received signal of the APs in that region. The corresponding estimated trajectories are included in Fig. 8. Positions in the corridor near the added wall are estimated closer to the APs, indicating that the attenuated signals are interpreted as originating from more distant locations relative to the blocked AP.

The results in Table V and Fig. 7 indicate that the *Power*, *APP*, and *DPP* features remain robust under distribution shifts that do not significantly alter the received signal at the APs.



(a) *Power*. (b) *CSI fingerprinting*.

Fig. 7: Estimated trajectories for the case where the UE height has been decreased during inference from 1.5 m to 0.8 m.



(a) *Power*. (b) *CSI fingerprinting*.

Fig. 8: Estimated trajectories under a distribution shift of the data during testing time that includes a wooden wall in the top horizontal corridor.

V. Conclusions and Outlook

In this work, we have proposed DT-aided CC for self-supervised positioning in an uplink D-MIMO scenario. The proposed framework preserves both the local and global geometry of the estimated positions without requiring labeled data, accurate AP synchronization, affine transformations, or LoS area estimation of each AP. Our framework leverages a DT to compute real-time large-scale feature vectors that help train conventional CC and provides global coordinates. We have investigated several feature vectors and showed that the *Power*, *APP*, and *DPP* features outperform state-of-the-art results, with the *Power* feature providing an overall better performance and

the *APP* and *DPP* features more robustness in areas covered by only one AP. These results indicate that the proposed approach is an attractive alternative to CSI fingerprinting when a floor plan or DT of the environment is available. We have also investigated the impact of the density of predefined DT positions, demonstrating that the *Power* feature provides similar results with fewer computations from the DT. Additionally, we have compared the performance of the proposed framework under DT modeling mismatches during training with two DT-based fingerprinting strategies that rely on synthetic CSI and received power, respectively. We concluded that, for slight mismatches, DT-based CSI fingerprinting achieves the best performance at the cost of increased computational complexity in the DT, while the proposed approach achieves the best performance when a computationally efficient DT is harnessed. Additionally, the proposed approach is the most robust under severe DT mismatches. These results are the first step towards efficient, automatic, and real-time DT modeling, e.g., from pictures of the environment, without substantially degrading the positioning performance.

Finally, we have tested different datasets at inference time compared to the training data. We considered an unexpected decrease in the UE height or the introduction of reflecting elements in the environment. The results showcased that the proposed approach remains robust under distribution shifts that do not severely affect the large-scale features, still outperforming state-of-the-art methods. Under very severe distribution shifts, the proposed approach degrades in performance and fine-tuning with new collected unlabeled data must be performed to improve the overall positioning performance.

Accordingly, future research in this area can explore the continuous training of the proposed approach under changes in the environment and the integration of tracking algorithms to further improve positioning performance while estimating the uncertainty of the predictions. Moreover, the proposed approach can be validated on real hardware with real measurements, where the NN can also account for hardware impairments. Additionally, the proposed DT-based approach can be extended to other positioning modalities, e.g., matching features in camera images with simulated view scenes from the DT using ray-tracing algorithms.

ACKNOWLEDGMENTS

The authors would like to thank Remcom for providing a license for the Wireless InSite ray-tracing software. The authors also thank Sueda Taner for discussions on channel charting in real-world coordinates [27].

APPENDIX
EVALUATION METRICS

In what follows, we list the metrics that we used to assess the positioning performance of the proposed method and the baselines. We denote by $\mathcal{J}^{(n)}$ and $\hat{\mathcal{J}}^{(n)}$ the sets of the J closest true and estimated positions to $\mathbf{x}^{(n)}$ and $\hat{\mathbf{x}}_{\boldsymbol{\theta}}^{(n)}$, respectively. We denote the ranking of how close $\mathbf{x}^{(n)}$ is to $\mathbf{x}^{(j)}$ and how close $\hat{\mathbf{x}}_{\boldsymbol{\theta}}^{(n)}$ is to $\hat{\mathbf{x}}_{\boldsymbol{\theta}}^{(j)}$ according to the Euclidean distance as $r(n,j)$ and $\hat{r}(n,j)$, respectively. We define $\gamma = 2/(NJ(2N-3J-1))$ as a normalization factor [62]. We set $J = 0.05N$ as in [15].

