Optimizing Vehicle Positioning via Multi-model Image and Radio Frequency Fusion

Ouwen Huan*,Mingzhe Chen†,Tao Luo* *Beijing Laboratory of Advanced Information Network,
Beijing University of Posts and Telecommunications,
Beijing 100876, China, Emails: {ouwenh,tluo}@bupt.edu.cn.
†Department of Electrical and Computer Engineering and Institute for Data Science and Computing,
University of Miami, Coral Gables, FL, 33146, USA, Email: mingzhe.chen@miami.edu.

Abstract

In this paper, a multi-modal vehicle positioning framework that jointly localizes vehicles with channel state information (CSI) and images is designed. In particular, we consider an outdoor scenario where each vehicle can communicate with only one BS, and hence, it can upload its estimated CSI to only its associated BS. Each BS is equipped with a set of cameras, such that it can collect a small number of labeled CSI, a large number of unlabeled CSI, and the images taken by cameras. To exploit the unlabeled CSI data and position labels obtained from images, we design an expectation-maximization (EM) based deep learning (DL) algorithm. Specifically, since we do not know the corresponding relationship between unlabeled CSI and the multiple vehicle locations in images, we formulate the calculation of the log-likelihood function as a maximum matching problem. Subsequently, the model parameters are updated according to the maximum matching between unlabeled CSI and position labels obtained from images. Simulation results show that the proposed method can reduce the positioning error by up to 64% compared to a baseline that does not use images and uses only CSI fingerprint for vehicle positioning.

I. Introduction

Due to the fundamental role in the intelligent transportation systems, vehicle positioning technologies have gained great attention constantly. As the default solution for outdoor positioning, current global navigation satellite system (GNSS) based vehicle localization methods suffer from serious performance deterioration in urban environments where satellite signals are severely attenuated or blocked, and affected by multi-path propagation [1]. To achieve higher localization accuracy in urban areas, radio frequency (RF) fingerprint based localization is drawing increasing interests [2]. Compared to GNSS based positioning methods, RF fingerprint based methods have lower latency and higher localization accuracy in urban areas due to the densely distributed base stations (BSs) and road side units (RSUs). However, using RF fingerprints to localize vehicles still confronts with a number of challenges, such as strong dependence on the amount of labeled

training data (i.e. RF fingerprints and the corresponding positions), and low transferability of the model between different scenarios due to the different propagation environments.

Recently, a number of existing works [2]–[5] have studied RF fingerprint based methods for indoor and outdoor positioning. In [2], the authors designed an efficient angle-delay channel amplitude matrix (ADCAM) fingerprint, and then employed a convolutional neural network (CNN) to localize the user equipments. The work in [3] introduced two effective methods to process CSI for positioning. In [4], an attention-augmented residual CNN with a larger receptive field is presented for indoor localization. However, these works [2]–[4] only considered the RF fingerprint based localization with single BS. The authors in [5] studied the fingerprint based localization with multiple BSs by comparing the positioning performances of using early fusion and late fusion. However, the work in [5] assumed that all the BSs can obtain the accurate CSI of a user, which is difficult to achieve in a practical cellular communication system. Besides, all of these works [2]–[5] require a large amount of labeled data to train deep learning (DL) models. Since the corresponding positions need to be estimated with other methods such as GNSS, a large labeled dataset may not be available especially in some urban areas where GNSS based methods are not effective. Considering that multi-modal data such as images can carry rich information about the positions of users, several works in [6]–[8] studied the use of multi-modal data to aid localization or location based applications. In [6], the authors designed a fusion-based DL framework operating on multi-modal data to predict the top-K optimal beam pairs. The work in [7] proposed an algorithm to combine the features of RF data and images for indoor smartphone localization. In [8], the authors introduced an image-driven representation method to represent all the received signals with an RF image. Then, this RF image was fused with an image captured by the camera for positioning. However, none of these works [6]–[8] considered using multi-modal data to generate position labels for RF fingerprints to reduce the costs of labeled data collection.

The main contribution of this work is a novel multi-modal multi-BS vehicle positioning framework that jointly utilizes images and CSI fingerprints to determine the vehicle locations. In particular, we consider an outdoor scenario where each vehicle can communicate with only one BS, and hence, it can upload its estimated CSI to only its associated BS. Each BS is equipped with a set of cameras, such that it can collect a small number of labeled CSI, a large number of unlabeled CSI, and the images taken by cameras. To exploit the unlabeled CSI data and position labels obtained from images, we design an expectation-maximization

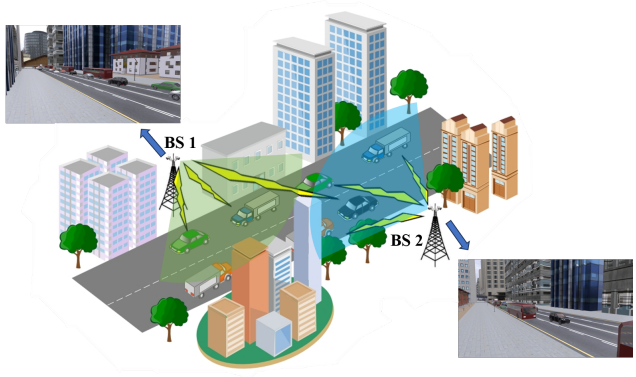

Fig. 1. Considered scenario.

(EM) [9] based DL algorithm that iteratively trains the DL model. Specifically, since we do not know the corresponding relationship between unlabeled CSI and the multiple vehicles in images, we formulate the calculation of the log-likelihood function as a maximum matching problem [10]. Subsequently, the model parameters are updated according to the maximum matching between unlabeled CSI and position labels obtained from images. Simulation results show that the proposed method can reduce the positioning error by up to 64% compared to a baseline that does not use images for vehicle positioning.

II. System Model

As shown in Fig. 1, we consider a multi-modal millimeter wave (mmWave) vehicle positioning system where the locations of vehicles are jointly determined by B base stations (BSs) using downlink channel status information (CSI) and red-green-blue (RGB) images. Specifically, each BS is equipped with a uniform linear array (ULA) while each vehicle has an omni-directional antenna. To localize a vehicle, the BS will first send a pilot signal to its served vehicles which will send the estimated downlink CSI back. Each BS is equipped with a set of C cameras such that it can use both CSI and images to localize vehicles. We assume that a vehicle can communicate with only one BS at each time slot, and hence, it can upload its estimated CSI to only its associated BS. Next, we first introduce the channel model. Then, we introduce the processes of coordinate transformation from pixel coordinates to real-world coordinates, and image processing. Subsequently, we present the datasets and positioning models. Finally, we formulate our multi-modal positioning problem.

