DeepAoA+: Online Cross-Domain Vehicular Relative Direction Estimation via Deep Learning

Facheng Hu, *Graduate Student Member, IEEE*, Yunxiang Cai, Hongzi Zhu, *Senior Member, IEEE*, Shan Chang, *Member, IEEE*, Xudong Wang, *Fellow, IEEE*, and Minyi Guo, *Fellow, IEEE*

***Abstract*—Relative direction estimation among neighboring vehicles in urban environments is essential to a wide variety of driving safety applications. To obtain accurate direction information solely from vehicle-to-vehicle (V2V) communications is desirable but very challenging due to enormous multipath effects. In this paper, we propose an online cross-domain vehicular relative direction estimation method, called DeepAoA+, based on a domain adaptive deep learning method. More specifically, the channel state information (CSI) is estimated from a set of synchronized receiving radios at a receiver vehicle. By taking the CSI phase difference between a pair of such radios, CSI phase errors in the baseband can be effectively eliminated, which makes a compelling feature to represent the direction of incident radio frequency (RF) signals and the dynamic channel characteristics. Moreover, to conquer the non-*i.i.d* (independent and identically distributed) problem caused by dynamic environments, DeepAoA+ discovers latent domains in the CSI phase difference dataset. For each identified domain, a domain-specific model is trained to estimate the Angle-of-Arrival (AoA) of incoming signals. To this end, a framework based on an expectation-maximization (EM) algorithm is proposed to obtain domain partitions considering downstream AoA estimation tasks. Furthermore, a pre-training technique is used to enhance the robustness of AoA estimation with misclassified test samples. We implement a prototype of DeepAoA+, using four synchronized Universal Software Radio Peripherals (USRPs) from various scenarios, and conduct extensive real-world experiments. The results demonstrate that DeepAoA+ can achieve the best AoA estimation with low latency.**

***Index Terms*—AoA, CSI, vehicular network, cross-domain, deep learning.**

I. Introduction

ACCURATE relative direction information among neighboring vehicles can be the cornerstone of many driving safety applications. For instance, the cooperative collision warning system [1] relies on vehicle-to-vehicle (V2V) communication to exchange crucial driving status information including the relative direction and range of approaching vehicles. Though dedicated sensors can be mounted on vehicles to obtain relative direction information, one major drawback is that such sensors are isolated from V2V radios, which makes it hard to inform those vehicles concerned through V2V communication. Besides, sensors like Light Detection and Ranging (LiDAR) [2], [3] and cameras [4] are sensitive to harsh weather or poor light conditions. Localization systems such as Global Positioning System (GPS) can be used to calculate relative location or direction information but the accuracy degrades a great deal when encountering urban canyons [5]. Therefore, it is of great benefit to directly estimate the Angle-of-Arrival (AoA) information from radio frequency (RF) signals in V2V communication, which is referred to as *the online vehicular relative direction finding problem*.

A practical scheme for the online vehicular direction-finding problem should meet the following four stringent requirements: 1) the estimated AoA should be extremely precise, especially for driving safety applications; 2) the scheme should be robust to various vehicular environments where RF signals may be severely faded due to the multipath effect; 3) the scheme should have a short response time as outdated AoA information would be ineffective; 4) the scheme should be easily deployed without the need of dedicated sensors beyond affordable V2V radios.

In the literature, a wide spectrum of RF-based direction-finding methods have been proposed. A rich set of schemes based on interferometry, which can be used to determine the AoA of a wave by directly measuring the phase difference of signals received at different elements of an antenna array [6]. For instance, small-aperture systems, with the spacing between adjacent antennas no greater than half of the wavelength, can achieve unambiguous determination of the azimuth and elevation but they are vulnerable to multipath fading. Another category of direction-finding methods is subspace-based, which aims at eliminating the effect of noise by splitting up the m-dimensional space spanned by the element outputs into subspaces with an m-antenna array. For example, Multiple Signal Classification (MUSIC) [7] is based on the fact that signals lie perpendicular to the noise subspace. The reciprocal value is usually used as the direction-finding function so that distinct peaks occur in the signal directions. Subspace methods require that the number of antenna elements is larger than the number of incoming signals. Consequently, such methods cannot achieve high

Received 30 October 2023; revised 2 March 2024 and 15 June 2024; accepted 21 July 2024. Date of publication 19 August 2024; date of current version 19 December 2024. This work was supported by the National Natural Science Foundation of China under Grant 62432008. The review of this article was coordinated by Dr. Ebrahim Bedeer. *(Facheng Hu and Yunxiang Cai are co-first authors.) (Corresponding author: Hongzi Zhu.)*

Facheng Hu, Hongzi Zhu, and Minyi Guo are with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: facheng_hu@sjtu.edu.cn; hongzi@sjtu.edu.cn; guo-my@cs.sjtu.edu.cn).

Yunxiang Cai is with Huawei Technologies Company Ltd. Shanghai 200121, China (e-mail: caiyunxiang@sjtu.edu.cn).

Shan Chang is with the School of Computer Science and Technology, Donghua University, Shanghai 201620, China (e-mail: changshan@dhu.edu.cn).

Xudong Wang is with UM-SJTU Joint Institute, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: wxudong@ieee.org).

Digital Object Identifier 10.1109/TVT.2024.3445722

accuracy in highly dynamic settings. One recent scheme [8] mounts four antennas on a vehicle and roughly determines the relative location of other vehicles by comparing the Received Signal Strength Indication (RSSI) measured on each antenna. This method is simple but lacks good accuracy. Previously, we have proposed a deep-learning-based system that can achieve good performance under controlled scenes [9], but the performance drops dramatically when switching to a scenario unfamiliar to the trained model. As a result, to the best of our knowledge, there is no existing solution to the online vehicular relative direction-finding problem.

In this paper, we propose a cross-domain vehicular relative direction finding scheme, called *DeepAoA+*, which is solely based on normal Orthogonal Frequency Division Multiplexing (OFDM)-based V2V communication (e.g., Dedicated Short-Range Communications (DSRC) or Cellular Vehicle-to-Everything (C-V2X)) packets. The core idea of DeepAoA+ is to train compelling deep neural network (DNN) models to extract effective features from channel state information (CSI) of V2V packets so that each model can well characterize the correlation between AoAs and received CSIs in a particular dynamic vehicular scenario. Instead of putting all available training data collected in various vehicular scenarios in training one single DNN model, DeepAoA+ first automatically identifies different latent feature domains and trains a model separately for each domain scenario. By adaptively selecting the best fit pre-trained DNN model for online AoA estimation in mobile settings, DeepAoA+ can achieve accurate AoA estimation even in highly dynamic vehicular environments.

There are three main challenges in designing DeepAoA+. First, it is non-trivial to obtain stable and accurate CSI phase information from each element of an antenna array. CSI phase errors are prevalent due to hardware imperfection in wireless signal processing [10], which leads to the measured CSIs containing unpleasant noise for AoA estimation. To address this challenge, we calculate the CSI phase difference between each pair of two synchronized radios at 52 sub-carriers so that linear phase errors can be eliminated. Therefore, the obtained pairwise CSI phase differences are mainly related to the direction of incident waves and the channel characteristics.

Second, CSI phase difference data collected from highly dynamic vehicular environments are normally non-*i.i.d.* (independent and identically distributed), making it hard to train one single model that fits all scenarios. To deal with this challenge, we employ an expectation-maximization (EM)-based framework to iteratively train an encoder. Specifically, each EM iteration consists of an E-step and an M-step. In the E-step, the encoder transforms CSI phase difference data into embeddings in feature space, which are then grouped into distinct domains using K -means. In the M-Step, we design a two-factor loss function to evaluate the current encoder. More specifically, we first use a contrastive-learning-based loss to assess the disparity between domains, referred to as the *domain disparity loss*. Moreover, a DNN model is first trained on a specific domain. Then, we conduct trial tests on the trained DNN model within the same domain and use the loss of trial tests to assess domain quality with respect to the specific downstream task, referred to as the *domain quality loss*. We combine both losses to optimize the encoder with a Stochastic Gradient Descent (SGD) method. After the EM algorithm converges, an effective encoder can be obtained, with which obtained embeddings can be easily partitioned into latent channel-specific domains and each domain can be well characterized by a DNN model for AoA estimation.

Third, during online AoA estimation, it is inevitable that some test samples are misclassified into a wrong domain and fed into a wrong domain model, resulting in catastrophic prediction errors. To tackle this issue, we employ a pre-training approach to equip each domain model with the basic ability to handle the samples of other domains, making AoA estimation results more stable. To this end, we first train a basic model on data from all domains for a small number of epochs, thus giving it a comprehensive global view. Then we train each channel-specific model by fine-tuning the pre-trained model on corresponding domains. Therefore, DeepAoA+ can maintain reliable estimation accuracy even for misclassified samples.

We implement a prototype system of DeepAoA+, consisting of a receiver and a transmitter. The receiver is composed of four synchronized Universal Software Radio Peripheral (USRP) N210s and an Ubuntu PC running GNU Radio for signal processing. The transmitter is equipped with an Arada Locomate DSRC On-Board Unit (OBU) to broadcast Wireless Access in Vehicular Environment (WAVE) Short Messages (WSMs). The ground truth is labeled with a 32-line RoboSense LiDAR. We collect CSI traces of real-world V2V communication in four typical scenarios in Shanghai including the rural region, the parking lot, the surface road, and the highway. Results of extensive experiments demonstrate that DeepAoA+ can achieve low AoA estimation errors of less than 12° with an 80% confidence interval.