1) Trustworthiness (TW): This metric penalizes close estimated positions that do not correspond to close true positions and it is defined as [63]

$$\mathrm{TW}(J) = 1 - \gamma \sum_{n\in\mathcal{N}} \sum_{\substack{j\notin\mathcal{J}^{(n)} \\ j\in\hat{\mathcal{J}}^{(n)}}} (r(n,j) - J). \tag{28}$$

TW takes values in $[0,1]$ with optimal value 1.

2) Continuity (CT): This metric penalizes close true positions that do not correspond to close estimated positions and it is defined as [63]

$$\mathrm{CT}(J) = 1 - \gamma \sum_{n\in\mathcal{N}} \sum_{\substack{j\in\mathcal{J}^{(n)} \\ j\notin\hat{\mathcal{J}}^{(n)}}} (\hat{r}(n,j) - J). \tag{29}$$

CT takes values in $[0,1]$ with optimal value 1.

3) Kruskal Stress (KS): This metric measures the dissimilarity between pairwise distances for true positions, defined as $d(n,j) = \|\mathbf{x}^{(n)} - \mathbf{x}^{(j)}\|$, and pairwise distances for the estimated positions, defined as $\hat{d}(n,j) = \|\hat{\mathbf{x}}_{\boldsymbol{\theta}}^{(n)} - \hat{\mathbf{x}}_{\boldsymbol{\theta}}^{(j)}\|$. KS is defined as [64]

$$\mathrm{KS} = \min_{\eta\in\mathbb{R}} \sqrt{\frac{\sum_{n,j\in\mathcal{N}} (\hat{d}(n,j) - \eta d(n,j))^2}{\sum_{n,j\in\mathcal{N}} \hat{d}(n,j)^2}}. \tag{30}$$

KS takes values in $[0,1]$ with optimal value 0.

4) Rajski Distance (RD): We denote as V and Q the discrete random variables representing the distribution of pairwise distances for the true positions $d(n,j)$ and the estimated positions $\hat{d}(n,j)$, respectively. The RD measures the difference between the mutual information and joint entropy of V and Q and it is defined as [65]

$$\mathrm{RD} = 1 - \frac{I(V,Q)}{H(V,Q)}, \text{for } H(V,Q) \neq 0, \tag{31}$$

where $I(V,Q)$ is the mutual information given by

$$I(V,Q) = \sum_{v\in V, q\in Q} P_{V,Q}(v,q) \log_2 \frac{P_{V,Q}(v,q)}{P_V(v) P_Q(q)} \tag{32}$$

and $H(Q,V)$ is the joint entropy given by

$$H(Q,V) = -\sum_{v\in V, q\in Q} P_{V,Q}(v,q) \log_2 P_{V,Q}(v,q). \tag{33}$$

In (32) and (33), $P_{V,Q}(v,q)$ is the joint distribution and $P_V(v)$ and $P_Q(q)$ are the marginal distributions of V and Q, respectively. The RD takes values in $[0,1]$ with optimal value 0. We quantize the pairwise distances for the true and estimated positions into 20 uniform bins.

5) Mean Distance Error (MDE): This metric measures the average error between the true and estimated positions and it is defined as

$$\mathrm{MDE} = \frac{1}{N} \sum_{n\in\mathcal{N}} \|\mathbf{x}^{(n)} - \hat{\mathbf{x}}_{\boldsymbol{\theta}}^{(n)}\|. \tag{34}$$

The MDE is nonnegative with optimal value 0.

