Metric Learning based Positioning

Santeri Kokkonen, Xinze Li, Ashvin Srinivasan, Hanan Al-Tous, and Olav Tirkkonen

Department of Information and Communications Engineering, Aalto University, Finland
Email: {santeri.kokkonen, xinze.li, ashvin.1.srinivasan, hanan.al-tous, olav.tirkkonen}@aalto.fi

***Abstract*—We predict the physical distance between two locations based on the Channel State Information (CSI) wireless channels between a multiantenna base station and the locations. We propose a metric learning framework using a neural network that ensures the properties of a metric: zero distance between a point and itself, non-negativity, symmetry, and the triangle inequality. Neural network performance improves with larger datasets, and training on small datasets can result in lower accuracy. However, in applications like fingerprint localization, creating large datasets is often impractical. In such cases, learning the metric from a small dataset may still be feasible because the number of pairs of data points increases quadratically as compared to the number of points, which would be used in direct fingerprint regression. We utilize metric learning with Weighted K-Nearest Neighbour (WKNN) localization. The learned metric is used to find the neighbours and compute the weighting vector. Simulation results show that the 80th percentile error can be improved by 29 cm using the learned metric with WKNN regression compared to using the Euclidean distance with WKNN regression.**

***Index Terms*—Channel state information, non-line-of-sight communication, fingerprint localization, metric learning, weighted k-nearest-neighbours regression.**

I. Introduction

Accurate localization is an important feature of modern wireless communication systems, enabling location-based services. In outdoor scenarios fully developed localization technologies, such as Global Navigation Satellite System (GNSS), can meet accuracy demands of most services. In indoor environments, localization accuracy faces many challenges due to the presence of multipath localization. Fifth Generation New Radio (5G NR) shows a substantial improvement in indoor cellular positioning, benefiting from a wide selection of standardized techniques [1], [2].

In indoor environments where positioning is particularly challenging, fingerprint positioning emerges as a viable solution [3], [4]. Fingerprint positioning involves collecting radio measurements, e.g., Received Signal Strength (RSS) or Channel State Information (CSI), at labeled locations to form a data set. A fingerprint model is trained based on the created data set, based on a supervised machine leanring method such as Weighted K-Nearest Neighbor (WKNN) regression, a Support Vector Machine (SVM), or a Deep Neural Network (DNN) [2]. WKNN regression is widely used, because of its high accuracy, especially when the data set is small, and due to its low computational cost [5]–[7]. The WKNN method shows better localization results than a Deep Neural Network (DNN) for a measured data set in [6].

WKNN performance is affected by the feature distance and weight fucntion used, and the number of neighbours. Typically the Euclidean distance and an eponential weight function are chosen. In [8], it is shown that the Sørensen distance outperforms other distances when RSS features are used. In [9] a neighbour selecting criterion and enhanced weighting function for WKNN localization are considered. Different CSI features and DNN structures have been considered for fingerprint localization [10]–[13].

In the literature, metric learning has been based on reducing intra-class distance and increasing inter-class distance in a classification problem. The problem can be formulated as an optimization problem. Since the problem is non-convex, relaxation approaches have been considered to obtain a sub optimal solution [14], [15]. A triplet neural structure (TNS) is considered to parameterize the metric in [16]. The TNS is utilized for localization and channel charting in [17], [18].

In [19], the authors consider unmanned aerial vehicle geometric localization. A DNN that maps the CSI feature distance between pairs of nodes to their physical distance is considered. A multilateration algorithm is then applied to find the physical location. The distance obtained from DNN is symmetric. However, the distance does not satisfy other axioms of a metric, i.e., non-negativity, the vanishing of the distance from a point to itself, and the triangular inequality.

In this paper we consider a DNN that outputs a metric. We utilize the obtained metric for WKNN localization in a challenging non-line-of-sight indoor factory scenario. For this, the covariance CSI feature from several base stations is considered. The proposed metric learning structure consists of two similar networks, where the CSI features of two points are input. Data points do not need to be classified into far and near. We compute the CSI-induced distance of two points we considering the Euclidean distance of the outputs of the two networks. The mean squared error between the CSI-induced distance and true distance is used as the loss function. The metric structure in this paper has some similarity with Siamese neural networks, which have been used for dimensionlity reduction method [20], [21]. However, the loss function is different. We compare the localization performance of the metric learning based WKNN approach with Euclidean distance based WKNN, and DNN localization.