We highlight our main contributions in this paper as follows:

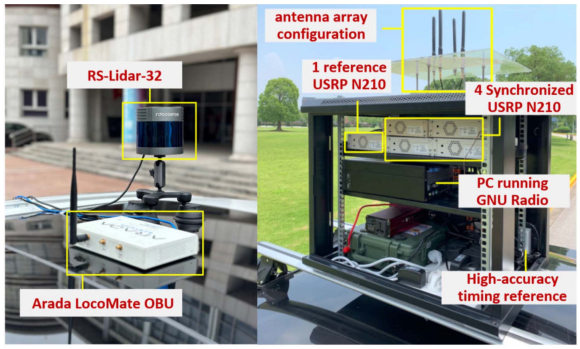
- We propose a cross-domain vehicular relative direction estimation scheme, called DeepAoA+, which solely relies on V2V communication and can automatically identify the current channel scenario and select a best-trained DNN model for reliable and accurate AoA estimation.
- We implement a prototype system to collect CSI traces in four typical urban scenarios with a LiDAR-based ground truth system. The dataset is open-sourced to promote the development of solving the online vehicular relative direction estimation problem.[1]
- We evaluate DeepAoA+ with real-world traces and the results demonstrate the efficacy of DeepAoA+.

II. Related Work

We classify the existing work that is relevant to our work into two categories, i.e., AoA estimation schemes aiming to estimate AoA with RF signal, and domain partition schemes aiming to establish cross-domain models.

AoA Estimation Schemes: Interferometry utilizes the phase difference between the signals collected from different antennas

[1]The DOI Link of the dataset is *https://dx.doi.org/10.21227/452q-hk60*



(a) (b)

Fig. 1. The transmitter vehicle with an Arada LocoMate OBU and a LiDAR and the receiver vehicle with four synchronized USRPs and an antenna array. (a) Tx vehicle. (b) Rx vehicle.

on the received signals to determine the AoA of a wave. Tzur et al. [11] uses multiple receiving antennas and OFDM-based devices to implement an interferometry direction-finding system. Their scheme utilizes only two antennas but lacks accuracy. Some schemes utilize RSSI to achieve AoA estimation. Niculescu et al. [12] adds a rotating directional antenna to the base station to measure both angles and ranges. Wang et al. [13] uses two specialized antennas that keep rotating to find the azimuth and elevation with the strongest RSSI. Those methods can achieve high accuracy but require special hardware and need to overhear many packets to collect enough RSSI data. Another category of methods is subspace-based AoA estimation. Subspace-based techniques produce the spatial spectrum by using eigen-decomposition of the covariance matrix of input signals, from which AoA is estimated. One of the earliest algorithms within the family of subspace methods for AoA estimation is MUSIC [7]. However, MUSIC requires the number of antenna elements larger than the number of incoming signals and the number of incoming signals must be known as prior knowledge, which is not possible in most vehicular cases. ArrayTrack [14] uses a variant of MUSIC algorithm to estimate AoA which requires eight antennas. It achieves high accuracy but is hard to deploy. SpotFi [15] proposes a super-resolution AoA estimation algorithm based on MUSIC algorithm, which jointly estimates the AoA and ToF from the CSI information of the Wireless Fidelity (Wi-Fi) OFDM symbols. It performs well in localization problems but cannot work effectively in online AoA estimation due to high computation overheads. Besides, previously we have proposed a deep-learning-based scheme that can achieve good performance under controlled scenes [9], but the accuracy decreases dramatically when switching to a scenario unfamiliar to the trained. To the best of our knowledge, no existing estimation schemes can achieve high accuracy efficiently without dedicate devices.

Domain Partition Schemes: In the realm of data partitioning, one natural approach is to perform clustering before training, such as in the case of Clustered Partition [16]. These approaches can be further divided into two categories: prior information-based and historical sample-based. The first kind of method uses prior information to bring similar domains closer [17], [18]. However, obtaining such prior information can be difficult, limiting the practicality of this method in real-world applications. Some schemes use samples for automatic clustering [19], [20]. This kind of method relies solely on samples or features, which can lead to missing intrinsic information in complex systems [21], [22]. Another option is to use downstream task labels for task-specific data partitioning [23], [24]. TForest [23] proposes a task-oriented data grouping strategy based on a greedy method, while LEON [24] proposes an online updating method for task-specific data partitioning. However, all these methods still require prior information for initial data partitioning. As a result, we investigate whether dynamic vehicular environments can be automatically partitioned into distinct domains without the reliance on prior information.

III. System Model and Design Goals

A. System Model

We consider the online vehicular relative direction estimation problem in urban vehicular networks, where a node can be a moving vehicle or a static roadside unit (RSU). We do not consider the differences of vehicle heights because the differences are significantly smaller than the horizontal distances between vehicles. A node is considered to have a set of synchronized radios, a stable power supply, and sufficient computational capability. No other dedicated infrastructure or sensors are required. Moreover, nodes communicate with each other by OFDM-based V2V communication protocols such as IEEE 802.11p and C-V2X. Each vehicle plays the following two roles in the network:

- *Transmitter:* a vehicle needs to periodically broadcast its driving status information to its neighbors, using one of its radios. For example, according to IEEE 1609 standards, a vehicle broadcasts WSMs ten times per second on the Control Channel (CCH).
- *Receiver:* when a vehicle is not transmitting packets, it listens to the channel and receives packets sent from other vehicles in its vicinity. In addition, all synchronized radios are used to perceive the AoA of a received packet at the same time.

B. Design Goals

An online vehicular direction-finding system should satisfy the following requirements:

- *High accuracy:* The derived AoA estimation should be accurate. For example, at a distance of 40 meters, an error of 15° will cause a localization error of over 10 meters, which could lead to false negative errors or even car accidents.
- *Good reliability:* In urban vehicular environments, RF signals can be reflected or blocked by moving vehicles and roadside buildings. The system should reliably work in such dynamic environments.
- *Short response time:* The system should only use normal small V2V communication packets and should give an accurate AoA estimation in a short response time. Out-of-date information may mislead drivers to make wrong decisions.
- *Low cost:* The system should be cost-efficient and easy to deploy. As V2X radios would be mandatory for future

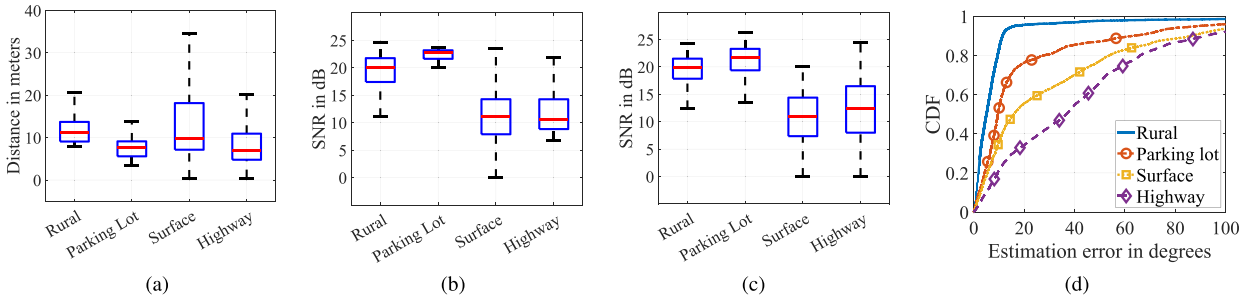


Fig. 2. (a) Distributions of the distance d between the Tx and Rx vehicles; (b) the distributions of SNR when $d <$10 m; (c) the distributions of SNR when $d >$ 10 m; (d) CDFs of AoA estimation errors using the MUSIC algorithm in four vehicular scenarios.

vehicles, the system should solely rely on V2X radios and minimize the hardware cost.

IV. Empirical Study

A. Real-World V2V CSI Trace Collection

As illustrated in Fig. 1, we deploy two experimental vehicles, i.e., one transmitter (Tx) vehicle and one receiver (Rx) vehicle. For the Tx vehicle, an off-the-shelf Arada LocoMate OBU is mounted on the roof of the vehicle, which implements IEEE 802.11p and IEEE 1609 standards and periodically sends ten WSM packets per second. For the Rx vehicle, five USRP N210s constitute a set of four synchronized radios with omnidirectional antennas, forming a uniform linear array (ULA) of the small aperture by placing antennas with a displacement of half of the wavelength $\lambda = 5.093$ cm of the central frequency 5.89 GHz. Specifically, four Rx N210s with CBX daughterboards are synchronized with an Ettus Octo-Clock-G and their initial PLL phase differences are calibrated with the fifth N210. All five USRPs are controlled by a PC running GNU Radio. We revise the IEEE 802.11p transceiver flow graph to detect 802.11p packets and extract CSI from all receiving USRPs. In addition, each vehicle is equipped with a Robosense 32-line LiDAR.

