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 users based on the Channel State Information (CSI) of wireless channels. The CSI of each user is measured at several multiantenna base stations. We consider a supervised metric learning framework using a neural network that ensures that the properties of a metric are fulfilled: zero distance between a point and itself, non-negativity, symmetry, and the triangle inequality. The training data set consists of CSI from pairs of points and their physical distance. As an example use case, we consider fingerprint localization, where creating large datasets is impractical. The metric can be learned from a small dataset because the number of training data pairs increases quadratically in the number of CSI-fingerprints. We use the learned metric for Weighted K-Nearest Neighbor (WKNN) localization, to find neighbors in the dataset and to compute the weighting vector. Simulation results show that the 80th percentile error can be improved by some 70% using the learned metric as compared to the Euclidean distance for WKNN regression.**

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

I. Introduction

Accurate localization is an important feature of modern wireless communication systems, enabling a wide range of location-based services. In outdoor environments, well-established localization technologies, such as the Global Navigation Satellite System (GNSS), generally meet the accuracy demands of most applications. However, achieving high-precision localization in indoor environments remains challenging due to the presence of multipath propagation and signal attenuation. The advent of Fifth Generation New Radio (5G NR) has significantly enhanced indoor cellular positioning with a broad set of standardized techniques [1], [2].

Among various indoor positioning methods, fingerprint-based localization has emerged as a viable solution [3], [4]. This approach 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, using a supervised machine learning 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 localization accuracy, particularly when the dataset is small, and its low computational cost [5]–[7]. The WKNN method shows better localization results than a DNN for a measured data set in [6]. The performance of WKNN is influenced by factors such as the choice of feature distance metric, the weight function, and the number of nearest neighbors. Typically the Euclidean distance and an exponential weight function are used. In [8], it is shown that the Sørensen distance outperforms other distances when RSS features are used. In [9] an enhanced weighting function and a neighbor selection criterion for WKNN localization are proposed. Additionally, different CSI features and DNN architectures have been explored for fingerprint-based localization [10]–[13].

In the literature, metric learning has been based on reducing intra-class and increasing inter-class distance in classification problems, leading to a non-convex optimization problem. Relaxations have been considered to obtain sub-optimal solutions [14], [15]. A Triplet neural network is used for parameterizing the metric in [16], and applied for localization and channel charting in [17], [18]. In [19], [20] Siamese neural network is used to learn a distance/ similarity for classifications. In [21], a Siamese neural network is used to parameterize Sammon's mapping and learn a 2D feature representation.

Unmanned aerial vehicle geometric localization is considered in [22], based on a DNN that maps the CSI feature distance between pairs of nodes to their physical distance. 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 triangle inequality. The proposed metric neural network structure has two branches with common weights similar to [19]–[21]. However, the loss function and input distance are different.

In this paper, we propose a DNN-based metric learning framework for wireless localization in challenging non-line-of-sight (NLoS) indoor factory environments. The main contributions are:

- **Metric Learning:** Our framework learns a distance that satisfies the axioms of non-negativity, symmetry and triangle inequality, addressing limitations in [22], and not limited to learning a metric for classifying data points into two categories [16], [19].
- **Enhanced Localization Performance:** By integrating the learned metric into WKNN localization, we demonstrate significant improvements over conventional Euclidean distance-based WKNN and direct DNN localization methods, particularly in challenging NLoS indoor environments.

The remainder of this paper is organized as follows: The system model is introduced in Section II, the metric learning problem in Section III and the localization framework

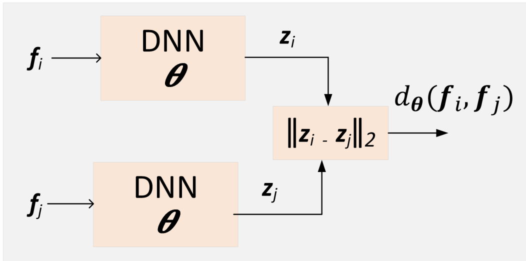

Fig. 1. DNN structure for metric learning consisting of two DNN with the same parameters $\boldsymbol{\theta}$. The output is a learned low dimensional feature. The structure is used to predict the physical distance.

in Section IV. Section V provides performance evaluation and complexity analysis, and Section VI simulation results. Section VII concludes the paper.