The remainder of this paper is organized as follows: In Section II, the system model is introduced. In Section III, the metric learning problem is presented. In Section IV, the localization framework is discussed. In Section V, the performance evaluation and complexity analysis are provided. Simulation results are presented and discussed in Section VI. Finally, conclusions are drawn in Section VII.

II. System Model

We consider a communication system with B Base Stations (BSs), each BS having M antennas, e.g., a Uniform Linear Array (ULA). User Equipments (UEs) have one omnidirectional antenna.

We assume transmissions using Orthogonal Frequency-Division Multiplexing (OFDM) with N subcarriers, where the cyclic prefix is longer than the maximum delay spread of the channels. The channel vector between UE u and BS b over subcarrier n at time-sample s is $\mathbf{h}_{u,b,n,s} \in \mathbb{C}^{M\times 1}$. The channel coefficients model path-loss as well as large and small scale multipath fading effects.

The selection and preprocessing of the CSI features is essential for enhancing data quality in analytical and machine learning tasks. We consider the covariance matrix, which captures statistical spatial characteristics. The empirical covariance CSI feature of the channel between UE u and BS b is computed as

$$\mathbf{R}_{u,b} = \frac{1}{S\,N} \sum_{s=0}^{S-1} \sum_{n=0}^{N-1} \mathbf{h}_{u,b,n,s}\, \mathbf{h}^{\mathrm{H}}_{u,b,n,s}\,, \tag{1}$$

where $\mathbf{R}_{u,b} \in \mathbb{C}^{M\times M}$, and S is the number of time samples. In $\mathbf{R}_{u,b}$, the impact of small-scale fading is averaged, making covariance-based features more robust to small-scale fading effects.

In this paper, we will consider covariance logarithm feature as well as the raw covariance feature without any processing, and keep other operations for future work.

The Eigendecomposition of covariance matrix $\mathbf{R}$ of size $M \times M$ is

$$\mathbf{R} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^{\mathrm{H}}\,,$$

where $\boldsymbol{\Lambda} = \mathrm{diag}\left([\lambda_1, \ldots, \lambda_M]\right)$ a diagonal matrix, and $\mathbf{U}$ is a unitary matrix. For a positive definite matrix all Eigenvalues are positive. The logarithm of the covariance matrix then becomes:

$$\log(\mathbf{R}) = \mathbf{U}\,\mathrm{diag}\left([\log\lambda_1, \ldots, \log\lambda_M]\right)\mathbf{U}^{\mathrm{H}}\,. \tag{2}$$

The logarithmic transform provides scale invariance. For a positive semi-definite matrix, a practical approach is to find the matrix logarithm of the $M' < M$ largest Eigenvalues, with $\lambda_{m'} > 0$ for $m' = 1, \ldots, M'$ and use the corresponding Eigenvectors.

III. Metric Learning

In the literature, several feature distances/similarities are considered. The simplest distance between two matrices $\mathbf{M}$ and $\mathbf{M}^{'}$ is the Euclidean distance

$$d_{\mathrm{Euc}}\left(\mathbf{M}, \mathbf{M}^{'}\right) = \left\|\mathbf{M} - \mathbf{M}^{'}\right\|_{\mathrm{F}}\,, \tag{3}$$

where $||.||_F$ is the Frobenius norm. We will consider the Euclidean distance to measure the distance between covariance matrices as well as between logarithms of covariance matrices.

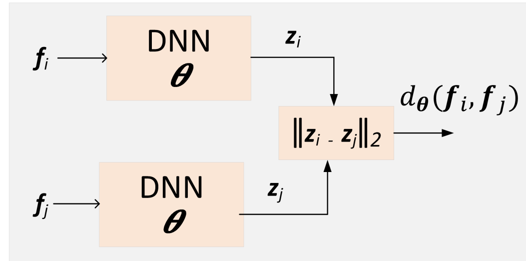

Fig. 1. DNN structure for metric learning. Consisting of two DNN with the same parameters $\boldsymbol{\theta}$. The output is a learned low dimensional feature. The structure is used to predict the physical distance.

A metric d on a set X is a function $f : X \times X \to \mathbb{R}$ such that for $x, y, z \in X$, the following conditions are satisfied

- Non-negativity: $d(x, y) > 0$ and $d(x, y) = 0$ if and only if $x = y$.
- Symmetry: $d(x, y) = d(y, x)$,
- Triangle inequality: $d(x, y) \leq d(x, z) + d(z, y)$.

A metric space is an ordered pair (X, d), where X is the set and d is the metric on X. If $d(\circ, \circ)$ on the set X is non-negative, symmetric, and satisfies the triangle inequality, but $d(x, y) = 0$ for some $x \neq y$, it is called a *pseudo metric*. The Euclidean distance is a metric.

