IEEE ComSoc | IEEE COMPUTER SOCIETY | IEEE Signal Processing Society | VTS | IEEE Transactions on Machine Learning in Communications and Networking

Received 15 January 2024; revised 5 June 2024; accepted 5 August 2024.
Date of publication 9 August 2024; date of current version 26 August 2024.

The associate editor coordinating the review of this article and approving it for publication was K. Seddik.

Digital Object Identifier 10.1109/TMLCN.2024.3441521

Deep Learning-Based Positioning With Multi-Task Learning and Uncertainty-Based Fusion

ANASTASIOS FOLIADIS[1,2], MARIO H. CASTAÑEDA GARCIA[1] (Member, IEEE), RICHARD A. STIRLING-GALLACHER[1] (Member, IEEE), AND REINER S. THOMÄ[2] (Life Fellow, IEEE)

[1]Munich Research Center, Huawei Technologies Duesseldorf GmbH, 80992 Munich, Germany
[2]Electronic Measurements and Signal Processing, Technische Universität Ilmenau, 98693 Ilmenau, Germany
CORRESPONDING AUTHOR: A. FOLIADIS (anastasios.foliadis@huawei.com)

ABSTRACT Deep learning (DL) methods have been shown to improve the performance of several use cases for the fifth-generation (5G) New radio (NR) air interface. In this paper we investigate user equipment (UE) positioning using the channel state information (CSI) fingerprints between a UE and multiple base stations (BSs). In such a setup, we consider two different fusion techniques: early and late fusion. With early fusion, a single DL model can be trained for UE positioning by combining the CSI fingerprints of the multiple BSs as input. With late fusion, a separate DL model is trained at each BS using the CSI specific to that BS and the outputs of these individual models are then combined to determine the UE's position. In this work we compare these different fusion techniques and show that fusing the outputs of separate models achieves higher positioning accuracy, especially in a dynamic scenario. We also show that the combination of multiple outputs further benefits from considering the uncertainty of the output of the DL model at each BS. For a more efficient training of the DL model across BSs, we additionally propose a multi-task learning (MTL) scheme by sharing some parameters across the models while jointly training all models. This method, not only improves the accuracy of the individual models, but also of the final combined estimate. Lastly, we evaluate the reliability of the uncertainty estimation to determine which of the fusion methods provides the highest quality of uncertainty estimates.

INDEX TERMS Deep learning, wireless positioning, late fusion, early fusion, multi-task learning, uncertainty estimation.

I. INTRODUCTION

ACCURATE user positioning is an enablers of several future services and technologies [1], [2], [3], [4] such as location-aware communication, vehicle to everything (V2X) applications, industrial internet of things (IIOT), cooperating robots, commercial applications, etc. For this purpose, radio-based positioning of user equipment (UE) in wireless communication networks can be considered [5]. Multiple base stations (BSs) deployed in such networks allow the collection of channel state information (CSI) over distributed links, which can be exploited for positioning of a UE. The CSI consists of the channel across the spatial and frequency domain, where the large number of antennas and large available bandwidth of current and future communication networks [4], e.g., fifth generation (5G) or upcoming sixth generation (6G), can provide a high angular and temporal resolution to enable high accuracy positioning.

Conventional radio-based positioning methods are generally model-based and usually follow a two-step approach. With CSI estimated at one BS [6] or at multiple BSs [7], relevant parameters or measurements e.g., path delay, angle of arrival (AoA), reference signal receive power (RSRP), time difference of arrival (TDoA), etc, are first determined to subsequently compute the UE's position in a second step. Recently, machine learning (ML) and artificial intelligence (AI)-based techniques have also been proposed for

radio-based UE positioning [8], [9], [10], [11], [12] which are primarily data-driven and not model-based. In particular, deep learning (DL) methods, particularly convolutional neural networks (CNNs) have shown promising results [13], [14], [15], [16], being able to achieve sub-meter accuracy. In such data-driven models, the CSI over subcarriers and antennas of a UE at a given position is considered as a fingerprint associated with the UE's position. By leveraging the ability of wireless networks to collect large amounts of data, a database of CSI fingerprints associated with different UE's positions along with the respective UE's position label can be constructed. With the DL-based positioning methods, a neural network (NN) can be trained on a given database, such that afterwards the NN can be employed for estimating a UE's position by providing the CSI of the UE as its input. Different types of fingerprints have been considered in the literature, including the received signal strength (RSS), the magnitude and/or phase of the CSI over subcarriers in the frequency domain and across antennas in the spatial domain [15], [16], [17], [18], [19].

With the CSI of a UE available across multiple BSs, early fusion or late fusion can be considered for the DL-based positioning methods [20], [21]. In early fusion, the CSI fingerprints from multiple BSs are collected and bundled together to constitute a single CSI fingerprint associated with the UE's position. Thus with early fusion, only one NN needs to be trained with a database comprising with fingerprints of the CSI across multiple BSs [20]. On the other hand, with late fusion, one NN is assumed at each BS where the CSI is considered as a fingerprint of the UE's location associated only with the given BS [21]. The NN associated with that BS is trained with a database of CSI fingerprints from that BS, enabling the NN to determine the UE's position based only on the CSI estimated by that BS. Afterwards, a final UE's position estimate is obtained by combining the position estimates obtained by the NNs across the multiple BSs [21], [22], e.g., with a weighted average.

The choice between early or late fusion generally depends on the application [23]. However, when considering changes in the UE-BS channel between the training phase and deployment phase, e.g., due to a blockage of the line of sight (LOS) between a UE and a BS, late fusion can benefit from uncertainty estimation [21]. In particular, the NN at each BS can be trained to estimate the uncertainty in the UE's position determined by each NN. This enables the late fusion approach, to determine the final position estimate for the UE considering the uncertainty of the multiple position estimates obtained across multiple BSs. In practice the most reliable position estimates have a larger impact in determining the final UE's position. Uncertainty estimation can be computed based on simple approaches like Monte Carlo Dropout (MCD) [24] and Deep Ensembles (DEs) [25], which characterize the uncertainty based on the variance of the positioning error obtained with multiple NNs, i.e., similar position estimates across the different NNs indicates lower uncertainty estimation. Uncertainty estimation methods have also been proposed in [26] and [27] to detect corrupted fingerprints.

Most conventional approaches to positioning require a strong line-of-sight (LOS) path and may be impaired in non-LOS (NLOS) conditions or when there is a strong multipath. Recent works such as [6] and [28] have shown how to take advantage of the multipath information for single anchor UE positioning but are limited to multiple-input multiple-output (MIMO) systems and require prior knowledge of the nature of the incoming paths (i.e., LOS or NLOS). On the other hand DL-based methods can still be employed in strong multipath scenarios and don't require multiple antennas at both receiver and transmitter. Despite this fact, with the multipath profile being susceptible to environmental changes, a DL model trained with CSI fingerprints from one environment may achieve a poor performance for the UE positioning in another environment [21], [29].

The lack of direct transferability of the knowledge acquired in one environment to other environments is one of the challenges of DL-based positioning [30]. The most straightforward way to address this is to retrain the NN from scratch with CSI fingerprints from the new environment, which may however be resource expensive and may not always be feasible.The resource intensive nature of position labeling that is required can be reduced by employing channel charting [31] and by considering distance metrics between CSI fingerprints to create a map of the deployment scenario [32], [33] using no or very few position labels. On the other hand, several approaches can be considered for improving the generalizability of a trained model to adapt it to environmental changes or to a new environment including transfer learning, domain generalization, multi-task learning and meta learning. With transfer learning, a previously trained model is used as an initial model that is fine-tuned with reduced training data from a new environment [19], [29], which allows to speed up the training and to improve the performance compared to training from scratch.

Furthermore, with multi-task learning (MTL) the aim is to jointly learn multiple models by training them while also sharing some or all of their parameters, thereby benefiting from regularization [34]. Consequently, by considering positioning in different environments as different tasks, the positioning across multiple environments can be improved. When training a MTL scheme the choice of the relative importance of each task has to be considered. The hardest to learn tasks should be weighted less, so that the model focuses more on tasks that are easier to learn. Based on the uncertainty of each task, a method was proposed in [35] that takes into account the importance of each task. This method, not only provides a way to tune the importance of different tasks but also simultaneously learns the uncertainty for each task, which as shown in [21] is beneficial for the DL-based position using CSI fingerprints.

Another approach aiming at improving the generalizability of NN models is meta-learning. With meta-learning, a model

is trained on multiple tasks or environments such that the minimization of the loss function in an unseen task is done more efficiently. Training is done by considering a meta-level objective such as the average positioning error across the multiple environments [30], [36]. Meta-learning aims at having a trained model that generalizes better not only across the trained tasks but also facilitates learning an unseen task with a lower number of training samples, in contrast to MTL which only aims at learning better the trained tasks.

Motivated by the two-step approach of conventional positioning methods, i.e., with parameter extraction from the CSI in a first step and a position determination in a second step, a two-part model trained with multi-task learning and a meta-level objective has been recently proposed in [37]. For UE positioning in different environments, i.e., different training tasks, different models are assumed with the first part of the models being common across all task and trained with CSI samples from all tasks (multi-task learning) aiming at minimizing the sum positioning error across all tasks (meta-level objective). The second part of the model of each task is trained to be environment specific by using only training data from each environment. The proposed approach in [37] is able to improve the positioning accuracy of the trained environments, as well as achieve a better generalizability when transferring the first part of the model and fine tuning the two-part model with CSI samples of a new environment.

A. CONTRIBUTIONS

As proposed in [35], MTL benefits from uncertainty estimation. The training in MTL can be improved by determining the relative weighting of the losses of each task based on the associated uncertainty estimate [35]. For this reason, in this paper we combine the results from [21] and [37] to benefit from the MTL of different positioning tasks and from late fusion using uncertainty estimation. For a setup with multiple BSs and considering the positioning of a UE using each BS as a separate task, we show that employing a MTL scheme with uncertainty estimation and late fusion achieves high positioning accuracy. Additionally, even though this is outside the scope of the current paper, it was shown in [37] that a model trained with the MTL scheme can be further used for transfer learning in a new environment, reducing the time and amount of data that needs to be gathered.

Moreover, we extend the work in [21] by employing a method described in [38] for sensor fusion that takes into account the possibility that one or more sensors may be spurious. In the case of DL-based positioning, a model estimate could be spurious if the purported uncertainty is low but the real error is high. We employ this method in a late fusion scheme and show that it is beneficial in improving the positioning accuracy especially in dynamic environments.