We collect a CSI trace of V2V communication in four typical vehicular scenarios, i.e., a rural region with no surrounding buildings and passing vehicles, a parking lot with many parked vehicles, the urban surface road, and the highway, for two hours per day from July 6 to July 22 in the year 2022 in Shanghai. Specifically, both the rural region and the parking lot scenarios are static, where the Rx vehicle is parked at a fixed position and the Tx vehicle passes the receiver vehicle slowly while transmitting packets. In contrast, both the surface road and the highway scenarios are dynamic, where both vehicles, one in front and one behind, move in adjacent lanes with various surrounding traffic conditions.

In all scenarios, the Tx vehicle keeps sending ten WSM packets per second while the Rx vehicle keeps extracting CSI from each of its four radios and recording the signal-to-noise ratio (SNR) at the same time. In addition, the two vehicles also record LiDAR point clouds for obtaining the ground truth of the AoA and the distance between both vehicles. We remove samples without ground truth due to the obstruction of passing vehicles.

As a result, we obtain a CSI trace of 51,477 samples, denoted as $\mathcal{T}_{CSI}$, with each sample consisting of a set of four CSI estimates extracted from the receiver's antenna array. The numbers of samples in the rural region, the parking lot, the surface road, and the highway scenarios are 17,508, 8,307, 17,409, and 8,253, respectively.

B. AoA Estimation Analysis With MUSIC Algorithm

We first analyze the V2V channel conditions in these scenarios. Specifically, Fig. 2(a) depicts the box plots of the distance between experimental vehicles, denoted as d, in the aforementioned four scenarios. It can be seen that both dynamic scenarios have larger distance ranges and variances than static scenarios. For example, the distance between the transmitter and the receiver varies from 0 to 35 meters on the surface road. Fig. 2(b) and (c) depict the SNR of received signals when $d < 10m$ and $d > 10m$, respectively. It can be seen that in both cases dynamic scenarios have much lower SNR than static scenarios, indicating that V2V channels are more severely impaired in dynamic scenarios.

We then examine AoA estimation in the aforementioned four vehicular scenarios using the MUSIC algorithm. Fig. 2(d) illustrates the cumulative distribution functions (CDFs) of the estimation error in each scenario. In general, the MUSIC algorithm cannot obtain reliable and accurate AoA estimation with our antenna array configuration in vehicular scenarios. For instance, it can achieve low AoA estimation errors of 10° with an 80% confidence interval in the rural region scenario but result in huge AoA estimation errors of 60° with an 80% confidence interval in the highway scenario.

C. Difficulty of AoA Estimation in Vehicular Environment

We further demonstrate the difficulty of AoA estimation in the vehicular environment. Specifically, we pull over our Tx and Rx vehicles on a surface road segment in an urban area of Shanghai, where, as illustrated in Fig. 3(a) and (b), both vehicles experience three following scenes and the Rx vehicle estimates the AoA in each scene with MUSIC:

- *Scene 1 (S_1):* The signal transmitted from the Tx can directly reach the Rx with few reflections from nearby objects. It can be seen that, in this case, existing direction

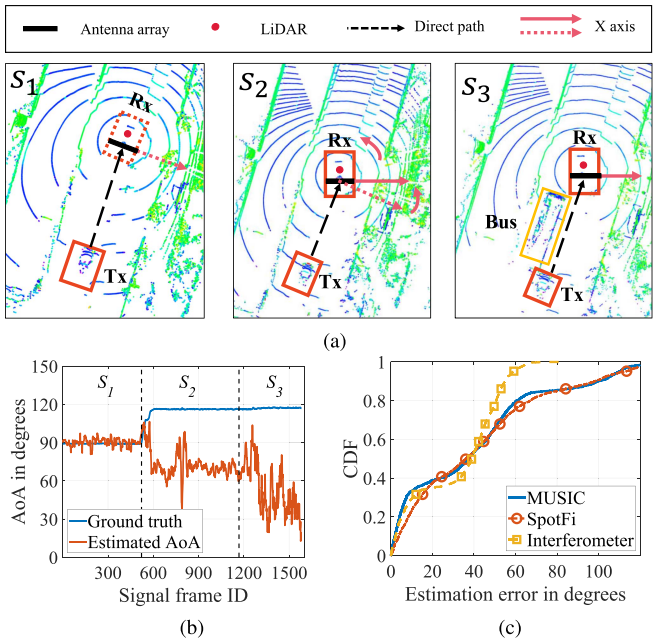


Fig. 3. Illustration of the difficulty of AoA estimation in vehicular environments. (a) Illustration of three distinct scenes encountered during real-world study. (b) AoA estimated with MUSIC in each scene. (c) CDFs of AoA estimation errors over all scenes.

estimation methods such as MUSIC can achieve accurate AoA estimation with a low average error of 6°.

- *Scene 2 (S_2):* We slightly change the orientation of the Rx vehicle, which makes the AoA ground truth slowly increase to near 120° while the AoA estimated with MUSIC changes to around 60°. It shows that the multipath profile may change dramatically even with a subtle variation between the Tx and Rx pair.
- *Scene 3 (S_3):* The Tx and Rx vehicles remain unchanged except that there is a bus passing by in the next lane, which causes the estimated AoA to fluctuate vigorously from around 30° to 90°. It demonstrates that the vehicular environment brings severe challenges to classic AoA estimation algorithms.

Fig. 3(c) depicts the CDFs of absolute AoA estimation errors across all scenes above obtained with three classic AoA estimation algorithms, i.e., MUSIC, SpotFi, and interferometer. It can be seen that none of them can achieve reliable and accurate AoA estimation in the vehicular environment.

V. Design of DeepAoA+

A. System Overview

The core idea of DeepAoA+ is for a receiver to utilize normal V2V packets to perceive the relative directions of respective transmitters. Specifically, DeepAoA+ trains different DNN models to capture the correlation between AoAs and CSI phase information of received packets in various scenarios. When predicting the AoA of a received packet, the best pre-trained DNN can be selected, achieving accurate AoA estimation even in highly dynamic vehicular environments. The architecture of DeepAoA+, as depicted in Fig. 4, consists of four components as follows.

CSI Phase Differences (CPD): Upon receiving a packet, m raw CSIs can be obtained from a m-antenna array. DeepAoA+ eliminates CSI phase errors by subtracting the CSI phase between each pair of antennas, resulting in a sample of C_m^2 groups of pairwise CSI phase differences. Such samples of CSI phase differences are used for AoA estimation.

Channel-specific Domain Estimation (CDE): For the same AoA, different channel conditions lead to distinct CSI phase difference patterns. CDE utilizes an iterative EM algorithm to identify latent channel-specific CSI phase difference domains in the feature space. In the E-step, samples of CSI phase differences are converted to features via an encoder $\mathcal{M}_{enco.}$, and similar features are then grouped into domains using K-means clustering. In the M-step, the quality of the identified domains is evaluated, which is used to optimize the parameters of the encoders $\mathcal{M}_{enco.}$. This process repeats until it converges.

Domain Model Training (DMT): Given K identified domains, DMT trains K corresponding domain-specific models, denoted as $\{\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_K\}$, for online prediction. Specifically, DMT is based on model pre-training where the initial model pre-trained with all available data is finetuned on each identified domain, obtaining these domain-specific models.

Online AoA Estimation (OAE): Domain-specific models $\{\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_K\}$ and $\mathcal{M}_{enco.}$ can be downloaded to vehicles for online AoA estimation. Specifically, a test sample of CSI phase differences is converted into a feature using $\mathcal{M}_{enco.}$ and classified into a particular domain by comparing the L_2 distance of the feature and centers of existing domains. Then, the corresponding domain-specific model is then selected for AoA estimation.

B. CSI Phase Differences

Fig. 5 illustrates the principle of using CSI phase information for AoA estimation. Given a pair of antennas with a displacement of d meters, if the phase of an incoming wave of frequency f with light speed c can be accurately measured at both antennas, denoted as ϕ_1 and ϕ_2, respectively, the direction of incidence θ can be determined by the difference between ϕ_1 and ϕ_2 with the equation $e^{j(\phi_1-\phi_2)} = e^{j(2\pi d sin\theta f/c)}$. The physical layer protocol in V2V communication, such as 802.11p and C-V2X, is based on OFDM modulation, making it possible for a receiver to measure the phase of all sub-carrier frequencies by calculating the CSI when receiving such a V2V packet. For an 802.11p packet, a CSI consisting of 52 used sub-carriers including 48 used for data transmission and 4 used as pilot sub-carriers, can be obtained. DeepAoA+ does not directly calculate the direction of a transmitter using the measured CSI phases but uses samples of CSI phase differences to train DNN models. Without loss of generality, we adopt the structure of IEEE 802.11p to describe the acquisition of the CSI phase difference signal.

