Chapter

Self-Supervised Learning for Wireless Localization

Artan Salihu, Markus Rupp and Stefan Schwarz

Abstract

In this chapter, we provide an overview of several data-driven techniques for wireless localization. We initially discuss shallow dimensionality reduction (DR) approaches and investigate a supervised learning method. Subsequently, we transition into deep metric learning and then place particular emphasis on a transformer-based model and self-supervised learning. We highlight a new research direction of employing designed pretext tasks to train AI models, enabling them to learn compressed channel features useful for wireless localization. We use datasets obtained in massive multiple-input multiple-output (MIMO) systems indoors and outdoors to investigate the performance of the discussed approaches.

Keywords: wireless localization, dimensionality reduction, massive MIMO, metric learning, transformer, self-supervised, deep learning, CSI, channel representations

1. Introduction

As we advance toward the sixth-generation (6G), the capabilities of mobile communication networks are anticipated to evolve far beyond their present role of connecting individuals or machines [1, 2]. Among different foreseen application scenarios, wireless localization, sensing, and artificial intelligence (AI) are the most critical ones for advanced future communication systems [2]. AI, in more general terms, incorporates a range of techniques, primarily machine learning (ML) methods, which allow machines to *learn* from data and prior experiences. As such, AI-enhanced physical layer (PHY) may allow adaptive data-driven decisions for various signal-processing parts in the communication chain. Integrating AI with wireless localization presents an opportunity to optimize advanced communication systems to leverage the surroundings better. This integration can improve their ability to accurately locate the users, allow context-aware transmission, and more efficiently utilize processing and energy resources.

1.1 Wireless localization with deep learning

Wireless localization methods can be divided into two categories: model-based and data-driven [3]. Model-based techniques require knowledge of the geometric relationship between the estimated parameters of the received signal and the position of the transmitter [4]. Therefore, the performance of the existing model-based methods is heavily degraded when such a relationship is not available, for example, in non-line-of-sight (NLOS) or dense multi-path propagation conditions [4–7]. Consequently, data-driven approaches, specifically ML, have emerged over the years [8]. Among the various ML approaches, deep neural network (DNN)-based models have demonstrated exceptional localization accuracy [9–15].

A common approach in cellular-based localization systems is to use derived features of the estimated channel at the base station (BS), such as signal time of arrival (ToA), angle of arrival (AoA), received signal strength (RSS), or a combination of them [6, 16, 17]. On the other hand, DNN-based methods prefer utilizing the channel state information (CSI) in its acquired form [9, 10, 13, 18, 19]. The acquisition of CSI is particularly important for advanced communication systems that use massive multiple-input multiple-output (MIMO) [20, 21]. Hence, CSI is, in general, readily available for localization. Massive MIMO, characterized by a large antenna array at the BS, is widely regarded as the key technology for the fifth-generation (5G) [22, 23] and forthcoming communication systems [24]. Employing a considerable number of antennas enhances the angular resolution of the received multipath signal, thereby benefiting localization methods [25–27].

ML-based methods can generally be supervised or unsupervised, depending on whether labeled training data are necessary. A vast majority of DNN localization techniques are supervised and constrained to a task-specific feature learning, raising concerns particularly about the ability to work across diverse scenarios and adapt in data-scarce environments. On the other hand, developing DNN methods that sustain good accuracy and transferability is an essential research topic for future wireless localization systems.

In this chapter, we cover three approaches for wireless localization. We investigate conventional dimensionality reduction (DR) techniques, like PCA and basic manifold approaches. Afterward, we discuss a supervised learning method that uses multi-layer perceptrons (MLP). Furthermore, we extend the basic MLP into a deep metric learning method. Finally, we transition into advanced DNN architectures such as transformers, combined with sophisticated learning frameworks like self-supervised learning, emphasizing its growing research importance in wireless localization and beyond.

Specifically, we structure the rest of this chapter as follows. First, in Section 2, we outline the system model to generate synthetic datasets for validating DNNs. In Section 3, we investigate classical DR approaches. Then, in Section 4, we discuss supervised DNN methods and deep metric learning. In Section 5, we elaborate on more advanced neural network architectures and learning frameworks. Finally, we draw our conclusions in Section 6.

2. System model

We consider a massive MIMO uplink setup as illustrated in **Figure 1**. The base station has N_r antenna elements. In addition to the co-located MIMO setup, we also consider a distributed antenna system (DAS) with N_r antennas spatially distributed across M remote radio heads (RRHs), positioned at $\mathbf{b}_m = [b_{m,1}, b_{m,2}, b_{m,3}]^T$, where $m \in \{1, \dots, M\}$. For a DAS, we assume that every RRH is connected to the central unit (CU) via high-speed fronthaul links, implying no synchronization delays between the RRHs and the central unit (CU). There are R single-antenna user locations at

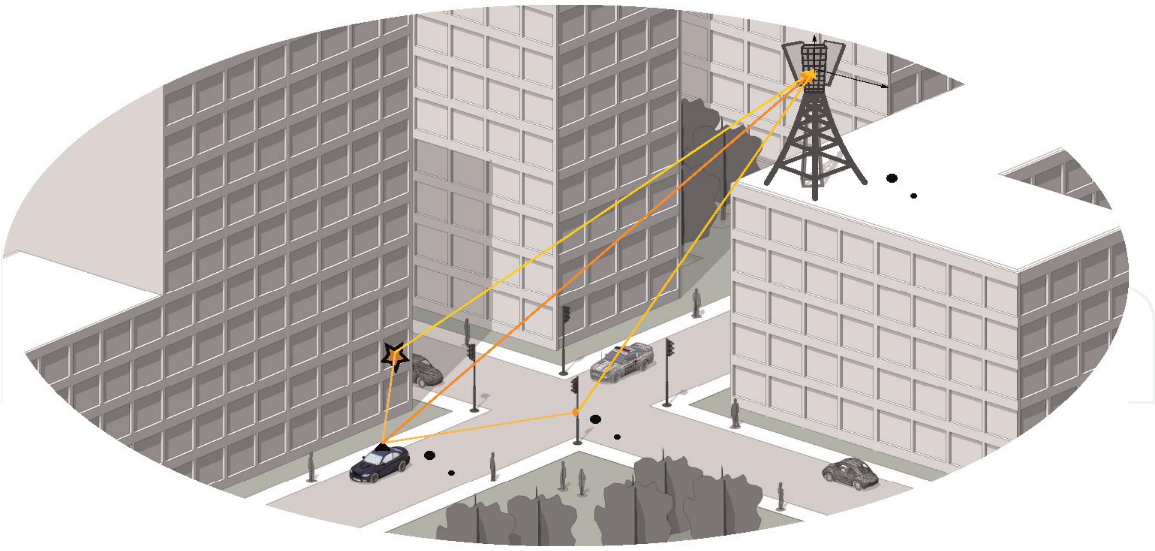


Figure 1.
Example scenario considered for the system model in a co-located massive MIMO setup.

$\mathbf{u}_r = [u_{r,1}, u_{r,2}, u_{r,3}]^T$, where $r \in \{1, \dots, R\}$. Furthermore, we account for G scatterers in the region of interest (ROI) with positions at $\mathbf{p}_g = \left[p_{g,1}, p_{g,2}, p_{g,3}\right]^T$, where $g \in \{1, \dots, G\}$.

2.1 Signal model

The uplink input-output relationship of a single-user transmission from $\mathbf{u}_r$ on subcarrier $n \in \{1, \dots, N_c\}$ is

$$\mathbf{y}_n = \sqrt{P_{\text{tx}}}\tilde{\mathbf{h}}_n x_n + \mathbf{n}_n. \tag{1}$$

Here, P_{tx} is the average transmit power, x_n is the normalized transmitted signal with $|x_n|^2 = 1$, $\tilde{\mathbf{h}}_n \in \mathbb{C}^{N_r \times 1}$ is the channel vector, and $\mathbf{n}_n \sim \mathcal{CN}(\mathbf{0}, N_0 W_{\text{sc}} \mathbf{I}_{N_r})$, where N_0 is the noise power spectral density and W_{sc} is the subcarrier bandwidth. We assume that the BS is able to estimate the channel vectors $\tilde{\mathbf{h}}_n$ through uplink pilot signals. We denote the estimated CSI as $\mathbf{h}_n$.

2.2 Channel model

The DNN techniques detailed in this chapter are data-driven and therefore are not constrained to a particular channel model. Nevertheless, it is beneficial to clarify the channel using location-related parameters like distances and angles for investigation. Consequently, we utilize a commonly adopted geometric channel model to characterize the estimated channel of the received signal for such scenarios as that illustrated in **Figure 1** [28],

$$\tilde{\mathbf{h}}_n = \sum_{g=1}^{G} \eta_g e^{j2\pi n \Delta f \tau_g} \mathbf{a}\left(\varphi_{az,g}, \varphi_{el,g}\right), \tag{2}$$

where η_g and τ_g represent the complex gain and propagation delay (ToA) of the $g-$th path between the $r-$th user equipment (UE) position and the $m-$th BS. The angles of arrival (AoA) for azimuth and elevation are represented as $\varphi_{az,g}$ and $\varphi_{el,g}$, respectively, while the subcarrier spacing is denoted by Δf. Assuming a uniform planar array with antennas arranged along the $x-$ and $z-$axis, the array response vector at the receiver is expressed as $\mathbf{a}(\varphi_{\mathrm{az}}, \varphi_{\mathrm{el}}) = \mathbf{a}_z(\varphi_{\mathrm{el}}) \otimes \mathbf{a}_x(\varphi_{\mathrm{az}}, \varphi_{\mathrm{el}})$, where $\mathbf{a}_x(\cdot)$ and $\mathbf{a}_z(\cdot)$ are given as

$$\begin{aligned}\mathbf{a}_x(\varphi_{\mathrm{az}}, \varphi_{\mathrm{el}}) &= \left[1, e^{j\frac{2\pi}{\lambda_c} d \sin(\varphi_{\mathrm{el}}) \sin(\varphi_{\mathrm{az}})}, \dots, e^{j\frac{2\pi}{\lambda_c} d (M_x - 1) \sin(\varphi_{\mathrm{el}}) \sin(\varphi_{\mathrm{az}})}\right]^T, \\ \mathbf{a}_z(\varphi_{\mathrm{el}}) &= \left[1, e^{j\frac{2\pi}{\lambda_c} d \cos(\varphi_{\mathrm{el}})}, \dots, e^{j\frac{2\pi}{\lambda_c} d (M_z - 1) \cos(\varphi_{\mathrm{el}})}\right]^T.\end{aligned} \tag{3}$$

Here, $\lambda_c = c_l / f_c$, where f_c is the carrier frequency, and c_l denotes the speed of light. The antenna element spacing is given by $d = \lambda_c/2$.