Most metric learning methods aim to find a linear transform of the features, which provides the best metric properties. For this, a matrix $\mathbf{A}$ is searched for, such that the distance between features $\mathbf{f}_i$ and $\mathbf{f}_j$ is

$$d_{\mathbf{A}}(\mathbf{f}_i, \mathbf{f}_j) = \sqrt{\left(\mathbf{f}_i - \mathbf{f}_j\right)^{\mathrm{T}} \mathbf{A}\left(\mathbf{f}_i - \mathbf{f}_j\right)}\,. \tag{4}$$

A postive definite matrix $\mathbf{A}$ parameterizes the metric. When $\mathbf{A}$ is positive semi-definite, the result is a pseudo metric. $\mathbf{A}$ is learned with the goal of minimizing a cost function subject to constraints defined by the data set. Most algorithms work either with pairwise relationships or with proximity relation triplets derived from labels [15]. Most approaches seek a distance that brings similar samples "closer," while it "pushes away" dissimilar ones. The problem, can be formulated in various ways using different objective functions. If the matrix $\mathbf{A}$ is decomposed in terms of a $\mu \times J$-matrix $\mathbf{L}$ as $\mathbf{A} = \mathbf{L}^{\mathrm{T}}\mathbf{L}$, the distance becomes

$$d_{\mathbf{L}}(\mathbf{f}_i, \mathbf{f}_j) = \|\mathbf{L}\left(\mathbf{f}_i - \mathbf{f}_j\right)\|_2\,. \tag{5}$$

When $\mu < J$, the features have been projected to a lower-dimensional space μ. Several optimization algorithms have been used to find $\mathbf{L}$ [14], [15].

Here, we instead consider a nonlinear model:

$$d_{\boldsymbol{\theta}}(\mathbf{f}_i, \mathbf{f}_j) = \|\boldsymbol{\Phi}_{\boldsymbol{\theta}}(\mathbf{f}_i) - \boldsymbol{\Phi}_{\boldsymbol{\theta}}(\mathbf{f}_j)\|_2\,, \tag{6}$$

where $\boldsymbol{\Phi}_{\boldsymbol{\theta}}(\mathbf{f}_i) \in \Re^{\mu\times 1}$ is a mapping function parameterized by $\boldsymbol{\theta}$. We consider a DNN structure as shown in Figure 1, and use the squared distance error loss function

$$\mathcal{L}_{i,j} = \left(d_{\boldsymbol{\theta}}(\mathbf{f}_i, \mathbf{f}_j) - d_{\mathrm{Phy}}(\mathbf{p}_i, \mathbf{p}_j)\right)^2\,, \tag{7}$$

where $d_{\mathrm{Phy}}(\mathbf{p}_i, \mathbf{p}_j) = \|\mathbf{p}_i - \mathbf{p}_j\|_2$ is the physical distance between physical locations $\mathbf{p}_i$ and $\mathbf{p}_j$ in the training data

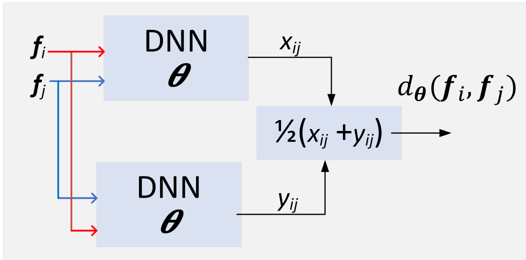

Fig. 2. DNN structure for learning the physical distance [19].

set. The training of the DNN takes place during an offline phase. In the online phase, the DNN is used to predict the low dimensional feature for a given CSI feature.

It is easy to show that $d_{\boldsymbol{\theta}}(\mathbf{f}_i, \mathbf{f}_j)$ satisfies the properties of a pseudometric. From the definition, it clear that it is non-negative and symmetric. The triangle inequality can be proved as follows. First adding and subtracting the same term

$$\|\boldsymbol{\Phi}_{\boldsymbol{\theta}}(\mathbf{f}_i)-\boldsymbol{\Phi}_{\boldsymbol{\theta}}(\mathbf{f}_j)\|_2^2 = \|\boldsymbol{\Phi}_{\boldsymbol{\theta}}(\mathbf{f}_i)-\boldsymbol{\Phi}_{\boldsymbol{\theta}}(\mathbf{f}_k)+\boldsymbol{\Phi}_{\boldsymbol{\theta}}(\mathbf{f}_k)-\boldsymbol{\Phi}_{\boldsymbol{\theta}}(\mathbf{f}_j)\|_2^2 ,$$