Lastly, we aim not only to minimize the positioning error, but also evaluate the reliability of the uncertainty estimation. It would be beneficial if the estimated uncertainty truly reflects the model's uncertainty about the current measurement, such that a high uncertainty should indicate high positioning error and vice-versa. To evaluate the quality of uncertainty estimates we consider the area under sparsification error (AUSE) metric [27], [39]. In addition to the AUSE metric, we evaluate the integrity of the positioning results with respect to the integrity risk (IR) which is used in global navigation satellite system (GNSS) applications and has been recently proposed in the Third Generation Partnership Project (3GPP) as a positioning key performance indicator (KPI) for 5G positioning [40].

The paper is structured as follows. In Section II the considered system model is described along with the different types of fusion and the MTL scheme is introduced. In Section III the simulation setup is described and the DL-model structure. The results and conclusion are then presented in sections IV and V respectively.

II. SYSTEM MODEL

We consider an uplink setup with N_B BSs each with N_R receive antennas and a single transmit antenna at the UE. The UE transmits a reference signal on N_C subcarriers within an orthogonal frequency division multiplexing (OFDM) symbol. The received uplink signal is used to estimate the CSI matrix between UE and each BS. The estimated channel at the n-th BS over the N_C subcarriers is described as:

$$\tilde{\boldsymbol{H}}_n = [\tilde{\boldsymbol{h}}_0^n, \tilde{\boldsymbol{h}}_1^n, \ldots, \tilde{\boldsymbol{h}}_{N_C-1}^n] \in \mathbb{C}^{N_R \times N_C}, \tag{1}$$

where $\tilde{\boldsymbol{h}}_l^n \in \mathbb{C}^{N_R \times 1}$ is a column vector that describes the estimated uplink channel between the UE and the N_R antennas of the n-th BS at the l-th subcarrier. The estimated channels can be considered as a unique fingerprint of the position of the UE and depend on the multipath between the UE and each BS. To transform the raw complex CSI data to meaningful inputs for the NN, we stack the matrices $\Re\{\tilde{\boldsymbol{H}}_n\}$ and $\Im\{\tilde{\boldsymbol{H}}_n\}$ in the third dimension to obtain a new real-valued 3D matrix $\boldsymbol{H}_n \in \mathbb{R}^{N_R \times N_C \times 2}$. The symbols $\Re\{\cdot\}$ and $\Im\{\cdot\}$ denote the real and imaginary values of each of the matrix elements respectively. The values of each matrix are then normalized in the range [0, 1]. This transformation is a widely adopted practice in the literature for AI positioning using fingerprints as it allows the network to learn from both the magnitude and phase information, which are crucial for exploiting the multipath propagation effects captured by CSI [16], [19], [41]. We input the matrix $\boldsymbol{H}_n$ to the DL-model without applying manual feature engineering and we leverage the model's ability to autonomously learn relevant features from the data [42].

A. DL BASED POSITIONING WITH FINGERPRINTS

Deep learning based localization using CSI fingerprints as inputs consists of two phases, namely the training and the deployment phase, which are often alternatively termed as offline and online phases, respectively. During the training phase, CSI fingerprints are collected throughout the area of interest along with a label corresponding to the UE position associated with each CSI fingerprint. In order to collect

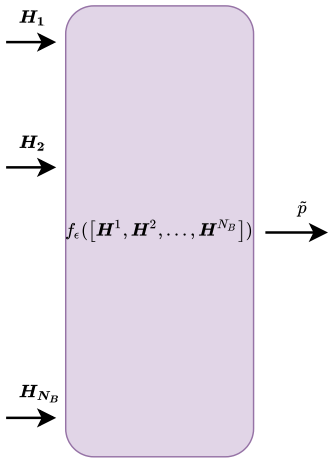

FIGURE 1. Early fusion.

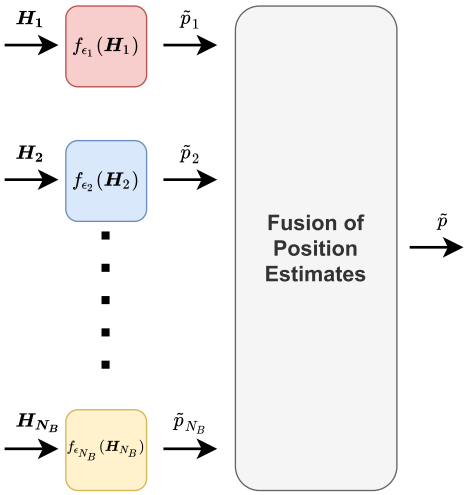

FIGURE 2. Late fusion.

fingerprints along with their labels, the use of positioning reference units (PRUs) can be employed, which consist of a device with known position, i.e., obtained with another positioning method or with sensors [43]. Without loss of generality, we assume that the UEs lie on a two dimensional plane. Subsequently, the CSI fingerprints and the position labels $\boldsymbol{p} = [x, y] \in \mathbb{R}^2$ are used to train the parameters ϵ of a neural network (NN) $f_\epsilon(.)$. Training is accomplished by minimizing the mean squared error between the position labels and the output of the NN with the labeled CSI fingerprints as input. Eventually, the trained NN is then used during the deployment phase to estimate the position $\tilde{\boldsymbol{p}}$ of a UE based on the estimated CSI fingerprint, where $\tilde{\boldsymbol{p}} = [\tilde{x}, \tilde{y}] \in \mathbb{R}^2$ is the position estimate for the UE.

The key idea behind positioning with CSI fingerprints is that the CSI for each position is considered unique for that specific position. This stems from the fact that the channel between UE and BS is a rich source of information since it is influenced by various environmental factors such as walls objects or other obstacles. All this information is indirectly incorporated into the multipath propagation of the channel, which includes direct paths (LOS) and indirect paths (NLOS), and is extracted during the training phase of the NN. Consequently, positioning using fingerprints is part of modern positioning techniques such as [28], which leverages both LOS and NLOS paths. Additionally, as shown in [44], there is not necessarily a need for a LOS path at all since NLOS paths already contain information that can make the fingerprints unique and useful for positioning. The basic assumption is that the propagation environment should not significantly change between the training and deployment phases since that would degrade the performance of the NN.

Two different approaches for positioning using CSI fingerprints from multiple N_B BSs can be considered [21], namely early and late fusion.

1) EARLY FUSION

In early fusion, a single DL-model is trained for the UE positioning, having as input the concatenation of the CSI fingerprints from all BSs, i.e., the single NN model $f_\epsilon(\boldsymbol{H}) = \tilde{\boldsymbol{p}}$ where $\boldsymbol{H} = [\boldsymbol{H}_1, \boldsymbol{H}_2, \ldots, \boldsymbol{H}_{N_B}] \in \mathbb{R}^{(N_B \cdot N_R) \times N_C \times 2}$. Although this is a straightforward way to combine the information from all BSs and perform localization using a DL-model, it has some disadvantages. Firstly, a large signaling overhead is required in order to transmit the relevant CSI data to a central server which has the single NN model and second, if the setup changes (e.g. a BS is removed), then a new NN model has to be trained from scratch. A block diagram of early fusion is shown on Fig. 1.

2) LATE FUSION

With late fusion, a UE's position estimate is determined based on the CSI fingerprint at each BS. For this purpose, a separate NN model is trained at each BS based on the CSI at each BS as input. The parameters ϵ_n of the model $f_{\epsilon_n}(\boldsymbol{H}_n) = \tilde{\boldsymbol{p}}_n$ of the n-th BS are trained, with the CSI measurements $\boldsymbol{H}_n$ of that BS. During deployment, the position estimates obtained across all the BSs are appropriately combined to produce a single position estimate. This type of fusion is shown in Fig. 2. We refer to this type of late fusion as, single task learning (STL) late fusion, since each DL-model is trained on a single specific task, namely the task of positioning based on CSI data from a specific BS.

Compared to early fusion, this method necessitates much less traffic for the network. The reason for that is that only the output of each of the models needs to be collected instead of the whole CSI fingerprints as in the case of early fusion. On the other hand, an appropriate model for the combination of the multiple estimates has to be developed. In this paper we built upon the work in [21] and propose and compare methods to appropriately combine the position estimates considering the uncertainty.

B. UNCERTAINTY ESTIMATION

Normally, for DL based positioning with CSI fingerprints, the parameters ϵ of the NN are optimized with respect to the mean

squared error (MSE) loss:

$$\mathcal{L}(f_\epsilon) = \mathcal{L}_x + \mathcal{L}_y, \tag{2}$$

where $\mathcal{L}_x = E[|\tilde{x} - x|^2]$, $\mathcal{L}_y = E[|\tilde{y} - y|^2]$ and $E[.]$ denotes the mean value over the samples in the training set.

The drawback of using such a loss function is that the model does not acquire any knowledge about the uncertainty that is present in the measurements. In the following, we discuss different types of uncertainties.

1) ALEATORIC UNCERTAINTY

The data dependent uncertainty is called aleatoric uncertainty and it reflects the uncertainty that a measurement has about the specific task. In a case of positioning using CSI fingerprints, a particular CSI measurement would have high uncertainty if it has low receive SNR for example. This type of aleatoric uncertainty, i.e. instance-dependent uncertainty, is called heteroscedastic uncertainty. Since the aleatoric uncertainty in positioning using fingerprints is data-dependent, it can also be learned from the data. In [45] a modification to the MSE loss was proposed in order to train a model to simultaneously calculate the position and the aleatoric uncertainty of the current position estimate. The loss function which shall be minimized with respect to the model parameters ϵ is the negative log-likelihood (NLL) function:

$$\mathcal{L}'(f_\epsilon) = \frac{1}{2(\sigma_x^\alpha)^2}\mathcal{L}_x + \frac{1}{2(\sigma_y^\alpha)^2}\mathcal{L}_y + \log(\sigma_x^\alpha \sigma_y^\alpha), \tag{3}$$

where σ_x^α and σ_y^α are the aleatoric uncertainties for the outputs x and y respectively.

Subsequently, the output of the model has to be modified to include the learned aleatoric uncertainty $\boldsymbol{\sigma}^\alpha = [\sigma_x^\alpha, \sigma_y^\alpha]$ of $\tilde{\boldsymbol{p}} = [\tilde{x}, \tilde{y}]$, i.e., $f'_\epsilon(\boldsymbol{H}) = [\tilde{x}, \tilde{y}, \sigma_x^\alpha, \sigma_y^\alpha]$. The modified loss function $\mathcal{L}'(f_\epsilon)$ consists firstly of two regression terms that describe the inverse relationship between the MSE loss of each estimated coordinate and its corresponding aleatoric uncertainty. When the MSE loss for a particular coordinate cannot be minimized further, the model increases the respective aleatoric uncertainty to compensate. On the other hand the aleatoric uncertainty remains low for instances where the MSE loss is low. The last term is a regularization term that is used to limit the infinite increase of the outputs σ_x^α and σ_y^α.