A. Channel Model

We assume that BS b ($b = 1, \cdots, B$) is serving V_b vehicles in one time slot. The pilot symbol sequence transmitted by BS b to vehicle v ($v = 1, \cdots, V_b$) over subcarrier k is $\boldsymbol{x}_{b,v}^k \in \mathbb{C}^{N^{\mathrm{P}} \times 1}$, where N^{P} is the number of symbols in the symbol sequence. Then, the received pilot symbol sequence $\boldsymbol{y}_{b,v}^k \in \mathbb{C}^{N^{\mathrm{P}} \times 1}$ is

$$\boldsymbol{y}_{b,v}^k = \left[\boldsymbol{x}_{b,v}^k \cdot \left(\boldsymbol{f}_{b,v} \right)^T \right] \cdot \boldsymbol{h}_{b,v}^k + \boldsymbol{n}_{b,v}^k, \tag{1}$$

where $\boldsymbol{h}_{b,v}^k \in \mathbb{C}^{N^{\mathrm{B}} \times 1}$ is the channel from BS b to the vehicle v, with N^{B} being the number of antennas of the ULA. $\boldsymbol{f}_{b,v}$ is the beamforming vector, and $\boldsymbol{n}_{b,v}^k$ is the additive noise vector following a zero-mean Gaussian distribution with a covariance matrix $\boldsymbol{\Sigma}_{b,v}^k \in \mathbb{R}^{N^{\mathrm{P}} \times N^{\mathrm{P}}}$. Since both $\boldsymbol{x}_{b,v}^k$ and $\boldsymbol{f}_{b,v}$ are known by vehicle v, the channel $\boldsymbol{h}_{b,v}^k$ can be estimated at vehicle v based on $\boldsymbol{y}_{b,v}^k$, and will be transmitted back to BS b. The received CSI matrix from vehicle v is $\boldsymbol{H}_{b,v} = \left[\boldsymbol{h}_{b,v}^1, \boldsymbol{h}_{b,v}^2, \cdots, \boldsymbol{h}_{b,v}^{N^{\mathrm{C}}} \right] \in \mathbb{C}^{N^{\mathrm{B}} \times N^{\mathrm{C}}}$, with N^{C} being the number of valid subcarriers. Then, the set of CSI matrices collected by all BSs at one certain time slot is $\mathcal{H} = \Big\{ \boldsymbol{H}_{b,v} | b \in \{1, 2, \cdots, B\}, v \in \{1, 2, \cdots, V_b\} \Big\}$.

B. Coordinate Transformation Model

Since images can only provide pixel coordinates of vehicles, we need to transform these vehicle pixel coordinates to real-world coordinates, such that they can be used for positioning. To explain the coordinate transformation, we consider three coordinate systems as shown in Fig. 2, which are the 3D world coordinate system (WCS) (o^w, x^w, y^w, z^w), the 2D image coordinate system (ICS) (o^i, x^i, y^i) and the 2D pixel coordinate system (PCS) (o^p, u^p, v^p) [11]. Both the ICS and PCS are on the image plane but have their own coordinate origins (i.e. $o^i \neq o^p$). In particular, we assume that both axis $o^p u^p$ and axis $o^i x^i$ are parallel to plane $x^w o^w y^w$. Given the line-of-sight (LoS) direction and viewing angles of a camera, and the width and height of the images, the pixel coordinate $[u, v]^{\mathrm{T}}$ of each vehicle in PCS can be transformed to a polar coordinate $[\phi, \theta]^{\mathrm{T}}$ in the WCS [12], where ϕ and θ respectively denote the azimuth and elevation angles of the polar coordinate. As shown in Fig 3, the goal of positioning is to estimate the coordinate $\boldsymbol{p} = [x, y]^{\mathrm{T}} = [d^{\mathrm{H}} \cos\phi, d^{\mathrm{H}} \sin\phi]^{\mathrm{T}}$ of each vehicle in the $x^w o^w y^w$ plane of WCS with d^{H} being the horizontal distance between vehicle and the BS. We assume that the height difference between camera and each vehicle is Δh. Then, the horizontal distance is $d^{\mathrm{H}} = \Delta h \cdot \tan\theta$. Therefore, if

we can obtain the pixel coordinate of a vehicle, we can approximately localize this vehicle in 3D world coordinate system through coordinate transformation.

C. Image Processing

In this section, we introduce how to process the images captured at each BS to obtain vehicle position coordinates from images. Since each BS is equipped with C cameras to capture a set of C images in each time slot, the set of C images captured by BS b at a time slot is $\mathcal{I}_b = \{\boldsymbol{I}_b^c | c = 1, 2, \cdots, C\}$, with $\boldsymbol{I}_b^c$ being an RGB image captured by camera c. All the images have the same dimension of $3 \times W \times H$ with W and H being the width and height of the images. Given $\mathcal{I}_b$, we first utilize YOLO [13] to detect vehicles from these images in order to obtain their pixel coordinates, such that we can then transform these pixel coordinates of vehicles to their coordinates in WCS. Here, we can also employ other mature object detection models to detect vehicles in images. The set of vehicle position coordinates obtained from all the images in $\mathcal{I}_b$ is defined as $\mathcal{P}_b = \left\{\boldsymbol{p}_{b,i} | i = 1, 2, \cdots, \hat{V}_b\right\}$, where $\hat{V}_b$ denotes the number of vehicles detected from images. Note that $\hat{V}_b$ is not certainly equal to V_b, since several vehicles served by BS b may not be captured by the cameras due to obstructions or be not in the camera's field of view (FoV). However, the undetected vehicles of BS b may be captured by the cameras of other BSs. By integrating $\mathcal{P}_1, \mathcal{P}_2, \cdots, \mathcal{P}_B$ and removing the duplicate position coordinates, the set of vehicle locations captured by all the images of all BSs at one certain time slot is $\mathcal{P} = \left\{\boldsymbol{p}_j | j = 1, 2, \cdots, \hat{V}\right\}$, with $\hat{V}$ being the number of vehicles captured by images of all BSs in one time slot.

D. Labeled and Multi-modal Datasets

We assume that a small number of vehicles can provide their locations and CSI to their associated BSs. Hence, each BS b has a small-sized labeled dataset expressed as $\mathcal{D}_b^{\mathrm{L}} = \{\boldsymbol{H}_{b,j}, \boldsymbol{p}_{b,j}\}_{j=1}^{N_b^{\mathrm{L}}}$, with $\boldsymbol{H}_{b,j} \in \mathbb{C}^{N^{\mathrm{B}} \times N^{\mathrm{C}}}$ being CSI sample j, $\boldsymbol{p}_{b,j} \in \mathbb{R}^{2 \times 1}$ being the corresponding coordinate of $\boldsymbol{H}_{b,j}$, and N_b^{L} being the number of data samples of labeled dataset $\mathcal{D}_b^{\mathrm{L}}$.