6) 95th Percentile Distance Error (PDE): This metric measures the 95th percentile of the norm between the true and estimated positions. It is defined as the value ρ_{95} such that

at least 95% of the distance errors $\{\|\mathbf{x}^{(n)} - \hat{\mathbf{x}}_{\boldsymbol{\theta}}^{(n)}\|\}_{n\in\mathcal{N}}$ are less than ρ_{95}.

REFERENCES

[1] A. Behravan, V. Yajnanarayana, M. F. Keskin, H. Chen, D. Shrestha, T. E. Abrudan, T. Svensson, K. Schindhelm, A. Wolfgang, S. Lindberg, and H. Wymeersch, "Positioning and sensing in 6G: Gaps, challenges, and opportunities," *IEEE Veh. Technol. Mag.*, vol. 18, no. 1, pp. 40–48, Mar. 2023.

[2] S. Lu, F. Liu, Y. Li, K. Zhang, H. Huang, J. Zou, X. Li, Y. Dong, F. Dong, J. Zhu, Y. Xiong, W. Yuan, Y. Cui, and L. Hanzo, "Integrated sensing and communications: Recent advances and ten open challenges," *IEEE Internet Things J.*, vol. 11, no. 11, pp. 19 094–19 120, Jun. 2024.

[3] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of wireless indoor positioning techniques and systems," *IEEE Trans. Syst., Man, Cybern., C (Appl. Rev.)*, vol. 37, no. 6, pp. 1067–1080, Nov. 2007.

[4] A. Yassin, Y. Nasser, M. Awad, A. Al-Dubai, R. Liu, C. Yuen, R. Raulefs, and E. Aboutanios, "Recent advances in indoor localization: A survey on theoretical approaches and applications," *IEEE Commun. Surveys Tut.*, vol. 19, no. 2, pp. 1327–1346, Q2 2017.

[5] S. He and S.-H. G. Chan, "Wi-Fi fingerprint-based indoor positioning: Recent advances and comparisons," *IEEE Commun. Surveys Tut.*, vol. 18, no. 1, pp. 466–490, 2016.

[6] X. Wang, L. Gao, S. Mao, and S. Pandey, "CSI-based fingerprinting for indoor localization: A deep learning approach," *IEEE Trans. Veh. Technol.*, vol. 66, no. 1, pp. 763–776, Jan. 2017.

[7] J. Ott, J. Pirkl, M. Stahlke, T. Feigl, and C. Mutschler, "Radio foundation models: Pre-training transformers for 5g-based indoor localization," in *14th Int. Conf. Indoor Positioning Indoor Navigation (IPIN)*, 2024, pp. 1–6.

[8] A. Salihu, M. Rupp, and S. Schwarz, "Self-supervised and invariant representations for wireless localization," *IEEE Trans. Wireless Commun.*, vol. 23, no. 8, pp. 8281–8296, 2024.

[9] X. Gong, A.-A. Lu, X. Fu, X. Liu, X. Gao, and X.-G. Xia, "Semisupervised representation contrastive learning for massive MIMO fingerprint positioning," *IEEE Internet of Things Journal*, vol. 11, no. 8, pp. 14 870–14 885, 2024.

[10] N. Yu, S. Zhao, X. Ma, Y. Wu, and R. Feng, "Effective fingerprint extraction and positioning method based on crowdsourcing," *IEEE Access*, vol. 7, pp. 162 639–162 651, 2019.

[11] C. Xiang, S. Zhang, S. Xu, and G. C. Alexandropoulos, "Self-calibrating indoor localization with crowdsourcing fingerprints and transfer learning," in *IEEE Int. Conf. Commun.*, Virtual, 2021, pp. 1–6.

[12] X. Tong, Y. Wan, Q. Li, X. Tian, and X. Wang, "CSI fingerprinting localization with low human efforts," *IEEE/ACM Trans. Networking*, vol. 29, no. 1, pp. 372–385, 2021.