then using Cauchy–Schwarz inequality, we get

$$\begin{aligned} &\|\boldsymbol{\Phi}_{\boldsymbol{\theta}}(\mathbf{f}_i) - \boldsymbol{\Phi}_{\boldsymbol{\theta}}(\mathbf{f}_k) + \boldsymbol{\Phi}_{\boldsymbol{\theta}}(\mathbf{f}_k) - \boldsymbol{\Phi}_{\boldsymbol{\theta}}(\mathbf{f}_j)\|_2^2 \leq \\ &(\|\boldsymbol{\Phi}_{\boldsymbol{\theta}}(\mathbf{f}_i) - \boldsymbol{\Phi}_{\boldsymbol{\theta}}(\mathbf{f}_k)\|_2 + \|\boldsymbol{\Phi}_{\boldsymbol{\theta}}(\mathbf{f}_k) - \boldsymbol{\Phi}_{\boldsymbol{\theta}}(\mathbf{f}_j)\|_2)^2 \ . \end{aligned} \tag{8}$$

Taking square roots on both sides results in the triangular inequality.

It is worth to compare this to the DNN structure discussed in [19] to learn the physical distance. It is illustrated in Figure 2. The learned distance satisfies only the property of symmetry. The other metric properties are not fulfilled.

The idea of using several copies of the same neural network is used in Triplet DNN based metric learning and Siamese DNN based dimesionality reduction. The loss function for such networks is different from (7). In contrast to normal loss functions that typically focus on classifying individual samples, contrastive loss in Siamese networks optimizes the distance between pairs of similar and dissimilar examples, while triplet loss focuses on relative distances between an anchor, a positive, and a negative sample to enforce better separation between similar and dissimilar data.

An important property of the metric learning structure in Figure 1 is that the two features are separately input to a DNN. This results in learning a low dimensional representation of the feature. This has an advantage when saving the feature. E.g., for fingerprinting localization there is no need to save CSI features of the offline phase to be used in the online phase. It is sufficient to save the learned low-dimensional feature representation $\mathbf{z}_i$. To find the k-nearest neighbours of a CSI feature in the online phase, its low dimensional feature needs to be found and then the feature distances are computed using the Euclidean distance on a lower dimensional space.

IV. Localization Framework

We focus on fingerprint-based wirless localization in a Non-Line-of-Sight (NLoS) scenario, which is more challenging than in LoS. The CSI features are measured at the network side and the location is estimated at the network side based on an machine leaning approach. The same framework can be used when the CSI feature is measured at the UE side and then fed back to the network.

We will consider both the raw covariance CSI feature and its logarithm. We consider a real vector representation of a CSI features which are Hermitian matrices, and stack the CSI features from B BSs. The length of the CSI vector then becomes BM^2.

A data set consisting of U fingerprints $\{\mathbf{f}_u\}$ is created in the offline phase, with the corresponding physical positions $\{\mathbf{p}_u\}$. The created data set will be used in the online phase to estimate the position $\mathbf{p}_{u'}$ of UE $u^{'} \notin \{1, \ldots, U\}$ using $\mathbf{f}_{u'}$ and a machine leaning algorithm.

The focus on this paper on having a small data set, which is of high importance from practical point view. In such a case, WKNN outperforms DNN based localization as reported in [6].

We aim to reveal the benefit of using a DNN based metric learning framework compared to a direct machine learning approach, as exemplified by WKNN with Euclidean distance and direct DNN based localization. The metric learning approach utilizes U^2 pairs of points to learn the metric, whereas direct DNN based localization utilizes only U points. Therefore metric learning mitigates the problem of having a small data set.

We will consider WKNN localization with Euclidean distance, WKNN localization with the learned metric in (6), and DNN fingerprint based localization.

A. Weighted K Nearest Neighbour Localization

WKNN regression is a non-parametric supervised learning method. The output is the weighted average of the k nearest neighbors. The neighbours are determined based on a distance function. The weight vector is computed based on a feature distance, using a weighting function such as the inverse distance, an exponential function or a Gaussian function.