Due to the the insufficient positioning accuracy of GNSS in urban environments [1], and the concerns of user privacy [14], most vehicles will only upload their CSI without sharing their location information. Hence, each BS can collect a large number of CSI samples without corresponding location labels. To create labels for these CSI data, one can obtain vehicle locations

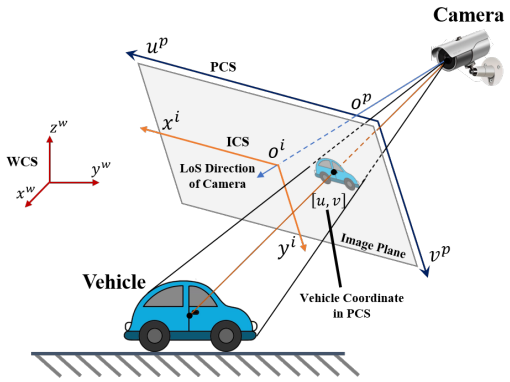

Fig. 2. Camera FoV.

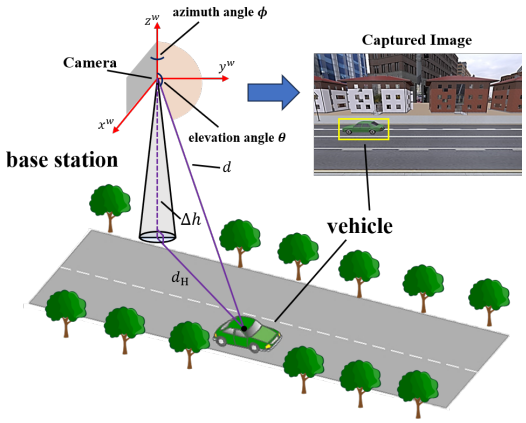

Fig. 3. Diagram of vehicle positioning.

from RGB images. Therefore, the multi-modal dataset consists of unlabeled CSI and vehicle locations obtained from images can be defined as

$$\mathcal{D}^{\mathrm{M}} = \left\{\mathcal{H}^k, \mathcal{P}^k\right\}_{k=1}^{N^{\mathrm{M}}}, \tag{2}$$

where N^{M} is the number of samples in $\mathcal{D}^{\mathrm{M}}$, $\mathcal{H}^k$ is the set of CSI collected by all the BSs, and $\mathcal{P}^k$ represents the position coordinates obtained from images of all BSs. Since $\mathcal{H}^k$ are unlabeled, we use a random variable vector $\boldsymbol{r}_{b,v} \in \mathbb{R}^{2\times 1}$ to represent the position coordinate of $\boldsymbol{H}_{b,v} \in \mathcal{H}^k$, and the set that includes all $\boldsymbol{r}_{b,v}$ of CSI in $\mathcal{H}^k$ is defined as $\mathcal{R}^k$. We assume that the set of vehicle position coordinates obtained from images (i.e. $\mathcal{P}^k$) is a subset of $\mathcal{R}^k$. Nevertheless, the exact corresponding relationship between each $\boldsymbol{r}_{b,v} \in \mathcal{R}^k$ (or $\boldsymbol{H}_{b,v} \in \mathcal{H}^k$) and $\boldsymbol{p}_j \in \mathcal{P}^k$ is unknown.

E. Positioning Model

Each BS uses a neural network (NN) to localize its served vehicles. We model the NN of BS b as the conditional probability density function (CPDF) of vehicle location when given the CSI matrix of a vehicle. Hence, the positioning model can be expressed as

$$F_{\boldsymbol{\omega}_b}(\cdot|\cdot) : \mathbb{R}^{2\times 1} \times \mathbb{C}^{N^{\mathrm{B}}\times N^{\mathrm{C}}} \to \mathbb{R}, \tag{3}$$

where $\mathbb{R}^{2\times 1}\times\mathbb{C}^{N^{\mathrm{B}}\times N^{\mathrm{C}}}$ denotes the cartesian product of $\mathbb{R}^{2\times 1}$ and $\mathbb{C}^{N^{\mathrm{B}}\times N^{\mathrm{C}}}$, and $\boldsymbol{\omega}_b$ is the parameter vector of the model. The parameter matrix of all the NNs is defined as $\boldsymbol{\Omega} = [\boldsymbol{\omega}_1, \boldsymbol{\omega}_2, \cdots, \boldsymbol{\omega}_B]$.

F. Problem Formulation

Given the defined system model, our goal is to together maximize the log-likelihood function of the positioning model at each BS using the multi-modal dataset $\mathcal{D}^{\mathrm{M}}$ and each labeled dataset $\mathcal{D}_b^{\mathrm{L}}$. Specifically, the log-likelihood function of $\boldsymbol{\omega}_b$ on $\mathcal{D}_b^{\mathrm{L}}$ is

$$L_b^{\mathrm{L}}(\boldsymbol{\omega}_b) = \sum_{j=1}^{N_b^{\mathrm{L}}} \log F_{\boldsymbol{\omega}_b}(\boldsymbol{p}_{b,j}|\boldsymbol{H}_{b,j}). \tag{4}$$

The joint log-likelihood function of the parameter matrix $\boldsymbol{\Omega}$ on dataset $\mathcal{D}^{\mathrm{M}}$ can be derived using the chain rule, which is given by

$$L^{\mathrm{M}}(\boldsymbol{\Omega}) = \sum_{\left\{\mathcal{H}^k,\mathcal{P}^k\right\}\in\mathcal{D}^{\mathrm{M}}} \log\Big[\sum_{\mathcal{R}^k} \mathbb{1}_{\{\mathcal{P}^t\subseteq\mathcal{R}^t\}} \prod_{\substack{\boldsymbol{r}_{b,v}\in\mathcal{R}^k \\ \boldsymbol{H}_{b,v}\in\mathcal{H}^k}} F_{\boldsymbol{\omega}_b}(\boldsymbol{r}_{b,v}|\boldsymbol{H}_{b,v})\Big], \tag{5}$$

where $\mathbb{1}_{\{\cdot\}}$ denotes the indicator function which returns 1 if the argument is true and 0 otherwise. Considering that the joint maximization of (4) and (5) is a multi-objective optimization problem (MOOP), we utilize the weighted sum scheme [] to merge different objective functions. Thus, our machine learning problem can be formulated as

$$\max_{\boldsymbol{\Omega}} \gamma\cdot L^{\mathrm{M}}(\boldsymbol{\Omega}) + (1-\gamma)\cdot\sum_{b=1}^{B} L_b^{\mathrm{L}}(\boldsymbol{\omega}_b), \tag{6}$$

where $\gamma\in[0,1]$ is the weight parameter.