[13] H. Si, X. Hou, J. Wang, G. Owusu Boateng, Z. Zhang, X. Guo, and D. Niyato, "Unsupervised localization toward crowdsourced trajectory data: A deep reinforcement learning approach," *IEEE Trans. Wireless Commun.*, vol. 24, no. 9, pp. 7970–7984, 2025.

[14] "Smartphone-based localization for blind navigation in building-scale indoor environments," *Pervasive and Mobile Computing*, vol. 57, pp. 14–32, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1574119218303316

[15] C. Studer, S. Medjkouh, E. Gonultaş, T. Goldstein, and O. Tirkkonen, "Channel charting: Locating users within the radio environment using channel state information," *IEEE Access*, vol. 6, pp. 47 682–47 698, Aug. 2018.

[16] P. Ferrand, M. Guillaud, C. Studer, and O. Tirkkonen, "Wireless channel charting: Theory, practice, and applications," *IEEE Commun. Mag.*, vol. 61, no. 6, pp. 124–130, Jun. 2023.

[17] P. Huang, O. Castañeda, E. Gönültaş, S. Medjkouh, O. Tirkkonen, T. Goldstein, and C. Studer, "Improving channel charting with representation-constrained autoencoders," in *Proc. IEEE SPAWC*, Cannes, France, 2019, pp. 1–5.

[18] E. Lei, O. Castañeda, O. Tirkkonen, T. Goldstein, and C. Studer, "Siamese neural networks for wireless positioning and channel charting," in *Proc. Allerton Conf. Commun., Control, Comput.*, Monticello, VA, USA, 2019, pp. 200–207.

[19] Q. Zhang and W. Saad, "Semi-supervised learning for channel charting-aided iot localization in millimeter wave networks," in *Proc. IEEE GLOBECOM*, Madrid, Spain, 2021, pp. 1–6.

[20] J. Deng, O. Tirkkonen, J. Zhang, X. Jiao, and C. Studer, "Network-side localization via semi-supervised multi-point channel charting," in *Proc. IWCMC*, Harbin City, China, 2021, pp. 1654–1660.

[21] P. Q. Viet and D. Romero, "Implicit channel charting with application to uav-aided localization," in *Proc. IEEE SPAWC*, Oulu, Finland, 2022, pp. 1–5.

[22] V. Palhares, S. Taner, and C. Studer, "CSI2Vec: towards a universal csi feature representation for positioning and channel charting," *arXiv preprint arXiv:2506.05237*, 2025.

[23] 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.

[24] F. Euchner, P. Stephan, and S. ten Brink, "Uncertainty-aware dimensionality reduction for channel charting with geodesic loss," in *Proc. Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, USA, 2024, pp. 1665–1672.

[25] L. Zhao, Y. Yang, Q. Xiong, H. Wang, B. Yu, F. Sun, and C. Sun, "A signature based approach towards global channel charting with ultra low complexity," in *Proc. ICC Workshops*, Denver, CO, USA, 2024, pp. 667–672.

[26] P. Stephan, F. Euchner, and S. t. Brink, "Angle-delay profile-based and timestamp-aided dissimilarity metrics for channel charting," *IEEE Trans. Commun.*, vol. 72, no. 9, pp. 5611–5625, Sep. 2024.

[27] S. Taner, V. Palhares, and C. Studer, "Channel charting in real-world coordinates with distributed MIMO," *IEEE Trans. Wireless Commun.*, vol. 24, no. 9, pp. 7286–7300, Sep. 2025.

[28] O. Esrafilian, M. Ahadi, F. Kaltenberger, and D. Gesbert, "Global scale self-supervised channel charting with sensor fusion," *arXiv preprint*, 2024, arXiv:2405.04357.

[29] L. Cazzella, F. Linsalata, M. Maleki, D. Badini, M. Matteucci, and U. Spagnolini, "Chartwin: A case study on channel charting-aided localization in dynamic digital network twins," *arXiv preprint*, 2025, arXiv:2508.09055.