1) Channel Estimation: CSI characterizes the properties of a communication channel and represents the combined effect of fading, scattering, and power delay over distance. The preamble structure of IEEE 802.11p packet is utilized by the receiver radio

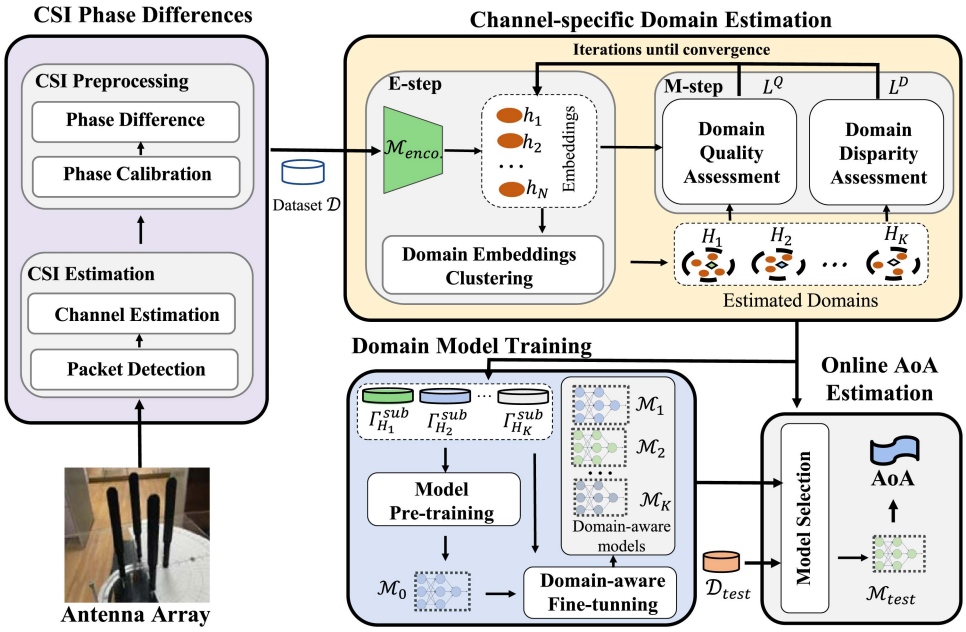


Fig. 4. System architecture of DeepAoA+, which consists of four main components, i.e., CSI phase difference, channel-specific domain estimation, domain model training, and online AoA estimation.

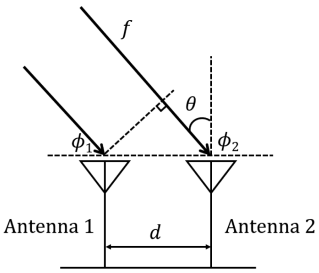


Fig. 5. An illustration example of the relation between the direction θ of an incident wave of frequency f and phases measured on two parallel antennas, denoted as ϕ_1 and ϕ_2, respectively.

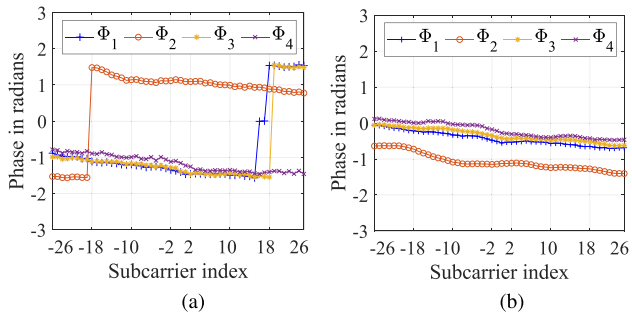


Fig. 6. Two groups of CSI phase measured for two packets with a temporal interval of 10 ms on four synchronized radios, which are fluctuating and not stable as features for estimation. (a) Using packet one. (b) Using packet two.

to detect a packet through autocorrelation using the short training sequence. Then two identical long training sequences are utilized to estimate the CSI. To ensure stability, the spectral and time averaging (STA) scheme [25] is adopted in our estimation method.

We denote $\phi_{i,j}$ as the received phase of sub-carrier j from a given receiving antenna i, and $H_i(j)$ as the CSI for sub-carrier j from the aforementioned antenna. The received phase is represented mathematically as $\phi_{i,j} = arg(H_i(j))$, where $arg(\cdot)$ refers to the get the angle of a complex number. Fig. 6(a) illustrates the CSI phase measurement obtained when a packet is received by four synchronized radios, while Fig. 6(b) shows the CSI phase measurement after 10 ms received by the same radios. It can be observed that different radios yield different CSI phase measurements for the same packet. Besides, the CSI phase measurements fluctuate on sequential packets.

2) CSI Preprocessing: From the example presented in Fig. 6, it is evident that raw CSI phases cannot represent stable characteristics of the channel between a transmitter and receiver. It is mainly caused by frequency response errors in signal processing circuits. In addition to measurement noise, a prior study [10] has identified several sources of CSI phase errors, such as:

- *Carrier Frequency Offset (CFO):* The central frequencies of a transmitter and receiver pair may not be in perfect alignment, leading to temporal variations in the CSI phase offset across sub-carriers.
- *Sampling Frequency Offset (SFO):* As a result of unsynchronized clocks, there may be a discrepancy between the sampling frequencies of the transmitter and receiver, which can lead to the temporal displacement of the received signal in relation to the transmitted signal, as well as errors in phase rotation.
- *Packet Detection Delay (PDD):* Detection of packets incurs an additional temporal shift relative to the transmitted signal, resulting in errors of variable phase rotation per packet.

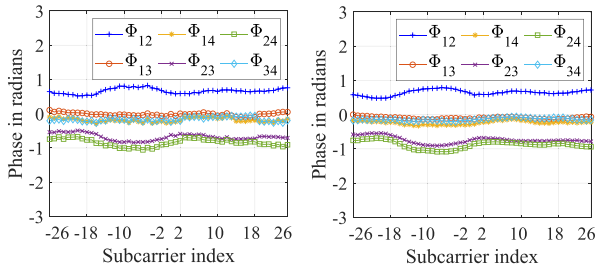


Fig. 7. Two groups of CSI phase difference measured for two packets with a temporal interval of 10ms on four synchronized radios, which remain stable between adjacent packets. (a) Using packet one. (b) Using packet two.

- *PLL Phase Offset (PPO):* The utilization of a phase-locked loop (PLL) in the generation of the central frequency for both the transmitter and the receiver can result in an initial random phase offset that subsequently leads to additional phase deviation at the receiver.

From the above-identified sources of error, CSI phase measures are primarily affected by various phase rotation and offset errors. For a given receiving antenna i, the measured phase of sub-carrier j, $\phi_{i,j}$, can be represented as

$$\phi_{i,j} = \theta_{i,j} - 2\pi \cdot j \cdot f_s \cdot \delta_i + \beta_i + Z_i, \tag{1}$$

where δ_i denotes the timing offset at the receiver, which includes time shift due to PDD and SFO, f_s represents the sub-carrier spacing between two adjacent sub-carriers (e.g. 156.25KHz), β_i represents the total phase offset and Z_i represents the additive white Gaussian measurement noise. It is noteworthy that, except Z_i, the reported phase errors are linear to sub-carrier indexes.

To use the CSI phase, it is essential to eliminate all phase errors in the baseband. If two receiving radios are synchronized, meaning they share the same center frequency and sample rate, they possess CFO, SFO, PDD, and PPO relative to the same transmitter. As a result, the measured phase difference of sub-carrier j between radio 1 and radio 2 can be calculated by $arg(H_1(j)/H_2(j))$ and can be expressed as

$$\Delta\phi_{12} = \phi_{1,j} - \phi_{2,j} = \theta_{1,j} - \theta_{2,j} + Z', \tag{2}$$

where $\theta_{1,j} - \theta_{2,j}$ represents the desired passband phase difference of sub-carrier j, and $Z' = Z_1 - Z_2$ represents the residual Gaussian noise.

Based on the above discussion, we can construct an accurate CSI phase difference signal by subtracting the CSI phase between each pair of antennas. We denote the CSI phase difference between two radios as

$$\Delta\Phi_{12} = [\phi_{12,-26}, \phi_{12,-25}, \ldots, \phi_{12,26}]. \tag{3}$$

As depicted in Fig. 7, the CSI phase differences between each pair of four synchronized receiving radios remain stable across the same two packets as discussed in Fig. 6. It can be observed that CSI phase differences not only remain smooth and stable despite unstable raw CSI phases within one packet but also exhibit consistency between adjacent packets. As a result, we consider pairwise CSI phase differences as a reliable signal for representing the direction of incident RF waves and the channel characteristics. In general, for m synchronized receiving radios, the number of radio pairs would be $C_m^2 = \frac{m(m-1)}{2}$. For instance, with four synchronized radios, six groups of distinct CSI phase differences can be obtained when receiving a packet, denoted as x,

$$x = [\Delta\Phi_{12}, \Delta\Phi_{13}, \Delta\Phi_{14}, \Delta\Phi_{23}, \Delta\Phi_{24}, \Delta\Phi_{34}]. \tag{4}$$

C. Channel-Specific Domain Estimation

The dataset of CSI phase differences encompasses signals from all directions, covering both the direct path and multi-paths, constituting a composite of channels. While traditional methods such as MUSIC may encounter challenges in distinguishing multipath signals, we utilize DNN-based models for its better performance in learning discrepancies through more effective feature extraction. The dynamic multipath effects in the vehicular environment introduce non-*i.i.d.* characteristics to the CSI phase difference dataset, making it challenging to train a single sophisticated DNN model with satisfactory performance. It is better to automatically partition $\mathcal{D}$ into domains $\{\Gamma_1, \Gamma_2, \ldots, \Gamma_K\}$ with distinct channel conditions and to train individual models $\{\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_K\}$ on each domain. To this end, DeepAoA+ employs an EM algorithm [26] to achieve this goal. The EM algorithm is a technique used to estimate the parameters of statistical models that involve latent variables. It works by iteratively improving the estimates of the latent variables and maximizing the likelihood of the data given those estimates. In DeepAoA+, we apply this process to optimize the domain estimates and maximize the likelihood of the channel-specific domain likelihood until convergence.