We denote a feature vector by $\mathbf{f}_i$. Let $\mathbf{p}_i$ be the corresponding physical location in the data set, and $d(\circ, \circ)$ the distance measure between two features. Here, we use the Euclidean distance. To estimate the physical location corresponding to feature $\mathbf{f}_u$, the distances $d(\mathbf{f}_i, \mathbf{f}_u)$ to all points in the data set are computed, and the set $\mathcal{I}_u$ of the k feature points nearest to $\mathbf{f}_u$ are determined, with $|\mathcal{I}_u| = k$. The weight $\omega_{u,i}$ of UE $i \in \mathcal{I}_u$ when localizing UE u is

$$\omega_{u,i} = \frac{g\left(d(\mathbf{f}_i, \mathbf{f}_u)\right)}{\sum_{j \in \mathcal{I}_u} g\left(d(\mathbf{f}_j, \mathbf{f}_u)\right)}, \tag{9}$$

where function $g(d)$ maps a distance d to a similarity. The exponential function

$$g(d) = \exp\left(-\frac{d}{\tau}\right) ,$$

with tuning parameter $\tau > 0$ is the most commonly used function. The location corresponding to feature $\mathbf{f}_u$ is then estimated as:

$$\hat{\mathbf{p}}_u = \sum_{i \in \mathcal{I}_u} \omega_{u,i}\, \mathbf{p}_{u,i}\,, \tag{10}$$

where $\mathbf{p}_{u,i}$ is the location of neighbouring point i in $\mathcal{I}_u$.

WKNN with metric learning is based on using the trained network to learn the low-dimensional feature, i.e., to find the location of CSI feature $\mathbf{f}_i$. With the low-dimensional feature $\mathbf{z}_i = \mathbf{\Phi}_{\boldsymbol{\theta}}(\mathbf{f}_i)$ given by the DNN, the distance $\|\mathbf{z}_i - \mathbf{z}_u\|_2$ to all points in the data set is found. The set of k nearest neighbours $\mathcal{I}_i$ is then determined, the weight vector is computed and the location is estimated as in (10).

B. Direct DNN based Localization

We train a DNN to directly infer the location of a UE from the CSI feature, i.e.,

$$\hat{\mathbf{p}}_i = \mathbf{\Gamma}_{\boldsymbol{\theta}}(\mathbf{f}_i)\,,$$

where $\boldsymbol{\theta}$ is the vector of weight and bias values of the DNN. The DNN takes in a CSI feature and passes it through several fully connected layers until the output layer, which outputs the estimated position.

In the training phase, the loss function of the predicted value against the ground truth value is computed. The MSE is considered as the loss function. The training of the DNN takes place during an offline phase. The trainable parameters are updated by back-propagation. In the online phase, the DNN is used to predict the position for a given CSI feature.

V. Performance Evaluation and Complexity Analysis

To evaluate localization performance, we consider the statistics of the distance error, i.e., the difference of the predicted and ground truth locations in the test data set. We consider the 80th and 90th percentiles of the error and the Root Mean Squared Error (RMSE).

For the metric leaning framework, we consider in addition the correlation coefficient between the learned and the true distance. Pearson's correlation coefficient between a pair of random variables P and V is given as

$$\rho(P, V) = \frac{\mathbb{E}\left[(P-\mu_P)(V-\mu_V)\right]}{\sigma_P \sigma_V}\,, \tag{11}$$

where $\mathbb{E}$ is the expectation, μ_P and σ_P are the mean and standard deviation of P, respectively. The correlation is between -1 and 1.

When evaluting complexity, we consider online phase complexity only, which is crucial for real-time operation. In ν-digit computation, the complexity of an addition operation is $O(\nu)$ and a multiplication is $O(\nu^2)$. We neglect additions for simplicity.

In WKNN, the most computationally intensive step is the feature distance computation when finding the nearest neighbors. For Euclidean distance, the complexity is

$$\mathcal{C}(L_\mathrm{f}, U) = L_\mathrm{f} U\,,$$

where U is the data set size and L_f is the length of the feature.

In DNN, assuming a fully connected layer q has size L_q, the number of multiplications is a function of the number of layers and the number of neurons at each layer,

$$\mathcal{C} = \sum_{q=1}^{Q-1} L_q\, L_{q+1}\,,$$

where Q is the number of layers.

For metric learning with WKNN localization, the computation complexity includes the complexity of DNN to obtain the low dimension feature, i.e., $\mathbf{z}_i = \mathbf{\Phi}_{\boldsymbol{\theta}}(\mathbf{f}_u)$ and then using the Euclidean distance to measure the distance between $\mathbf{z}_i$ and $\mathbf{z}_u$.