[30] J. Pihlajasalo, M. Koivisto, J. Talvitie, S. Ali-Löytty, and M. Valkama, "Absolute positioning with unsupervised multipoint channel charting for 5G networks," in *Proc. IEEE VTC-Fall*, Victoria, BC, Canada, 2020, pp. 1–5.

[31] M. Stahlke, G. Yammine, T. Feigl, B. M. Eskofier, and C. Mutschler, "Velocity-based channel charting with spatial distribution map matching," *IEEE J. Indoor Seamless Positioning Navig.*, vol. 2, pp. 230–239, Jul. 2024.

[32] Y. Zhang, G. Pan, M. F. Keskin, O. Kaltiokallio, M. Valkama, and H. Wymeersch, "UNILoc: Unified localization combining model-based geometry and unsupervised learning," *arXiv preprint*, 2025, arXiv:2504.17676.

[33] Y. Vindas and M. Guillaud, "Multi-site wireless channel charting through latent space alignment," in *Proc. IEEE SPAWC*, Lucca, Italy, 2024, pp. 826–830.

[34] F. Euchner, P. Stephan, and S. t. Brink, "Augmenting channel charting with classical wireless source localization techniques," in *Proc. Asilomar Conf. Signals Syst. Comput.*, Pacific Grove, CA, USA, 2023, pp. 1641–1647.

[35] F. Euchner, P. Stephan, and S. Ten Brink, "Leveraging the doppler effect for channel charting," in *Proc. IEEE SPAWC*, Lucca, Italy, 2024, pp. 731–735.

[36] I. Karmanov, F. G. Zanjani, I. Kadampot, S. Merlin, and D. Dijkman, "WiCluster: Passive indoor 2D/3D positioning using WiFi without precise labels," in *Proc. IEEE GLOBECOM*, Madrid, Spain, 2021, pp. 1–7.

[37] F. Tao, H. Zhang, A. Liu, and A. Y. C. Nee, "Digital twin in industry: State-of-the-art," *IEEE Trans. Ind. Inform.*, vol. 15, no. 4, pp. 2405–2415, Apr. 2019.

[38] F. G. Zanjani, I. Karmanov, H. Ackermann, D. Dijkman, S. Merlin, M. Welling, and F. Porikli, "Modality-agnostic topology aware localization," in *Proc. NeurIPS*, vol. 34, 2021, pp. 10 457–10 468.

[39] F. A. Aoudia, J. Hoydis, and A. Keller, "Fast and differentiable radio maps with NVIDIA instant RM," NVIDIA, Tech. Rep., 2024, [Online]. Available: https://developer.nvidia.com/blog/fast-and-differentiable-radio-maps-with-nvidia-instant-rm/.

[40] A. Goldsmith, *Wireless communications*. Cambridge university press, 2005.

[41] P. Ferrand, A. Decurninge, L. G. Ordoñez, and M. Guillaud, "Triplet-based wireless channel charting: Architecture and experiments," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 8, pp. 2361–2373, Aug. 2021.

[42] 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. IEEE ISWCS*, 2021, pp. 1–6.

[43] T. Yassine, L. L. Magoarou, S. Paquelet, and M. Crussière, "Leveraging triplet loss and nonlinear dimensionality reduction for on-the-fly channel charting," in *Proc. IEEE SPAWC*, Oulu, Finland, 2022, pp. 1–5.

[44] 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.

[45] N. Rahaman, A. Baratin, D. Arpit, F. Draxler, M. Lin, F. Hamprecht, Y. Bengio, and A. Courville, "On the spectral bias of neural networks," in *Proc. ICML*, vol. 97, Long Beach, CA, USA, 2019, pp. 5301–5310.

[46] Y. Cao, Z. Fang, Y. Wu, D.-X. Zhou, and Q. Gu, "Towards understanding the spectral bias of deep learning," in *Proc. IJCAI*, Montreal, QC, Canada, 2021, pp. 2205–2211.