1) Latent Domain Estimation (E-Step): The E-step of DeepAoA+ aims to estimate latent domains in $\mathcal{D}$, which involves the following two sub-steps.

Embeddings Extraction: Following the EM algorithm's procedure, we first initialize a Convolutional Neural Network (CNN)-based encoder $\mathcal{M}_{enco.}$ randomly. The encoder $\mathcal{M}_{enco.}$ can transform data $\{x_1, x_2, \ldots, x_N\} \in \mathcal{X}$ into embeddings. It is worth noting that $\mathcal{M}_{enco.}$ can use any neural network with non-linear representation extracting ability. We use CNN [9] as the encoder in this paper.

Embeddings Clustering: We use the K-means method to cluster the extracted embeddings into K domains with similar distributions. The number of domains K in the data is a hidden feature of the dataset itself. If K is too small, it is impossible to divide all the data into several independent and identically distributed domains; if K is too large, the model performance will degrade due to insufficient data and bring a higher computation burden. Therefore, we utilize Hill Climbing [27] to find a suitable K to balance performance and computation resources.

2) Domain Likelihood Maximization (M-Step): In the M-step, we evaluate the clustering results and optimize the parameters of $\mathcal{M}_{enco.}$ using joint gradient descent based on the evaluation results. Specifically, we use two assessment metrics for joint optimization to maximize the inter-class differences in the encoding results and improve the performance of the AoA estimation task.

Domain Quality Assessment: To obtain the task-oriented domain quality loss $\mathcal{L}^Q$, we conduct trial tests. We further partition channel-specific domains $\{\Gamma_1, \Gamma_2, \ldots, \Gamma_K\}$ into a training set and a validation set for downstream task training and validation. The validation loss is then referred to as the domain quality loss $\mathcal{L}^Q$. This allows us to directly assess the impact of CSI-based data partitioning in terms of the AoA estimation task. In order to maintain the computability of the loss calculation while reducing the complexity of the experimentation, we use the extracted embeddings rather than the raw data for the experimentation. For each partitioned domain in $\Gamma_i \subset \mathcal{D}$, we first split the embeddings belonging to Γ_i into a training set and a validation set in a 7:3 ratio. Specifically, we treat AoA estimation as a classification problem, where the angle labels and the model outputs are discretized into integers ranging from 0° to 180°. The rationale behind this choice is that a regression problem is more susceptible to interference from multipath effects. For example, signals with two incident angles, 10° and 120°, may lead to a larger loss when the regression model yields an inaccurate inference. Then, we train the model on the training set and calculate the cross-entropy loss $\mathcal{L}_i^Q, i \in [1, K]$ on each validation set $\Gamma_i, i \in [1, K]$ and sum them to obtain $\mathcal{L}^Q$ as

$$\mathcal{L}^Q = \sum_{i=1}^{K} \mathcal{L}_i^Q, \tag{5}$$

where $\mathcal{L}_i^Q$ is the domain quality loss on domain Γ_i and K is the number of estimated domain clusters.

Domain Disparity Assessment: The evaluation of the clustering results aims to increase the disparity between the representations of different domains by contrastive learning so that the data can be better distinguished into different domains. The ProtoNCE loss [28] is used to evaluate the clustering results, which involves augmenting sequences with an augmentation function, denoted as $\mathcal{A}$, to create positive views for a data sample and calculating the mutual information between a data sample and its corresponding domain. We use the Gaussian noise function as the augmentation function. By the way, there is a lot of ongoing work studied to design a better augmentation function but this topic is out of the scope of this paper. Formally, the domain disparity loss is computed as

$$\mathcal{L}^D = \mathcal{L}_{NCE}\left(e_u, e_{c_u}\right) + \mathcal{L}_{NCE}\left(\mathcal{A}(e_u), e_{c_u}\right), \tag{6}$$

where e_u and e_{c_u} represent the vector representations of data sample x_u and the cluster center of its corresponding cluster Γ_u, respectively. $\mathcal{L}_{NCE}$ denotes the ProtoNCE loss as introduced in [29]:

$$\mathcal{L}_{NCE}\left(e_u, e_{c_u}\right) = -\log \frac{\exp\left(e_{c_u} \cdot e_u\right)}{\sum_{v=1}^{N_B} \mathbb{I}_{v \notin \Gamma_u^e} \exp\left(e_{c_u} \cdot e_v\right)}, \tag{7}$$

where Γ_u^e represents the vector representation of Γ_u in the hidden feature space $\mathcal{E}$, and N_B is the batch size.

Stochastic-gradient-descent Optimization: Given the domain quality loss $\mathcal{L}^Q$ and the domain disparity loss $\mathcal{L}^D$, the encoder $\mathcal{M}_{enco.}$ can be optimized using SGD optimization method. Formally, the joint optimization of the encoder parameter, denoted as $W_{enco.}$, using gradient descent is as

$$W_{enco.} = W_{enco.} - \eta \cdot \nabla_{W_{enco.}} \left(\mathcal{L}^Q + \alpha \cdot \mathcal{L}^D\right), \tag{8}$$

where $\eta > 0$ denotes the learning rate for encoder training, and α is a hyperparameter to adjust the weight of trial loss.

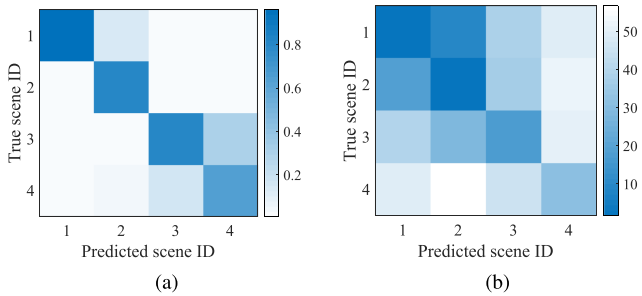


Fig. 8. (a) The accuracy of the selector is limited with an accuracy of 86.7%, resulting in inevitable misclassification; (b) Wrong domain selection may cause catastrophic performance degradation on non-naive test samples, e.g., the degrading from blue (10°) to white (50°) in scene 4. (a) Confusion matrix of domain selection with scene ID. (b) Heatmap of estimation errors on test domains with prior scenes.

D. Domain Model Training

Given a set of estimated domains $\{\Gamma_1, \Gamma_2, \ldots, \Gamma_K\}$, within the training dataset $\mathcal{D}$, the objective of domain model training is to train a collection of domain-specific models, represented as $\{\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_K\}$.

1) Model Pre-Training: An intuitive approach to train domain models is to train $\mathcal{M}_i$ only based on a domain Γ_i. However, this approach poses a significant risk, as it relies on the assumption that the clustering results are error-free. Each domain model $\mathcal{M}_i$ has no experience in handling data from other domains $\Gamma_j, j \neq i$. If the clustering results are biased, the results can be disastrous. For example, we partitioned four domains according to human-labeled scenes and trained DNN models on each domain. A Multi-Layer Perceptron (MLP)-based selector is trained to feed samples into the predicted domain for inference. As shown in Fig. 8(a), some data may inevitably be processed by the wrong domain. Fig. 8(b) demonstrates that the inference results will be highly inaccurate if a domain model handles misclassified samples. To mitigate this risk, we adopt a pre-training approach, where a base model $\mathcal{M}_0$ is first trained on all data with a small number of epochs, thereby gaining a global perspective of the data. Thereafter, even if the data is processed by the wrong domain model, the results will still be relatively reliable.

2) Domain-Aware Model Fine-Tuning: At this stage, we copy and fine-tune the base model $\mathcal{M}_0$ with parameters W_0 as a training starting point for $\mathcal{M}_i$ in the domain Γ_i, $i \in [1, K]$. $\mathcal{M}_i$ is fine-tuned by further training it on Γ_i by minimizing the cross-entropy loss between estimated AoA and ground truth with respect to W_i. The obtained domain-aware models are able to achieve good performance on all test samples due to the global information inherited from the base model $\mathcal{M}_0$.