2.3 Dynamic scenario

We consider that the environment may change over the time. When using synthetic data to evaluate the proposed methods in this chapter, we account for situations where some scattering objects change their positions over the time interval T, assuming, for example, a dynamic setting [29]. We formulate $p_{g,i}^{(t)} = p_{g,i} + w_{g,i}$, with $w_{g,i}$ representing the Gaussian noise with zero-mean and variance σ_w^2 at the $i-$th coordinate. Doing so leads to a non-additive distortion model [24], where the angle, delay, and channel gain of the individual multi-bounce NLOS paths change for each channel realization. We also assume the uncertainty in the position of antenna for the UE, modeled as $u_{r,i}^{(t)} \triangleq u_{r,i} + \dot{w}_{r,i}^{(t)}$. Similar to the uncertainty in the position of the scattering objects, $\dot{w}_{g,i}$ is the Gaussian noise with variance $\sigma_{\dot{w}}^2$ at $i-$th coordinate. The variability in antenna position serves as a means to address potential imperfect channel estimates and other system impairments, which may arise from, for instance, a lack of perfect synchronization between the UE and the BS.

Next, we stack $N_{c'} \leq N_c$ subcarriers in a matrix form for the $r-$th UE location,

$$\mathbf{H}_r^{(t)} = \left[\mathbf{h}_1^{(t)}, \mathbf{h}_2^{(t)}, \dots, \mathbf{h}_{N_c'}^{(t)}\right] \in \mathbb{C}^{N_r \times N_c'}, \tag{4}$$

where N_c' may correspond to pilot subcarriers. The input to a neural network-based model is real-valued; hence, we handle complex-valued channel coefficients by stacking their real and imaginary parts and, in some of our architectures, their absolute values. Throughout the chapter, we omit the index t from the expressions as we consider a single-time snapshot in the algorithms presented in here. Similarly, we drop r if not relevant to the discussion or analysis. Finally, the localization performance in this chapter is reported in terms of the root-mean squared error, RMSE,

$$\mathrm{RMSE} \quad = \sqrt{\frac{\sum_{r'=1}^{R_{\mathrm{test}}} \|\hat{\mathbf{u}}_{r'} - \mathbf{u}_{r'}\|^2}{R_{\mathrm{test}}}}, \tag{5}$$

where R_{test} denotes the number of test locations, and $\hat{\mathbf{u}}_{r'}$ is the estimated UE location.

3. Shallow DR techniques

In this section, we discuss dimensionality reduction approaches to obtain a channel representation useful for positioning the UE. Specifically, we first investigate shallow approaches based on principle component analysis and iterative scaling.

3.1 Principle component analysis and iterative scaling

Since principle component analysis (PCA) is the most widely used technique for lossy data compression or feature extraction, we first illustrate the efficacy of the PCA-derived channel subspace when used to determine the UE location. To do so, let us consider a single-path line-of-sight (LOS) channel, that is, $G = 1$ in (2). We also consider a single subcarrier $n = 1$ and assume a uniform linear array (ULA) at the BS. Therefore, we assume that the channel coefficients of $\mathbf{h}_r$ are dependent only on the distance $\|\mathbf{b}_m - \mathbf{u}_r\|$ and the direction of arrival φ_{az}. Furthermore, we consider that a dataset from R distinct UE locations is available, construct an $R \times N_r$ matrix,

$$\mathbf{H}_{\text{ref}} = \begin{bmatrix} \mathbf{h}_1^T \\ \mathbf{h}_2^T \\ \vdots \\ \mathbf{h}_R^T \end{bmatrix}, \tag{6}$$

and center it, that is, $\overline{\mathbf{H}}_{\text{ref}} = \mathbf{H}_{\text{ref}} - \mathbf{1}\left(\frac{1}{R}\mathbf{H}_{\text{ref}}^T\mathbf{1}\right)^T$ is the zero-mean sample channel matrix, and $\mathbf{1}$ is a column vector of ones.

Our goal is to construct a representation map of (pseudo-)locations, $\mathcal{Z} = \left\{\mathbf{z}_r \in \mathbb{R}^D\right\}_{r=1}^R$, only from the high-dimensional CSI samples. The derived map can be thought of as extracting a low-dimensional representation of UEs spatial information from corresponding high-dimensional channel vectors $\{\mathbf{h}_r\}_{r=1}^R$. We can compute the PCA to get a low-rank approximation of the channel from R locations by eigenvalue decomposition on the empirical covariance matrix $\mathbf{R}_{\mathbf{H}_{\text{ref}}} = \mathbf{U}\Sigma\mathbf{U}^T$, where $\mathbf{U}$ are the left singular vectors of $\overline{\mathbf{H}}_{\text{ref}}$ and the eigenvectors of $\mathbf{R}_{\mathbf{H}_{\text{ref}}}$. Similarly, Σ are the eigenvalues of $\mathbf{R}_{\mathbf{H}_{\text{ref}}}$, that is, squared singular values of $\overline{\mathbf{H}}_{\text{ref}}$. The low-dimensional map is considered as

$$\mathbf{Z}_{\text{PCA}} = \left[\sqrt{\sigma_1}\mathbf{u}_1, \dots, \sqrt{\sigma_D}\mathbf{u}_D\right], \tag{7}$$

matching the first D largest eigenvalues from $\{\sigma_1, \dots, \sigma_R\}$ with the corresponding eigenvectors $\{\mathbf{u}_r\}_{r=1}^R$.

As an optimization problem, PCA is closely related to the so-called multidimensional scaling (MDS) [30], or metric MDS. MDS is another reconstruction method to obtain a representation *map* based on the dissimilarities between the points. This is particularly appropriate when such dissimilarities can, exactly or approximately, be expressed in terms of the Euclidean distances. Classical scaling, that is, metric MDS, like PCA, results in maximizing the variance in a reduced-dimensional space, which is a widely recognized relationship between the two [31, 32].

In addition to using PCA to construct a low-dimensional map representation by optimizing a convex objective using eigendecomposition, we next look into an alternative to classical scaling, which is known as MDS with Sammon mapping [33]. Sammon mapping is an iterative, gradient-based approach to evaluate a non-convex objective function and is considered a generalization of metric MDS. In this case, we seek to find an optimal representation by normalizing the squared errors using the pairwise distance in the original features space. To obtain the channel features while reducing the distance between low- and high-dimensional representations, we minimize the cost function.

$$\mathcal{L}(\mathbf{Z}_{\text{MDS}}) = \sum_{i=1}^{R-1} \sum_{j=i+1}^{R} \frac{\left(d_{i,j}^2 - \|\mathbf{z}_i - \mathbf{z}_j\|^2\right)}{d_{i,j}^2}. \tag{8}$$

Assuming a subset of CSI has labels, its low-dimensional features can serve as either a reference map or an input to a task-specific model to derive the final location of the UEs. Below, we aim to capture the CSI manifold and investigate the localization performance when CSI is projected onto, for instance, a $D = 2$ dimensional channel map. Provided such maps only yield pseudo-locations, we use a matching algorithm like k-nearest neighbors (k-NN) to compare a new transmitter's channel features to those of known locations in the reference map. Specifically, we form a distance metric between the features of the unknown location $\mathbf{z}_{r'}$ and all reference locations $\mathbf{z}_r$ for each r as

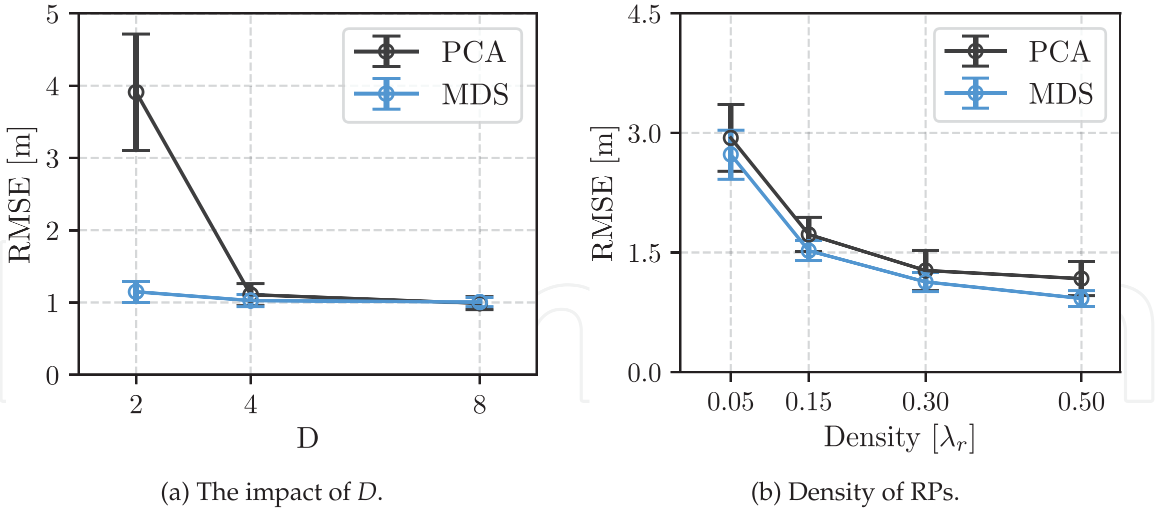
$$d_{\mathbf{z}_r} = \|\mathbf{z}_{r'} - \mathbf{z}_r\|, \tag{9}$$

and choose $r_{NN} = \arg\min_r d_{\mathbf{z}_r}$. Finally, the known location with the minimum distance is selected as the estimated position of the transmitter, $\hat{\mathbf{u}}_{r'} = \mathbf{u}_{r_{NN}}$.

