

Article

Generative Adversarial Network-Based Joint Mapping and Localization for Millimeter-Wave Communication Systems

Zexu Zhao, Zhigang Chen *  and Lu Chen

School of Information and Communications Engineering, Xi'an Jiaotong University, Xi'an 710049, China

* Correspondence: zgchen@mail.xjtu.edu.cn

Abstract

In this paper, we propose a novel generative adversarial network (GAN)-based joint localization and mapping (JLAM) method using angle difference of arrival (ADOA) measurements for millimeter-wave (mmWave) communication systems. The proposed method adopts a deep auto-encoder neural network as the discriminator of the GAN and models the generator as an explicit geometric ADOA function of the access point (AP) positions and the mobile terminal (MT) position, rather than as a conventional black-box neural network. By exploiting the two-dimensional distribution characteristics of high-dimensional ADOA vectors collected at a large number of random and unknown MT positions, the proposed method learns the ADOA data distribution and transforms it into the AP geometric topology. Then, the MT positions and the indoor map are estimated based on the recovered physical and virtual AP topology. The simulation results show that, under the representative setting with $N = 2000$ measured ADOA vectors and $\sigma = 2^\circ$ AOA measurement noise, the proposed method achieves an average localization error of about 0.25 m, compared with about 0.60 m for the JADE algorithm, corresponding to an error reduction of approximately 58%. The proposed method also provides more accurate room boundary estimation than JADE, confirming its effectiveness for mmWave JLAM.

Keywords: angle difference of arrival (ADOA); deep neural network; generative adversarial network (GAN); joint localization and mapping (JLAM); millimeter-wave communication systems

1. Introduction

Indoor wireless localization has become an important research topic in recent years, driven by its broad potential applications in areas such as smart construction, health-care services, pedestrian navigation, and indoor guidance. Existing indoor positioning techniques are generally categorized into two representative groups: geometry-based localization methods and fingerprint-based localization methods [1]. The geometrical location methods usually employ the multipath parameters, such as time-of-arrival (TOA) or angle-of-arrival (AOA), to realize positioning through geometric principles [2,3]. Due to the serious non-line-of-sight (NLOS) multipath effects and the limited system bandwidth, such geometric location methods suffer from limited estimation accuracy of multipath parameters [2–4]. In contrast, fingerprinting-based localization usually determines the target position by matching real-time RSS or CSI measurements with an offline radio map constructed in advance. Typical examples include nearest-neighbor (NN) and K-nearest-neighbor (KNN)-based methods [1,5]. Since the measured fingerprints implicitly contain the propagation characteristics of both LOS and NLOS components, such methods are able



Academic Editor: Gia Khanh Tran

Received: 10 April 2026

Revised: 17 June 2026

Accepted: 21 June 2026

Published: 7 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

to exploit location-dependent information embedded in multipath signals. However, the multipath effects make the received signal strength (RSS) or channel impulse response (CIR) fingerprints spatially unstable, leading to the unstable localization performance. Even worse, the fingerprinting positioning methods require a prohibitively huge workload for constructing an offline fingerprint database [5]. Hence, the conventional positioning methods, including the geometric positioning methods and fingerprinting methods, are subject to multipath effects.

Millimeter-wave (mmWave) communication has been widely regarded as one of the enabling technologies for supporting multi-Gbit/s data transmission in 5G and future wireless networks [6,7]. Owing to the large available bandwidth and the compact antenna size associated with millimeter wavelengths, mmWave systems can provide fine resolution in both the delay and angular domains [8,9]. In addition, mmWave signals generally exhibit quasi-optical propagation behavior and sparse multipath characteristics, mainly because of their weak scattering capability in typical indoor environments [10,11]. To sum up, the mmWave communication systems can provide great potential for achieving high positioning accuracy due to the multipath sparsity and high temporal and spatial resolution of mmWave channels, which is not the case for sub-6GHz microwave propagation and means more information can be extracted from the multipath parameters [12]. As a result, lots of mmWave positioning methods through processing RF multipath measurements have been proposed to enhance the positioning accuracy [13].