2) EPISTEMIC UNCERTAINTY

Aleatoric uncertainty is not the only type of uncertainty present in a DL model. The other type is called epsitemic uncertainty and it accounts for uncertainty in the model's parameters [45]. Estimates with high epistemic uncertainty indicate that the input comes from a distribution that was not learned by the model. In a DL localization model with fingerprints, the epistemic uncertainty would be high for a region in space where no data were collected or when a CSI measurement was corrupted.

In [24], an approach for capturing a model's epistemic uncertainty called Monte Carlo dropout (MCD) was introduced. When employing dropout, random neurons in every weight layer of the deep learning model are deactivated with a predefined probability. Typically, dropout is utilized solely for training purposes as a regularization technique [42], but with MCD, this same dropout probability is retained even during the deployment phase. Each successive forward pass through the deep DL model with MCD generates a unique configuration, and conducting multiple forward passes is akin to sampling from an approximate posterior distribution of the model's parameters given the dataset [24]. The variance of the estimates from the different model configurations during these forward passes serves as an indicator of the model's epistemic uncertainty.

After T forward passes, the combined aleatoric and epistemic uncertainty of the coordinate x is [45]:

$$\sigma_x = \frac{1}{T}\sum_{t=1}^{T}\tilde{x}_t^2 - \left(\frac{1}{T}\sum_{t=1}^{T}\tilde{x}_t\right)^2 + \sigma_x^\alpha, \tag{4}$$

where t indicates the current forward pass and $\tilde{x}_t$ indicates the estimate of the coordinate x at the t-th forward pass. The combined aleatoric and epistemic uncertainty for the y position coordinate is calculated similarly.

Even though the epistemic uncertainty estimation is an efficient way for the DL model to report on its own knowledge about the current measurement, it is not always accurate. There are cases where the epistemic uncertainty is low but the mapping to the position is highly inaccurate. In those instances the model would provide a spurious estimate which has to be identified and eliminated from the fusion scheme. For the late fusion in [21], the method employed to combine the results from the different estimates is based on the assumption that each estimate follows a known Gaussian distribution, whose variance corresponds to the estimated combined aleatoric and epistemic uncertainty. As this assumptions does not always hold, we take into account such model inconsistencies by incorporating a method described in [38] to fuse measurement from multiple sensors. The basic idea of this method is to weigh less the estimate that is most inconsistent with the other estimates. We should note that this method leverages multiple position and uncertainty estimates from different BSs, and therefore can only be employed in a late fusion scheme as described next.

In our setup we consider N_B different models, corresponding to the models trained at each of the N_B BSs. The authors of [38] assume that the probability that a measurement from the n-th sensor $n \in [1, \ldots, N_B]$, is not spurious with probability

$$p_n = \exp\left(\frac{-(x - \tilde{x}_n)^2}{\alpha_n}\right) \tag{5}$$

where x is the true state, $\tilde{x}_n$ is the estimate of the n-th sensor and α_n is a parameter that depends on the variances of each separate model and the difference between the output of the

n-th model with respect to other sensors:

$$\alpha_n = \frac{b_n}{\prod_{l=1,l\neq n}^{N_B}(\tilde{x}_n - \tilde{x}_l)^2} \tag{6}$$

where b_n is a hyperparameter.

From α_n we see that when the estimate of the n-th model is very different from the estimates of the other models, i.e., $\prod_{l=1,l\neq n}^{N_B}(\tilde{x}_n - \tilde{x}_l)^2 \to \infty$, then $\alpha_n \to 0$ and subsequently $p_n = 0$, meaning that the n-th estimate is definitely spurious. On the other hand when the n-th estimate largely agrees with the other estimates, i.e., $\prod_{l=1,l\neq n}^{N_B}(\tilde{x}_n - \tilde{x}_l)^2 \approx 0$ then it follows that $p_n \approx 1$, which means that the n-th estimate is not spurious. To reflect this intuition he authors of [38] show that the variance of the n-th model can be modified from $\sigma_{x,n}$ to:

$$\sigma'^2_{x,n} = \frac{{\sigma_{x,n}}^2 b_n^2}{b_n^2 - 2\sigma_{x,n}^2 \prod_{l=1,l\neq n}^{N_B}(\tilde{x}_n - \tilde{x}_l)^2} \tag{7}$$

where it must hold that $b_n^2 > 2\sigma_{x,n}^2 \prod_{l=1,l\neq n}^{N_B}(\tilde{x}_n - \tilde{x}_l)^2$, so that $\sigma'_n > 0$. In our model we choose:

$$b_n^2 = 2\sigma_{x,n}^2 \prod_{l=1,l\neq n}^{N_B} (\tilde{x}_n - \tilde{x}_l + \lambda)^2 \tag{8}$$

where λ is a small valued hyperparameter.

By choosing the parameter b_n as such, we make sure that when $\prod_{l=1,l\neq n}^{N_B}(\tilde{x}_n - \tilde{x}_l)^2 \to \infty$, then $\sigma'_n \to \infty$, reflecting very high uncertainty. On the other hand, when $\prod_{l=1,l\neq n}^{N_B}(\tilde{x}_n - \tilde{x}_l)^2 \approx 0$, then $\sigma'_n \approx \sigma_n$.

C. MULTI TASK LEARNING

When considering the late fusion approach, i.e. a separate NN for each of the BSs, we propose sharing some parameters across models of different BSs as described in [37]. The n-th model which corresponds to the n-th BS $f_{\theta,\epsilon_n}(\boldsymbol{H}_n)$ is parametrized by the common parameters θ and the BS specific parameters ϵ_n. By defining $f_{\theta,\epsilon_n}(\boldsymbol{H}_n) = g_{\epsilon_n}(\phi_\theta(\boldsymbol{H}_n))$ and training the parameters θ only on data from multiple BS we are forcing θ to be the same, regardless of the input data. This means that we can deal with the training of the different models as a MTL scheme and jointly minimize the loss for the positioning with the model at each BS. MTL with parameter sharing allows for information flow between tasks which eventually may help each individual task [46]. The block diagram of a the MTL scheme considered in this work is shown in Fig. 3.

This method of training is not possible when considering early fusion, since there is only one available model. Furthermore, when comparing STL late fusion to MTL late fusion we see that MTL requires the data from all BSs to be collected in order to train the models since the models share some parameters. For STL late fusion each BS trains its own model and then only shares the result of the model so there is no need to share input data between them. However, the fact that the models in MTL late fusion share parameters can enable training by means of federated learning [47]. Federated learning refers to the technique whereby multiple nodes can train a model by partially training it locally and then sharing the model's parameters instead of sharing the data. This method can reduce data transfer requirements between nodes and also preserve privacy. Federated learning is also not applicable in the early fusion case since no single BS is able to do partial training on the model (see Fig. 1) as the it needs CSI data from all BSs on its input to predict a single UE position.

The naive approach of optimizing a MTL scheme is to minimize a linear sum of the loss for each individual task, i.e. for the positioning at each BS. Thus, the loss for the training of the models in the MTL scheme would be would be:

$$\mathcal{L}_{\text{MTL}}(f_{\theta,\epsilon_1}, f_{\theta,\epsilon_2}, \ldots, f_{\theta,\epsilon_{N_B}}) = \sum_{n=1}^{N_B} \mathcal{L}_n(f_{\theta,\epsilon_n}), \tag{9}$$

where $\mathcal{L}_n(f_{\theta,\epsilon_n})$ is the MSE loss of the n-th model as described in eq. (2). The authors of [35] observed that by weighting each task appropriately the performance of each task can be greatly improved and proposed to set the weighting based on the aleatoric uncertainty [45] of each task.

In the context of DL based localization using fingerprints each model is generates a 2-dimensional position estimate. By assuming that each output of the n-th DL model follows a Gaussian distribution $\mathcal{N}(\tilde{x}_n, {\sigma_{x,n}}^2)$, and similarly for the y variable, the authors of [45] derive the minimization objective for the multi task learning as follows:

$$\begin{aligned}
&\mathcal{L}'_{\text{MTL}}(f_{\theta,\epsilon_1}, f_{\theta,\epsilon_2}, \ldots, f_{\theta,\epsilon_{N_B}}) \\
&= \sum_{n=1}^{N_B} \mathcal{L}'_n(f_{\theta,\epsilon_n}) \\
&= \sum_{n=1}^{N_B} \left[\frac{1}{2(\sigma^\alpha_{x,n})^2} \mathcal{L}_{x,n} + \frac{1}{2(\sigma^\alpha_{y,n})^2} \mathcal{L}_{y,n} + \log(\sigma^\alpha_{x,n}\sigma^\alpha_{y,n}) \right]
\end{aligned} \tag{10}$$

where $\mathcal{L}'_n(f_{\theta,\epsilon_n})$ is the NLL loss of the n-th model as defined in (3) and it includes each model's aleatoric uncertainty $\sigma^\alpha_{x,n}$ and $\sigma^\alpha_{y,n}$ which is implicitly weighing the losses for each task.