Acquiring the reference map poses a challenge in determining the optimal number of low-dimensional features, D. In terms of minimum error formulation, using $D = \text{rank}(\mathbf{H}_{\text{ref}})$ would, naturally, result in a reconstruction error of zero. However, this does not necessarily ensure a minimized localization error. Furthermore, our objective is to attain $D \ll N_r$. A straightforward approach to select D is to quantify the fraction of variance explained by the selected eigenvalues in $\overline{\mathbf{H}}_{\text{ref}}$, that is, $\sum_{i=1}^{D}\sigma_i / \sum_{i'=1}^{R}\sigma_{i'}$, for example, choosing D to capture 90% of the energy. Alternatively, we can directly evaluate the localization error while sweeping over the value of D. To illustrate the capability of classical DR techniques in retaining low-dimensional features useful for the localization task, in the following, we describe a numerical experiment to evaluate the impact of D.

3.2 Location estimation using D-dimensional features

We set up a simple two-dimensional ROI with a layout of 40m × 40m and assume that the user locations are distributed based on a Poisson point process (PPP) with the density λ_r. Hence, the probability of having R users in the plane is $\mathbb{P}(R, \mathcal{A}) = [\lambda_r S(\mathcal{A})]^r e^{-\lambda_r S(\mathcal{A})}/R!$ [34], where $S(\mathcal{A})$ is the area of the bounded ROI, $\mathcal{A}$. We place the BS at the origin $(0, 0)$ with respect to $y-$axis. The BS employs a large ULA with $N_r = 64$. The carrier frequency considered is 3.5 GHz; the pathloss exponent is $\rho = 2$, and the SNR is set to 30 dB to account for a channel estimation error. We assume that



(a) The impact of D.

(b) Density of RPs.

Figure 2.
Shallow DR techniques in a pure LOS. In (a), the impact of D is evaluated while $\lambda_r = 0.5$, *and (b) shows the performance accuracy with increasing density of reference locations when* $D = 4$.

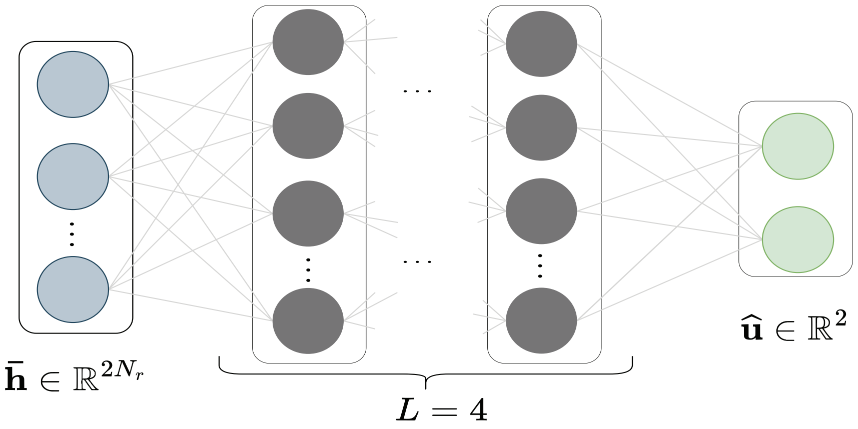

$\bar{\mathbf{h}} \in \mathbb{R}^{2N_r}$
$L = 4$
$\hat{\mathbf{u}} \in \mathbb{R}^2$

Figure 3.
A basic MLP, termed as the base DNN throughout the chapter, illustrated in the role of a regressor.

there are $\lambda_r = \{0.05, 0.15, 0.30, 0.50\}$ users/m^2 and keep 100 user locations for testing data. In **Figure 2a**, we observe that the performance of PCA improves significantly when increasing D from 2 to 4, but then for larger D, it saturates. On the other hand, the iterative MDS with Sammon mapping approach is more robust against changes in the dimensionality, maintaining a similar performance irrespective of the value of D. Yet, the accuracy hardly improves for $D > 4$. We also investigate the localization accuracy while varying the density of reference points (RPs) and keeping $D = 4$. As one can expect, and as shown in **Figure 2b**, increasing the density of known locations improves the localization performance for both PCA and MDS.

In the next section, we discuss wireless localization with neural networks in general and deep metric learning in particular.

4. Basic DNN methods and deep metric learning

Numerous works propose supervised training methods for CSI-based localization, where the utilized channel features are labeled with a corresponding target location information (e.g., position coordinates of the transmitter) [10, 13, 18, 19, 35–37].

These techniques establish a mapping function between the obtained CSI and the corresponding position coordinates, intending to achieve accurate localization performance on new, unseen data.

In **Figure 3**, we show a straightforward and relatively low-complexity localization method based on a feedforward neural network, that is, an MLP. Throughout this section of the chapter, we will refer to it as the base DNN, as we will use it to compare it with a metric-learning approach or as a basis for the proposed model. In the following, we set the number of hidden layers to $L = 4$ and the number of units for each hidden layer to 650. Selecting the number of hidden layers and units is more of an experimental decision than one grounded in theory. We make adjustments often based on empirical performance and computational considerations. The standard components (i.e., the input layer and the activation functions) of the fully connected feedforward MLP require fixed-size, flattened real-valued inputs. Hence, for the base DNN, we treat the complex-valued channel as two independent real numbers and average over subcarriers to maintain a lower complexity of the DNN. Therefore, the channel for the input layer is $\overline{\mathbf{h}}_r \in \mathbb{R}^{2N_r}$. Selecting a single subcarrier is typically sufficient representation in the presence of a strong LOS path and large coherence bandwidth. In a more pronounced selective fading channel, averaging over N_c subcarriers can potentially smooth out the effects of deep fading. However, one should note that both approaches inevitably result in information loss. Hence, we consider the whole estimated channel for the advanced models presented later in the chapter.

The wireless localization problem for supervised models is often formulated as a regression task. In such a case, we consider the base DNN as a function $f^{(\text{Base})}_{\Psi_b} : \mathbb{R}^{2N_r} \mapsto \mathbb{R}^2$ parameterized by Ψ_b. Given the input of DNN is the channel state vector $\overline{\mathbf{h}}_r$, the goal is to directly map it into position information, $\mathbf{u}_r$. For a training dataset of R sample pairs, the set of optimal parameter values Ψ_b is learned by minimizing a given loss function, $\mathcal{L}_b(\cdot)$. Usually, the supervised training for the regression is performed to minimize the sum of squared errors,

$$\mathcal{L}_b\left(\mathbf{u}_r, \overline{\mathbf{h}}_r, \Psi_b\right) = \arg\min_{\Psi} \mathbb{E}\left[\left\|\mathbf{u}_r - f^{(\text{Base})}_{\Psi_b}\left(\overline{\mathbf{h}}_r\right)\right\|^2\right], \tag{10}$$

estimated by averaging over the batch of training samples.

Alternatively, we consider a classifier that can learn to separate R_{rps} locations from which we obtain the channel by employing the cross-entropy loss,

$$\mathcal{L}\left(\mathbf{y}, \overline{\mathbf{h}}, \Psi_b\right) = \arg\min_{\Psi_b} \mathbb{E}\left[-\sum_{i=1}^{R_{\text{rps}}} \mathbf{y}_i \log f^{(\text{Base})}_{\Psi_b}\left(\overline{\mathbf{h}}_i\right)\right]. \tag{11}$$

Here, $f^{(\text{Base})}_{\Psi_b}$ is modified to behave as a classifier rather than a regressor. More specifically, instead of an identity layer function at the last layer (i.e., a linear activation), we use a softmax activation function at the last layer. Hence, the output values for $\mathbf{z}_i = f^{(\text{Base})}_{\Psi_b}\left(\overline{\mathbf{h}}_i\right)$ in (11) fall within the range of zero to one. These are the normalized probability *scores* corresponding to the predicted RP location. The number of units for the output layer is equivalent to the number of RP locations, R_{rps}. On the other hand, the true labels, corresponding to R_{rps} locations, are one-hot encoded, that is, $\mathbf{y}_i \in \{0, 1\}^{R_{\text{rps}}}$.

In general, supervised DNN-based methods excel in any task where extensive labeled datasets are readily available for training. However, collecting large-scale geo-tagged channel estimates for different mobile network tasks can be time-consuming, error-prone, and, in many cases, impractical. Thus, in the next section, we discuss more advanced methods that can learn channel features from unlabeled data.

4.1 Metric learning DNN and contrastive task

In contrast to the DR approaches we discussed in Section 3, in deep metric learning, we obtain an embedding space using neural networks. Consequently, the objective of the DNN becomes to learn a D-dimensional embedding that discriminates dissimilar point clouds while simultaneously bringing similar ones closer together. The DNN can map the estimated channel into a metric-defined subspace. Within this space, any distance metric $d(.,.) : \mathbb{R}^D \to \mathbb{R}$ can be applied to match the derived embedding with the closest RP. Furthermore, we can also incorporate a distance metric-based loss function directly in the embedding space. Among the most recognized deep metric learning methods are those based on the Siamese networks [38]. Such methods have been proposed as feature extractors across different domains, like the works extracting features from images of faces [39]. DNN methods, founded on Siamese architecture, consist of multiple equivalent networks sharing identical weights. Similarly, we introduce a Siamese-based network for obtaining low-dimensional channel representations useful for localization [40], depicted in **Figure 4**.

The number of networks used in a Siamese-based method can be any. However, it is usually two or three. A Siamese architecture with three equivalent networks is also called a triplet. Hence, the name triplet network for the model. Specifically, the network is composed of three branches. Each network employs the same hyperparameters as the base DNN for the intermediate computation layers. For the final layer (i.e., the output), the identity function is utilized, $\sigma(\mathbf{z}_i) = \mathbf{z}_i$. The goal of the *triplet network* is to learn an embedding in the way that the parameters of each network result in the decrease of variance between the channel obtained within a sub-region and increase otherwise. In other words, our goal is to find an objective function that promotes similarity between the embeddings of two CSI samples acquired from the same region $r \in \{1, \dots, R_{\text{rps}}\}$ while pushing apart embeddings corresponding to CSI samples from different regions.