However, the existing multipath measurement processing-based mmWave positioning methods are severely subject to extra requirements for deterministic or statistical information beyond mmWave communications for overcoming the problem of unknown virtual AP (access point) positions, which are the reflection mirrors of the physical AP through indoor plane surfaces. On the one hand, fingerprint-based mmW positioning methods can make full use of multipath information without estimating multipath parameters and thus can achieve better performance [6,14–17], whereas this category of methods requires laborious and complex offline database work. On the other hand, mmW multipath parameter-based geometrical positioning methods can improve the positioning accuracy by using the extra geometric relationship between indirect path measurements and the mobile terminal (MT) position, while these methods still suffer from the rigorous assumptions or requirements on the indirect path parameters due to the unknown complex environments. For example, refs. [9,18–20] exploit both the direct path angle and indirect path angles to realize location by strictly assuming information of indirect path reflectors or the known floorplan, and [21,22] estimate the MT position by geometrically resolving the information redundancy provided by a parameter triplet of each path, TOA, AOA and AOD (angle-of-departure), which require not only extra-strict synchronization between mmWave transmitters and receivers but also the known bearings of antenna arrays at the AP and/or the MT.

As a viable solution to the problem of virtual APs' unknown positions for practical mmW communication systems, a few zero-initial-information-based mmW joint localization and mapping (JLAM) methods have been proposed to estimate AP locations and MT locations jointly, these methods usually gather information of multipath parameters at a large number of positions and then employ the information redundancy of such multipath measurements to localize the MT as well as the APs [9,19]. Since the angle difference of arrival (ADOA) can be obtained from the beam-training results already available in practical mmWave devices, several ADOA-based zero-initial-information JLAM methods have been developed. For example, the Joint Anchor and Device location Estimation (JADE) algorithm in [9] reformulates the originally NP-hard joint estimation problem of AP coordinates and random MT positions into an iterative procedure consisting of two alternating

least-squares (LS) estimations. One LS step updates the AP positions, while the other estimates the MT positions. However, this simplification is achieved by relaxing the original geometric consistency among the AP topology, MT locations, and ADOA observations, which inevitably discards part of the information carried by the ADOA measurements. In addition, the communication-driven localization and mapping (CLAM) method in [19] first recovers the physical and virtual AP positions by solving a minimum set of equations that relate the ADOA observations to the AP shape, where these equations are obtained offline through automatic symbolic manipulation. The MT position is then inferred using a robust ADOA localization procedure. Nevertheless, in order to keep the computational burden manageable, CLAM does not exploit all the available interdependencies among the ADOA measurements, which may also lead to a degradation in estimation performance. Therefore, the existing zero-initial-information-based mmW SLAM methods have to compromise between low complexity and high information utilization since a large number of ADOA measurements are gathered to accumulate enough information.

In this paper, a novel generative adversarial network (GAN)-based mmWave JLAM algorithm has been proposed. Without requiring any extra information beyond the capability of MMW communication devices, the proposed algorithm first employs a deep neural network-based discriminator in the GAN to learn the 2-dimensional statistical characteristics of crowd-sourcing or online multipath ADOA vectors measured at a large number of random and unknown positions, which is completely dependent on the AP geometric topology. Further, it can transform such extracted 2-dimensional statistical characteristics into the AP geometric topology or AP positions through optimizing the generator in the GAN, since the generator is explicitly modeled as the ADOA function of the AP topology and the MT position, rather than a deep neural network. Based on the estimated AP positions, the MT position corresponding to each ADOA vector can be estimated through classical triangulation location schemes and the room boundary can be geometrically calculated as the reflection points, i.e., the environment map is constructed. Considering the GAN can take more advantage of the geometrical information of the crowd-sourcing and/or online ADOA data, our GAN-based SLAM method can achieve high information utilization, thus can outperform the existing mmW JLAM methods. This work extends our previous research on mmWave SLAM for joint positioning and mapping [23]. Our previous study introduced an Expectation Maximization (EM)-based SLAM method for estimating AP and MT positions. However, this EM based approach is highly sensitive to initial parameter settings and relies on the assumption of Gaussian-distributed data. In contrast, this paper introduces a novel GAN-based JLAM algorithm, which learns the 2D distribution of high-dimensional ADOA vectors through adversarial training, enabling improved robustness and generalization without strong distributional assumptions.

Different from existing GAN-based localization methods that mainly use GANs for fingerprint generation, data augmentation, or semi-supervised location regression [24], the proposed method constructs the generator as an explicit geometric ADOA function. Therefore, the GAN is used to estimate the physical and virtual AP topology from unlabeled ADOA vectors, rather than to directly generate fingerprints or regress MT positions.