II. System Model

We consider a communication system with B Base Stations (BSs), each BS having M antennas, e.g., arranged as 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.

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 a Hermitian $M \times M$ matrix computed from S time samples as

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

Here, the impact of small-scale fading is averaged, making covariance-based features more robust to small-scale fading effects. The Eigen-decomposition of a Hermitian matrix $\mathbf{R}$ is

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

where $\mathbf{\Lambda} = \mathrm{diag}\left([\lambda_1, \ldots, \lambda_M]\right)$ a diagonal matrix of non-negative eigenvalues, and $\mathbf{U}$ is a unitary matrix. If the matrix is positive definite, with all Eigenvalues positive, the logarithm of the matrix becomes:

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

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 $\|.\|_{\mathrm{F}}$ 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.

A metric d on a set X is a function $d : X \times X \to \mathbb{R}$ 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 positive definite matrix $\mathbf{A}$ is searched for, such that the distance between features $\mathbf{f}_i$ and $\mathbf{f}_j$ is

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

If $\mathbf{A}$ 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]. A distance is sought for that brings similar samples closer, while it pushing away dissimilar ones. The problem can be formulated in various ways using different objective functions. If $\mathbf{A}$ is decomposed in terms of a $\mu \times J$-matrix $\mathbf{L}$ as $\mathbf{A} = \mathbf{L}^{\mathrm{T}}\mathbf{L}$, the distance becomes

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

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:

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

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

$$\mathcal{L}_{i,j} = \left(d_{\boldsymbol{\theta}}\left(\mathbf{f}_i, \mathbf{f}_j\right) - \|\mathbf{p}_i - \mathbf{p}_j\|_2\right)^2 \,, \tag{7}$$

in terms of the physical distance between physical locations $\mathbf{p}_i$ and $\mathbf{p}_j$ in the training data set. The DNN learns the physical distance between two users based on the their CSI.

It is easy to show that $d_{\boldsymbol{\theta}}(\mathbf{f}_i, \mathbf{f}_j)$ inherits the properties of a pseudometric directly from the Euclidean distance used in (6). From the definition, it is clear that it is non-negative and symmetric. The triangle inequality can be proved as follows. Defining $\mathbf{z}_i = \boldsymbol{\theta}(\mathbf{f}_i)$, we first add and subtract the same term and then use the Cauchy–Schwarz inequality:

$$\begin{aligned} \|\mathbf{z}_i - \mathbf{z}_j\|_2^2 &= \|\mathbf{z}_i - \mathbf{z}_k + \mathbf{z}_k - \mathbf{z}_j\|_2^2 \\ &\leq \left(\|\mathbf{z}_i - \mathbf{z}_k\|_2 + \|\mathbf{z}_k - \mathbf{z}_j\|_2\right)^2 \,. \end{aligned}$$

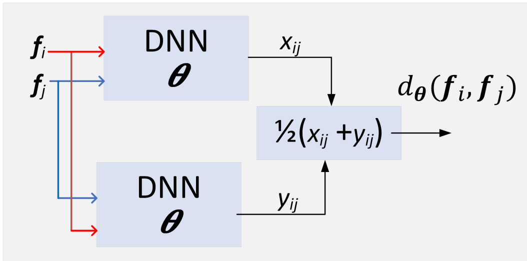

Fig. 2. DNN structure for learning the physical distance [22].

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

It is worthwhile to compare the proposed DNN structure to that discussed in [22], illustrated in Figure 2. The distance learned in [22] satisfies only the symmetry property. The other metric properties are not fulfilled.