D. LATE FUSION WITH UNCERTAINTY ESTIMATION

The assumption that the outputs follow a Gaussian distribution, for which we have estimates of the mean value and variance, can be leveraged during the data fusion process. The N_B estimates all refer to a single UE's position in a 2-dimensional plane and the fused probability distribution can be calculated using Bayesian inference [48]. The resulting maximum likelihood (ML) fused estimate $\tilde{\boldsymbol{p}} = [\tilde{x}, \tilde{y}] \sim \mathcal{N}([\tilde{x}_{\text{ML}}, \tilde{y}_{\text{ML}}], \begin{bmatrix} \sigma_x^2, 0 \\ 0, \sigma_y^2 \end{bmatrix})$ has a variance of:

$$\sigma_x^2 = \frac{1}{\sum_{n=1}^{N_B} 1/\sigma_{x,n}^2}, \quad \sigma_y^2 = \frac{1}{\sum_{n=1}^{N_B} 1/\sigma_{y,n}^2} \tag{11}$$

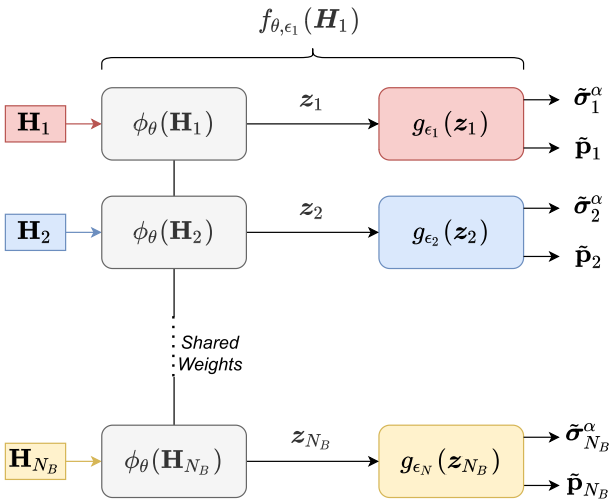

FIGURE 3. Two-part models with multi-task learning. Models' output comprise the UE's position estimate and the aleatoric uncertainty (when considered).

and a mean value:

$$\tilde{x}_{\text{ML}} = \frac{\sum_{n=1}^{N_B} \tilde{x}_n/\sigma_{x,n}^2}{1/\sigma_x^2},$$
$$\tilde{y}_{\text{ML}} = \frac{\sum_{n=1}^{N_B} \tilde{y}_n/\sigma_{y,n}^2}{1/\sigma_y^2} \tag{12}$$

and similarly for $\tilde{y}$. This type of estimate weighting is called inverse variance weighting. The value of $\sigma_{x,n}$ is given by (4) in the case of MCD uncertainty estimation, or (7) when considering spurious estimates (SP). Additionally, if we don't take into account the uncertainty of each estimate we can consider $\sigma_{x,n} = \sigma_{y,n} = 1$ which this results to a simple averaging of the position estimates.

E. QUALITY OF UNCERTAINTY ESTIMATION

After presenting different ways to calculate the uncertainty for each estimate, we now discuss how to assess the quality of these uncertainty estimates. As shown in the previous section, instead of providing a single position estimate, each model provides a different probability distribution for each individual input, for which the variance corresponds to the uncertainty. Normally to assess whether the output of the model indeed conforms to a probability distribution we would repeatedly produce samples from the model for a single input, calculate the empirical mean and variance and determine how close they are to the estimated model's distribution. However, in the context of localization using CSI fingerprints we have at most a couple of CSI samples for a given location, therefore any empirical calculation would be unreliable. Instead we use a method to determine the reliability of the uncertainty estimation process which is called the area under sparsification error curve (AUSE).

1) AREA UNDER SPARSIFICATION ERROR CURVE

The idea behind this metric is to use the so-called sparsification plots as a quality metric and the sparsification error [39].

Before defining the sparsification error we first need to define the oracle error. We define an array of the errors of each position sample as:

$$e = \left[||\tilde{x}_0 - x_0||_2, ||\tilde{x}_1 - x_1||_2, \ldots, ||\tilde{x}_{N_{\text{test}-1}} - x_{N_{\text{test}-1}}||_2\right] \tag{13}$$

where $\tilde{x}_i$ is the estimated value of the i-th sample and x_i is the real value and N_{test} is the number of samples in the test set. We also define the function sort(.) which is used to sort the elements of an array in descending order, such that $e' = \text{sort}(e)$. With that, the oracle error can be defined as:

$$\text{O}_N = \sqrt{\frac{\sum e'_{N:N_{\text{test}}-1}}{N_{\text{test}} - N}}, \tag{14}$$

where $e'_{N:N_{\text{test}}-1} = [e'_N, e'_{N+1}, \ldots, e'_{N_{\text{test}}-1}]$ and $1 \leq N \leq N_{\text{test}}$. The value O_N is decreasing monotonically with N since the errors are removed from the array e' in decreasing order.

To define the sparsification error, we first define an uncertainty vector $s = [\sigma_0, \sigma_1, \ldots, \sigma_{N_{\text{test}}-1}]$ and also, the function $\text{argsort}_s(.)$ which sorts the elements of an array with respect to the descending order of the elements of the array s, and $e^S = \text{argsort}_s(e)$. With that, the sparsification error is defined as:

$$S_N = \sqrt{\frac{\sum e^S_{N:N_{\text{test}}-1}}{N_{\text{test}} - N}}, \tag{15}$$

The sparsification error shows how much the estimated uncertainty coincides with the true errors on the test set. By removing gradually from the test set the estimates with the highest uncertainty and if the estimated uncertainty is of high quality the mean error S_N should decrease monotonically when increasing N. We compare S_N to O_N, by calculating the curve under the function $S_N - O_N$ for $N \in [1, N_{\text{test}} - 1]$. The value AUSE is calculated as:

$$\text{AUSE} = \frac{\sum_{N=1}^{N_{\text{test}}-1} S_N - O_N}{N_{\text{test}-1}}. \tag{16}$$

A small value indicates that the sparsification error is close to the oracle error, meaning that the uncertainty estimation is a good indicator for the actual error in the test set.

2) INTEGRITY RISK

We additionally use the integrity risk (IR) metric which was recently proposed by 3GPP as a key performance indicator for positioning integrity [40]. Normally, if the uncertainty is high, the system should give a warning that the respective error is also high. The integrity risk is defined as the probability that the unknown positioning error exceeds an application specific alert limit (AL) without warning. The available information from each user is the position and the uncertainty estimate, therefore we define an indicator function $\mathbb{1}_{\text{AL}}[||\boldsymbol{\sigma}_i||_2]$ which gives the aforementioned warning when the euclidean norm of the uncertainty $||\boldsymbol{\sigma}_i||_2$ vector of the i-th measurement is

larger than some threshold γ which corresponds to an position error equal to the AL:

$$\mathbb{1}_{\text{AL}}[||\boldsymbol{\sigma}_i||_2] = \begin{cases} 1 & \text{for } ||\boldsymbol{\sigma}_i||_2 \leq \gamma \\ 0 & \text{for } ||\boldsymbol{\sigma}_i||_2 > \gamma \end{cases}. \quad (17)$$

The IR is then defined as:

$$\text{IR} = \frac{\sum_{\{i|\mathbb{1}_{\text{AL}}(||\boldsymbol{\sigma}_i||_2)\}} \mathbb{1}_{\text{AL}}[e_i]}{N_{\text{test}}}, \quad (18)$$

In words, the IR is calculated as the ratio of samples that exceeded the AL but no warning was given, to the total number of test samples.

F. DATABASE DESCRIPTION

To evaluate our proposals we use the Dichasus channel measurements described in [49], that were collected at four antenna arrays distributed on the corners of an industrial area shown in Fig. 4. Each of the antenna arrays consists of a 4×2 uniform rectangular array (URA) with vertical and horizontal antenna spacing of half a wavelength. The measurements in [49] were collected with a single-antenna UE transmitting an OFDM signal in the uplink with a bandwidth of 50 MHz and with a pilot sent at every tenth subcarrier out of 1024 subcarriers, i.e., $Nc = 103$. The carrier frequency is $f_C = 1.272$ GHz. The ground truth positions are measured with a tachymeter robotic total station, a very precise instrument that tracks the robot's antenna with a laser with at least centimeter-level accuracy. This method aligns with the 3GPP work item [50], where position labels for a PRU are available using a different positioning sensor.

In a real deployment of a positioning system using fingerprints, the CSI fingerprints are influenced by variations in the environment such as movement of objects or people throughout the area of interest. To model a change in the environment for a UE at a given position, i.e., between the training and deployment phase, we consider the attenuation of the strongest path from the UE to a BS, i.e., due to blocking by a nearby person or object. Please note that the strongest path to a BS may correspond to a NLOS path, as some areas do not have a LOS to a BS.

III. SIMULATION SETUP

A. DYNAMIC SCENARIO

We assume that the wireless signal propagates along a number of different paths to the BSs. Each path is associated with a complex gain, time-of-arrival and an angle-of-arrival. The totality of the paths and their parameters can fully describe the channel between a UE and a BS. In order to model the attenuation of the strongest path we first transform the channel from the antenna-subcarrier domain to the angle-delay domain by means of the discrete Fourier transform (DFT) as described in [51].

After the matrix transformation to the angle-delay domain we identify the strongest path as the largest element of the matrix and we attenuate it by 20dB which corresponds to the attenuation effect caused by a human body at similar carrier

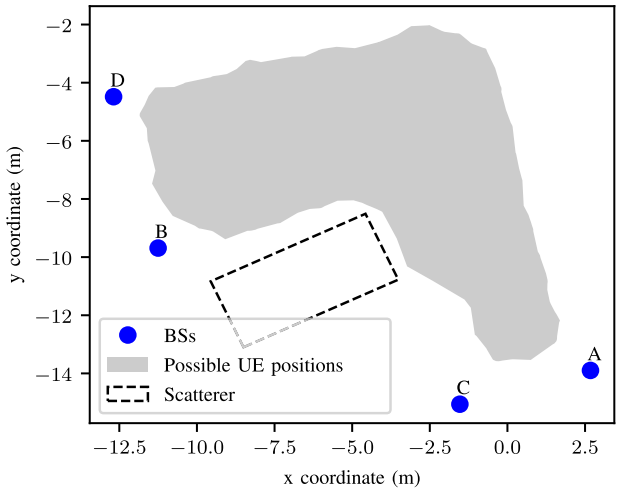

FIGURE 4. Layout of considered industrial environment in Dichasus [49].