The rest of this paper is organized as follows. In Section 2, the virtual AP-based system model is described and the multipath ADOA vector is defined. We develop a novel ADOA and GAN-based SLAM method, mainly including the AP topology estimation by training the GAN and positioning MTs through classical schemes, and analyze the Cramer–Rao lower bound for positioning the MT in Section 3. The simulation results are presented in Section 4 to evaluate the performance of the proposed method. Finally, conclusions are drawn in Section 5.

2. System Model

Consider a 2-dimensional indoor scenario (shown in Figure 1) in which a single mmWave AP, denoted as AP_1 , is deployed and its location $\mathbf{a}_1$ is unknown. The following system model can be simply extended to a 3-dimensional scenario. Because mmWave signals suffer from severe propagation loss and approximately follow optical-like propagation in air [10], this work considers only the LOS path and the single-bounce reflected paths, following the modeling assumptions in [9,19]. Reflected paths involving two or more bounces are assumed to be negligible, since they usually undergo much stronger attenuation. A first-order NLOS component can be equivalently represented as an LOS component transmitted from a virtual AP, which is the mirror image of the physical AP with respect to the corresponding reflecting surface, such as an indoor wall. Let these virtual APs be denoted by $AP_l (l = 2, 3, \dots, L)$, with unknown coordinates $\mathbf{a}_l = [a_{l,x}, a_{l,y}]^T, (l = 2, 3, \dots, L)$. The complete AP set, including both the physical AP and the virtual APs, is then written as $\mathbb{A} = \{AP_1, AP_2, \dots, AP_L\}$, where L denotes the total number of APs.

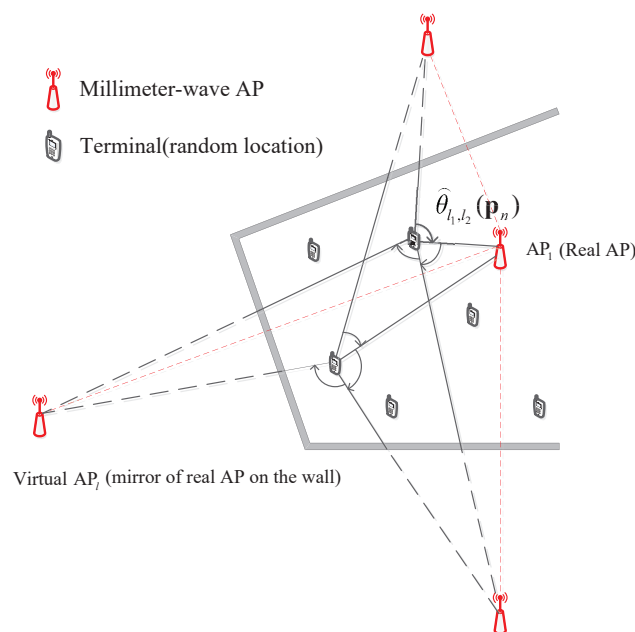


Figure 1. Virtual AP-based mmW system model.

When the MT is located at $p_n (n = 1, 2, \dots, N)$, it can utilize the beam-training results naturally provided by the mmWave communication procedure to extract the AOAs of dominant multipath components. The ADOA measurements are then obtained by calculating the angular differences between the paths associated with different AP pairs, where the APs may correspond to either the physical AP or the virtual APs. In this work, the AOA/ADOA measurement vectors are used as the input of the proposed GAN-based JLAM method and are assumed to have already been obtained from the mmWave communication process, such as beam training or channel estimation. Therefore, this paper focuses on the localization and mapping algorithm based on the collected ADOA measurements, rather than on the detailed antenna-array design or the specific AOA estimation procedure. Denote the AOA of the signal from AP_l to the MT located at $\mathbf{p}_n = [p_{n,x}, p_{n,y}]^T$ as $\theta_l(\mathbf{p}_n)$, thus the ADOA for the AP pair $\{AP_{l_1}, AP_{l_2}\}$ at position $\mathbf{p}_n$ can be expressed as $\check{\theta}_{l_1, l_2}(\mathbf{p}_n) = \theta_{l_1}(\mathbf{p}_n) - \theta_{l_2}(\mathbf{p}_n)$. According to the properties of analytic geometry, the ADOA $\check{\theta}_{l_1, l_2}(\mathbf{p}_n)$ satisfies

$$\cos \check{\theta}_{l_1, l_2}(\mathbf{p}_n) = \frac{(\mathbf{a}_{l_1} - \mathbf{p}_n) \bullet (\mathbf{a}_{l_2} - \mathbf{p}_n)}{|\mathbf{a}_{l_1} - \mathbf{p}_n| \cdot |\mathbf{a}_{l_2} - \mathbf{p}_n|} \quad (1)$$

where $\bullet$ denotes dot-product operation.