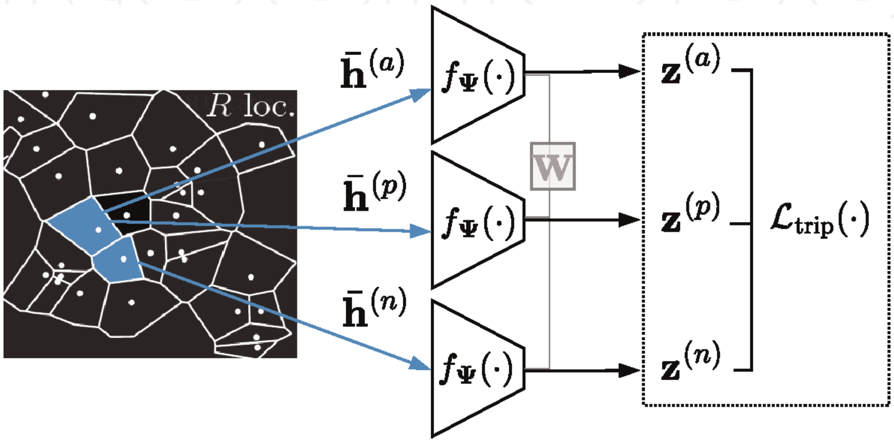


Figure 4.
Illustration of the triplet-based DNN model.

4.1.1 Contrastive task

In order to be able to design the contrastive task and therefore sample channels from different regions, we partition the ROI into RPs's sub-regions. We sample similar and non-similar CSI; that is, if two CSI vectors correspond to the same sub-region, they are considered similar; otherwise, they are not similar. For obtaining the triplets, we consider every $r \in \{1, \dots, R_{\text{rps}}\}$ as the anchor node, denoted by $\overline{\mathbf{h}}_i^{(a)}$. Then, we sample one neighboring channel realization belonging to the same region, that is, a positive sample, $\overline{\mathbf{h}}_i^{(p)}$. Finally, for the negative (not similar CSI), we only need to randomly select a channel vector that does not belong to the same quantized location as the anchor node, that is, the negative sample, denoted by $\overline{\mathbf{h}}_i^{(n)}$. For large-scale scenarios and/or very high density of RPs, it might not be computationally efficient to sample all the possible triplets. Therefore, one could sample only a portion of the triplets for the sake of the training process. Finally, the obtained triplets for training are given as $\left\{\left(\overline{\mathbf{h}}_i^{(a)}, \overline{\mathbf{h}}_i^{(p)}, \overline{\mathbf{h}}_i^{(n)}\right)\right\}_{i=1}^{R_{\text{tri}}}$, where R_{tri} is the number of triplets providing the triplet network for training.

Since all three networks illustrated in **Figure 4** share the same weights, we implement the three branches using a single network, the base DNN. During the training phase, we successively feed the channel realizations within each triplet. Consequently, we obtain their respective embeddings $\mathbf{z}_i^{(a)} = f_\Psi\left(\overline{\mathbf{h}}_i^{(a)}\right)$, $\mathbf{z}_i^{(p)} = f_\Psi\left(\overline{\mathbf{h}}_i^{(p)}\right)$, and $\mathbf{z}_i^{(n)} = f_\Psi\left(\overline{\mathbf{h}}_i^{(n)}\right)$. By collecting all the possible triplets from the ROI under investigation, we then minimize the loss function,

$$\mathcal{L}_{\text{tri}}\left(\mathbf{z}_i^{(a)}, \mathbf{z}_i^{(p)}, \mathbf{z}_i^{(n)}, \Psi\right) = \arg\min_\Psi \sum_{i=1}^{R_{\text{tri}}} \left[\|\mathbf{z}_i^{(a)} - \mathbf{z}_i^{(p)}\|^2 - \|\mathbf{z}_i^{(a)} - \mathbf{z}_i^{(n)}\|^2 + \delta_{\text{tri}}\right]_+, \tag{12}$$

where δ_{tri} enforces a margin between the similar and non-similar pairs, and $[.]_+ = \max(0, .)$. This triplet margin, δ_{tri} parameter, is tuned during the training process, and for most of our experiments, $\delta_{\text{tri}} = 0.2$ yields the best results. During the testing phase, the channel realization of a new UE is passed through the network to obtain an embedding, $\mathbf{z}_{r'}$. In this case, given that a new UE appears in the ROI, we evaluate $d_{\mathbf{z}_r} = \|\mathbf{z}_{r'} - \mathbf{z}_r\|$. Then, we choose $r_{NN} = \arg\min_r d_{\mathbf{z}_r}$. Finally, the location of the RP corresponding to the most similar embedding is retrieved and considered as the position of the transmitter, $\hat{\mathbf{u}}_{r'} = \mathbf{u}_{r_{NN}}$.

To evaluate the performance gains of the triplet network, we compare it to the supervised classifier discussed in Section 4.

4.1.2 Impact of LOS and density of reference locations

In the simulations conducted, we choose the model that demonstrates the lowest validation loss over 100 epochs. If the LOS is not present, the margin of the triplet loss during the training process is adjusted to $\delta_{\text{tri}} = 10$. We consider 80% of the dataset for training and the remaining for validation and testing. The scenario covers a 40m × 40m area with the BS positioned at the origin $(0, 0)$, relative to the $y-$ axis. Users are spread in the far field, at least 20m away from the BS. The NLOS paths are set to $G = 3$, and the total subcarriers are $N_c' = 512$.

The desired positioning accuracy is primarily determined by the density of RPs. Increasing RPs, the system's ability for higher accuracy increases, too. However, the presence or the absence of a LOS path in a multipath channel also influences the accuracy of the prediction. Consequently, it limits the accuracy even when a high density of RPs is used.

In **Figure 5**, we show the impact of LOS with the increased density of RPs for both DNNs, the classifier and the triplet network. In the presence of a LOS path, a higher density of RPs enhances the overall location estimation accuracy. The triplet network surpasses the base DNN classifier when $\lambda_r > 0.05$. Conversely, in the absence of the LOS path, a denser RPs distribution results in a degradation of the localization accuracy compared to scenarios where the LOS path is available.

5. Advanced DNN localization methods

Predominantly, previous works use raw CSI to feed their respective proposed DNN architectures. Convolutional neural networks (CNNs) and MLPs are the main components of such neural network architectures. As the system bandwidth and the number of antenna elements at the BS become larger [41], the dimensionality of CSI also increases. This can pose a challenge for methods that rely solely on MLPs, like an insufficient number of units in the input layer. Increasing the units would increase the number of parameters. As a consequence, the capacity of the model would become higher, and therefore, a more extensive training set is needed. Furthermore, MLPs cannot capture local correlations [42], for example, the channel at neighboring antennas or subcarriers, information that would be common even for hand-feature extractors. Finally, due to their fully connected nature, they have no mechanism to ensure

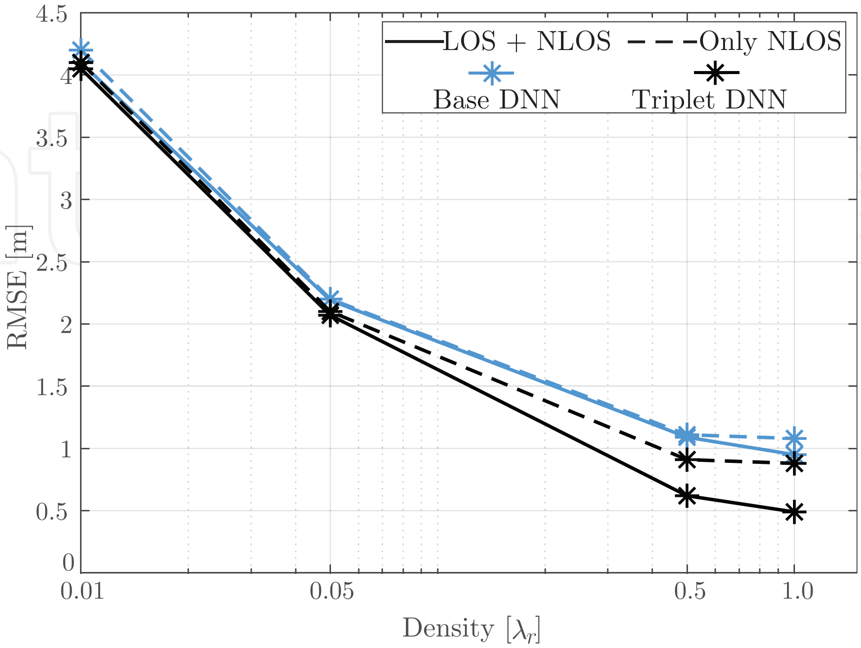


Figure 5.
The influence of line-of-sight path and the density of reference point locations for the base DNN and triplet network architecture. Reproduced with permission from [40]. © 2023 IEEE.

invariance with respect to small-scale variations in the input channel. Hence, they might not be the optimal choice for channel-feature learning or channel-to-location mapping.

To cope with the issues of imperfect channel estimates and other system impairments, various works suggest the conventional option of hand-designing feature extractors for more robust models. A common idea in the literature for hand-engineered features is to leverage the channel transform domains, such as angle, delay, or Doppler, for example, [35, 43]. However, hand-crafting input features limit the expressive capacity of DNN models, hence constraining the generalization of learned representations or the trained model.

Several studies have recommended the use of convolutional neural networks (CNNs) to better learn channel attributes essential for mapping the channel-to-location [10, 14, 44]. Nevertheless, CNNs introduce a significant inductive bias by employing filters to *slice* the channel and learn various parts of it.

In contrast to long-standing approaches, transformer-based architectures proposed more recently in natural language processing [45], computer vision [46], or wireless communications [29] adopt the *attention* mechanism [47]. Consequently, they show greater learning capacity [48] and less inductive bias and can capture local and wide-range dependencies. Next, we discuss a transformer-based model for wireless localization, that is, wireless transformer (WiT).