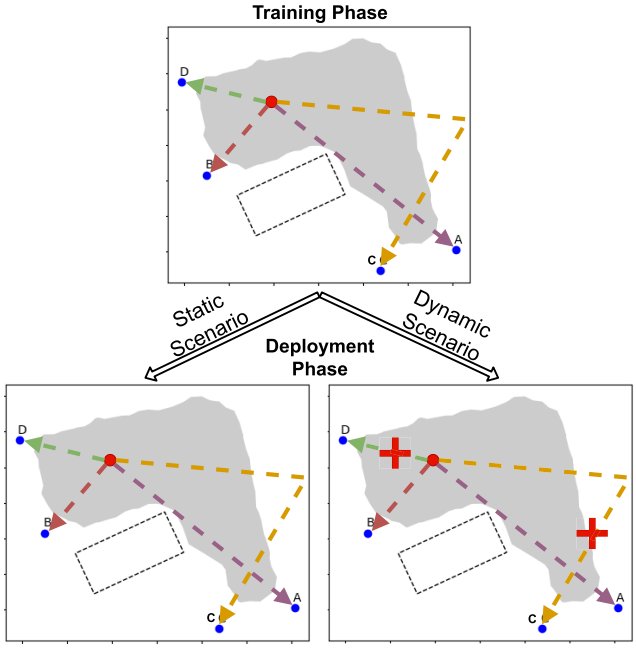

FIGURE 5. Static and dynamic scenarios example (strongest paths of 2 BSs attenuated). The UE is depicted as red dot inside the area of interest and the arrows indicate the strongest paths between UE and BSs.

frequency [43]. For a real system as the one considered, there is power leakage of each path to the neighboring elements although the matrix still remains mostly sparse. To take into account the power leakage we attenuate by the same amount a grid of 3×3 elements around the strongest path. The matrix is then transformed back to the antenna-subcarrier domain, resulting in a modified CSI fingerprint due to the attenuation. Although a realistic blockage of the strongest path may not actually result in the same modified CSI fingerprint, our aim is to evaluate the performance of the considered approaches

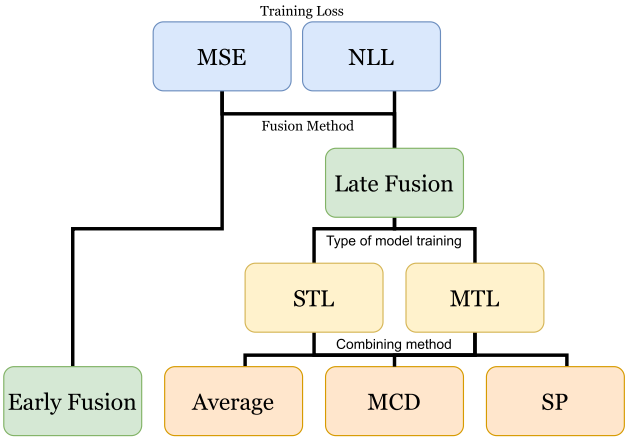

FIGURE 6. **Overview of different fusion methods and training schemes.**

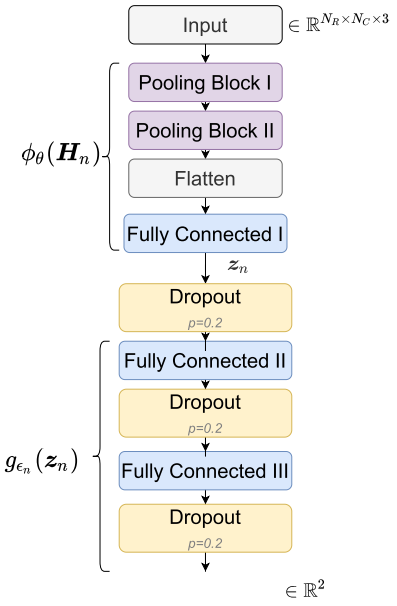

FIGURE 7. **Complete DL model.**

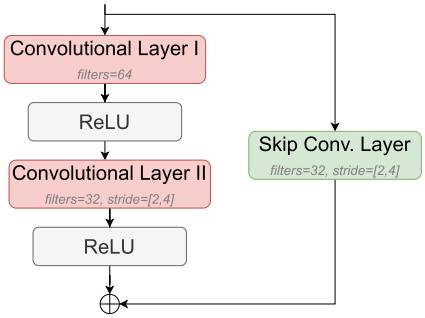

FIGURE 8. **Pooling block.**

considered a change in the environment. This is regardless whether the change is realistic or not, as the point is that the CSI fingerprint learned for a given position has changed.

Consequently, we consider scenarios with or without the above mentioned human body attenuation. When no attenuation is present we assume that there is no change in the environment between training and deployment phases, i.e. the environment is static as shown in Fig. 5. Additionally we consider scenarios where attenuation affects the signal between UE and each of the BSs, but only in the deployment phase. This implies a change in the environment between the training and the deployment phases which we define as a dynamic scenario. We consider 4 different cases for the dynamic case considering a path attenuation from the UE to one, two, three or to all BSs. In the dynamic scenario example in Fig. 5 the strongest paths of BSs C and D are attenuated.

Lastly, we compare all the different fusion approaches. An overview of all the considered approaches is shown in Fig. 6 considering different types of training loss, i.e., MSE or NLL loss, and different fusion methods, i.e., early or late fusion. For the late fusion specifically we have different types of model training. Firstly, the STL late fusion which is the same method as described in [21] where each BS corresponds to a single DL-model and each model is trained only on data from that BS. We also consider the MTL late fusion which assumes that the models of each BS share the parameters of their initial layers and are trained using the MTL scheme described in Section II-C. Finally we also consider different types of combining of the estimates from the multiple models, namely averaging, MCD or SP. With early fusion only one model is trained, with either MSE or NLL loss.

B. NEURAL NETWORK CONFIGURATION

The considered neural network is shown in Fig. 7. In the MTL late fusion we consider that the models across BSs share the parameters of the first four blocks. Both early and late fusion use the same overall structure but with different input size, depending on the considered fingerprint.

The basis of the considered neural network is the convolutional layer as it has shown promising results for positioning using CSI fingerprints [8], [11], [13], [14], [16], [21], [37]. In general, the convolutional layer is followed by a pooling layer whose purpose is to downsample its input but recently [52] it has been shown that using strided convolution instead of a pooling layer may improve the model's performance. Therefore, for the DL-model in this work we only use strided convolution and no pooling layers. A strided convolution can be thought of as a learned pooling layer, where the input is downsampled but the method of downsampling is learned during training [42, chapter 9.5].

Additionally, to further reduce information loss during downsampling, we implement the method of pooling blocks introduced in [53] which was also used for CSI based positioning in [41]. In a pooling block, a convolutional layer doubles the number of learned convolutional filters before downsampling and then the spatial size is reduced by a strided convolution. A final convolutional layer is used to reduce

TABLE 1. Fusion schemes.

Training scheme	Input dim.	DL models	Total Parameters
Early fusion	$32 \times 103 \times 2$	1	357 954
STL Late Fusion	$8 \times 103 \times 2$	4	743 688
MTL Late Fusion	$8 \times 103 \times 2$	4	285 768

again the number of learned convolutional filters to the original size. In this way, it is expected that the pooling block will learn to transfer the important information from the spatial dimension to the convolutional filters and preserve it.

Lastly, in order to avoid any problems with vanishing gradients we employ skip connections [54]. By combining all the aforementioned methods we create a pooling block which is shown in Fig. 8. Two pooling blocks are placed one after the other and are followed by 3 dense layers with 128 neurons each and finally with a dense layer with 2 neurons that outputs the estimated 2-dimensional position. When considering the aleatoric uncertainty the last dense layer has 4 neurons which correspond to the 2-dimensional position plus the NLL loss for the x and y position coordinates. Each convolutional layer has 32 filters with a 3×3 kernel. The models are trained using the Adam optimizer with a batch size of 64 for 1000 epochs in total. The initial learning rate is 10^{-3} and is reduced to 10^{-4} after 100 epochs of no improvement of the validation loss.

Depending on the method used, i.e., early fusion, STL late fusion or MTL late fusion, the total number of model parameters are different. For the early fusion method, the input's dimension is $32 \times 103 \times 2$ which results from the CSI over all 32 antennas, i.e. 4 BSs with 8 antennas each, and 103 subcarriers. Therefore the total number of parameters in the early fusion case is 357 954 parameters. When using STL late fusion, the CSI over the 8 antennas of a BS is used as input to each model. As the input dimensionality is then $8 \times 103 \times 2$, the number of parameters per model is reduced to 185 922. Since we have 4 models the total number of parameters is 743 688. Lastly when jointly training the multiple models in a MTL scheme some parameters are shared so in this case the total number of parameters is 285 768. The different configurations are summarized in Table 1.

IV. SIMULATION RESULTS

We test the different proposed schemes using the Dichasus database [49] in the deployment area shown in Fig. 4 by incrementally increasing the number of training samples N_{train} from $N_{\text{train}} = 10\,000$ to $N_{\text{train}} = 60\,000$. The validation set is 10% of the training set i.e. $N_{\text{val}} \in [1000, 6000]$. The hyperparameter λ in Eq. (8) is chosen as 0.01. We compare the different schemes with respect to the mean error (ME), which is given by the mean euclidean distance between the estimated position and the true position in the test set. The number of test samples is $N_{\text{test}} = 59\,137$, regardless of the number of training and validation samples. Furthermore we compare the quality of the uncertainty estimation of the different schemes by the AUSE metric and the IR.

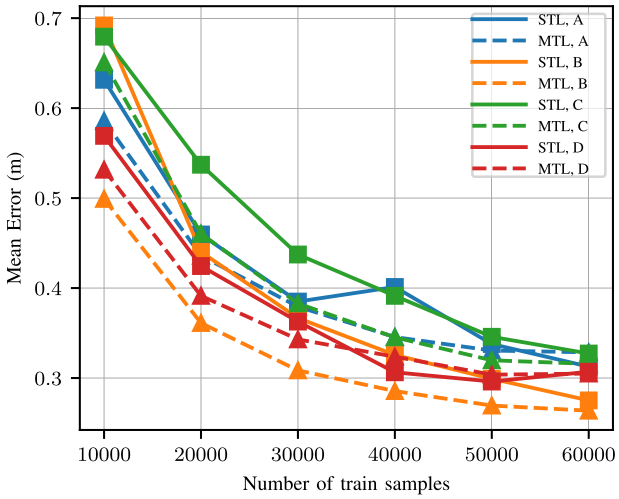

FIGURE 9. Comparison of ME of each BS in a static scenario when using STL or MTL scheme and MSE loss.

A. STATIC SCENARIO

Initially we compare the different results for a static environment, i.e., when there is no change between training and deployment phases. Before comparing the different fusion approaches listed in Table 1, we first show the gain of the training performed with the MTL training on the late fusion approach. For both the STL and MTL late fusion approaches, one model is trained at each BS by using the MSE loss in Eq. (2) as the objective function. However, for the STL late fusion the model at each BS is trained only with data from that BS, while for MTL late fusion the first part of the model at each BS is trained with data from all 4 BSs. In the following, when mentioning MTL late fusion, we refer to joint training of the first common part of the models. Fig. 9 shows the performance of the models at each BS with the STL and MTL. The gain of joint training can clearly be seen across the models at each BS. By training the models jointly, their common part incorporates information from multiple BSs effectively increasing its training size. Thus, for a late fusion scheme, joint training the models at each BS (using MTL), instead of separately (STL), leads to a decreased ME at each BS.