Based on the observed ADOAs for different AP pairs, we construct the measured ADOA vector at the position $\mathbf{p}_n$ as $\boldsymbol{\psi}(\mathbf{p}_n) = [\hat{\theta}_{1,2}(\mathbf{p}_n), \hat{\theta}_{2,3}(\mathbf{p}_n), \dots, \hat{\theta}_{L-1,L}(\mathbf{p}_n)]$, where $\{\hat{\theta}_{l-1,l}(\mathbf{p}_n)\}_{l=1}^{L-1}$ are the observed ADOAs. Further, we collect the ADOA observation vectors at a larger number of random and unknown positions and set up a dataset of ADOA vectors as $\mathcal{Q} = \{\boldsymbol{\psi}(\mathbf{p}_n)\}_{n=1}^N$. The preliminary ADOA measurement database is acquired as follows. When an MT appears at a random and unknown position, it performs the standard mmWave beam-training or beam-refinement procedure to identify the dominant LOS/NLOS paths and estimate their AOAs. The ADOA values are then obtained by subtracting the AOAs of different path/AP pairs, and the corresponding ADOA vector is stored as one unlabeled sample in $\mathcal{Q}$. Repeating this process for many random MT positions, either through crowd-sourcing users or online measurements during normal communication, produces the ADOA dataset used for GAN training. Different from RSSI fingerprinting, the proposed ADOA database does not require pre-surveyed reference-point coordinates or location-labeled RSSI fingerprints; it only requires unlabeled ADOA vectors measured at random positions.

As indicated by (1), the constructed ADOA vector depends only on the relative geometry between the MT and the APs, and is therefore not affected by the unknown orientation offset of the MT. In typical mmWave multipath scenarios, the number of dominant paths, or equivalently the number of physical and virtual APs L , is usually greater than three. Consequently, if the AP positions are given, the ADOA vector contains redundant angular constraints for determining a two-dimensional MT position. When ADOA vectors are collected from many different and unknown MT locations, this redundancy provides sufficient geometric information to jointly infer the AP topology and the corresponding MT positions, including both the physical AP and the virtual APs.

Therefore, the problem of mmWave simultaneous localization and mapping for a mmWave communication system in an unknown indoor environment can be described as *simultaneously estimating the geometric topology of APs (including both physical APs and virtual APs) and the MT location based on the measured ADOA data $\mathcal{Q} = \{\boldsymbol{\psi}(\mathbf{p}_n)\}_{n=1}^N$, i.e., a large number of collected ADOA vectors at unknown MT locations.*

3. Algorithm

In this section, a novel generative adversarial network (GAN)-based AP topology estimation method is first proposed. Then, the structure and training method of the discriminator and generator of the GAN is described in detail. Further, based on the estimated AP topology and the measured ADOA vectors, a least-square positioning method for the MT is developed and the environment map is geometrically calculated as the reflection points.

3.1. GAN-Based AP Topology Estimation

For a given indoor environment or a given AP topology, the set of high-dimensional ADOA vectors observed at a large number of random MT locations can be regarded as a two-dimensional manifold inside a space of $L - 1$ dimension, since an $L - 1$ dimensional ADOA vector is mapped from a two-dimensional position coordinate. In other words, the dataset of ADOA vectors possess two-dimensional distribution characteristics. Such two-dimensional distribution characteristics are completely determined by the AP topology. Hence, the estimation of AP topology based on the collected ADOA dataset can be realized through learning the low-dimensional distribution characteristics of the ADOA dataset and transforming the low-dimensional distribution characteristic into the AP shape.

It should be noted that the two-dimensional characteristic mentioned here refers to the intrinsic manifold structure of the ADOA dataset, rather than a strict lossless reduction of all high-dimensional information to two independent variables. For a fixed AP topology, each ADOA vector is generated by a nonlinear geometric mapping from the two-dimensional MT position. Therefore, although the ADOA vector lies in an $L - 1$ -dimensional measurement space, its dominant intrinsic degrees of freedom are mainly determined by the two-dimensional MT coordinate. The higher-order dependencies among different ADOA components are still contained in the high-dimensional ADOA vector and are used by the discriminator during reconstruction and adversarial classification.