TABLE I
Simulation Parameters

Parameter	Value	Parameter	Value
Center Freq.	3.5 GHz	Subcarrier Spa.	30 kHz
Scenario	InF-SL	Bandwidth	10 MHz
BS Height	1.5 m	UE Height	1 m
BS Array	8 ULA	UE Array	1
Num. of BSs	4		

TABLE II
DNN Structures for Different Frameworks

Framework	DNN Layers
Metric learning	$[256, 2048, 1024, 512, 256, 128, 64]$
Distance learning [19]	$[256, 2048, 1024, 512, 256, 128, 64, 1]$
DNN localization	$[256, 256, 128, 64, 2]$

VI. Simulation

We evaluate the localization performance in an NLoS environment, specifically an Indoor Factory Sparse Low (InF-SL) scenario of [22]. The simulation parameters are summarized in Table I. The environment layout consists of 4 BSs located at xy-coordinates $[-10,\, 10]$ m, $[-10,\, -10]$ m, $[10,\, 10]$ m and $[10,\, -10]$ m, where 2000 UEs are on a grid with 0.4 m spacing. The basis of evaluation is synthetic channel data generated with the QuaDRiGa simulator, considering large-scale and small-scale effects including multi-path fading [23]. We adopt the values for delay spread, angle-of-arrival and angle-of-departure distributions for the InF-SL scenario discussed in [22]. The log-covariance and log-power features are computed with 50 time samples.

We consider 200 points on a grid of 1.2 m for training, and 1800 points for testing, unless otherwise stated. We consider both the covariance and log covariance features and apply the following localization methods:

- WKNN based localization with the Euclidean distance.
- Metric-WKNN: WKNN based localization with metric learning, as illustrated in Figure 1. The DNN architecture consists of $[2048, 1024, 512, 256, 128, 64]$ layers.
- [19]-WKNN: WKNN based localization with the learned distance in [19] as shown in Figure 2. The DNN consists $[2048, 1024, 512, 256, 128, 64, 1]$ layers.

TABLE III
LOCALIZATION PERFORMANCE (TRAIN=200)

Feature	Method	80%	90%	RMSE
Covariance	WKNN	1.20	1.60	1.11
log-Covariance	WKNN	0.65	0.85	0.55
Covariance	DNN	1.00	1.36	1.24
log-Covariance	DNN	1.28	1.70	1.07
Covariance	Metric-WKNN	1.12	1.57	1.06
log-Covariance	Metric-WKNN	**0.36**	**0.65**	**0.49**
Covariance	[19]-WKNN	1.34	1.72	1.14
log-Covariance	[19]-WKNN	0.41	0.73	0.51

TABLE IV
LOCALIZATION PERFORMANCE CONSIDERING 20 NEAREST NEIGHBORS FOR EACH POINT (TRAIN=200)

Feature	Method	80%	90%	RMSE
Covariance	Metric-WKNN	1.08	2.07	0.87
log-Covariance	Metric-WKNN	**0.32**	**0.58**	**0.46**

- DNN based localization. The DNN takes the CSI feature and passes it through three fully connected layers of size $[256, 128, 64]$.

We consider an exponential weight function for WKNN with $\tau = 1$ after normalizing the feature distance by its maximum value, and consider $k = 5$ nearest neighbours.

Table II provides a summary of DNN structures for metric learning, for the distance in [19] and direct fingerprint localization. The rectified linear unit (ReLU) activation is used at the layers, except a linear activation is applied at the output layer.

We evaluate the localization performance using 80th and 90th percentiles and RMSE. The localization performance results for several approaches are summarized in Table III. The performance of WKNN with metric learning outperforms other approaches. The log-covariance feature outperforms the covariance feature. The gain in terms of 80th, percentile is 29 cm, as compared to WKNN localization without learning. The metric learning structure in this paper outperforms the distance proposed in [19].

As the performance of WKNN is based on the k-nearest distances, in order to improve the learning of short distances, we train the DNN metric structure by considering a subset of points, we select for each data point, the nearest 20 points, this results on 200×20 data points for learning the metric. The results are summarized on Table IV. The gain in terms of 80%, and RMSE are 37 cm and 11 cm, respectively, compared to WKNN localization without learning using the log-covariance feature.

TABLE V
LOCALIZATION PERFORMANCE CONSIDERING A LARGE DATA SET (TRAIN=1700)

Feature	Method	80%	90%	RMSE
Covariance	WKNN	0.56	0.72	0.38
log-Covariance	WKNN	**0.37**	**0.41**	**0.21**
Covariance	DNN	0.62	0.82	0.61
log-Covariance	DNN	0.67	0.86	0.40

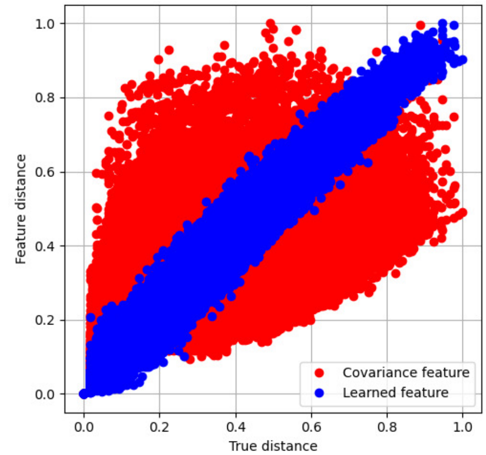

Fig. 3. Feature pairwise distance versus true pairwise distance for covariance feature and the learned covariance feature.