The idea of using several copies of the same neural network is used in Siamese DNN [19]–[21], where two copies are used, and in triplet DNN [16]–[18], where three copies are used. The loss functions for these networks are different from (7). In [19], contrastive loss is considered. In [20], the cross-entropy loss function is used. In [21], the Sammon mapping loss function is used. In triplet DNNs, the loss enforces better separation between similar and dissimilar points, by considering an anchor, a positive, and a negative samples as inputs. For Siamese DNN several loss functions have been considered.

IV. Localization Framework

We focus on fingerprint-based wireless localization in a NLoS scenario, which is more challenging than in LoS. At the network side, the CSI features are measured and the location is estimated based on a machine leaning approach. The same method can be used if the CSI feature is measured at the UE side and fed back to the network.

We will consider both the raw covariance CSI feature and its logarithm. The $M \times M$ covariances of interest are Hermitian matrices, and can be represented as M^2-dimensional real vectors. We stack the covariance matrices from B BSs into a feature vector, which then becomes BM^2-dimensional.

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.

This paper focus 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.

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 or feeding back 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 neighbors 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. 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 Neighbor Localization

WKNN regression is a non-parametric supervised learning method. The output is the weighted average of the k nearest neighbors. The neighbors are determined based on a distance function.

Let $\mathbf{p}_i$ be the physical location in the data set corresponding to feature $\mathbf{f}_i$, and $d(\circ, \circ)$ the feature distance used. Here, for conventional WKNN 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

$$\omega_{u,i} = \frac{g\left(d(\mathbf{f}_i, \mathbf{f}_u)\right)}{\sum_{j \in \mathcal{I}_u} g\left(d(\mathbf{f}_j, \mathbf{f}_u)\right)}, \tag{8}$$

where function $g(d)$ maps a distance d to a similarity. The location corresponding to feature $\mathbf{f}_u$ is then estimated as:

$$\hat{\mathbf{p}}_u = \sum_{i \in \mathcal{I}_u} \omega_{u,i}\, \mathbf{p}_{u,i}\,, \tag{9}$$

where $\mathbf{p}_{u,i}$ is the location of neighboring 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 neighbors $\mathcal{I}_i$ is then determined, the weight vector is computed and the location is estimated as in (9).

B. Direct DNN based Localization

We compare WKNN with/without a learned feature distance with direct DNN localization. For this, we train a DNN to directly infer the location of a UE from the CSI feature:

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

where $\boldsymbol{\theta}$ is parameter vector 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

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		

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 80^{th} and 90^{th} percentiles of the error, as well as 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

$$\rho(P,V) = \frac{\mathbb{E}\left[(P-\mu_P)(V-\mu_V)\right]}{\sigma_P \sigma_V}\,, \tag{10}$$

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

When evaluating 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 $\mathbf{z}_u$.

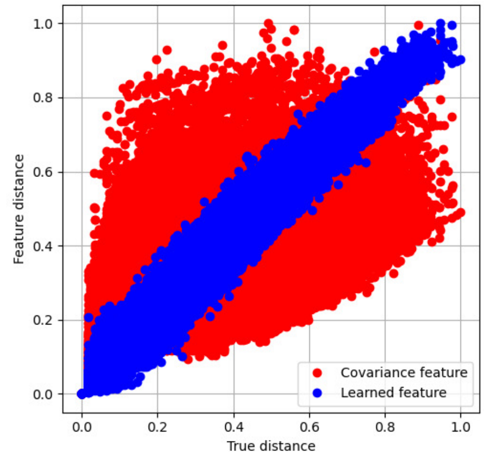

Fig. 3. Pairwise feature distance versus true pairwise distance for the covariance feature and the learned covariance feature.

VI. Simulation

We evaluate the localization performance in an NLoS environment, specifically an Indoor Factory Sparse Low (InF-SL) scenario of [23]. 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 [24]. We adopt the values for delay spread, angle-of-arrival and angle-of-departure distributions for the InF-SL scenario discussed in [23]. 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.