As an unsupervised learning-based generative network [25,26], the typical GAN including two neural networks (generator and discriminator) can learn low-dimensional features from a large number of ADOA data samples and then generates data with the same distribution as the original data. By using such unsupervised learning capability of the GAN, we present a GAN-based AP topology estimation method, which jointly learns the low-dimensional distribution characteristics of the unlabeled ADOA vectors and transforms it into the AP shape. The structure of the proposed GAN is shown in Figure 1.

The proposed GAN is composed of two parts: the generator that tries to generate the ADOA vector as real as possible and the discriminator that distinguishes the real ADOA vectors from the generated ones. As in typical GANs, our discriminator is implemented by a deep neural network, which first learns the low-dimensional characteristics from the real but unlabeled high-dimensional ADOA vectors, and then distinguishes the real vector from the generated vector. Different from the common generative models in GANs, our generator is realized explicitly based on the geometric relationship among the AP topology, the random indoor position and the ADOA vector in (1), rather than using a deep neural network. Hence, the generator's parameters are the AP topology parameters. It should be emphasized that the generator in the proposed GAN is not a neural network-based black-box mapping. For a given random MT position, the generator is a deterministic and explicit geometric function derived from the ADOA relationship. The only trainable parameters of this function are the coordinates of the physical AP and virtual APs. Therefore, the adversarial training of the generator is actually a parameter estimation process for the AP coordinates. In other words, "transforming the low-dimensional distribution characteristics into the AP shape" means adjusting the AP-coordinate parameters in the explicit ADOA generator until the generated ADOA vectors have the same distribution as the measured unlabeled ADOA vectors.

As shown in Figure 1, with the input of randomly drawn 2-dimension positions of the MT, the generator generates fake ADOA vectors according to the ADOA function in (1). Further, the discriminator receives both the generated ADOA vector and the real ADOA vector and aims to tell them apart. Since the error signal to the discriminator is provided through the simple ground truth of knowing whether the ADOA vector came from the real measurements or from the generator, the error signal can be used to train both the generator and the discriminator simultaneously, leading the generator and the discriminator toward being able to produce forgeries of better quality and distinguish the real vectors from the generated vectors, respectively. Through the iterative training of the discriminator and the generator, the GAN can create ADOA vectors following the same two-dimensional distribution of the real ADOA vectors, thus the optimal estimation of AP coordinates is obtained.

The training of the discriminator and the generator is respectively developed in detail as follows. Without losing generality, the AP topology is represented as the AP coordinates with two AP coordinates fixed in the following training, since an AP topology has the ambiguity of rotation, translation and scaling.

3.1.1. Training the Discriminator

To learn the two-dimensional feature of high-dimensional measured ADOA vectors, the discriminator in the proposed GAN system is modeled as a deep auto-encoder neural network, which is shown in Figure 2, and the Sigmoid function is adopted as the activation function of the hidden layers and the logistic regression layer.

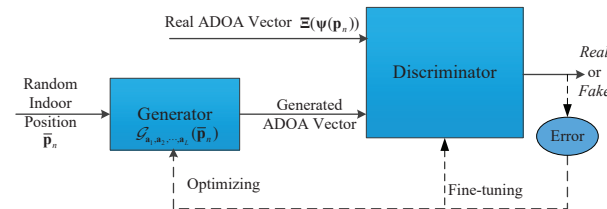


Figure 2. The GAN system for AP topology estimation.

In order to simplify the explicit function of the generator and ensure the data consistency between the generated samples and the training samples for the discriminator, we define the training sample of the discriminator as the following vector based on the ADOA measurements: $\Xi(\psi(\mathbf{p}_n)) = [\cos(\hat{\theta}_{1,2}(\mathbf{p}_n)), \cos(\hat{\theta}_{2,3}(\mathbf{p}_n)), \dots, \cos(\hat{\theta}_{L-1,L}(\mathbf{p}_n))]$.