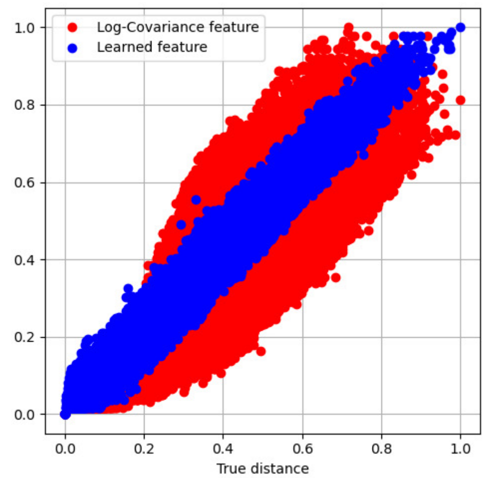

Fig. 4. Feature pairwise distance versus true pairwise distance for log-covariance feature and the learned log-covariance feature.

To evaluate the gain of metric learning in terms of reducing the size of the data set. We consider a data set of 1700 points in a grid of 0.4 m for the same environment. We apply WKNN and DNN localization. The performance results are summarized in Table V. As expected, having a large data set improves localization performance of DNN and WKNN. Comparing the results of WKNN with metric learning for the log-covariance feature using nearest 20 points for a data set of 200 points as in Table IV with DNN localization for a data set of 1700 points as in Table V, shows that the 80th and 90th percentiles of metric leaning with WKNN are much smaller. This means that WKNN with metric learning has the capability of exploiting a small data set to produce a quadratic number of samples.

To understand performance of the metric learning framework itself, we study the linear relation between CSI feature distance and true distance using correlation coefficient. We consider covariance feature, log-covariance feature and learned features based on covariance as well as log-covariance features. Figure 3

TABLE VI
NUMBER OF MULTIPLICATIONS IN MILLIONS

Euclidean-WKNN	Metric-WKNN	[19]-WKNN	DNN
0.066	3.33	1698.72	0.107

shows the relation between 1) the covariance feature distances and the true distances in red colour. The correlation coefficient is 0.56. 2) the learned feature distance for covariance feature and the true distances in blue colour. The correlation coefficient is 0.99. The learned feature has a much stronger correlation with the true distances.

Similarly, Figure 4 illustrates the log covariance feature distances in relation to the true distances (correlation is 0.81), with the associated learned feature distance again achieving a correlation of 0.99. These results demonstrate that while log covariance features have a better correlation with the true distances than covariance features, learned metric consistently achieve the highest correlation. This underscores that metric learning is significantly more effective than relying on raw covariance or log covariance features.

Table VI summarizes the computation complexity in millions of multiplications. Metric-WKNN localization has negligible complexity compared to [19]-WKNN localization.

VII. CONCLUSIONS

In this paper, we addressed predicting the physical distance between two locations based on their Channel State Information (CSI). We have considered a metric leaning framework using a deep neural network. The proposed approach benefits from the fact that the number of data points is squared. We have utilized the metric for Weighted K-Nearest Neighbours (WKNN) localization. We evaluated the performance of the learned metric using the correlation coefficient between true distance and the learned metric. A correlation coefficient of 0.99 can be achieved for the considered data set. We considered the localization performance of WKNN based on metric learning, WKNN based on Euclidean distance, and DNN based localization. For a small data set of a high dimension feature, the localization of DNN is limited. WKNN with metric learning showed a good performance in this case. In future work, we will investigate advance neural network structures such as convolution and transformers and address the scalability of the network when a large number of base stations is considered.

ACKNOWLEDGMENT

This work was supported by a CHIST-ERA grant for project CHASER (CHIST-ERA-22-WAI-01) by Research Council of Finland (RCF) grant 359837, and by RCF grant 345109. We acknowledge the computational resources provided by the Aalto Science-IT project.