VI. Evaluation

A. Methodology

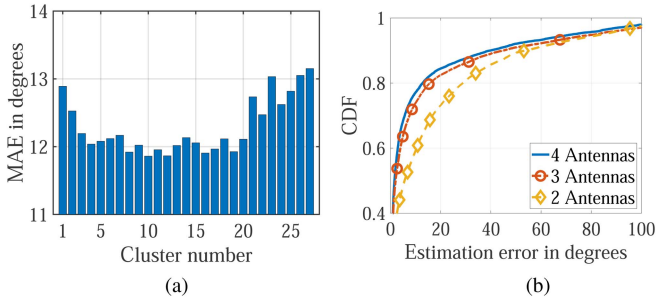
We use $\mathcal{T}_{CSI}$, a real-world V2V CSI dataset with true AoA labeled with our experimental vehicles, to evaluate DeepAoA+ and other candidate methods. Each sample contains 6×52-dimensional CSI phase differences, 4-dimensional SNRs on each antenna, and 1-dimensional ground truth labeled by LiDAR. The dataset is divided into a training set, a test set, and a validation set, all mutually exclusive, with a ratio of 6:2:2. The encoder $\mathcal{M}_{enco.}$ comprises three 1D convolutional layers, followed by two fully connected layers with the dimension of encoded embeddings set to 64. The domain-specific models $\{\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_K\}$ contain a three-layer MLP on each domain respectively. The Gaussian noise function is used as the augmentation function for domain disparity loss in (6). The trial learning rate and the SGD learning rate η are set to 1e-3 with the weight α set to 0.2. Thereafter, domain-specific model training has the same settings as domain quality assessment except for a pre-training process. The training process is offline executed on a GPU server with 128 GB RAM and 4 Nvidia GTX 3090 GPUs running Ubuntu 18.04, while the inference is carried out on a PC with an Nvidia T600, an i7-8700 CPU, and 16 GB RAM, running Ubuntu 16.04. We compare DeepAoA+ with the following candidate methods:

- *MUSIC [7]:* is a subspace-based AoA estimation algorithm, which formulates the correlation matrix of signals and searches the AoA through spectral decomposition.
- *SpotFi [15]:* contains a super-resolution AoA estimation component which is an improvement of the MUSIC algorithm. We modify the original SpotFi AoA estimation algorithm for four ULA antennas.
- *DeepAoA-Naive [9]:* trains one single model with all training data in $\mathcal{D}_{train}$. The architecture of DeepAoA-Naive is set to be consistent with the domain-specific models of DeepAoA+.
- *DeepAoA-Scene:* partitions the training set $\mathcal{D}_{train}$ into four domains according to the data-collecting scenario, i.e., the rural region, the parking lot, the surface road, and the highway. Domain-specific models are trained on the four domains while an MLP is utilized to select the corresponding model at the inference stage.
- *DeepAoA-Raw:* performs clustering in the raw data space, partitioning all training samples into different domains. Critical models are trained on these domains, and during inference, testing samples select the closest clustering centers and the corresponding model for prediction.

We consider the absolute error between the estimated AoA and the ground truth obtained by LiDAR as the performance metric.

B. Parameters Configurations

1) Effect of the Number of Domain Clusters K: We first investigate the effect of the number of identified domains by varying K from 1 to 27 on all training data. Fig. 9(a) depicts the estimation error of AoA inference with different cluster numbers on the test set. It can be observed that, on the one hand, as the number of domains increases, the mean absolute error (MAE) of inference decreases due to the non-*i.i.d.* data being partitioned. The minimum value of MAE achieves 11.8° when K is set to 10. On the other hand, as the number of domain partitions becomes larger, the MAE generally increases instead. The reason is that the cluster number K is beyond the number of intrinsic latent domains and more fine-grained division leads to smaller sub-datasets, thus damaging model performance. Given a specific dataset, the optimal number of domain clusters K can be determined.



(a) (b)

Fig. 9. (a) The MAE in degrees is plotted against different numbers of estimated latent domains K. It can be seen that as K increases, the MAE initially decreases and finally increases. (b) As the antenna number increases, the performance of DeepAoA+ also improves. (a) Cluster number K. (b) The number of antennas.

2) Effect of Antenna Array Configuration: To investigate the effect of the number of antennas, we vary the antenna number m from 2 to 4 over all scenarios, where the distance between two adjacent antennas is set to half of the wavelength $\lambda = 5.093$ cm. The domain models of DeepAoA+ are trained and tested over all scenarios with different feature columns. Specifically, the dimensions of CSI phase signals are 1×52, 3×52, and 6×52 with 2, 3, and 4 antennas, respectively. Fig. 9(b) plots the CDFs of the AoA estimation errors using DeepAoA+ with different antenna configurations. It can be observed that as m increases, DeepAoA+ can provide more accurate online AoA inference results. However, it should be noted that the utilization of more antennas also entails a higher cost in terms of hardware deployment. Therefore, the strategy is to determine the minimal number of antennas that can meet the Quality of Service (QoS) requirements of the application scenarios.

3) Effect of Model Pre-Training: Each individual domain-specific model should have a basic capability to deal with test samples belonging to other domains, referred to as *non-native* test samples, while having the best capability to handle samples within its native domain, referred to as *native* test samples. In this experiment, we examine the effect of model pre-training when training domain-specific models $\{\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_K\}$. We train domain-specific models with/without model pre-training and compare their inference results. Fig. 10 shows the MAE of AoA estimation achieved by DeepAoA+ with and without model pre-training tested on native and non-native domains, respectively. From Fig. 10(a), it can be seen that the MAE decreases on the corresponding domains, albeit to a limited extent, due to the use of more data for pre-training. From Fig. 10(b), it can

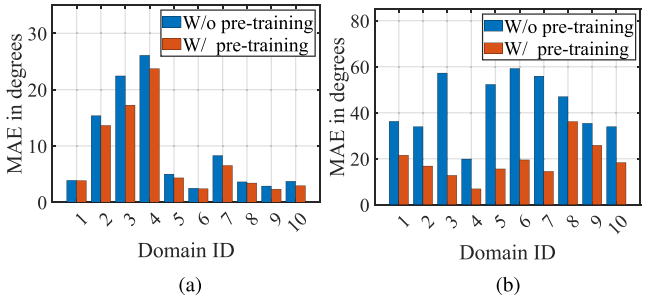


Fig. 10. The utilization of pre-training results in a lower MAE and the performance improvement is more pronounced on non-native test samples due to the comprehensive view obtained by pre-training. (a) Native test samples. (b) Non-native test samples.

be observed that the improvement from pre-training is quite significant for non-native test samples, as the domain-specific models are strengthened with a global view in the pre-training process. The results demonstrate the necessity of using model pre-training.

C. Performance Comparison

In the rest experiments, the cluster number K and the antenna number m are set to 10 and 4 respectively. We investigate the impact of SNR conditions and vehicular scenarios among DeepAoA+ and other candidate methods. In this experiment, the encoder and domain-aware models of DeepAoA+ are trained and validated on the training set and the validation set over all scenarios. Then we conduct AoA inference on the test set and evaluate the impact of factors in the inference results.

1) Impact of SNR Conditions: We first investigate the impact of SNR conditions over all scenarios for DeepAoA+. The test samples are divided into four categories in terms of the SNR values, including $SNR >$20 dB, 15 dB $< SNR <=$ 20 dB, 10 dB $< SNR <=$ 15 dB, and $SNR <=$ 10 dB . Fig. 11(a) illustrates the performance of DeepAoA+ in terms of estimation error over different SNR values. The SNR values are obtained from the demodulated antenna signals. It can be observed that as the SNR increases, DeepAoA+ achieves smaller estimation errors due to better signal quality. At an SNR greater than 20dB, the estimation error at an 80% confidence interval is 10°. However, as the SNR drops below 10, the estimation error at an 80% confidence interval reaches nearly 50°. In practical deployment, SNR values can serve as an indicator of signal quality. As a result, packets that meet a desired QoS can be selected for inference depending on the SNR. Fig. 11(b) also indicates that data samples with similar SNR values are more inclined to be divided into closer clusters as DeepAoA+ iteratively estimates the channel-specific domains. As a result, Fig. 11(c) shows that DeepAoA+ also outperforms all other methods in different SNR values, e.g., DeepAoA+ obtain an MAE of 22° when the SNR of signals is lower than 10 dB while the MAEs of other methods are nearly larger than 30°.

2) Impact of Vehicular Scenarios: We then evaluate the impact of vehicular scenarios with the same parameter settings. We

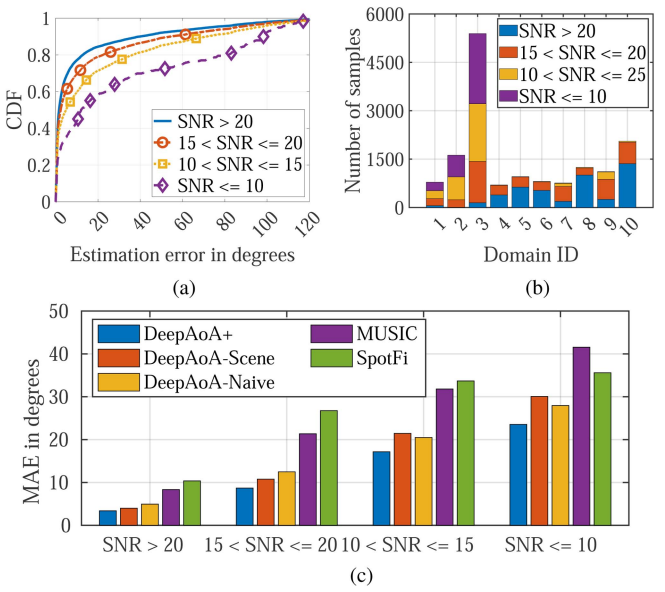


Fig. 11. (a) CDFs of the estimation error over different SNR values in the inference result, where the estimation MAE increases as the SNR values decrease. (b) SNR compositions of each estimated domain, where data samples with similar SNR values are automatically gathered by latent domain estimation. (c) Performance of methods over different SNR values, where DeepAoA+ remains the best in all SNR values compared with other methods. (a) Error of different SNR values. (b) SNR compositions of domains. (c) Performance of methods over different SNR values.

adopt the same four traffic scenes as Section IV, including the rural region, the parking lot, the surface road, and the highway. Fig. 12(a) depicts the performance of DeepAoA+ in terms of estimation error over different scenarios. It can be observed that the performance is particularly good in the rural region, with an 80% confidence interval of AoA estimation error within 3°. This is attributed to the presence of marginal multipath effects and noises in the rural region. In contrast, the worst performance is observed in the highway scenarios due to data collection during rush hours, which results in complex and dynamic multipath scenarios caused by reflections from numerous metallic surfaces of vehicles. Fig. 12(b) demonstrates that during the latent domain estimation, features of different scenes are captured to form a better partition scheme. Fig. 12(c) shows that DeepAoA+ outperforms all other methods in all traffic scenes, demonstrating that DeepAoA+ can well handle the non-*i.i.d.* problem due to the channel-specific domain estimation.