Auto-Encoder Neural Network

As an unsupervised learning network, the auto-encoder neural network can automatically extract features from unlabeled data and give better feature description than original data. Assume that the auto-encoder neural network includes one input layer and $M - 1$ hidden layers. Denote the parameters of m -th layer neural network as $(\omega^{(m)}, \mathbf{b}^{(m)})$, where $\omega_{i,j}^{(m)}$ denotes the weight between the i -th neuron of the m -th layer and the j -th neuron of the $(m + 1)$ -th layer, and $b_j^{(m)}$ denotes the bias of the j -th neuron of the $(m + 1)$ -th layer. Denote the input of the m -th layer of auto-encoder neural network as $\tilde{\mathbf{X}}^{(m)}(n)$, the input $\tilde{\mathbf{X}}^{(m+1)}(n)$ and the decoding output $\mathbf{Y}^{(m+1)}(n)$ of the $(m + 1)$ -th layer can be expressed as

$$\tilde{\mathbf{X}}^{(m+1)}(n) = \mathcal{F}(\omega^{(m)} \tilde{\mathbf{X}}^{(m)}(n) + \mathbf{b}^{(m)}) \quad (2)$$

and

$$\mathbf{Y}^{(m+1)}(n) = \mathcal{F}((\omega^{(m)})^T \tilde{\mathbf{X}}^{(m+1)}(n) + \mathbf{b}^{(m)}), \quad (3)$$

respectively, wherein $\mathbf{b}^{(m)}$ represents the bias item of the corresponding decoding layer and $\mathcal{F}(\bullet)$ represents the activation function, i.e., the Sigmoid function

$$\mathcal{F}(z) = \frac{1}{1 + \exp(-z)}$$

Given the ADOA dataset, the auto-encoder neural network can be trained without supervision through minimizing the following loss function:

$$\begin{aligned} J(\omega^{(m)}, \mathbf{b}^{(m)}, \mathbf{b}^{(m)}) &= \frac{1}{N} \sum_{n=1}^N \left\| \mathbf{Y}^{(m+1)}(n) - \tilde{\mathbf{X}}^{(m)}(n) \right\|^2 \\ &+ \lambda \sum_i \sum_j \left\| \omega_{ij}^{(m)} \right\|^2 + \gamma \sum_{n=1}^N \left| \tilde{\mathbf{X}}^{(m+1)}(n) \right| \end{aligned} \quad (4)$$

where the first term denotes the average of the square reconstruction error, the second term and the third term are added to prevent the neural network from overfitting and to impose a sparsity constraint on the outputs of the neurons, respectively. Correspondingly, λ and γ denote the weight decay coefficient and the sparsity parameter, respectively.

The sparsity constraint in the auto-encoder is used to reduce overfitting to sample-specific noise in the spatial angle dataset. Let denote the average activation of the i -th hidden neuron, and let denote the desired sparsity level. The sparsity penalty can be written as

$$\Omega_{\text{sparse}} = \sum_j KL(\rho \parallel \hat{\rho}_j) = \sum_j \left[\rho \log \frac{\rho}{\hat{\rho}_j} + (1 - \rho) \log \frac{1 - \rho}{1 - \hat{\rho}_j} \right] \quad (5)$$

This penalty encourages only a small subset of hidden neurons to be activated for each ADOA vector, so that the discriminator learns stable manifold features rather than memorizing noise in the measured spatial angles. Standard dropout is not adopted because ADOA vectors are structured geometric observations, and randomly dropping hidden units may disturb the geometric continuity of the learned ADOA manifold. In contrast, sparsity regularization preserves deterministic feature propagation while controlling the model capacity.

The input of the discriminator includes the real ADOA vectors and the generated vectors, which are respectively labeled as true or false. As the features of the real vectors and the generated vectors are extracted automatically by training the auto-encoder network, the probability of being true or false can be further judged by a follow-up logistic regression network.

Logistic Regression Network

As shown in Figure 3, the output of a multilayer auto-encoder neural network $\tilde{\mathbf{X}}^{(M)}(n)$ is input to a logistic regression network with a weight vector $\tilde{\omega}$. The sigmoid function is also adopted as the activation function in this logistic regression network. Thus the output of logistic regression network can be written as

$$\mathcal{D}_{\tilde{\omega}}(\tilde{\mathbf{X}}^{(M)}(n)) = \frac{1}{1 + \exp(-\tilde{\omega} \tilde{\mathbf{X}}^{(M)}(n))} \quad (6)$$