Next, we compare the different fusion schemes with respect to the ME in the test set. Specifically for both STL and MTL late fusion we consider three different methods of combining the estimates of the multiple models as described in section II-D. The first combining method is averaging where a simple average of all the model outputs is performed to calculate the overall estimate. The second combining method uses MCD-based combining where the variance of each estimate is estimated using the MCD method shown in eq. (4). Lastly, we also consider the SP-based combining where the variance, shown in Eq (7), of the different estimates is modified by taking into account spurious measurements. The fused estimate for both MCD and SP is calculated using eq. (12). From Fig. 10 we can observe that using early fusion

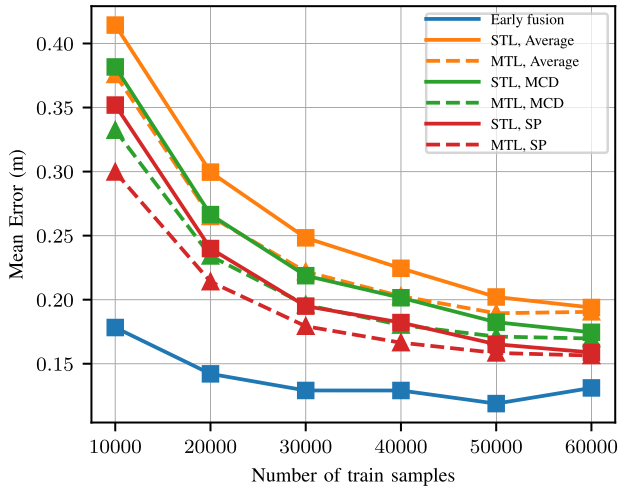

FIGURE 10. Comparison of ME for different fusion methods in a static scenario when training using STL or MTL scheme and MSE loss.

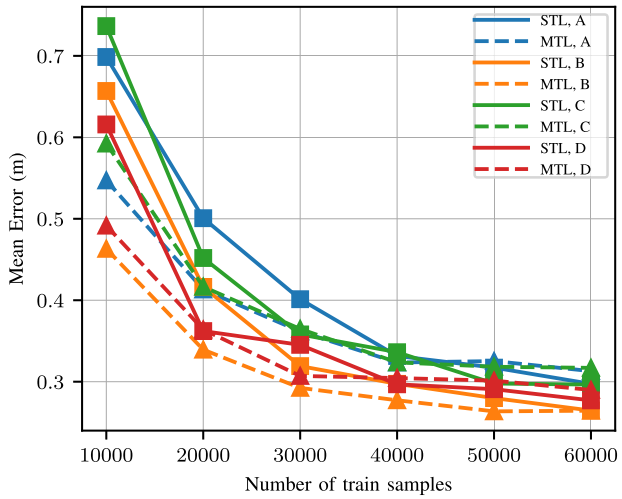

FIGURE 11. Comparison of ME of each BS in a static scenario when using STL or MTL scheme and NLL loss.

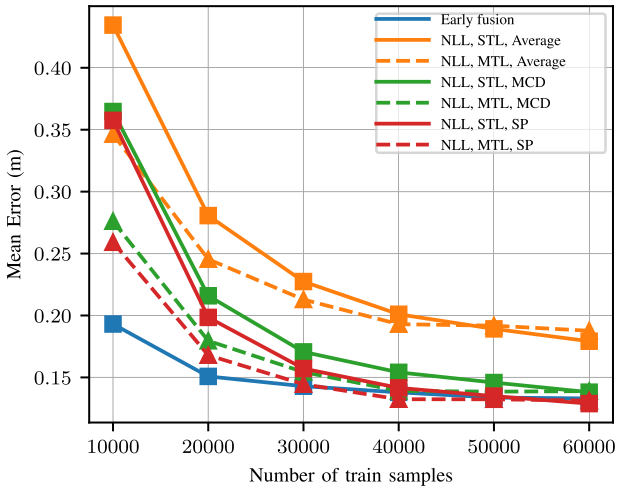

FIGURE 12. Comparison of ME for different fusion methods in a static scenario when using STL or MTL scheme and NLL loss.

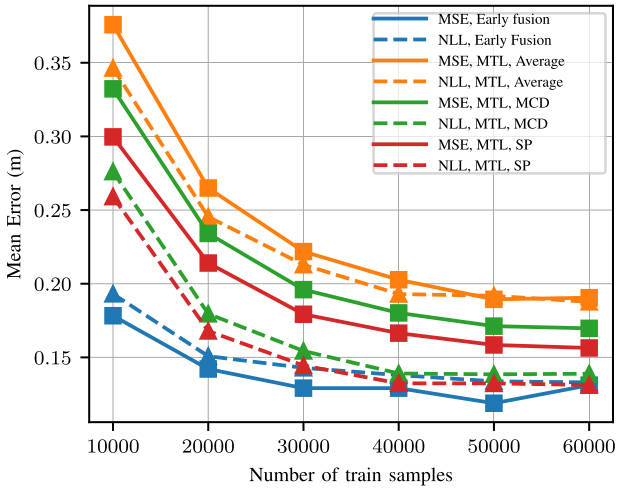

FIGURE 13. Comparison of ME for different fusion methods in a static scenario when training jointly with MSE or aleatoric loss.

outperforms all late fusion methods in a static environment, i.e., when there are no changes in the environment between the training and the deployment phase. This is in contrast to the result from [21], where it was shown that late fusion outperformed early fusion in the static case. The difference in the conclusion of the results of [21] and the ones shown in Fig. 10 can be explained due to the different considered environments. Whereas in [21] there is always a LOS between the UE and each BS, in this work the link between the UE and each BS can either be LOS or NLOS.

We consider the performance of the different fusion approaches when the models are trained using the NLL loss function shown in Eq. (3). Fig. 11 shows the comparison of the models at each BS with separate and joint learning when using the NLL loss function. Similar to the MSE loss function case, there is an improvement in the performance for each model when training them jointly in a MTL scheme. As explained in [35], using a MTL scheme which enables joint late fusion, the aleatoric uncertainty is implicitly used as a learned weighting parameter between the losses corresponding to each task, i.e. the positioning at each BS, which can increase the performance for each task.

We further compare the STL and MTL late fusion and the early fusion methods when training the models using the NLL loss. The results are shown in Fig. 12. Similar to the results in Fig. 11 when considering the MSE loss for the training, jointly training the models shows a significant improvement in performance compared to when training each model separately. This effect is particularly strong for a small number of training samples. We also see that even though the early fusion still outperforms the late fusion approaches for a small number of training samples, this is no longer the case when the number of training samples increases, i.e. the late fusion methods perform similar or better than the early fusion method. Similar to when training with the MSE loss, the late fusion with averaging is the worst performing option.

TABLE 2. AUSE in static scenario.

Training Loss	Early Fusion	Separate MCD	Separate SP	Joint MCD	Joint SP
MSE	0.51345	0.42626	0.38907	0.40984	0.38676
NLL	0.38434	0.33162	0.30825	0.32853	0.31335

TABLE 3. AUSE in dynamic scenario.

Blocked BSs	Training Loss	Early Fusion	Separate MCD	Separate SP	Joint MCD	Joint SP
A	MSE	0.49605	0.42946	0.37902	0.41478	0.37673
	NLL	0.31853	0.33723	0.29964	0.31908	0.30472
A, B	MSE	0.36338	0.42794	0.36777	0.40224	0.35451
	NLL	0.21424	0.32243	0.29882	0.28941	0.2768
A, B, C	MSE	0.25143	0.53462	0.41211	0.41538	0.32425
	NLL	0.1995	0.36986	0.32064	0.31521	0.26985
A, B, C, D	MSE	0.28687	0.52786	0.43868	0.38177	0.34691
	NLL	0.26857	0.44355	0.39339	0.38311	0.35556

Additionally, we compare the different fusion methods when training the models using the MSE (2) or the NLL (3) loss functions. We see in Fig. 13 that every late fusion scheme benefits when training the models based on the NLL function regardless of the number of training samples. Essentially, the inclusion of the aleatoric uncertainty improves the MTL late fusion schemes with MCD or SP-based combining, such that they are able to come close to the performance of the early fusion scheme, assuming an adequate number of training samples. For the early fusion scheme we see the opposite effect, namely that the inclusion of the aleatoric uncertainty during training impairs the performance, albeit slightly. The reason for this different behavior between late and early fusion may come from the fact the early fusion measurements always have some BSs having a LOS to the UE. This fact limits the usefulness of the aleatoric uncertainty since in a LOS measurement the uncertainty is anyway low. For the late fusion this is not the case since every BS may experience NLOS conditions which have high aleatoric uncertainty and positioning is then more challenging than the LOS case. As explained in section II-B.1, the model prioritizes cases where the aleatoric uncertainty is low, i.e. LOS cases.

Furthermore, we compare the quality of the uncertainty estimation in the different fusion methods in Table 2. During our evaluations, we noticed that the AUSE value remains mostly constant over training samples and therefore, we show the average over the training samples in Table 2. The averaging late fusion method it is not included in the table as it does not have uncertainty information. A lower AUSE value means that the sorting of the positioning errors across each measurement more closely corresponds with the sorting of the uncertainty, making the uncertainty a good indicator for the actual positioning error. We see from the table that for every fusion method, training using the NLL loss function improves the quality of the uncertainty estimates according to AUSE. This makes sense since there is no aleatoric uncertainty information when training MSE loss function, and instead only the epistemic uncertainty is used.

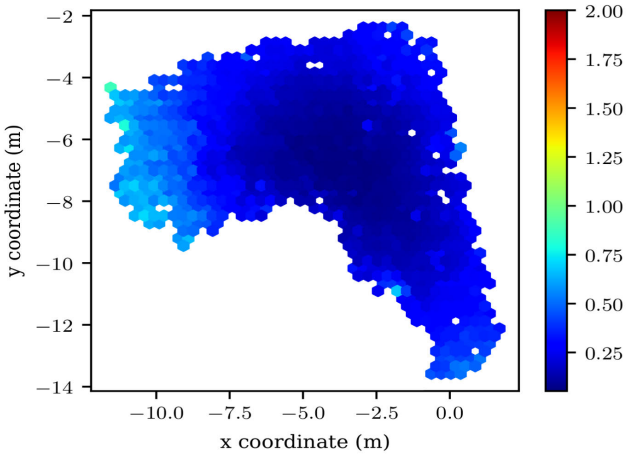

FIGURE 14. Uncertainty of model of BS A in a static scenario over all positions.