From (6), we see that the maximization of log-likelihood function jointly depends on the log-likelihood function of each positioning model parameter $\boldsymbol{\omega}_b$ with its own dataset $\mathcal{D}_b^{\mathrm{L}}$, and the joint log-likelihood function of $\boldsymbol{\Omega}$ with the shared dataset $\mathcal{D}^{\mathrm{M}}$. However, since $\mathcal{R}^k$ is a random

variable set, directly solving (6) with optimization algorithms (e.g. gradient ascent) requires computing the sum of $\prod F_{\boldsymbol{\omega}_b}\left(\boldsymbol{r}_{b,v}|\boldsymbol{H}_{b,v}\right)$ over all possible $\mathcal{R}^k$, which will result in exponential computational complexity, and high difficulty of model convergence. To effectively solve (6), we propose a meta-learning based Viterbi expectation-maximization (V-EM) algorithm.

III. Proposed Meta-learning based V-EM Algorithm

In this section, we introduce the proposed algorithm that can effectively solve (6). Compared to current positioning algorithms [2]–[5], [7], [8], the proposed algorithm can use images and unlabeled CSI to train the positioning model, and thereby has lower dependence on the amount of labeled training data. Next, we first explain the preparation work of the meta-learning based V-EM algorithm. Subsequently, we introduce the three main steps in each training iteration. Then, the inference procedure is presented. Finally, we discuss the computational complexity and convergence of the proposed algorithm.

A. Preparation Work

The preparation work contains two procedures: 1) construction of validation datasets, 2) NN initialization, which are specified as follows.

1) Construction of validation datasets: In our designed algorithm, a small validation dataset of each BS is needed to effectively implement meta-learning. Therefore, we randomly separate a small portion of $\mathcal{D}_b^{\mathrm{L}}$ as the validation dataset of BS b, which is defined as $\mathcal{D}_b^{\mathrm{V}} = \{\boldsymbol{H}_{b,j}, \boldsymbol{p}_{b,j}\}_{j=1}^{N_b^{\mathrm{V}}}$, with N_b^{V} being the number of validation samples.

2) NN initialization: We train the NN at each BS with the labeled dataset $\mathcal{D}_b^{\mathrm{L}}$ to obtain good initial NN parameters. In particular, we assume that each $F_{\boldsymbol{\omega}_b}\left(\boldsymbol{p}_{b,j}|\boldsymbol{H}_{b,j}\right)$ is a Gaussian distribution whose mean is equal to the predicted position coordinate of $\boldsymbol{H}_{b,j}$ from the NN. Then, the minimization of the mean square error (MSE) between the predicted and ground truth locations will be equivalent to maximizing the log-likelihood function. Hence, we utilize MSE as the loss function of NN initialization, which is given by

$$L_b^{\mathrm{E1}} = \frac{1}{N_b^{\mathrm{L}}} \sum_{j=1}^{N_b^{\mathrm{L}}} \|\boldsymbol{p}_{b,j} - \hat{\boldsymbol{p}}_{b,j}\|_2^2, \tag{7}$$

where $\hat{\boldsymbol{p}}_{b,j}$ is the predicted position coordinate from the NN, and $\|\cdot\|_2$ denotes the $L2$-norm of a vector. We use the mini-batch gradient descent (MBGD) method to update the NN parameter of each BS, and the obtained initial parameter matrix is defined as $\boldsymbol{\Omega}^0 = [\boldsymbol{\omega}_1^0, \boldsymbol{\omega}_2^0, \cdots, \boldsymbol{\omega}_B^0]$.

B. Main Steps

The V-EM algorithm can be used for iteratively optimizing the log-likelihood function when both observable and unobservable data exist. In our considered problem, the observable data is each $\left\{\mathcal{H}^k, \mathcal{P}^k\right\} \in \mathcal{D}^{\mathrm{M}}$, while the unobservable data refers to the ground truth position coordinates $\mathcal{R}^k$. Instead of calculating the sum of $\prod F_{\boldsymbol{\omega}_b}\left(\boldsymbol{r}_{b,v}|\boldsymbol{H}_{b,v}\right)$ over all possible $\mathcal{R}^k$ in each training iteration, which will incur intractable computational complexity, the V-EM algorithm selects to only compute $\prod F_{\boldsymbol{\omega}_b}\left(\boldsymbol{r}_{b,v}|\boldsymbol{H}_{b,v}\right)$ subject to the maximum posterior estimation (MAPE) of $\mathcal{R}^k$, and thereby constructs a surrogate function of (5) []. Subsequently, this surrogate function will be utilized as the objective function for NN parameter optimization. Since the MAPE of $\mathcal{R}^t$ in each training iteration is not necessarily equal to the real ground truth labels of $\mathcal{H}^k$, a meta-learning based reweighting algorithm is introduced to reduce the impact of label noises. Next, we explain the three main steps of each training iteration in detail: 1) construction of the surrogate function, 2) meta-learning based reweighting, 3) optimization of the surrogate function.

1) Construction of the surrogate function: To obtain the surrogate function of (5) at each training iteration, we need to first determine the MAPE of $\mathcal{R}^k$. Given $\left\{\mathcal{H}^k, \mathcal{P}^k\right\} \in \mathcal{D}^{\mathrm{M}}$ and the NN parameters $\boldsymbol{\Omega}^{i-1} = \left[\boldsymbol{\omega}_1^{i-1}, \cdots, \boldsymbol{\omega}_B^{i-1}\right]$ at iteration $i-1$, the MAPE of $\mathcal{R}^k$ is derived using the Baysian rule, which can be expressed as

$$\begin{aligned}\hat{\mathcal{R}}^k\left(\boldsymbol{\Omega}^{i-1}\right) &= \arg\max_{\mathcal{R}^k} \frac{\mathbb{1}_{\{\mathcal{P}^t \subseteq \mathcal{R}^t\}} \prod F_{\boldsymbol{\omega}_b^{i-1}}\left(\boldsymbol{r}_{b,v}|\boldsymbol{H}_{b,v}\right)}{\sum_{\mathcal{R}^k} \mathbb{1}_{\{\mathcal{P}^t \subseteq \mathcal{R}^t\}} \prod F_{\boldsymbol{\omega}_b^{i-1}}\left(\boldsymbol{r}_{b,v}|\boldsymbol{H}_{b,v}\right)} \\ &= \arg\max_{\mathcal{R}^k} \left[\mathbb{1}_{\{\mathcal{P}^t \subseteq \mathcal{R}^t\}} \prod_{\substack{\boldsymbol{r}_{b,v} \in \mathcal{R}^k \\ \boldsymbol{H}_{b,v} \in \mathcal{H}^k}} F_{\boldsymbol{\omega}_b^{i-1}}\left(\boldsymbol{r}_{b,v}|\boldsymbol{H}_{b,v}\right)\right].\end{aligned} \tag{8}$$