[47] Z. J. Xu, Y. Zhang, L. Tao, Y. Xiao, and Z. Ma, "Frequency principle: Fourier analysis sheds light on deep neural networks," *Commun. Comput. Phys.*, vol. 28, no. 5, pp. 1746–1767, 2020.

[48] B. Barz and J. Denzler, "Deep learning on small datasets without pre-training using cosine loss," in *Proc. IEEE/CVF Conf. Appl. Comput. Vis.*, Seattle, WA, USA, March 2020.

[49] J. T. Hoe, K. W. Ng, T. Zhang, C. S. Chan, Y.-Z. Song, and T. Xiang, "One loss for all: Deep hashing with a single cosine similarity based learning objective," in *Proc. NeurIPS*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34. Curran Associates, Inc., 2021, pp. 24 286–24 298.

[50] M. Kuczma, *An introduction to the theory of functional equations and inequalities: Cauchy's equation and Jensen's inequality*. Springer, 2009.

[51] A. J. Laub, *Matrix analysis for scientists and engineers*. SIAM, 2004.

[52] D. Jones, C. Snider, A. Nassehi, J. Yon, and B. Hicks, "Characterising the digital twin: A systematic literature review," *CIRP Journal of Manufacturing Science and Technology*, vol. 29, pp. 36–52, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1755581720300110

[53] J. Li, W. Gao, Y. Wu, Y. Liu, and Y. Shen, "High-quality indoor scene 3D reconstruction with RGB-D cameras: A brief review," *Computational Visual Media*, vol. 8, no. 3, pp. 369–393, 2022.

[54] Z. Keqiang, Z. Yan, and W. Wei, "Automated 3D scenes reconstruction for mobile robots using laser scanning," in *Chinese Control and Decision Conference*, Guilin, China, 2009, pp. 3007–3012.

[55] *Material Property Determination of Vibration Fatigued DMLS and Cold-Rolled Nickel Alloys*, ser. Turbo Expo, vol. Volume 7A: Structures and Dynamics, 06 2014. [Online]. Available: https://doi.org/10.1115/GT2014-26247

[56] P. K. Majumdar, M. FaisalHaider, and K. Reifsnider, *Multi-physics Response of Structural Composites and Framework for Modeling Using Material Geometry*. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.2013-1577

[57] C. Bae, E. Choi, and S. Lee, "Technologies, applications, and challenges of digital twin across industries: A systematic review of the state-of-the-art literature," *IEEE Access*, vol. 13, pp. 152 843–152 869, 2025.

[58] S. Taner, M. Guillaud, O. Tirkkonen, and C. Studer, "Channel charting for streaming CSI data," in *Asilomar Conf. Signals, Systems, Computers*, Pacific Grove, CA, USA, 2023, pp. 1648–1653.

[59] R. Wiesmayr, F. Zumegen, S. Taner, C. Dick, and C. Studer, "CSI-based user positioning, channel charting, and device classification with an NVIDIA 5G testbed," *Proc. Asilomar Conf. Signals, Systems, Computers*, 2025.

[60] R. W. Bohannon and A. Williams Andrews, "Normal walking speed: a descriptive meta-analysis," *Physiotherapy*, vol. 97, no. 3, pp. 182–189, 2011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031940611000307

[61] J. Morais and A. Alkhateeb, "Localization in digital twin mimo networks: A case for massive fingerprinting," in *IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, Denver, CO, USA, 2024, pp. 276–281.

[62] H. Al-Tous, P. Kazemi, C. Studer, and O. Tirkkonen, "Channel charting with angle-delay-power-profile features and earth-mover distance," in *Proc. Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, USA, 2022, pp. 1195–1201.

[63] Á. Vathy-Fogarassy and J. Abonyi, *Graph-based clustering and data visualization algorithms*. Berlin, Germany: Springer, 2013.

[64] J. B. Kruskal, "Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis," *Psychometrika*, vol. 29, no. 1, pp. 1–27, 1964.

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