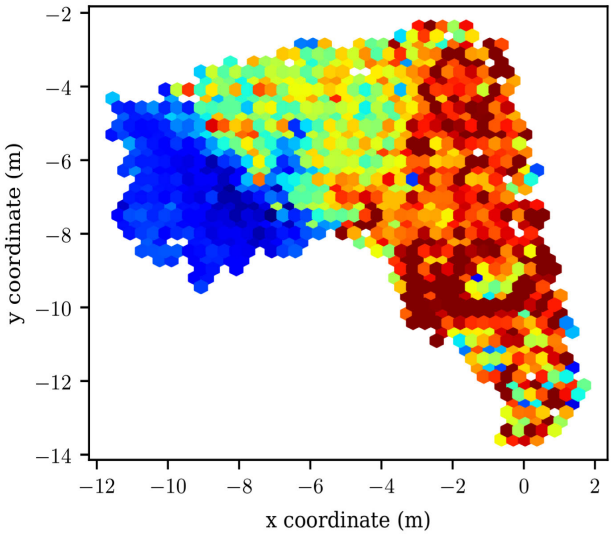

FIGURE 15. Uncertainty of model of BS A when it experiences a change.

B. DYNAMIC SCENARIO

Next we explore the positioning in a dynamic scenario, where the channel between the UE and one or more BSs experiences a change between the training and deployment phase, i.e., a 20 dB attenuation of the strongest path as described in Sec. III-A. In the following, we refer to this attenuation of the strongest path to a given BS as a change. The effect of this change on the uncertainty of the estimates can be seen in Fig. 15, which depicts the aleatoric uncertainty at BS A when the strongest path to BS is attenuated compared to the aleatoric uncertainty when no attenuation is considered shown in Fig. 14. While the uncertainty in the static case remains more or less low throughout the area, i.e., around 0.25, when there is a change the uncertainty can be up to 8 times larger. We see that the most affected regions are the ones that there is a LOS path to the BS. This path includes

most of the energy of the CSI thereby by reducing it the CSI is hugely affected. In the NLOS region we see that the uncertainty remains low since the position information is included in multiple paths, i.e., no single path contains most of the energy in the CSI fingerprint.

Fig. 16 depicts the mean error of different fusion schemes when the channel between the UE and the BS A experiences the described change above in the deployment phase. Compared to the static case (see Fig. 13), we observe a huge degradation in performance for the late fusion with averaging, as well as a large degradation in performance for the early fusion case. As expected the late fusion schemes with MCD and SP combining perform much better than the other fusion methods. Specifically the SP-based late fusion method is able to outperform all others since it is able to most reliably disregard the spurious measurements due to the change in the channel to BS A.

Furthermore, we now consider a change in the channels between the UE and multiple BSs. In Fig. 17, we depict the performance of early fusion and SP-based MTL late fusion when training with the MSE and NLL loss function as a function of the number of BSs with the attenuation of the strongest path, considering 40 000 training samples. The solid lines show the error over all positions in the test set. As the uncertainty at different positions varies, we also propose to consider the performance for the positions when the uncertainty is below a threshold. By using logistic regression in the static scenario we determine an uncertainty threshold for each method over which the positioning error is over 1m. Then for each method we exclude the measurements with uncertainty over this method-specific uncertainty threshold, and we see that the error decreases, depicted by the dashed lines in Fig. 17. The difference is more pronounced for the larger number of BSs with a change and similarly in those cases more measurements are over the uncertainty threshold and therefore excluded. Interestingly, even though the difference between using the MSE loss or the NLL loss is relatively high for a small number of blocked BSs, with NLL loss training performing better, the difference becomes smaller when blocking more BSs. The reason for that is that when more BSs experience a change then the epistemic uncertainty dominates, since the measurements differ more from the static scenario.

Next, we investigate the reliability of the uncertainty estimates in a dynamic scenario with respect to the AUSE as shown in Table 3. As in the static case, we provide the average over all training samples since similarly to the static case we noticed that the AUSE value remains mostly constant over the number of training samples. First we see that for almost every late fusion approach, training with NLL loss and MTL late fusion approach results in the most reliable uncertainty estimates.

Lastly, we depict in Fig. 18 the integrity risk for 40 000 training samples considering a channel change to one or more BSs. The integrity risk is described in equation (18) and we consider AL $=$ 1m and the threshold γ is calculated

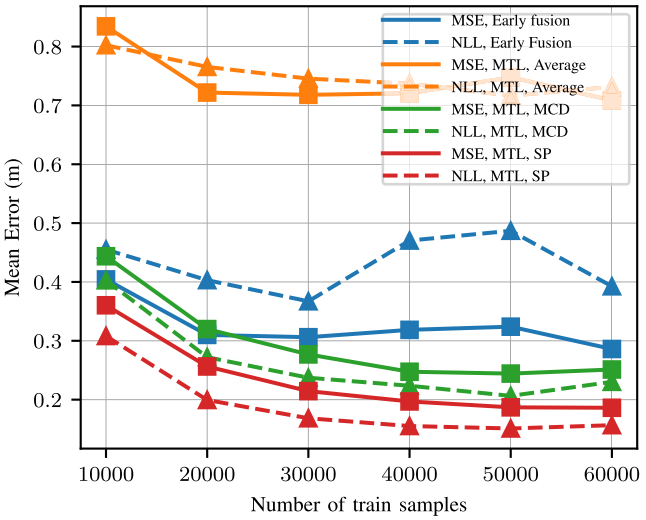

FIGURE 16. Comparison of ME for different fusion methods in a dynamic scenario when using MTL with MSE or NLL loss.

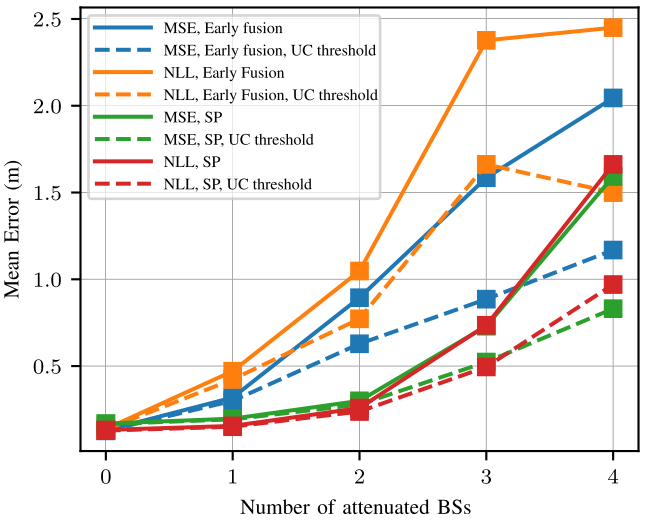

FIGURE 17. Comparison of the error of different error fusion for different number of blocked BSs, considering 40 000 training samples.

using logistic regression in the static scenario for each fusion method. A low integrity risk shows that the uncertainty estimation can be used to identify estimates that exceed the alert limit. We see in Fig. 18 that the SP method is the most reliable in this regard achieving an IR less than 7% over all cases. On the other hand the early fusion methods exhibit a quite high integrity risk. The high integrity risk of those methods combined with the relative low AUSE shows that the early fusion methods exhibit overconfidence in their estimates. In other words, they are able to sort their errors based on their uncertainty, as indicated by the low AUSE, but the error corresponding to each uncertainty estimate is underestimated. This implies that with early fusion, it is assumed that some estimates are under the alert limit even though this is not the case.

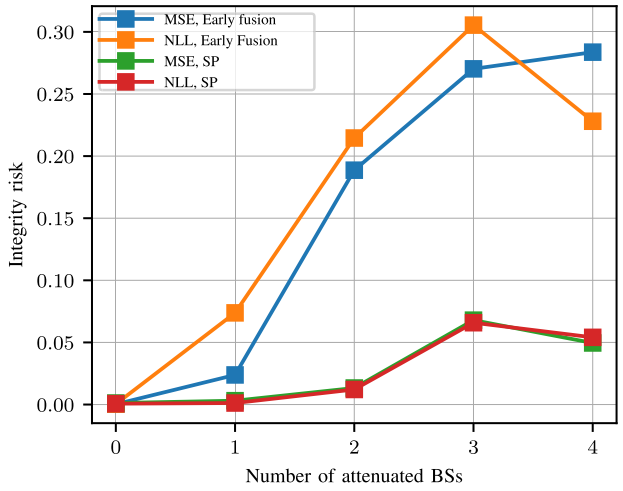

FIGURE 18. **Integrity risk.**

We note that here we provide only a simple method to calculate the uncertainty threshold based on the uncertainty vector l2-norm and provide an IR value to show the effectiveness of the uncertainty estimation. Other methods to better calculate the threshold can be developed using both uncertainty vector elements and using other classification methods such as support vector machines (SVMs). Moreover, the considered metric does not indicate how many estimates are over the uncertainty threshold for each method which may be something that needs to be considered in some use cases.

V. CONCLUSION

In this paper, we examined different fusion methods for positioning using deep learning and CSI fingerprints from multiple BSs. For early fusion, only one model is used for estimating the UE's position based on the CSI fingerprint across multiple base stations. For late fusion, one model per BS is employed and the overall UE's position is determined by combining the output of the models across the BSs. The performance of the trained models was evaluated considering a static scenario, where the channel between the training phase and the deployment phase remains the same, as well as in a dynamic scenario, where the channel between the UE and one or more BSs experience a change (attenuation of strongest path) in the deployment phase. While early fusion schemes may normally perform better in static scenarios, changes in the environment lead to a decrease positioning performance with early fusion, as the model is not able to adapt in a dynamic scenario. On the other hand, our results indicate that late fusion approaches are more robust to changes in the environment, which is an important aspect to be addressed for AI-based localization with CSI fingerprints in real deployments. Among the different considered late fusion approaches, we have shown the advantage of multi-task learning, by jointly training shared parameters of the models across the base stations, where the common part of the models benefits from a larger number of training samples.

For the late fusion approaches, different methods for combining the positioning estimates from the BS models have also been investigated. In particular, we have considered simple averaging as well as combining based on considering uncertainty estimation, namely MCD and SP, where the output of the different models are weighted based on the learned aleatoric uncertainty. We show that fusing the multiple estimates based on their uncertainty not only improves the positioning accuracy in both a static and dynamic scenario but also ultimately gives more reliable uncertainty estimates. The reliability of the uncertainty estimates is determined in terms of AUSE, which considers whether the uncertainty corresponds to the real positioning error, and in terms of the IR, which demonstrates a model's ability to discard unreliable estimates. Additionally we consider that some of the estimates may be spurious, i.e., falsely indicate low uncertainty but with an actual large positioning error, and we employ a technique to identify and disregard such estimates.

Overall, we show that late fusion scheme with multi task learning and uncertainty estimation is the most accurate and reliable in the considered scenarios. This holds also for the dynamic scenario, which is one of the main challenges limiting the deployment of AI-based localization with CSI fingerprints.

REFERENCES

[1] B. Zhou, A. Liu, and V. Lau, "Successive localization and beamforming in 5G mmWave MIMO communication systems," *IEEE Trans. Signal Process.*, vol. 67, no. 6, pp. 1620–1635, Mar. 2019.