Note that the second equality in (8) stems from the fact that $\sum_{\mathcal{R}^k} \mathbb{1}_{\{\mathcal{P}^t \subseteq \mathcal{R}^t\}} \prod F_{\boldsymbol{\omega}_b^{i-1}}\left(\boldsymbol{r}_{b,v}|\boldsymbol{H}_{b,v}\right)$ is a constant independent of $\mathcal{R}^k$. Since we assume that the possible $\mathcal{R}^k$ should be a superset of $\mathcal{P}^k$, solving (8) is equivalent to finding the appropriate corresponding relationship between $\boldsymbol{H}_{b,v} \in \mathcal{H}^k$ and $\boldsymbol{p}_j \in \mathcal{P}^k$. Specifically, we use $\boldsymbol{\alpha}_{b,v} = \left[\alpha_{b,v}^1, \alpha_{b,v}^2, \cdots, \alpha_{b,v}^{\hat{V}}, \alpha_{b,v}^{\hat{V}+1}\right]^{\mathrm{T}}$ with $\alpha_{b,v}^j \in \{0,1\}$, to represent the corresponding relationship between each $\boldsymbol{H}_{b,v}$ and $\boldsymbol{p}_j$. Here, $\alpha_{b,v}^j = 1$ implies that the coordinate $\boldsymbol{p}_j$ is the corresponding location of $\boldsymbol{H}_{b,v}$, otherwise, $\alpha_{b,v}^j = 0$, and $\alpha_{b,v}^{\hat{V}+1} = 1$ indicates that the corresponding location of $\boldsymbol{H}_{b,v}$ is not in $\mathcal{P}^k$. Hence, the corresponding location of $\boldsymbol{H}_{b,v}$ can be expressed as $\boldsymbol{p}_{b,v} = \left(1 - \alpha_{b,v}^{\hat{V}+1}\right) \sum_{j=1}^{\hat{V}} \alpha_{b,v}^j \boldsymbol{p}_j$. We formulate the problem of

determining the appropriate $\boldsymbol{\alpha}_{b,v}$ as a maximum matching problem [10], which is given by

$$\max_{\boldsymbol{\alpha}_{b,v}} \sum_{\boldsymbol{H}_{b,v}\in\mathcal{H}^k} \left(1-\alpha_{b,v}^{\hat{V}+1}\right) \log F_{\boldsymbol{\omega}_b^{i-1}}\left(\boldsymbol{p}_{b,v}|\boldsymbol{H}_{b,v}\right), \tag{9}$$

$$\text{s.t.}\ \alpha_{b,v}^{j} \in \left\{0,1\right\}, \tag{9a}$$

$$\sum_{b=1}^{B}\sum_{v=1}^{V_b} \alpha_{b,v}^{j} = 1, \tag{9b}$$

$$\sum_{j=1}^{\hat{V}+1} \alpha_{b,v}^{j} = 1, \tag{9c}$$

where (9a) indicates whether the corresponding location of CSI $\boldsymbol{H}_{b,v}$ is position coordinate $\boldsymbol{p}_j$, (9b) guarantees that each position coordinate can only correspond to one CSI, while (9c) ensures that each CSI has at most one corresponding location coordinate since the vehicle may not be captured by all cameras. Here, we use the Kuhn-Munkres (KM) algorithm [] to effectively solve (9) in polynomial time. The input of KM algoirthm at iteration i is the reward matrix $\boldsymbol{C}_i \in \mathbb{R}^{V\times\hat{V}}$ of the maximum matching problem, where $V=\sum_{b=1}^{B} V_b$ is the number of CSI matrices in $\mathcal{H}^k$. The element at row m and column n of $\boldsymbol{C}_i$ represents the reward of matching the mth CSI in $\mathcal{H}^k$ to the nth location coordinate in $\mathcal{P}^k$. Without loss of generality, we assume that the mth CSI matrix in $\mathcal{H}^k$ is $\boldsymbol{H}_{b_m,v_m}$. Then, the element at row m and column n is

$$c_i\left(m,n\right) = -\frac{\|\hat{\boldsymbol{p}}_{b_m,v_m}-\boldsymbol{p}_n\|_2^2}{\sigma^2_{\boldsymbol{\omega}_{b_m}^{i-1}}\left(\boldsymbol{p}_n\right)}, \tag{10}$$

where $\hat{\boldsymbol{p}}_{b_m,v_m}$ denotes the position coordinate of $\boldsymbol{H}_{b_m,v_m}$ predicted by the NN with parameters $\boldsymbol{\omega}_{b_m}^{i-1}$, and $\sigma_{\boldsymbol{\omega}_{b_m}^{i-1}}\left(\boldsymbol{p}_n\right)$ is the empirical positioning error at $\boldsymbol{p}_n$. Specifically, $\sigma_{\boldsymbol{\omega}_{b_m}^{i-1}}\left(\cdot\right)$ is obtained by polynomial fitting the positioning errors on the validation dataset $\mathcal{D}_{b_m}^{\mathrm{V}}$. Given $\boldsymbol{C}_i$, the optimal corresponding relationship vector $\hat{\boldsymbol{\alpha}}_{b,v} = \left[\hat{\alpha}_{b,v}^{1},\hat{\alpha}_{b,v}^{2},\cdots,\hat{\alpha}_{b,v}^{\hat{V}},\hat{\alpha}_{b,v}^{\hat{V}+1}\right]^{\mathrm{T}}$ of each $\boldsymbol{H}_{b,v}$ can be solved by using KM algorithm, and the MAPE of $\mathcal{R}^k$ can be further expressed as

$$\hat{\mathcal{R}}^k\left(\boldsymbol{\Omega}^{i-1}\right) = \left\{ \hat{\boldsymbol{r}}_{b,v} \middle| \hat{\boldsymbol{r}}_{b,v} = \hat{\alpha}_{b,v}^{\hat{V}+1}\hat{\boldsymbol{p}}_{b,v} + \left(1-\hat{\alpha}_{b,v}^{\hat{V}+1}\right)\sum_{j=1}^{\hat{V}} \hat{\alpha}_{b,v}^{j}\boldsymbol{p}_j \right\}, \tag{11}$$

Algorithm 1 EM algorithm.