3) Overall Performance: Fig. 13 plots the CDFs of AoA estimation error using different methods over all scenarios and data samples. It can be seen that DeepAoA+ can achieve the lowest estimation error of less than 12° with an 80% confidence interval. In contrast, for a confidence interval of 80%, the errors of DeepAoA-Scene, DeepAoA-Raw, DeepAoA-Naive, MUSIC, and SpotFi are 21°, 20°, 23°, 40°, and 45°, respectively. The poor accuracies of traditional methods, i.e., MUSIC algorithm, and SpotFi indicate the difficulty of handling dynamic traffic and multipath effects. DeepAoA-Scene is limited in representing the complexity of vehicular environments through

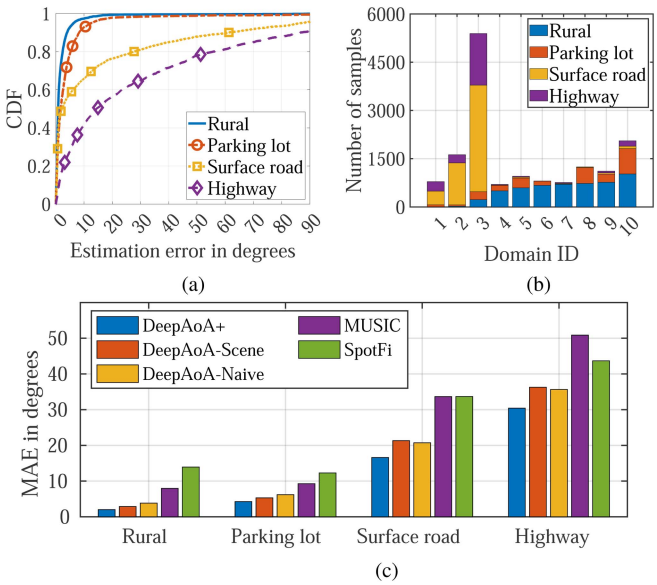


Fig. 12. (a) CDFs of the estimation error over four prior scenarios in the inference results, where data from the rural region and the parking lot obtain marginal errors while the other dynamic scenes lead to poor accuracy. (b) Scene compositions of each estimated domain, where each domain has different compositions of the prior scenarios, indicating that the domain estimation can recognize and classify the complex traffic scenarios. (c) Performance of methods over different traffic scenarios, where DeepAoA+ outperforms other methods in all four traffic scenarios. (a) Error of different scenarios. (b) Scene compositions of domains. (c) Performance of methods over different traffic scenarios.

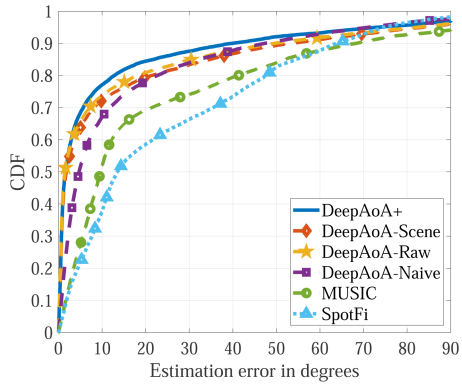


Fig. 13. Overall performance of DeepAoA+ and four candidate methods using all data samples, where DeepAoA+ outperforms other baselines and obtains a minimum MAE of 11.8° at an 80% confidence interval.

its pre-defined domain partitioning and selection. On the other hand, DeepAoA-Raw performs clustering in the raw data space, which does not inherently correlate with the AoA. Therefore, the results demonstrate the efficacy of utilizing domain-adaptive models for the direction-finding problem in dynamic vehicular environments.

D. Response Time

We measure the response time of one single AoA estimation required for DeepAoA+, DeepAoA-Scene, DeepAoA-Naive, MUSIC, and SpotFi. For each scheme, we conduct AoA estimation 15,000 times and take the average. The results are shown in Table I. On average, DeepAoA+ takes about 0.9 ms for each estimation. It is negligible compared to the 100 ms transmission interval as required in the WAVE protocols. In comparison, for each estimation, DeepAoA-Scene, DeepAoA, MUSIC, and SpotFi take about 0.7 ms, 0.6 ms, 2 ms, and 1065.9 ms, respectively. SpotFi is much slower and cannot satisfy the rigid requirement posed by the vehicular direction-finding problem.

TABLE I
RESPONSE TIME OF AOA ESTIMATION SCHEMES

Scheme	Response Time (ms)
DeepAoA+	0.9
DeepAoA-Scene	0.7
DeepAoA-Naive	0.6
MUSIC	2.0
SpotFi	1065.9

VII. DISCUSSION

A. Simultaneous Transmitters

We assume that only one transmitter is transmitting signals at a given time for DeepAoA+. This assumption is based on the consideration of the broadcast characteristics of practical vehicular communications. Both contention-based medium access control (MAC) protocols, e.g., 802.11p [25], and contention-free MAC protocols, e.g., time division multiple access (TDMA)-based MAC protocols [30] [31] are designed to optimize channel resource access control by ensuring that only one transmitter is active at any given time. However, if multiple transmitters are transmitting simultaneously, resulting in packet collisions, it is also beneficial to estimate the relative positions of multiple nodes to optimize such medium access control protocols. Therefore, we will continue to investigate the potential of DeepAoA+ for the case of multiple transmitters transmitting simultaneously in future research.

B. Utilization of Temporal Information

Considering that the location of the vehicle is continuous in the real world, temporal location information can be used to optimize the predicted results. For instance, it is possible to exclude predictions that are clearly out of the vehicle's movement trajectory based on the time series. In this paper, to focus on the AoA estimation scheme, we do not use the vehicle's time-series information for optimization. In addition, because our goal is to use only the "by-products" of in-vehicle communication for inter-vehicle positioning, we also have not used results from other sensors for fusion yet. In future work and commercial deployment, these aspects should be optimized.

VIII. CONCLUSION

In this paper, we have proposed a relative direction-finding scheme, called DeepAoA+, which leverages normal V2V communication packets to generate "by-products" AoA estimation between vehicles. DeepAoA+ involves a novel latent domain partition algorithm to deal with the generalized non-*i.i.d.* dataset

issue when applying deep learning methods. Except for V2V radios, DeepAoA+ does not require relatively expensive dedicated sensors and is lightweight for online AoA estimation. The results of real-world experiments demonstrate that the DeepAoA+ can achieve high accuracy and good reliability in the dynamic vehicular environment with low response time.

REFERENCES

[1] J. Yin et al., "Performance evaluation of safety applications over DSRC vehicular ad hoc networks," in *Proc. ACM MobiCom Workshop VANETs*, 2004, pp. 1–9.

[2] R. W. Wolcott and R. M. Eustice, "Robust LIDAR localization using multiresolution Gaussian mixture maps for autonomous driving," *Int. J. Robot. Res.*, vol. 36, no. 3, pp. 292–319, 2017.

[3] G. Wan et al., "Robust and precise vehicle localization based on multi-sensor fusion in diverse city scenes," in *Proc. IEEE IEEE Int. Conf. Robot. Automat.*, 2018, pp. 4670–4677.

[4] V. John, Z. Liu, S. Mita, and Y. Xu, "Stereo vision-based vehicle localization in point cloud maps using multiswarm particle swarm optimization," *Signal, Imag. Video Process.*, vol. 13, no. 4, pp. 805–812, 2019.

[5] H. Zhu, S. Chang, L. Lu, and W. Zhang, "RUPS: Fixing relative distances among urban vehicles with context-aware trajectories," in *Proc. IEEE Int. Parallel Distrib. Process. Symp.*, 2016, pp. 123–131.

[6] J.-H. Lee and J.-M. Woo, "Interferometer direction-finding system with improved DF accuracy using two different array configurations," *IEEE Antennas Wireless Propag. Lett.*, vol. 14, pp. 719–722, 2014.

[7] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Trans. Antennas Propag.*, vol. 34, no. 3, pp. 276–280, Mar. 1986.

[8] D. Li, T. Bansal, Z. Lu, and P. Sinha, "Marvel: Multiple antenna based relative vehicle localizer," in *Proc. ACM MobiCom*, 2012, pp. 245–256.

[9] Y. Zhu, Y. Cai, H. Zhu, and S. Chang, "DeepAoA: Online vehicular direction finding based on a deep learning method," in *Proc. IEEE 25th Int. Conf. Parallel Distrib. Syst.*, 2019, pp. 782–789.