5.1 Wireless transformer

As presented in [29], and depicted in **Figure 6**, WiT is a fully supervised technique and is trained to minimize

$$\Psi^{\star} = \arg\min_{\Psi} \mathbb{E}\left[\left\|\mathbf{u}_r - f_{\Psi}(\mathbf{H}_r))\right\|^2\right]. \tag{13}$$

In contrast to prior DNN methods, we view the input channel $\mathbf{H}_r \in \mathbb{C}^{N_r \times N_c'}$ as a series of subcarriers. Each of these subcarriers, denoted by $\left\{\mathbf{h}_n \in \mathbb{R}^{1\times 3N_r}\right\}_{n=1}^{N_c'}$ includes

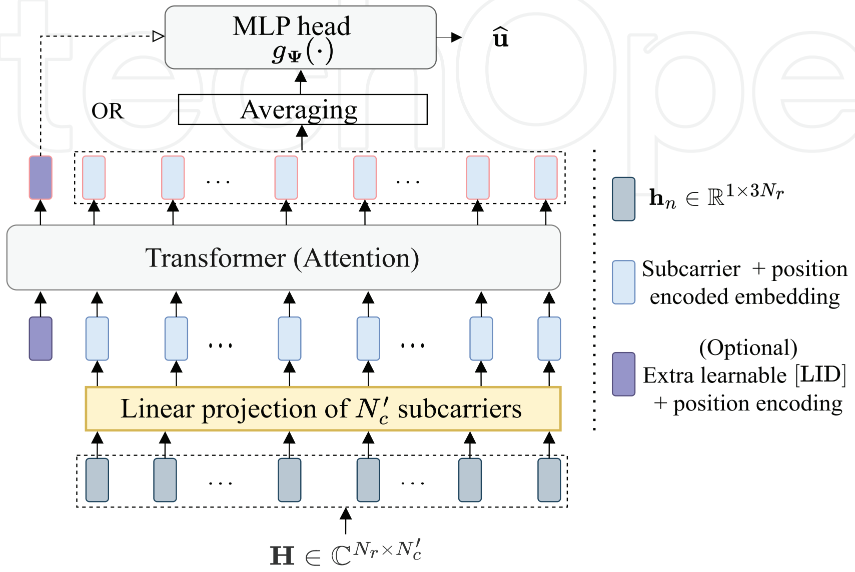
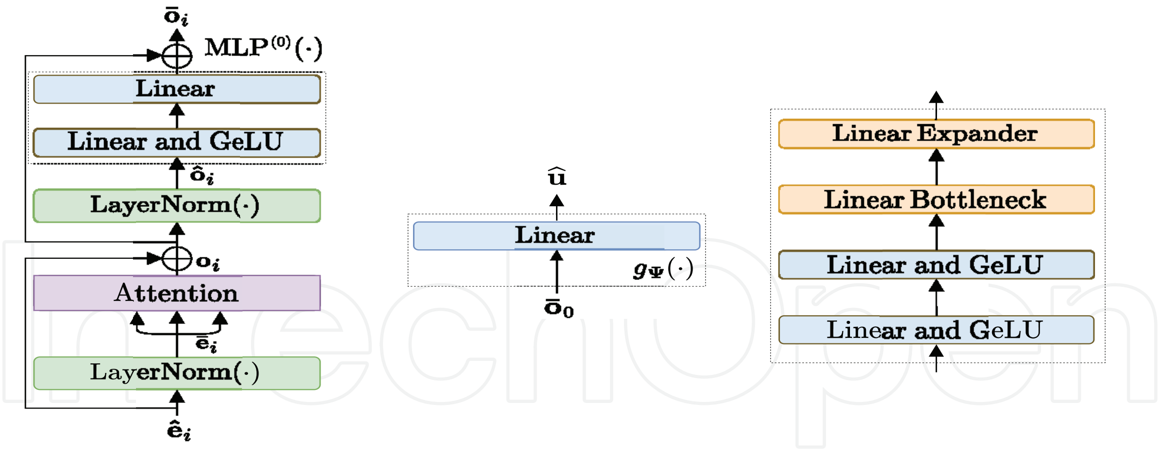

MLP head
$g_{\Psi}(\cdot)$
$\hat{\mathbf{u}}$
OR
Averaging
Transformer (Attention)
Linear projection of N_c' subcarriers
$\mathbf{H} \in \mathbb{C}^{N_r \times N_c'}$
$\mathbf{h}_n \in \mathbb{R}^{1\times 3N_r}$
Subcarrier + position encoded embedding
(Optional) Extra learnable [LID] + position encoding

Figure 6.
A transformer-based model, WiT. Used with permission from [29]. © 2023 IEEE.



(a) Transformer block.

(b) MLP head.

(c) Projectors.

Figure 7.
Details of (a) a transformer, (b) MLP-head used for location estimation, and (c) the projectors for SWiT.

the real, imaginary, and absolute elements of the row channel vectors, that is, $3N_r$. Further, individual subcarrier representations undergo a linear embedding process. For each subcarrier, we apply a linear transformation to convert it into an embedding using a linear layer. This layer has learnable parameters, $\mathbf{E} \in \mathbb{R}^{3N_r \times D}$, leading to the embedding representation $\mathbf{e}_i = \mathbf{h}_i\mathbf{E}$.

Transformer-based models, due to their lack of recurrence or standard convolutional operations, treat all subcarrier representations as permutation invariant, ignoring the sequence order of frequency-dependent subcarriers. To address this limitation, we incorporate positional encodings to represent the sequence position of each subcarrier. More specifically, we add a random and learnable real-valued vector embedding, $\mathbf{g}_i \in \mathbb{R}^{1\times D}$, to each subcarrier index i. Subsequently, given the representation of the channel, we combine $\mathbf{g}_i$ with the $i-$th embedding. The resulting subcarrier representation is $\hat{\mathbf{e}}_i = \mathbf{e}_i + \mathbf{g}_i$, which serves as the input to the subsequent transformer block.

As for the input to the MLP head, which follows the transformer block, we either average the derived features or utilize an extra symbol, [LID]. This symbol is analogous to the [CLS] token in [49]. The designated symbol [LID] represents a trainable vector, that is, $\hat{\mathbf{e}}_0 \in \mathbb{R}^D$. WiT uses $\hat{\mathbf{e}}_0$ to learn a compressed representation over all subcarriers. Moreover, the embedding $\bar{\mathbf{o}}_0 \in \mathbb{R}^D$ is input into a fully connected linear layer, as illustrated in **Figure 7b**, to map the channel features to location coordinates. The size of the set of input vectors to the transformer block is $C = N_c' + 1$.

5.1.1 Attention

Central to *every* transformer-based model is the attention mechanism [47]. Three identical inputs are employed for self-attention within the transformer block in the context of WiT and projected using the weights $\mathbf{W}_q = \mathbf{W}_k = \mathbf{W}_v$. The attention mechanism is then expressed as

$$\mathbf{o}_i = \sum_{j=1}^{C} \frac{\exp\left(\alpha_{i,j}\right)}{\sum_{j'=1}^{C} \exp\left(\alpha_{i,j'}\right)} \left(\bar{\mathbf{e}}_j \mathbf{W}_v\right) \tag{14}$$

where $\alpha_{i,j}$ denotes the attention weight between embeddings at positions i and j, given as

$$\alpha_{i,j} = \frac{1}{\sqrt{D}} \left(\overline{\mathbf{e}}_i \mathbf{W}_q\right) \left(\overline{\mathbf{e}}_j \mathbf{W}_k\right)^T. \tag{15}$$

Moreover, the embedding $\overline{\mathbf{e}}_i$ is derived from the layer normalization of $\hat{\mathbf{e}}_i$, represented as $\overline{\mathbf{e}}_i = \text{LayerNorm}(\hat{\mathbf{e}}_i; \zeta, \iota)$, where ζ and ι denote specific hyperparameters from [50].

WiT leverages the per-subcarrier channel structure and relies on learning the large-scale channel features either by averaging out the representations learned from subcarriers or by obtaining a unique representation across the entire channel. However, the wireless channel is characterized by both macroscopic and microscopic fading. Consequently, we extend WiT to a self-supervised learning framework, that is, self-supervised wireless transformer (SWiT) [51], an approach that utilizes both microscopic as well as macroscopic fading characteristics of the channel. Building upon the advantages of self-supervised training, the method in [51] may enable and facilitate several potential applications in wireless communications, extending beyond the localization task. Learned channel representations, that is, embeddings, can serve as pseudo-locations and facilitate different tasks, ranging from beamforming to localization for the purpose of different location-based services (LBS). For instance, it could be used to determine if two transmitters are close to a reference location, or a *spot*, by evaluating a *distance* metric between the points in the feature space.

5.2 Self-supervised wireless transformer

In contrast to other studies in wireless localization, in this part of the chapter, we discuss our approach to exploiting redundant and complementary information across the subcarriers. Doing so enables us to predict the channel from a single realization. While it might seem counterintuitive to learn to predict the information we possess, we later demonstrate that such an approach can be suitable for deriving meaningful representations for estimating different wireless communication tasks. In contrast to the triplet network detailed in Section 4.1, where we aim to distinguish channels from different sub-regions, here we avoid the necessity of sampling negative pairs and employing a contrastive loss. Furthermore, we show how to leverage the microscopic fading characteristics by designing subcarrier-level *pretext* tasks.

In the following, we use SWiT to derive a channel representation, denoted as $\overline{\mathbf{o}}_r$, without relying on labels or a contrastive objective function. The components of SWiT [51], depicted in **Figure 8**, are organized in a dual-branch neural network architecture. The first module, that is, channel transformations, comprises a set of stochastic augmentations represented as $\mathcal{Q}{:=}\{Q_1, \ldots, Q_A\}$, each accompanied by its occurrence probability. Furthermore, we construct $\mathcal{T}_i(\mathbf{H_r}) = (Q_A \circ Q_{A-1} \circ \ldots Q_1)(\mathbf{H}_r)$. The goal of the stochastic augmentation module is to produce multiple views of the channel, each accounting for different parts of it. Specifically, the first view is $\left\{\overline{\mathbf{h}}'_n \in \mathbb{R}^{3N_r}\right\}_{n=1}^{N_{c_1}} \triangleq \mathcal{T}_1(\mathbf{H_r})$, and similarly, the second view is where N_{c_1} denotes the total number of selected subcarriers for the first transformation. The channel transformation module generates additional N_s random views with fewer subcarriers than the first two. As a result, we obtain $V = 2 + N_s$ views of an input channel realization. The first two views are referred to as *global* views, and the others as *local*.