Initialize: Model parameters $\boldsymbol{\Omega}^0$, learning rate λ, the number of iteration T, iteration period N of re-estimating $\sigma_{\omega_b}(\cdot)$.
Input: Labeled datasets $\mathcal{D}_b^{\mathrm{L}}$, multi-modal dataset $\mathcal{D}^{\mathrm{M}}$
Separate validation dataset from each $\mathcal{D}_b^{\mathrm{L}}$.
for $i = 1 \rightarrow T$ **do**
  if $i - 1 \equiv 0 \pmod{N}$ **then**
    Estimate $\sigma_{\omega_b}(\cdot)$ on validation datasets.
  end if
  Calculate $\boldsymbol{C}_i$ based on (10).
  Solve each $\boldsymbol{\alpha}_{b,v}^i$ with KM algorithm.
  Calculate L_b^{E} based on (7), (12), and (13).
  Update $\boldsymbol{\omega}_b$ by (14).
end for

where $\hat{\boldsymbol{r}}_{b,v}$ denotes the MAPE of $\boldsymbol{r}_{b,v}$. Given $\hat{\mathcal{R}}^k\left(\boldsymbol{\Omega}^{i-1}\right)$, the surrogate function of (5) in iteration i is defined as

$$Q^i\left(\boldsymbol{\Omega}^{i-1}\right) = \sum_{\left\{\mathcal{H}^k,\mathcal{P}^k\right\}} \log\Big[\prod_{b,v} F_{\boldsymbol{\omega}_b^{i-1}}\left(\hat{\boldsymbol{r}}_{b,v}|\boldsymbol{H}_{b,v}\right)\Big] = \sum_{\left\{\mathcal{H}^k,\mathcal{P}^k\right\}} \sum_{b,v} \log F_{\boldsymbol{\omega}_b^{i-1}}\left(\hat{\boldsymbol{r}}_{b,v}|\boldsymbol{H}_{b,v}\right). \tag{12}$$

2) Meta-learning based reweighting: From (12), we see that the surrogate function $Q^i\left(\boldsymbol{\Omega}^{i-1}\right)$ is equivalent to the MSE loss with $\hat{\boldsymbol{r}}_{b,v}$ being the label of $\boldsymbol{H}_{b,v}$. Because $\hat{\boldsymbol{r}}_{b,v}$ is not necessarily equal to the real ground truth label of $\boldsymbol{H}_{b,v}$, we employ the meta-learning based reweighting algorithm proposed in [] to reduce the impact of noisy labels and thus improve the robustness of training. Specifically, we modify the surrogate function by reweighting the log-likelihood of each $\boldsymbol{H}_{b,v}$ and $\hat{\boldsymbol{r}}_{b,v}$. The reweighted surrogate function is defined as

$$\widetilde{Q}^i\left(\boldsymbol{\Omega}^{i-1}\right) = \sum_{\left\{\mathcal{H}^k,\mathcal{P}^k\right\}} \sum_{b,v} \hat{\epsilon}_{b,v} \log F_{\boldsymbol{\omega}_b^{i-1}}\left(\hat{\boldsymbol{r}}_{b,v}|\boldsymbol{H}_{b,v}\right), \tag{13}$$

where $\hat{\epsilon}_{b,v}$ is a weighting coefficient obtained through meta-learning. To derive $\hat{\epsilon}_{b,v}$, we first take $\left\{\mathcal{H}^k, \hat{\mathcal{R}}^k\left(\boldsymbol{\Omega}^{i-1}\right)\right\}$ as a mini-batch of training data. Then, we consider the NN parameter of BS b updated with the MBGD method as a function of the weighting coefficient vector $\boldsymbol{\epsilon}_b = \left[\epsilon_{b,1}, \epsilon_{b,2}, \cdots, \epsilon_{b,V_b}\right]^{\mathrm{T}} \in \mathbb{R}^{V_b \times 1}$, which is given by

$$\hat{\boldsymbol{\omega}}_b^i\left(\boldsymbol{\epsilon}_b\right) = \boldsymbol{\omega}_b^{i-1} - \lambda_i \nabla_{\boldsymbol{\omega}}\Big[-\sum_{v=1}^{V_b} \epsilon_{b,v} \log F_{\boldsymbol{\omega}}\left(\hat{\boldsymbol{r}}_{b,v}|\boldsymbol{H}_{b,v}\right)\Big]\Big|_{\boldsymbol{\omega}=\boldsymbol{\omega}_b^{i-1}} \tag{14}$$

where λ_i is the learning rate at iteration i, and ∇ denotes the gradient operator. Since we assume that the validation dataset $\mathcal{D}_b^{\mathrm{V}}$ is finely annotated, the optimal weighting coefficient vector $\hat{\boldsymbol{\epsilon}}_b = \left[\hat{\epsilon}_{b,1}, \hat{\epsilon}_{b,2}, \cdots, \hat{\epsilon}_{b,V_b}\right]^{\mathrm{T}}$ at iteration i should be able to locally minimize the validation

loss []. Therefore, the problem of solving $\hat{\boldsymbol{\epsilon}}_b$ can be formulated as

$$\min_{\boldsymbol{\epsilon}_b} \Big[-\sum_{j=1}^{N_b^{\mathrm{V}}} \log F_{\hat{\boldsymbol{\omega}}_b^i(\boldsymbol{\epsilon}_b)} \left(\boldsymbol{p}_{b,j} | \boldsymbol{H}_{b,j} \right) \Big], \tag{15}$$

$$\text{s.t.}\ \epsilon_{b,v} \geq 0, \tag{15a}$$

where (15a) guarantees that all the weighting coefficients are non-negative. Considering that directly solving (15) is time-consuming, we only take a single gradient descent step on the validation dataset with regard to $\boldsymbol{\epsilon}_b$ in order to obtain a cheap approximation of the optimal weighting coefficients at iteration i, which is given by

$$\begin{aligned} \widetilde{\epsilon}_{b,v} &= \frac{\partial}{\partial \epsilon_{b,v}} \Big[\sum_{j=1}^{N_b^{\mathrm{V}}} \log F_{\hat{\boldsymbol{\omega}}_b^i(\boldsymbol{\epsilon}_b)} \left(\boldsymbol{p}_{b,j} | \boldsymbol{H}_{b,j} \right) \Big] \Big|_{\epsilon_{b,v}=0} \\ &= \left\langle \nabla_{\boldsymbol{\omega}} \sum_{j=1}^{N_b^{\mathrm{V}}} \log F_{\boldsymbol{\omega}} \left(\boldsymbol{p}_{b,j} | \boldsymbol{H}_{b,j} \right), \nabla_{\boldsymbol{\omega}} \log F_{\boldsymbol{\omega}} \left(\hat{\boldsymbol{r}}_{b,v} | \boldsymbol{H}_{b,v} \right) \right\rangle \Big|_{\boldsymbol{\omega}=\boldsymbol{\omega}_b^{i-1}}, \end{aligned} \tag{16}$$

with $\langle \cdot, \cdot \rangle$ being the inner product of two vectors. However, since $\widetilde{\epsilon}_{b,v}$ may be a negative number, a rectification towards $\widetilde{\epsilon}_{b,v}$ is needed to satisfy the constrain of (15a). Here, we propose a new rectification method which works better than the rectified linear unit (ReLU) method used in [] especially when the validation dataset is small. Utilizing the proposed method, the finally obtained weighting coefficient $\hat{\epsilon}_{b,v}$ is given by

$$\hat{\epsilon}_{b,v} = 1 + \xi \cdot \frac{\widetilde{\epsilon}_{b,v}}{\max_v |\widetilde{\epsilon}_{b,v}|}, \tag{17}$$

where $\xi \in [0, 1]$ is a parameter used for controlling the scale of rectification.