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

Figure 3 shows the relation between the covariance feature distances and the true distances in red, and the relation of the corresponding learned feature distances and the true distances in blue. The correlation coefficient without learning is 0.56, while the learned distance has a correlation coefficient of 0.99.

Similarly, Figure 4 illustrates the log-covariance feature distances in relation to the true distances (correlation 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, the learned metrics consistently achieve the highest correlation. This underscores that metric learning is significantly more effective than relying on raw covariance or log-covariance features.

To visualize the effect of metric learning, we compare the following positioning methods:

- WKNN based localization with the Euclidean distance.

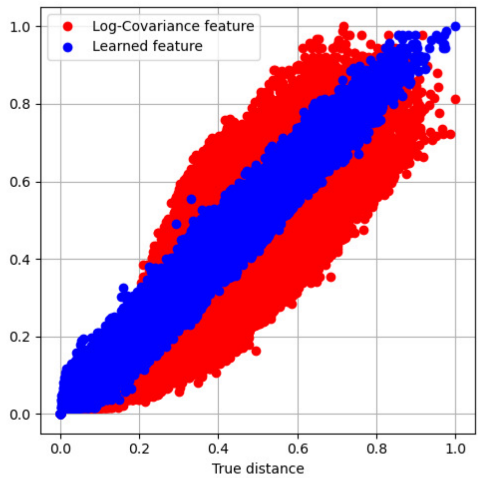

Fig. 4. Pairwise feature distance versus true pairwise distance for log-covariance feature and the learned log-covariance feature.

- 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.
- [22]-WKNN: WKNN based localization with the learned distance in [22] as shown in Figure 2. The DNN consists $[2048, 1024, 512, 256, 128, 64, 1]$ layers.
- 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 uniformed weight function for WKNN, and consider $k = 10$ nearest neighbors.

Table II provides a summary of DNN structures for metric learning, for the distance in [22] and direct fingerprint localization. Rectified linear unit (ReLU) activation is used at the layers, except for linear activation at the output layer.

We evaluate the localization performance using 80^{th} and 90^{th} percentiles and RMSE. The localization performances for several approaches are summarized in Table III. The result of WKNN with metric learning outperforms other approaches. The log-covariance feature outperforms the covariance feature. The gain in terms of 80^{th} percentile is 4 cm, as compared to WKNN localization without learning. The metric learning structure in this paper outperforms the distance proposed in [22].

Figure 5 illustrates the cumulative distribution function (CDF) of the errors for the metric-WKNN method utilizing log covariance features, evaluated for different values of k nearest neighbors. The results indicate that for $k = 3, 5, 10$, the CDF curves exhibit similar trends, suggesting comparable error distributions across these values. Notably, the configuration with $k = 5$ nearest neighbors demonstrates the best performance, achieving lower error values across the distribution.

The performance of WKNN is based on the precision of the k nearest distances. Thus to improve the metric learning for short distances we train the DNN metric structure by considering a subset of points— for each data point we select the nearest 20 points. This results in 200×20 data points for learning the metric. The resulting localization performance is summarized in Table IV. The gain in terms of the 80^{th} percentile and RMSE are 70 cm and 12 cm, respectively, compared to WKNN localization without learning using the 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 localization results are summarized in Table V. As expected, having a large data set improves localization performance of DNN and WKNN. WKNN with short-distance-enhanced metric learning in Table IV outperforms DNN localization for a data set of 1700 points with a wide margin, and provides comparable

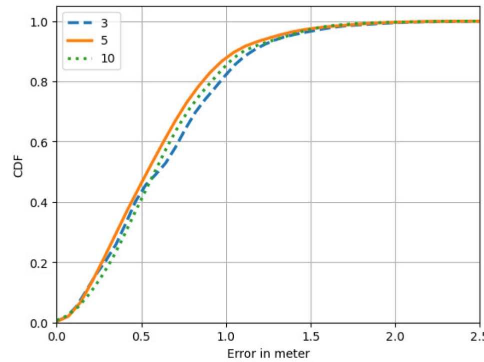

Fig. 5. Localization errors CDF for metric-WKNN based method using log-covariance features with different k=3,5, and 10 neighbors.