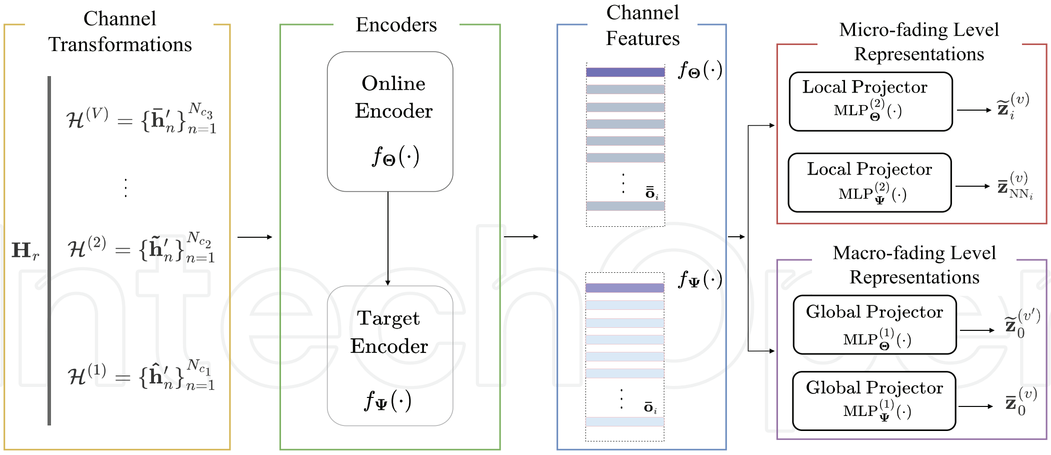

Channel Transformations
$\mathbf{H}_r$
$\mathcal{H}^{(V)} = \{\bar{\mathbf{h}}'_n\}_{n=1}^{N_{c_3}}$
⋮
$\mathcal{H}^{(2)} = \{\tilde{\mathbf{h}}'_n\}_{n=1}^{N_{c_2}}$
$\mathcal{H}^{(1)} = \{\hat{\mathbf{h}}'_n\}_{n=1}^{N_{c_1}}$
Encoders
Online Encoder
$f_{\mathbf{\Theta}}(\cdot)$
Target Encoder
$f_{\mathbf{\Psi}}(\cdot)$
Channel Features
$f_{\mathbf{\Theta}}(\cdot)$
$\bar{\bar{\mathbf{o}}}_i$
$f_{\mathbf{\Psi}}(\cdot)$
$\bar{\mathbf{o}}_i$
Micro-fading Level Representations
Local Projector $\mathrm{MLP}_{\mathbf{\Theta}}^{(2)}(\cdot)$
$\tilde{\mathbf{z}}_i^{(v)}$
Local Projector $\mathrm{MLP}_{\mathbf{\Psi}}^{(2)}(\cdot)$
$\bar{\mathbf{z}}_{\mathrm{NN}_i}^{(v)}$
Macro-fading Level Representations
Global Projector $\mathrm{MLP}_{\mathbf{\Theta}}^{(1)}(\cdot)$
$\tilde{\mathbf{z}}_0^{(v')}$
Global Projector $\mathrm{MLP}_{\mathbf{\Psi}}^{(1)}(\cdot)$
$\bar{\mathbf{z}}_0^{(v)}$

Figure 8.
Main components of SSL approach proposed to learn channel features.

Next, all channel views are processed sequentially via the encoders $f_{\Theta}(\cdot)$ and $f_{\Psi}(\cdot)$. We refer to the encoders as online and target encoder, respectively. This yields the respective representations for the first view, $\left\{\bar{\mathbf{o}}_n \in \mathbb{R}^D\right\}_{n=1}^{C} := f_{\Psi}\left(\left\{\bar{\mathbf{h}}'_n\right\}_{n=1}^{N_{c_1}}\right)$, and similarly for the others.

Given that the wireless channel exhibits both macroscopic and microscopic fading, our method is tailored to address this behavior. Therefore, the learning is split into two separate projectors, namely, micro-fading and macro-fading channel representation learning modules.

5.2.1 Micro-fading level representations

The micro-fading representation module uses a pretext task to obtain representations at the subcarrier level. In the case of SWiT, for each representation $\bar{\mathbf{o}}_i^{(v)}$, we measure its correlation with the embeddings in its vicinity, that is, $\left\{\bar{\mathbf{o}}_j^{(v)}\right\}_{j\in\mathcal{N}_i}$, where $\mathcal{N}_i$ denotes the set of adjacent subcarrier representations with $|\mathcal{N}_i| = K_n$, and $K_n \ll N_{c'}$. Next, to obtain the embeddings from the micro-fading level module, we also evaluate $\tilde{\mathbf{z}}_i^{(v)}$ and $\bar{\mathbf{z}}_{\mathrm{NN}_i}^{(v)}$, stemming from the online and target local projectors, respectively. Lastly, we evaluate the cross-entropy loss over all the N' representations; for each individual view,

$$\mathcal{L}_s := -\frac{1}{VN'}\sum_{v=1}^{V}\sum_{i=1}^{N'} \bar{\mathbf{z}}_{\mathrm{NN}_i}^{(v)} \log\left(\tilde{\mathbf{z}}_i^{(v)}\right). \tag{16}$$

5.2.2 Macro-fading level representations

In contrast to the micro-fading module, the macro-fading module of SWiT processes all channel views to output the respective embeddings $\bar{\mathbf{z}}_0^{(v)}$ and $\tilde{\mathbf{z}}_0^{(v)}$. In this case, we apply the cross-entropy to compare the global view representations originating from the target encoder and any alternative representation view stemming from the online encoder,

$$\mathcal{L}_c := -\frac{1}{2(V-1)}\sum_{v=1}^{2}\sum_{v'\neq v}^{V}\overline{\mathbf{z}}_0^{(v)}\log\left(\tilde{\mathbf{z}}_0^{(v')}\right). \tag{17}$$

The overall loss function is given by $\mathcal{L}_{\mathrm{SSL}} := \mathcal{L}_c + \beta\mathcal{L}_s$. Here, $\beta \in [0, 1]$ dictates the importance of learning macroscopic fading versus microscopic features.

5.3 Datasets and self-supervised training

We use real-world channel measurements and synthetic data to further benchmark the performance of the SWiT.

For actual measurements, we selected the *ultra dense indoor MaMIMO* [13]. This collection has multiple datasets obtained in different massive MIMO setups and propagation conditions in a laboratory. Three specific datasets from this collection were utilized: one featuring NLOS propagation with a uniform rectangular array setup (KUL-NLOS-URA-Lab), another in a LOS and a uniform linear array (KUL-LOS--ULA-Lab), and the third in a LOS and a distributed antenna system setup (KUL-LOS-DIS-Lab). Common parameters across these datasets include $N_r = 64$, $N_{c'} = 100, f_c = 2.61$GHz, and $R = 250{,}000$.

We also choose a synthetic dataset generated based on the discussion in Section 2.2. The datasets are referred to as S-200 and HB-200 [29], representing two dynamic railway scenarios, each with $T = 200$. The S-scenario is characterized by $M = 1$ and $R = 69\,212$. Conversely, the HB-scenario is characterized by $M = 8$ RRH and a larger sample size of $R = 81200$. For both scenarios, the common parameters are $N_r = 64$, $f_c = 3.5$GHz, $G = 4$, and $N_c' = 32$.

5.3.1 Self-supervised training

For both the online and target models, specifically $f_\Phi(\cdot), f_\Psi(\cdot)$, $\nu_\Phi(\cdot)$, and $\nu_\Psi(\cdot)$, we adopt the architectural specifics from WiT [29], excluding the details of MLP head. Furthermore, conforming to the norms in modern transformer-based models [46], we choose $D = 384$ as opposed to the previously utilized $D = 650$ in WiT. We also use a depth of single-transformer blocks $H_{\text{blck}} = 1$ with single-attention head $H_{\text{attn}} = 1$. Among other configurations, the LayerNorm parameters are set at $\zeta = 1$ and $\iota = 0.0001$. Further parameters include $N_s = 8$, $\tilde{N}_c = 36$, $\overline{N}_c = 16$, $K_n = 6$, and $K_k = 3$.

When training SWiT with a single GPU, it may require fine-tuning the weight decay and learning rate depending on the dataset size and the number of iterations (i.e., the batch size). For most of the experiments in this chapter, we trained SWiT without labels for varying epoch lengths, using a batch size of $B = 256$. Due to the considerable time complexity involved, our training utilizes approximately 10–25% of the KUL dataset. We employ the AdamW optimizer [52], utilizing a cosine learning rate schedule. We incorporate a linear warm-up phase over the first 10 epochs. The parameter β is set to 0.1. The base learning rate, ϖ_{base}, starts at 1.5e-4 and updates according to $\varpi \triangleq \varpi_{\text{base}} B/256$. For the updates in the target network, the base rate κ_{base} is set at 0.994 and adjusts with $\kappa \triangleq 1 - (1 - \kappa_{\text{base}})(\cos(\pi u/U) + 1)/2$. Furthermore, the weight decay scales between 0.04 and 0.5. We set χ_Ψ at 0.04, χ_Θ at 0.1, and Λ at 0.9. During the first epoch, the weights of the target encoder are not updated. As a non-linearity, we use GeLU and the number of units of 1 024 for standard layers in the

$\mathbf{Q}_a, \mathbb{P}(\mathbf{Q}_a)$	$\mathcal{T}_1(\cdot)$	$\mathcal{T}_2(\cdot)$	$\mathcal{T}_3(\cdot)$
Subcarrier selection (RSS)	1.0	1.0	1.0
Subcarrier flipping (RSF)	0.4	0.4	0.0
Gain offset (RGO)	0.2	0.8	0.0
Fading component (RFC)	0.0	0.1	0.0
Sign change (RSC)	0.0	0.2	0.0
Normalization	1.0	1.0	1.0
Gaussian noise	0.2	0.2	0.0

Table 1.
Transformation function $\mathcal{T}_i(\cdot)$ and $\mathbb{P}(Q_a)$.