REFERENCES

[1] R. Keating, M. Säily, J. Hulkkonen, and J. Karjalainen, "Overview of positioning in 5G New Radio," in *Proc. of ISWCS*, 2019, pp. 320–324.

[2] O. Kanhere and T. S. Rappaport, "Position location for futuristic cellular communications: 5G and beyond," *IEEE Commun. Mag.*, vol. 59, no. 1, pp. 70–75, 2021.

[3] P. Bahl and V. Padmanabhan, "RADAR: an in-building RF-based user location and tracking system," in *Proc. of INFOCOM*, vol. 2, 2000, pp. 775–784.

[4] P. Castro, P. Chiu, T. Kremenek, and R. Muntz, "A probabilistic room location service for wireless networked environments," in *Int. Conf. Ubiquitous Comput.* Springer, 2001, pp. 18–34.

[5] H. Zhang, Z. Wang, W. Xia, Y. Ni, and H. Zhao, "Weighted adaptive KNN algorithm with historical information fusion for fingerprint positioning," *IEEE Wirel. Commun. Lett*, vol. 11, no. 5, pp. 1002–1006, 2022.

[6] A. Sobehy, Éric Renault, and P. Muhlethaler, "CSI-MIMO: K-nearest neighbor applied to indoor localization," in *Proc. of ICC*, 2020, pp. 1–6.

[7] S. Liu, R. De Lacerda, and J. Fiorina, "Performance analysis of adaptive k for weighted k-nearest neighbor based indoor positioning," in *Proc. of VTC2022-Spring*, 2022, pp. 1–5.

[8] J. Torres-Sospedra, R. Montoliu, S. Trilles, Óscar Belmonte, and J. Huerta, "Comprehensive analysis of distance and similarity measures for Wi-Fi fingerprinting indoor positioning systems," *Expert Syst. Appl.*, vol. 42, no. 23, pp. 9263–9278, 2015.

[9] X. Li, H. Al-Tous, S. E. Hajri, and O. Tirkkonen, "Enhanced weighted K-nearest neighbor positioning," in *Proc. of VTC2024-Spring*, 2024, pp. 1–6.

[10] B. Berruet, O. Baala, A. Caminada, and V. Guillet, "Delfin: A deep learning based CSI fingerprinting indoor localization in IoT context," in *Proc. of IPIN*, 2018, pp. 1–8.

[11] B. Zhang, H. Sifaou, and G. Y. Li, "CSI-fingerprinting indoor localization via attention-augmented residual convolutional neural network," *IEEE Trans. Wireless Commun.*, 2023.

[12] X. Wang, X. Wang, and S. Mao, "Deep convolutional neural networks for indoor localization with CSI images," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 1, pp. 316–327, 2020.

[13] X. Li, H. Al-Tous, S. E. Hajri, and O. Tirkkonen, "Channel covariance based fingerprint localization," in *Proc. of VTC2024-Fall*, 2024, pp. 1–6.

[14] J. L. Suárez, S. García, and F. Herrera, "A tutorial on distance metric learning: Mathematical foundations, algorithms, experimental analysis, prospects and challenges," *Neurocomputing*, vol. 425, pp. 300–322, 2021.

[15] P. Moutafis, M. Leng, and I. A. Kakadiaris, "An overview and empirical comparison of distance metric learning methods," *IEEE Trans. Cybern.*, vol. 47, no. 3, pp. 612–625, 2017.

[16] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," in *Proc. of SIMBAD*, 2015.

[17] M. Alfaro, J. J. Cabrera, L. M. J. García, Ó. Reinoso, and L. Payá, "Hierarchical localization with panoramic views and triplet loss functions," *ArXiv*, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:269294039

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

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

[20] J. Sammon, "A nonlinear mapping for data structure analysis," *IEEE on Trans. Comput.*, vol. C-18, no. 5, pp. 401–409, 1969.

[21] E. Lei, O. Castaneda, O. Tirkkonen, T. Goldstein, and C. Studer, "Siamese neural networks for wireless positioning and channel charting," in *Proc. Allerton Conference on Communication, Control, and Computing*, 2019, pp. 200–207.

[22] 3GPP, "Study on channel model for frequencies from 0.5 to 100 GHz," 3rd Generation Partnership Project (3GPP), Technical Specification TS 38.901, Jan. 2018, version 14.3.0.

[23] S. Jaeckel, L. Raschkowski, K. Borner, and L. Thiele, "QuaDRiGa: A 3-D multi-cell channel model with time evolution for enabling virtual field trials," *IEEE Trans. Antennas Propag.*, vol. 62, no. 6, pp. 3242–3256, Jun. 2014.