[2] C. De Lima et al., "Convergent communication, sensing and localization in 6G systems: An overview of technologies, opportunities and challenges," *IEEE Access*, vol. 9, pp. 26902–26925, 2021.

[3] Z. Wang, Z. Liu, Y. Shen, A. Conti, and M. Z. Win, "Location awareness in beyond 5G networks via reconfigurable intelligent surfaces," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 7, pp. 2011–2025, Jul. 2022.

[4] H. Chen, H. Sarieddeen, T. Ballal, H. Wymeersch, M. Alouini, and T. Y. Al-Naffouri, "A tutorial on terahertz-band localization for 6G communication systems," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 3, pp. 1780–1815, 3rd Quart., 2022.

[5] F. Zafari, A. Gkelias, and K. K. Leung, "A survey of indoor localization systems and technologies," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2568–2599, 3rd Quart., 2019.

[6] A. Kakkavas, M. H. Castañeda García, R. A. Stirling-Gallacher, and J. A. Nossek, "Performance limits of single-anchor millimeter-wave positioning," *IEEE Trans. Wireless Commun.*, vol. 18, no. 11, pp. 5196–5210, Nov. 2019.

[7] P. Revisnyei, F. Mogyorósi, Z. Papp, I. Töros, and A. Pašic, "Performance of a TDOA indoor positioning solution in real-world 5G network," in *Proc. IEEE/IFIP Netw. Oper. Manage. Symp. (NOMS)*, May 2023, pp. 1–6.

[8] H. Chen, Y. Zhang, W. Li, X. Tao, and P. Zhang, "ConFi: Convolutional neural networks based indoor Wi-Fi localization using channel state information," *IEEE Access*, vol. 5, pp. 18066–18074, 2017.

[9] F. Yin and F. Gunnarsson, "Distributed recursive Gaussian processes for RSS map applied to target tracking," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 3, pp. 492–503, Apr. 2017.

[10] M. N. de Sousa and R. S. Thomä, "Enhancement of localization systems in NLOS urban scenario with multipath ray tracing fingerprints and machine learning," *Sensors*, vol. 18, no. 11, p. 4073, Nov. 2018, doi: 10.3390/s18114073.

[11] M. Widmaier, M. Arnold, S. Dorner, S. Cammerer, and S. T. Brink, "Towards practical indoor positioning based on massive MIMO systems," in *Proc. IEEE 90th Veh. Technol. Conf. (VTC-Fall)*, Honolulu, HI, USA, Sep. 2019, pp. 1–6.

[12] M. M. Butt, A. Pantelidou, and I. Z. Kovács, "ML-assisted UE positioning: Performance analysis and 5G architecture enhancements," *IEEE Open J. Veh. Technol.*, vol. 2, pp. 377–388, 2021.

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

[14] A. Niitsoo, T. Edelhäußer, and C. Mutschler, "Convolutional neural networks for position estimation in TDoA-based locating systems," in *Proc. Int. Conf. Indoor Position. Indoor Navigat. (IPIN)*, Sep. 2018, pp. 1–8.

[15] C.-H. Hsieh, J.-Y. Chen, and B.-H. Nien, "Deep learning-based indoor localization using received signal strength and channel state information," *IEEE Access*, vol. 7, pp. 33256–33267, 2019.

[16] A. Foliadis, M. H. C. Garcia, R. A. Stirling-Gallacher, and R. S. Thomä, "CSI-based localization with CNNs exploiting phase information," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Mar. 2021, pp. 1–6.

[17] Y. Chapre, A. Ignjatovic, A. Seneviratne, and S. Jha, "CSI-MIMO: Indoor Wi-Fi fingerprinting system," in *Proc. 39th Annu. IEEE Conf. Local Comput. Netw.*, Sep. 2014, pp. 202–209.

[18] J. Vieira, E. Leitinger, M. Sarajlic, X. Li, and F. Tufvesson, "Deep convolutional neural networks for massive MIMO fingerprint-based positioning," in *Proc. IEEE 28th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Oct. 2017, pp. 1–6.

[19] S. D. Bast, A. P. Guevara, and S. Pollin, "CSI-based positioning in massive MIMO systems using convolutional neural networks," in *Proc. IEEE 91st Veh. Technol. Conf. (VTC-Spring)*, May 2020, pp. 1–5.

[20] A. Niitsoo, T. Edelhäußer, E. Eberlein, N. Hadaschik, and C. Mutschler, "A deep learning approach to position estimation from channel impulse responses," *Sensors*, vol. 19, no. 5, p. 1064, Mar. 2019.

[21] A. Foliadis, M. H. Castañeda Garcia, R. A. Stirling-Gallacher, and R. S. Thomä, "Reliable deep learning based localization with CSI fingerprints and multiple base stations," in *Proc. IEEE Int. Conf. Commun.*, May 2022, pp. 3214–3219.

[22] E. Gönültas, E. Lei, J. Langerman, H. Huang, and C. Studer, "CSI-based multi-antenna and multi-point indoor positioning using probability fusion," *IEEE Trans. Wireless Commun.*, vol. 21, no. 4, pp. 2162–2176, Apr. 2022.

[23] D. Ramachandram and G. W. Taylor, "Deep multimodal learning: A survey on recent advances and trends," *IEEE Signal Process. Mag.*, vol. 34, no. 6, pp. 96–108, Nov. 2017.

[24] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," 2015, *arXiv:1506.02142*.

[25] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Proc. NIPS*, 2017, pp. 1–12.

[26] M. Stahlke, T. Feigl, S. Kram, B. M. Eskofier, and C. Mutschler, "Uncertainty-based fingerprinting model monitoring for radio localization," *IEEE J. Indoor Seamless Positioning Navigat.*, vol. 2, pp. 166–176, 2024.

[27] A. Salihu, S. Schwarz, and M. Rupp, "Towards scalable uncertainty aware DNN-based wireless localisation," in *Proc. 29th Eur. Signal Process. Conf. (EUSIPCO)*, Aug. 2021, pp. 1706–1710.

[28] A. Shahmansoori, G. E. Garcia, G. Destino, G. Seco-Granados, and H. Wymeersch, "Position and orientation estimation through millimeter-wave MIMO in 5G systems," *IEEE Trans. Wireless Commun.*, vol. 17, no. 3, pp. 1822–1835, Mar. 2018.

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

[30] A. Owfi, C. Lin, L. Guo, F. Afghah, J. Ashdown, and K. Turck, "A meta-learning based generalizable indoor localization model using channel state information," in *Proc. IEEE Global Commun. Conf.*, Dec. 2023, pp. 4607–4612.

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

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

[33] P. Stephan, F. Euchner, and S. ten Brink, "Angle-delay profile-based and timestamp-aided dissimilarity metrics for channel charting," 2023, *arXiv:2308.09539*.

[34] T. Hospedales, A. Antoniou, P. Micaelli, and A. Storkey, "Meta-learning in neural networks: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 9, pp. 5149–5169, Sep. 2022.

[35] R. Cipolla, Y. Gal, and A. Kendall, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 7482–7491.

[36] J. Gao, C. Zhang, Q. Kong, F. Yin, L. Xu, and K. Niu, "MetaLoc: Learning to learn indoor RSS fingerprinting localization over multiple scenarios," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2022, pp. 3232–3237.

[37] A. Foliadis, M. H. Castañeda Garcia, R. A. Stirling-Gallacher, and R. S. Thomä, "Multi-environment based meta-learning with CSI fingerprints for radio based positioning," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Mar. 2023, pp. 1–6.

[38] M. Kumar, D. P. Garg, and R. A. Zachery, "A method for judicious fusion of inconsistent multiple sensor data," *IEEE Sensors J.*, vol. 7, no. 5, pp. 723–733, May 2007.

[39] E. Ilg et al., "Uncertainty estimates and multi-hypotheses networks for optical flow," in *Computer Vision—ECCV*. Cham, Switzerland: Springer, 2018, pp. 677–693.

[40] 3GPP, *3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Study on NR Positioning Enhancements (Release 17)*, 3GPP, document TR 38.857 V17.0.0 (2021-03), 2006.

[41] B. Zhang, H. Sifaou, and G. Y. Li, "CSI-fingerprinting indoor localization via attention-augmented residual convolutional neural network," *IEEE Trans. Wireless Commun.*, vol. 22, no. 8, pp. 5583–5597, Aug. 2023.

[42] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org

[43] *Study on Artificial Intelligence (AI)/Machine Learning (ML) for NR Air Interface*, 3rd Generation Partnership Project (3GPP), document TR 38.843, v18.0.0, 2023. [Online]. Available: https://www.3gpp.org/ftp/Specs/archive/38_series/38.843/

[44] A. Foliadis, M. H. C. García, R. A. Stirling-Gallacher, X. Gong, and R. S. Thomä, "Deep learning based positioning with beamformed CSI fingerprints," in *Proc. 13th Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Sep. 2023, pp. 1–6.

[45] A. Kendall and Y. Gal, "What uncertainties do we need in Bayesian deep learning for computer vision?" 2017, *arXiv:1703.04977*.

[46] M. Crawshaw, "Multi-task learning with deep neural networks: A survey," 2020, *arXiv:2009.09796*.

[47] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2016, pp. 1–10.

[48] W. Li, Z. Wang, G. Wei, L. Ma, J. Hu, and D. Ding, "A survey on multisensor fusion and consensus filtering for sensor networks," *Discrete Dyn. Nature Soc.*, vol. 2015, no. 1, 2015, Art. no. 683701.

[49] F. Euchner, M. Gauger, S. Dörner, and S. ten Brink, "A distributed massive MIMO channel sounder for 'big CSI data'-driven machine learning," in *Proc. 25th Int. ITG Workshop Smart Antennas (WSA)*, Nov. 2021, pp. 1–6.

[50] *New WID on Artificial Intelligence (AI)/Machine Learning (ML) for NR Air Interface*, 3rd Generation Partnership Project (3GPP), document RP-234039, 2023. [Online]. Available: https://www.3gpp.org/ftp/tsg_ran/TSG_RAN/TSGR_102/Info_for_workplan/new_approved_WID_12/RAN1_5/RP-234039.zip

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

[52] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. A. Riedmiller, "Striving for simplicity: The all convolutional net," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, San Diego, CA, USA, Y. Bengio and Y. LeCun, Eds., 2015, pp. 1–14.

[53] G. Lin, A. Milan, C. Shen, and I. Reid, "RefineNet: Multi-path refinement networks for high-resolution semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5168–5177.

[54] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.