MLPs. The bottleneck layer uses 256 nodes, while the last layer has 25 000 for the global projectors and 1 024 for the local ones. The default augmentations from [51] are used with assigned probabilities, $\mathbb{P}(Q_a)$, for $\mathcal{T}_1(\cdot)$, $\mathcal{T}_2(\cdot)$, and $\mathcal{T}_3(\cdot)$, as shown in **Table 1**.

5.4 Localization performance and transferability

To assess the performance of learned representations, we perform linear and fine-tuning evaluation for the trained models on several localization tasks. We also investigate the transferability of the models to other tasks and datasets.

5.4.1 Localization accuracy with limited data

For linear analysis, we train a regressor $g_\Phi : \mathbb{R}^D \to \mathbb{R}^{D_{\text{out}}}$ atop frozen features from $f_{\Psi^\star}(\cdot)$. For localization, $D_{\text{out}} = 2$. We utilize a single-layer MLP and train for $500 - 1\,000$ epochs with $B = 128$ and SGD at $\varpi = 0.03$. Additionally, we evaluate the quality of the embedding with a k-NN classifier on frozen $f_{\Psi^\star}(\cdot)$. Performance metrics include top-1 and, when suitable, top-5 accuracy.

To assess the performance of the embeddings, we also perform fine-tuning using labeled data. Specifically, the backbone (i.e., target encoder) is initialized with the pre-trained weights, forming the new network $g_\Phi \circ f_{\Psi^\star}(\mathbf{H})$. Parameters are set as $B = 512$ and $\varpi = 3e - 4$, using AdamW with standard settings. We report accuracy on 32 subcarriers after normalization. A codebase, together with the evaluation results, are publicly available[1].

In **Figure 9**, we highlight the improvements of SWiT in a small data regime for the case of KUL-NLOS dataset. In contrast to the analysis in [51], we offer additional insights here, highlighting when a simple linear approach is sufficient and when the computationally demanding WiT and SWiT are justified. For the data regimes in the figure, we investigate the RMSE separately for six different randomly sampled datasets. Datasets for evaluation are sampled based on the Poisson point process (PPP) with varying density parameter values $\lambda_r = \{0.1, 0.5\}$. A value of $\lambda_{\text{data}} = 1.0 \approx 4\,500$ training examples. For testing, we sample $R_{\text{test}} = 5\,000$. We observe that when $R \approx 400$ samples, even the linear model, training on top of the channel features learned from

[1] https://github.com/ars205/ssl_wireless

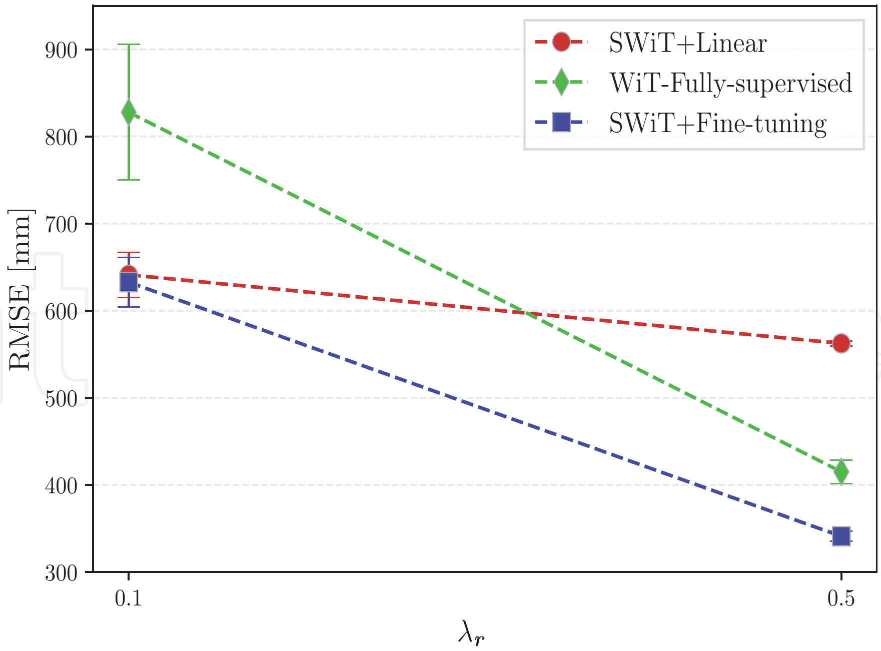


Figure 9.
Two-dimensional t-SNE embeddings of the combined LOS and NLOS datasets, each from four different spots. (a) Shows the representations from WiT with randomly initialized weights and (b) from SWiT.

SWiT, can outperform a fully supervised model, like WiT, which in here corresponds to the encoder of SWiT.

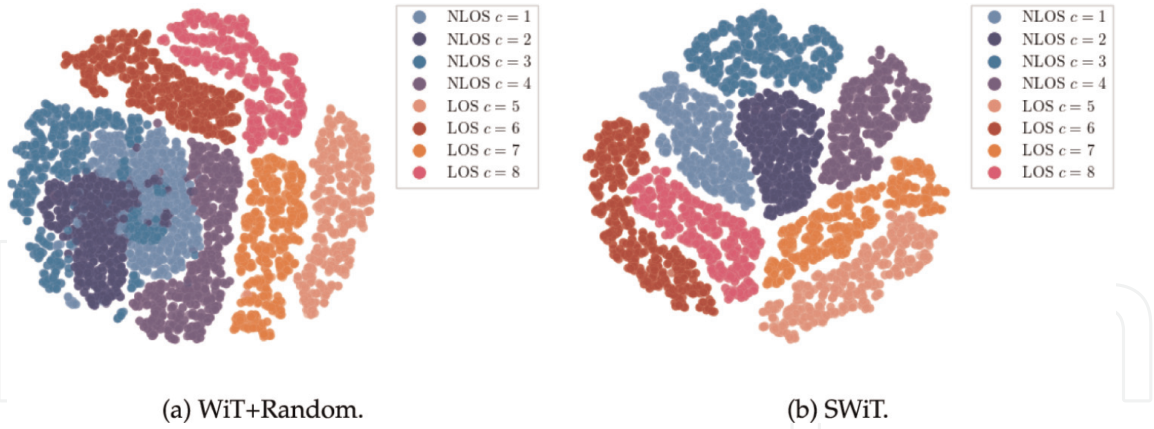
5.4.2 Spot-localization and transfer learning

Table 2 reports the model's ability to *cluster* channel representations while maintaining spatial neighborhood. The KUL datasets, divided into four sub-regions, have $\hat{C}_{\text{spot}} = 4$ spots, while S-200 and HB-200 possess $\hat{C}_{\text{spot}} = 360$ and $\hat{C}_{\text{spot}} = 406$, respectively. The SWiT encoder achieves nearly 100% classification accuracy for KUL datasets, indicating its superior representation learning capability compared to WiT with random weights. In the same table, we also show the transferability (SWiT + TF) of a model to other datasets. More specifically, the trained model on KUL-NLOS dataset is evaluated across other datasets, emphasizing its ability to adapt to diverse MIMO setups and propagation conditions.

In **Figure 10**, we show the two-dimensional t-SNE [53] embeddings of a merged KUL dataset, encompassing datasets in LOS and NLOS conditions, resulting in a total of $\hat{C}_{\text{spot}} = 8$ spots. For LOS conditions, both randomly initialized WiT and pre-trained models exhibit similar representations. However, under NLOS conditions, the SWiT

	KUL-NLOS	KUL-LOS	KUL-LOS-DIS	S-200		HB-200	
Method	↑ **Top-1**	↑ **Top-1**	↑ **Top-1**	↑ **Top-1**	↑ **Top-5**	↑ **Top-1**	↑ **Top-5**
Random	23.7	3.59	4.13	0.296	1.545	0.27	1.25
SWiT	99.99	99.99	99.99	18.70	52.38	37.38	82.85
SWiT + TF	99.99	99.97	99.98	16.1	44.34	20.68	53.58

Table 2.
Random weights versus SWiT.



(a) WiT+Random.

(b) SWiT.

Figure 10.
Two-dimensional t-SNE embeddings of the combined LOS and NLOS datasets, each from four different spots. (a) Shows the representations from WiT with randomly initialized weights and (b) from SWiT.

model demonstrates that it can clearly differentiate between users in various regions and determine if they are in LOS or NLOS. This is shown in **Figure 10a** and **b**, where each color denotes distinct *spots*.

6. Conclusion

In this chapter, we presented three different learning approaches and introduced multiple DNN-based models. First, we used classical PCA and metric scaling subspace methods to obtain useful channel features to determine the UE location. Then, we discussed neural networks as feature extractors and demonstrated the performance gains of the proposed triplet network when compared to an MLP classifier. Finally, we introduced self-supervised learning for wireless channel representation learning. We showed that we can design *pretext* tasks that exploit the channel's macroscopic- and microscopic-fading characteristics to train a model without labels and a contrastive objective. We showed that the SSL approach outperforms supervised techniques in data-limited scenarios, even when using a linear model. In this chapter, we also highlighted the ability of the SSL approach for *spot* estimation across different datasets, environments, and MIMO setups.

Acknowledgements

This work has been funded by the Christian Doppler Laboratory for Digital Twin assisted AI for sustainable Radio Access Networks, Institute of Telecommunications, TU Wien. The financial support by the Austrian Federal Ministry for Labour and Economy and the National Foundation for Research, Technology and Development and the Christian Doppler Research Association is gratefully acknowledged.