3) Optimization of the surrogate function: Given $Q^i(\boldsymbol{\Omega})$, the optimization problem at iteration i can be formulated as

$$\max_{\boldsymbol{\Omega}} \gamma \cdot Q^i(\boldsymbol{\Omega}) + (1-\gamma) \cdot \sum_{b=1}^{B} L_b^{\mathrm{L}}(\boldsymbol{\omega}_b). \tag{18}$$

Since $\hat{\boldsymbol{r}}_{b,v}$ can be considered as the label of the unlabeled CSI $\boldsymbol{H}_{b,v}$ according to (12), (18) can be directly solved using supervised learning algorithms. Specifically, the supervised loss function derived from $Q^i(\boldsymbol{\Omega})$ of BS b is

$$L_b^{\mathrm{E2}} = \frac{1}{N^{\mathrm{M}}} \sum_{\left\{\mathcal{H}^k, \mathcal{P}^k\right\} \in \mathcal{D}^{\mathrm{M}}} \frac{\sum_{v=1}^{V_b} \hat{\epsilon}_{b,v} \| \hat{\boldsymbol{r}}_{b,v} - \hat{\boldsymbol{p}}_{b,v} \|_2^2}{\sum_{v=1}^{V_b} \sum_{j=1}^{\hat{V}} \hat{\alpha}_{b,v}^j}, \tag{19}$$

TABLE I
SYSTEM PARAMETERS

Parameter	value	Parameter	Value
B	2	C	3
N^{B}	16	N^{C}	52
W	1280	H	720
λ	10^{-3}	Δh	3 m

where $\hat{\boldsymbol{p}}_{b,v}$ is the output of the NN. The integral loss function is given by

$$L_b^{\mathrm{E}} = \gamma \cdot L_b^{\mathrm{E2}} + (1-\gamma) \cdot L_b^{\mathrm{E1}}. \tag{20}$$

Here, we also take a single step gradient descent with regard to $\boldsymbol{\omega}_b$, and the updated NN parameter at iteration i can be expressed as

$$\boldsymbol{\omega}_b^i \leftarrow \boldsymbol{\omega}_b^{i-1} - \lambda_i \nabla_{\boldsymbol{\omega}} L_b^{\mathrm{E}}\left(\boldsymbol{\omega}\right)\big|_{\boldsymbol{\omega}=\boldsymbol{\omega}_b^{i-1}} \quad . \tag{21}$$

The specific meta-learning based V-EM algorithm is summarized in **Algorithm 1**.

C. Inference Procedure

In the testing stage, the locations of vehicles are jointly infered with CSI and images. In particular, given a pair of testing data $\left\{\mathcal{H}^k, \mathcal{P}^k\right\}$, we first utilize the NN to predict the position coordinate of $\boldsymbol{H}_{b,v} \in \mathcal{H}^k$, which is represented by $\hat{\boldsymbol{p}}_{b,v}$. Next, the corresponding relationship vectors $\hat{\boldsymbol{\alpha}}_{b,v}$ between each $\boldsymbol{H}_{b,v} \in \mathcal{H}^k$ and $\boldsymbol{p}_j \in \mathcal{P}^k$ are obtained by solving (9). Given $\hat{\boldsymbol{\alpha}}_{b,v}$, the estimated position coordinate of $\boldsymbol{H}_{b,v}$ is

$$\widetilde{\boldsymbol{p}}_{b,v} = \begin{cases} \hat{\boldsymbol{p}}_{b,v}, & \hat{\alpha}_{b,v}^{\hat{V}+1} = 1, \\ \sum_{j=1}^{\hat{V}} \hat{\alpha}_{b,v}^{j} \boldsymbol{p}_j, & \hat{\alpha}_{b,v}^{\hat{V}+1} = 0. \end{cases} \tag{15}$$

D. Complexity and Convergence of the Proposed Algorithm

IV. SIMULATION RESULTS AND ANALYSIS

In this section, we evaluate the performance of our proposed positioning algorithm on a public dataset called Vision-Wireless (ViWi) [15]. We first introduce the dataset and the baselines. Then, we analyze the performance of the algorithm.

A. Dataset and Baselines

In ViWi dataset, each data sample consists of six images from the two BSs, the CSI matrices, and position coordinates of the served vehicles. The specific values of the system parameters are given by Table I. The NN models of the two BSs are the same [16]. Here, considering that all the CSI matrices in $\mathcal{D}_b^{\mathrm{L}}$ and $\mathcal{D}^{\mathrm{M}}$ are complex-valued, we utilize the method in [3] to transform them to real-valued matrices. Then, these processed CSI matrices are used as inputs of the NNs, and the outputs are the predicted position coordinates of the CSI. The distances between the BS and its served vehicles are not larger than 55 m. For the vehicles locating in the overlapped serving area of the two BSs, we use the A3 event [17] to determine their connected BSs. After down-sampling and filtering out the unusable samples, we randomly select 2000 samples as $\mathcal{D}^{\mathrm{M}}$, and 500 samples as the testing data. We assume that the labeled dataset at each BS has a same amount of data. In particular, the size of each labeled dataset is set as an experimental variable, which will vary from 200 to 8000, and the size of validation dataset owned by each BS is $\frac{1}{3}$ of the labeled dataset. For comparison purposes, we consider two baselines. In baselines a), we directly use the models trained with $\mathcal{D}_b^{\mathrm{L}}$ to predict the vehicle locations in testing datasets. In baseline b), we utilize the same model as in baseline a). However, the output position coordinates of the NNs will be further calibrated with the locations obtained from images as described in (15).

B. Performance Evaluation

To evaluate the performance of our proposed method, we first calculate the mean Euclidean distances between the predicted and ground truth position coordinates [18] at each BS as the positioning error. Then, the average positioning error of all BSs is utilized as the performance measurement.

Fig. 4 shows an example of using images to assist vehicle positioning when $N_b^{\mathrm{L}} = 1000$. In particular, we use different colors and shapes to represent the ground truth locations, the locations obtained from images, and the predicted locations of the vehicles served by BS 1 and BS 2. From this figure, we see that most of the predicted vehicle locations are correctly corresponded to the locations obtained from images. Hence, compared to the method that localizes vehicles using only with CSI data, the mean positioning error of the proposed algorithm that uses multi-modal data is much lower.