TABLE III
LOCALIZATION PERFORMANCE, 200 POINTS FOR TRAINING

Feature	Method	80%	90%	RMSE
Covariance	WKNN	1.90	2.52	1.16
log-Covariance	WKNN	1.01	1.24	0.56
Covariance	DNN	1.03	1.45	1.44
log-Covariance	DNN	1.11	1.45	0.64
Covariance	Metric-WKNN	1.94	2.43	1.15
log-Covariance	Metric-WKNN	**0.97**	**1.11**	**0.53**
Covariance	[22]-WKNN	2.00	2.34	1.09
log-Covariance	[22]-WKNN	0.97	1.13	0.55

TABLE II
DNN STRUCTURES FOR DIFFERENT FRAMEWORKS

Framework	DNN Layers
Metric learning	$[256, 2048, 1024, 512, 256, 128, 64]$
Distance learning [22]	$[256, 2048, 1024, 512, 256, 128, 64, 1]$
DNN localization	$[256, 256, 128, 64, 2]$

TABLE IV
LOCALIZATION PERFORMANCE WITH SHORT-DISTANCE-ENHANCED METRIC LEARNING. 200 POINTS, 20 NEAREST NEIGHBORS FOR EACH POINT

Feature	Method	80%	90%	RMSE
Covariance	Metric-WKNN	0.4	0.64	0.50
log-Covariance	Metric-WKNN	**0.31**	**0.51**	**0.44**

TABLE V
LOCALIZATION PERFORMANCE WITH LARGE DATA SET, 1700 POINTS FOR TRAINING

Feature	Method	80%	90%	RMSE
Covariance	WKNN	0.64	0.85	0.43
log-Covariance	WKNN	**0.38**	**0.46**	**0.22**
Covariance	DNN	0.62	0.82	0.61
log-Covariance	DNN	0.67	0.86	0.40

TABLE VI
NUMBER OF MULTIPLICATIONS IN MILLIONS

Euclidean-WKNN	Metric-WKNN	[22]-WKNN	DNN
0.066	3.33	1698.72	0.107

performance to WKNN with 1700 points. This shows how WKNN with metric learning has the capability of exploiting a small data set by producing a quadratic number of samples.

Table VI summarizes the computation complexity in millions of multiplications. Metric-WKNN localization has negligible complexity compared to [22]-WKNN localization.

VII. CONCLUSIONS

In this paper, we addressed predicting the physical distance between two locations based on their Channel State Information. We have developed a metric leaning framework using a deep neural network, which guarantees that all the axioms of a pseudo-metric are fulfilled by the learned metric. The proposed approach benefits from the fact that the number of data points is squared.

As an example application, we have used the learned metrics for Weighted K-Nearest Neighbor localization. A correlation coefficient of 0.99 between true distance and the learned metric can be achieved for the considered data set. Localization performance of WKNN based on metric learning was compared to conventional WKNN with Euclidean distance, and DNN-based localization. For a small data set of a high-dimensional features, DNN provides poor localization performance, while WKNN with metric learning outperformed the benchmark schemes with a wide margin. In particular, limiting the training in metric learning to enhance short distances leads to excellent localization performance.

In future work, we shall investigate advanced neural network structures such as convolutional networks and transformers for metric learning, and address the scalability of the network when a large number of base stations is considered.

ACKNOWLEDGMENT

This work was supported in part 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 VTC-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 VTC-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 VTC-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, M. Flores, 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] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, 2005, pp. 539–546 vol. 1.

[20] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *Proc. International Conference on Machine Learning*, 2015.

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

[23] 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.

[24] 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.