Abbreviations

AI	artificial intelligence
6G	sixth-generation

ML	machine learning
DNN	deep neural network
CSI	channel state information
UE	user equipment
MIMO	multiple-input multiple-output
BS	base station
LOS	line-of-sight
NLOS	non-line-of-sight
DAS	distributed antenna system
RRH	remote radio head
PCA	principle component analysis
MDS	multidimensional scaling
k-NN	k-nearest neighbors
MLP	multi-layer perceptrons
RP	reference point
SSL	self-supervised learning
WiT	wireless transformer
SWiT	self-supervised wireless transformer

Author details

Artan Salihu[1,2]*, Markus Rupp[1] and Stefan Schwarz[1]

1 Institute of Telecommunications, Technische Universität (TU) Wien, Vienna, Austria

2 Christian Doppler Laboratory for Digital Twin Assisted AI for Sustainable Radio Access Networks, Vienna, Austria

*Address all correspondence to: artan.salihu@tuwien.ac.at

References

[1] Rong B. 6G: The next horizon: From connected people and things to connected intelligence. IEEE Wireless Communications. 2021;**28**(5):8-8

[2] You X, Wang C-X, Huang J, Gao X, Zhang Z, Wang M, et al. Towards 6G wireless communication networks: Vision, enabling technologies, and new paradigm shifts. Science China Information Sciences. 2021;**64**:1-74

[3] Wen F, Wymeersch H, Peng B, Tay WP, So HC, Yang D. A survey on 5G massive MIMO localization. Digital Signal Processing. 2019;**94**:21-28

[4] Wymeersch H, Seco-Granados G. Radio localization and sensing—Part ii: State-of-the-art and challenges. IEEE Communications Letters. 2022;**26**(12): 2821-2825

[5] Wylie M, P, Holtzman J. The non-line of sight problem in mobile location estimation. In: Proceedings of ICUPC-5th International Conference on Universal Personal Communications. Vol. 2. Cambridge, MA, USA: IEEE; 1996. pp. 827-831

[6] Sayed AH, Tarighat A, Khajehnouri N. Network-based wireless location: Challenges faced in developing techniques for accurate wireless location information. IEEE Signal Processing Magazine. 2005;**22**(4):24-40

[7] Witrisal K, Meissner P, Leitinger E, Shen Y, Gustafson C, Tufvesson F, et al. High-accuracy localization for assisted living: 5G systems will turn multipath channels from foe to friend. IEEE Signal Processing Magazine. 2016;**33**(2): 59-70

[8] Burghal D, Ravi AT, Rao V, Alghafis AA, Molisch AF. A comprehensive survey of machine learning based localization with wireless signals. arXiv. 2020

[9] Wang X, Gao L, Mao S, Pandey S. Deepfi: Deep learning for indoor fingerprinting using channel state information. In: 2015 IEEE Wireless Communications and Networking Conference (WCNC). New Orleans, LA, USA: IEEE; 2015. pp. 1666-1671

[10] Vieira J, Leitinger E, Sarajlic M, Li X, Tufvesson F. Deep convolutional neural networks for massive MIMO fingerprint-based positioning. In: 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC). IEEE; 2017. pp. 1-6

[11] Niitsoo A, Edelhäußer T, Mutschler C. Convolutional neural networks for position estimation in tdoa-based locating systems. In: 2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN). Nantes, France: IEEE; 2018. pp. 1-8

[12] Gante J, Falcao G, Sousa L. Deep learning architectures for accurate millimeter wave positioning in 5g. Neural Processing Letters. 2020;**51**(1): 487-514

[13] De Bast S, Guevara AP, Pollin S. CSI-based positioning in massive mimo systems using convolutional neural networks. In: 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring). Antwerp, Belgium: IEEE; 2020. pp. 1-5

[14] Ayyalasomayajula R, Arun A, Wu C, Sharma S, Sethi AR, Vasisht D, et al. Deep learning based wireless localization for indoor navigation. In: Proceedings of

the 26th Annual International Conference on Mobile Computing and Networking. New York, NY, USA: Association for Computing Machinery (ACM); 2020. pp. 1-14

[15] Salihu A, Schwarz S, Rupp M. Towards scalable uncertainty aware DNN-based wireless localisation. In: 2021 29th European Signal Processing Conference (EUSIPCO). Dublin, Ireland. 2021. pp. 1706-1710

[16] Zekavat R, Michael R, Buehrer. Handbook of Position Location: Theory, Practice and Advances. Vol. 27. New Jersey, USA: John Wiley & Sons; 2011

[17] Rupp M, Schwarz S. An LS localisation method for massive MIMO transmission systems. In: ICASSP - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Brighton, UK: IEEE; 2019. pp. 4375-4379

[18] Arnold M, Hoydis J, ten Brink S. Novel massive mimo channel sounding data applied to deep learning-based indoor positioning. In: SCC 2019; 12th International ITG Conference on Systems, Communications and Coding. Rostock, Germany: VDE; 2019

[19] Hoang MT, Yuen B, Ren K, Dong X, Lu T, Westendorp R, et al. A CNN-LSTM quantifier for single access point CSI indoor localization. arXiv. 2020

[20] Adhikary A, Nam J, Ahn J-Y, Caire G. Joint spatial division and multiplexing—The large-scale array regime. IEEE Transactions on Information Theory. 2013;**59**(10): 6441-6463

[21] Ngo HQ, Larsson EG, Marzetta TL. Energy and spectral efficiency of very large multiuser mimo systems. IEEE Transactions on Communications. 2013; **61**(4):1436-1449

[22] Lu L, Li GY, Lee Swindlehurst A, Ashikhmin A, Zhang R. An overview of massive mimo: Benefits and challenges. IEEE Journal of Selected Topics in Signal Processing. 2014;**8**(5):742-758

[23] Marzetta TL. Noncooperative cellular wireless with unlimited numbers of base station antennas. IEEE Transactions on Wireless Communications. 2010;**9**(11):3590-3600

[24] Schwarz S, Pratschner S. Multiple antenna systems in mobile 6G: Directional channels and robust signal processing. IEEE Communications Magazine. 2023;**61**(4):64-70

[25] Schmidt R. Multiple emitter location and signal parameter estimation. IEEE Transactions on Antennas and Propagation. 1986;**34**(3):276-280

[26] Krim H, Viberg M. Two decades of array signal processing research: The parametric approach. IEEE Signal Processing Magazine. 1996;**13**(4):67-94

[27] Liu W, Haardt M, Greco MS, Mecklenbräuker CF, Willett P. Twenty-five years of sensor array and multichannel signal processing: A review of progress to date and potential research directions. IEEE Signal Processing Magazine. 2023;**40**(4):80-91

[28] Heath RW, Gonzalez-Prelcic N, Rangan S, Roh W, Sayeed AM. An overview of signal processing techniques for millimeter wave mimo systems. IEEE Journal of Selected Topics in Signal Processing. 2016;**10**(3):436-453

[29] Salihu A, Schwarz S, Rupp M. Attention aided CSI wireless localization. In: 2022 IEEE 23rd International

Workshop on Signal Processing Advances in Wireless Communication (SPAWC). Oulu, Finland: IEEE; 2022. pp. 1-5

[30] Torgerson WS. Multidimensional scaling: I. Theory and method. Psychometrika. 1952;**17**(4):401-419

[31] Van Der Maaten L, Postma E, Van den Herik J. Dimensionality reduction: A comparative. Journal of Machine Learning Research. 2009;**10**(66–71):13

[32] Williams CKI. On a connection between kernel PCA and metric multidimensional scaling. Machine Learning. 2002;**46**(1–3):11-19

[33] Sammon JW. A nonlinear mapping for data structure analysis. IEEE Transactions on Computers. 1969;**18**(5): 401-409

[34] Moltchanov D. Distance distributions in random networks. Ad Hoc Networks. 2012;**10**(6):1146-1166

[35] Sun X, Chi W, Gao X, Li GY. Fingerprint-based localization for massive MIMO-OFDM system with deep convolutional neural networks. IEEE Transactions on Vehicular Technology. 2019;**68**(11):10846-10857

[36] Wang X, Wang X, Mao S. Deep convolutional neural networks for indoor localization with CSI images. IEEE Transactions on Network Science and Engineering. 2020;7(1):316-327

[37] Salihu A, Schwarz S, Rupp M. Learning-based remote radio head selection and localization in distributed antenna system. In: 2022 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit). Grenoble, France: IEEE; 2022. pp. 65-70

[38] Bromley J, Guyon I, LeCun Y, Säckinger E, Shah R. Signature verification using a "Siamese" time delay neural network. Advances in Neural Information Processing Systems. 1994: 737-744

[39] Schroff F, Kalenichenko D, Philbin J. Facenet: A unified embedding for face recognition and clustering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Boston, MA, USA: IEEE; 2015. pp. 815-823

[40] Salihu A, Schwarz S, Pikrakis A, Rupp M. Low-dimensional representation learning for wireless CSI-based localisation. In: 16th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob) (50308). Thessaloniki, Greece: IEEE; 2020. pp. 1-6

[41] ETSI. Study on New Radio (NR) Access Technology. Technical Specification (TS) 38.912. Valbonne, France: European Telecommunications Standards Institute (ETSI); 2021. Version 15.0.0

[42] LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proceedings of the IEEE. 1998;**86**(11): 2278-2324

[43] Ferrand P, Decurninge A, Guillaud M. DNN-based localization from channel estimates: Feature design and experimental results. In: GLOBECOM 2020–2020 IEEE Global Communications Conference. Taipei, Taiwan: IEEE; 2020. pp. 1-6

[44] De Bast S, Pollin S. MaMIMO CSI-based positioning using CNNs: Peeking inside the black box. In: 2020 IEEE International Conference on Communications Workshops (ICC

Workshops). Dublin, Ireland: IEEE; 2020. pp. 1-6

[45] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. Advances in Neural Information Processing Systems. 2017;**30**

[46] Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, et al. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv. 2020

[47] Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate. arXiv. 2014

[48] Zhao H, Jia J, Koltun V. Exploring self-attention for image recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Seattle, WA, USA: IEEE; 2020. pp. 10076-10085

[49] Devlin J, Chang M-W, Lee K, Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding. San Diego, United States: ICLR 2015; 2018

[50] Ba JL, Kiros JR, Hinton GE. Layer normalization. arXiv. 2016

[51] Salihu A, Schwarz S, Rupp M. Self-supervised and invariant representations for wireless localization. arXiv. 2023

[52] Loshchilov I, Hutter F. Decoupled weight decay regularization. arXiv. 2017

[53] Van der Maaten L, Hinton G. Visualizing data using t-sne. Journal of Machine Learning Research. 2008;**9**