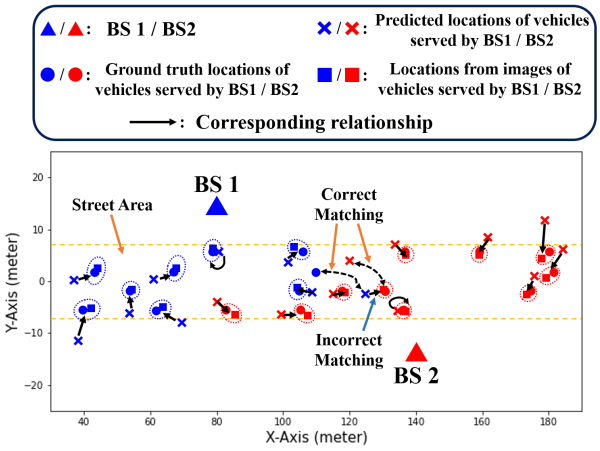

Fig. 4. An example of image aided vehicle positioning.

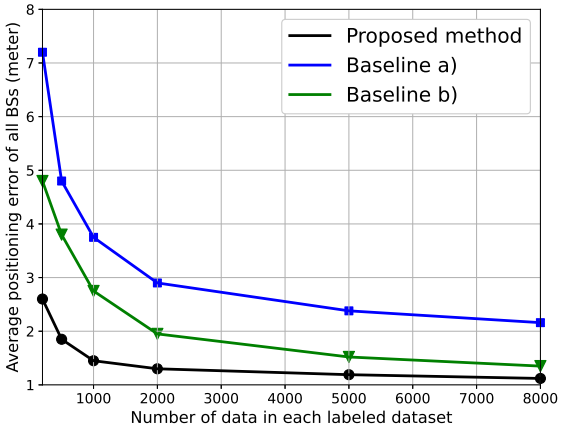

Fig. 5. Average positioning error with different amount of labeled data.

In Fig. 5, we show how the average positioning error changes as the number N_b^{L} of labeled data samples owned by each BS varies. From this figure, we see that the proposed algorithm can reduce the average positioning error by up to 64% when $N_b^{\mathrm{L}} = 200$. This is because of two reasons. First, the designed EM algorithm can effectively train the NN models with dataset $\mathcal{D}^{\mathrm{M}}$, and thus lower the positioning error. Second, the locations predicted by the NNs are calibrated by the position coordinates obtained from images, which further improves the positioning accuracies. Fig. 5 also shows that the proposed method can reduce the positioning error by up to 46% compared to baseline b). This is because the proposed algorithm trains the NNs using the multi-modal dataset $\mathcal{D}^{\mathrm{M}}$.

V. Conclusion

In this paper, we have developed a novel multi-modal multi-BS vehicle positioning framework that jointly utilize images and CSI fingerprints to localize vehicles. We have considered a practical communication scenario where each vehicle can communicate with only one BS at the same time, and hence, it can upload its estimated CSI to only its associated BS. Each BS is equipped with a set of cameras, and it can collect a large number of unlabeled CSI, the images taken by cameras, and only a small amount of labeled CSI for model training. To exploit the unlabeled CSI data and position labels obtained from images, we design an expectation-maximization (EM) based DL algorithm that iteratively trains the DL model of each BS. Simulation results show that the proposed method can outperform the baseline where the images are not used to aid the CSI fingerprint based vehicle positioning.

References

[1] C. Laoudias, A. Moreira, S. Kim, S. Lee, L. Wirola, and C. Fischione, "A survey of enabling technologies for network localization, tracking, and navigation," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 3607–3644, July 2018.

[2] X. Sun, C. Wu, X. Gao, and G. Y. Li, "Fingerprint-based localization for massive MIMO-OFDM system with deep convolutional neural networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 11, pp. 10 846–10 857, November 2019.

[3] 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 Communications and Networking Conference*, Nanjing, China, March 2021, pp. 1–6.

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

[5] A. Foliadis, M. H. C. Garcia, R. A. Stirling-Gallacher, and R. S. Thomä, "Reliable deep learning based localization with CSI fingerprints and multiple base stations," in *Proc. IEEE International Conference on Communications*, Seoul, Korea, May 2022, pp. 3214–3219.

[6] B. Salehi, G. Reus-Muns, D. Roy, Z. Wang, T. Jian, J. Dy, S. Ioannidis, and K. Chowdhury, "Deep learning on multimodal sensor data at the wireless edge for vehicular network," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 7, pp. 7639–7655, July 2022.

[7] J. Jiao, F. Li, Z. Deng, and W. Liu, "An indoor positioning method based on wireless signal and image," in *Proc. International Congress on Image and Signal Processing, BioMedical Engineering and Informatics*, Datong, China, October 2016, pp. 656–660.

[8] A. Alahi, A. Haque, and F. Li, "RGB-W: When vision meets wireless," in *Proc. IEEE International Conference on Computer Vision*, Santiago, Chile, December 2015, pp. 3289–3297.

[9] T. Nguyen and R. Raich, "Incomplete label multiple instance multiple label learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 3, pp. 1320–1337, March 2022.

[10] D. Feng, L. Lu, Y. Yuan-Wu, G. Y. Li, G. Feng, and S. Li, "Device-to-device communications underlaying cellular networks," *IEEE Transactions on Communications*, vol. 61, no. 8, pp. 3541–3551, August 2013.

[11] Z. Zhu, C. Guo, R. Bao, M. Chen, W. Saad, and Y. Yang, "Positioning using visible light communications: A perspective arcs approach," *IEEE Transactions on Wireless Communications*, to appear, March 2023.

[12] D. Kang and D. Kum, "Camera and radar sensor fusion for robust vehicle localization via vehicle part localization," *IEEE Access*, vol. 8, pp. 75 223–75 236, April 2020.

[13] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Computer Vision & Pattern Recognition*, 2016.

[14] O. Nazih, N. Benamar, H. Lamaazi, and H. Chaoui, "Challenges and future directions for security and privacy in vehicular fog computing," in *Proc. International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT)*, Sakheer, Bahrain, November 2022, pp. 693–699.

[15] M. Alrabeiah, J. Booth, A. Hredzak, and A. Alkhateeb, "ViWi vision-aided mmWave beam tracking: Dataset, task, and baseline solutions," *arXiv.2002.02445*, 2020.

[16] A. Foliadis, M. H. C. 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 Communications and Networking Conference (WCNC)*, Glasgow, United Kingdom, March 2023, pp. 1–6.

[17] Y. Chen, K. Niu, and Z. Wang, "Adaptive handover algorithm for LTE-R system in high-speed railway scenario," *IEEE Access*, vol. 9, pp. 59 540–59 547, April 2021.

[18] R. Kant, P. Saini, and J. Kumari, "Long short-term memory auto-encoder-based position prediction model for fixed-wing UAV during communication failure," *IEEE Transactions on Artificial Intelligence*, vol. 4, no. 1, pp. 173–181, February 2023.