[10] Y. Zhuo, H. Zhu, H. Xue, and S. Chang, "Perceiving accurate CSI phases with commodity WiFi devices," in *Proc. IEEE INFOCOM*, 2017, pp. 1–9.

[11] A. Tzur, O. Amrani, and A. Wool, "Direction finding of rogue Wi-Fi access points using an off-the-shelf MIMO-OFDM receiver," *Phys. Commun.*, vol. 17, pp. 149–164, 2015.

[12] D. Niculescu and B. Nath, "VOR base stations for indoor 802.11 positioning," in *Proc. 10th Annu. Int. Conf. Mobile Comput. Netw.*, 2004, pp. 58–69.

[13] J. Wang, Y. Chen, X. Fu, J. Wang, W. Yu, and N. Zhang, "3DLoc: Three dimensional wireless localization toolkit," in *Proc. IEEE Int. Conf. Distrib. Comput. Syst.*, 2010, pp. 30–39.

[14] J. Xiong and K. Jamieson, "ArrayTrack: A fine-grained indoor location system," in *Proc. 10th USENIX Symp. Netw. Syst. Des. Implementation*, 2013, pp. 71–84.

[15] M. Kotaru, K. R. Joshi, D. Bharadia, and S. Katti, "SpotFi: Decimeter level localization using WiFi," in *Proc. ACM Conf. Special Int. Group Data Commun.*, 2015, pp. 269–282.

[16] F. Wang, X. Wang, and T. Li, "Semi-supervised multi-task learning with task regularizations," in *Proc. IEEE Int. Conf. Data Mining*, 2009, pp. 562–568.

[17] T. Kato et al., "Multi-task learning via conic programming," in *Proc. Neural Inf. Process. Syst.*, 2008, pp. 1–8.

[18] L. Han et al., "Encoding tree sparsity in multi-task learning: A probabilistic framework," in *Proc. 28th AAAI Conf. Artif. Intell.*, 2014, pp. 1854–1860.

[19] X. Zhang, X. Zhang, and H. Liu, "Self-adapted multi-task clustering," in *Proc. Int. Joint Conf. Artif. Intell.*, 2016, pp. 2357–2363.

[20] A.-A. Liu, Y.-T. Su, W.-Z. Nie, and M. Kankanhalli, "Hierarchical clustering multi-task learning for joint human action grouping and recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 1, pp. 102–114, Jan. 2017.

[21] Z. Zheng, Y. Wang, Q. Dai, H. Zheng, and D. Wang, "Metadata-driven task relation discovery for multi-task learning," in *Proc. Int. Joint Conf. Artif. Intell.*, 2019, pp. 4426–4432.

[22] B. Daniel et al., "Tax fraud detection for under-reporting declarations using an unsupervised machine learning approach," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 215–222.

[23] Z. Zheng, Y. Li, H. Song, L. Wang, and F. Xia, "Towards edge-cloud collaborative machine learning: A quality-aware task partition framework," in *Proc. 31st ACM Int. Conf. Inf. Knowl. Manage.*, 2022, pp. 3705–3714.

[24] Z. Zheng, P. Luo, Y. Li, S. Luo, J. Jian, and Z. Huang, "Towards lifelong thermal comfort prediction with KubeEdge-sedna: Online multi-task learning with metaknowledge base," in *Proc. 13th ACM Int. Conf. Future Energy Syst.*, 2022, pp. 263–276.

[25] J. A. Fernandez, K. C. Borries, L. Cheng, B. V. K. V. Kumar, D. D. Stancil, and F. Bai, "Performance of the 802.11p physical layer in vehicle-to-vehicle environments," *IEEE Trans. Veh. Technol.*, vol. 61, no. 1, pp. 3–14, Jan. 2012.

[26] T. K. Moon, "The expectation-maximization algorithm," *IEEE Signal Process. Mag.*, vol. 13, no. 6, pp. 47–60, Nov. 1996.

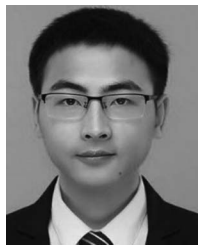
[27] S. J. Russell, *Artificial Intelligence a Modern Approach*. London, U.K.: Pearson Educ., 2010.

[28] J. Li, P. Zhou, C. Xiong, and S. Hoi, "Prototypical contrastive learning of unsupervised representations," in *Proc. Int. Conf. Learn. Representations*, 2020, pp. 1–16.

[29] Y. Chen, Z. Liu, J. Li, J. McAuley, and C. Xiong, "Intent contrastive learning for sequential recommendation," in *Proc. ACM Web Conf.*, 2022, pp. 2172–2182.

[30] H. A. Omar, W. Zhuang, and L. Li, "VeMAC: A TDMA-Based MAC protocol for reliable broadcast in VANETs," *IEEE Trans. Mob. Comput.*, vol. 12, no. 9, pp. 1724–1736, Sep. 2013.

[31] F. Lyu et al., "MoMAC: Mobility-aware and collision-avoidance MAC for safety applications in VANETs," *IEEE Trans. Veh. Technol.*, vol. 67, no. 11, pp. 10590–10602, Nov. 2018.



Facheng Hu (Graduate Student Member, IEEE) received the B.S. degree in 2017 and the master's degree from Shanghai Jiao Tong University, Shanghai, China, in 2023, under the supervision of Prof. Hongzi Zhu. He is currently working toward the Ph.D. degree with the Michigan-SJTU Joint Institute, Shanghai Jiao Tong University, under the supervision of Prof. Xudong Wang. His research interests include vehicular ad hoc networks and mobile computing.

Yunxiang Cai received the Ph.D. degree from the Department of Computer Science and Engineering from Shanghai Jiao Tong University, Shanghai, China, in September 2023, under the supervision of Prof. Hongzi Zhu. He is currently an Algorithm Engineer with Huawei Technologies Company, Ltd. His research interests include vehicular ad hoc networks, AI-empowered IoTs, and mobile computing.

Hongzi Zhu (Senior Member, IEEE) received the Ph.D. degree in computer science from Shanghai Jiao Tong University, Shanghai, China, in 2009. In 2009, he was a Postdoctoral Fellow with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong, and with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada, in 2010. He is currently a Professor with the Department of Computer Science and Engineering, Shanghai Jiao Tong University. His research interests include mobile sensing, mobile computing, and Internet of Things. He is also an Associate Editor for IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY and IEEE INTERNET OF THINGS JOURNAL. He was the recipient the Best Paper Award from IEEE Globecom 2016.

Shan Chang (Member, IEEE) received the Ph.D. degree in computer software and theory from Xian Jiaotong University, Xi'an, China, in 2013. From 2009 to 2010, she was a Visiting Scholar with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong. From 2010 to 2011, she was also a Visiting Scholar with BBCR Research Lab, University of Waterloo, Waterloo, ON, Canada. She is currently an Associate Professor with the Department of Computer Science and Technology, Donghua University, Shanghai. Her research interests include security and privacy in mobile networks and sensor networks.

Xudong Wang (Fellow, IEEE) received the Ph.D. degree in electrical and computer engineering from the Georgia Institute of Technology, Atlanta, GA, USA, in 2003. He was a Senior Research Engineer, Senior Network Architect, and Research and Development Manager with several companies. He is currently a John Wu and Jane Sun Chair Professor of engineering with UM-SJTU Joint Institute, Shanghai Jiao Tong University, Shanghai, China. He is also an Affiliate Professor with the Department of Electrical and Computer Engineering, University of Washington, Seattle, WA, USA. He holds a number of patents on wireless networking technologies and most of his inventions have been successfully transferred to products. His research interests include wireless communication networks, joint communications and sensing, edge computing, and federated learning. He was a voting member of IEEE 802.11 and 802.15 Standard Committees. He was the General Chair of 2017 IEEE 5G Summit in Shanghai and the TPC Co-Chair of the 32nd International Conference on Information Networking. He was an Editor of IEEE TRANSACTIONS ON MOBILE COMPUTING and IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, *Ad Hoc Networks* (Elsevier) and *China Communications*. He was also a Guest Editor of several international journals.

Minyi Guo (Fellow, IEEE) received the B.Sc. and M.E. degrees in computer science from Nanjing University, Nanjing, China, and the Ph.D. degree in computer science from the University of Tsukuba, Tsukuba, Japan. He was a Professor with the School of Computer Science and Engineering, University of Aizu, Aizuwakamatsu, Japan. He is currently the Zhiyuan Chair Professor and an ACM Distinguished Member. He is also the Department of Computer Science and Engineering Director of the Embedded and Pervasive Computing Center with Shanghai Jiao Tong University, Shanghai, China. He has more than 400 publications in major journals and international conferences in his research interests which include parallel/distributed computing, compiler optimizations, Big Data, and cloud computing. He was the recipient of National Science Fund for Distinguished Young Scholars from NSFC in 2007 and the 7 Best/Highlight Paper awards from International Conferences Including ALSPOS 2017 and ICCD 2018. He is also the Editor-in-Chief of IEEE TRANSACTIONS ON SUSTAINABLE COMPUTING, and is also on the Editorial Board of IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, IEEE TRANSACTIONS ON CLOUD COMPUTING, and *Journal of Parallel and Distributed Computing*.