The output of the logistic regression layer is a continuous probability rather than a hard binary value. For a discriminator input vector, the sigmoid output can be generally written as

$$\mathcal{D}(z) = \sigma(z) = \frac{1}{1 + \exp(-z)} \quad (7)$$

During adversarial training, the measured ADOA vectors and generated ADOA vectors are assigned target labels 1 and 0 respectively, and the loss function is optimized directly using the continuous probability $\mathcal{D}(z)$. Therefore, no hard threshold is applied during back-propagation. The threshold is only used for reporting the discriminator decision or classification accuracy. Specifically, the decision rule is

$$\hat{y} = \begin{cases} 1, & \mathcal{D}(z) \geq 0.5, \\ 0, & \mathcal{D}(z) < 0.5. \end{cases} \quad (8)$$

where the threshold 0.5 is used only for evaluation and does not participate in the gradient-based training process. It should be noted that the sigmoid activation may attenuate the gradient when the discriminator output approaches its asymptotic limits. The derivative of the sigmoid function is $\sigma'(z) = \sigma(z)(1 - \sigma(z))$. Therefore, when the discriminator probability output approaches 0 or 1, the scaling factor becomes close to zero. In this saturated region, the discriminator classifies the input ADOA vectors with very high confidence, and the back-propagated gradient through the logistic regression layer and the auto-encoder layers is weakened. This attenuation may also reduce the gradient transferred

to the geometric generator for updating the AP coordinates. Therefore, stable binary cross-entropy training and balanced discriminator–generator updates are used to avoid excessive discriminator saturation during adversarial training.

Due to

$$\begin{aligned}\tilde{\mathbf{X}}^{(M)}(n) &= \mathcal{F}(\omega^{(M-1)}\tilde{\mathbf{X}}^{(M-1)}(n) + \mathbf{b}^{(M-1)}) \\ &= \mathcal{F}(\omega^{(M-1)}\mathcal{F}(\dots\mathcal{F}(\omega^{(1)}\tilde{\mathbf{X}}^{(1)}(n) + \mathbf{b}^{(1)}) \dots + \mathbf{b}^{(M-2)}) \\ &\quad + \mathbf{b}^{(M-1)}),\end{aligned}\quad (9)$$

the output of logistic regression network can be denoted as $\mathcal{D}_{\bar{\omega}, \omega^{(1)}, \mathbf{b}^{(1)}, \dots, \omega^{(M)}, \mathbf{b}^{(M)}}(\tilde{\mathbf{X}}^{(1)}(n))$.

By labeling the real ADOA vector as 1 (true) and the generated vector as 0 (false), a supervised parameter estimation method can be employed to train the logistic regression network. Specifically, the objective function of the discriminator is constructed as follows:

$$\begin{aligned}J'(\bar{\omega}, \omega^{(1)}, \mathbf{b}^{(1)}, \dots, \omega^{(M)}, \mathbf{b}^{(M)}) &= \frac{1}{N} \sum_{n=1}^N \{ \log \mathcal{D}_{\bar{\omega}, \omega^{(1)}, \mathbf{b}^{(1)}, \dots, \omega^{(M)}, \mathbf{b}^{(M)}}(\Xi(\psi(\mathbf{p}_n))) \\ &\quad + \log(1 - \mathcal{D}_{\bar{\omega}, \omega^{(1)}, \mathbf{b}^{(1)}, \dots, \omega^{(M)}, \mathbf{b}^{(M)}}(\mathcal{G}_{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_L}(\bar{\mathbf{p}}_n))) \} \end{aligned}\quad (10)$$

where $\mathcal{G}_{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_L}(\bar{\mathbf{p}}_n)$ denotes the generated vector corresponding to random AP locations $\bar{\mathbf{p}}_n$.

This objective function achieves its maximum when the discriminator outputs “1” and “0” for inputting a real ADOA vector and a generated ADOA vector, respectively, i.e., $\mathcal{D}_{\bar{\omega}, \omega^{(1)}, \mathbf{b}^{(1)}, \dots, \omega^{(M)}, \mathbf{b}^{(M)}}(\Xi(\psi(\mathbf{p}_n))) = 1$ and $\mathcal{D}_{\bar{\omega}, \omega^{(1)}, \mathbf{b}^{(1)}, \dots, \omega^{(M)}, \mathbf{b}^{(M)}}(\mathcal{G}_{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_L}(\bar{\mathbf{p}}_n)) = 0$. Hence, the parameters of the logistic regression network can be updated by maximizing the above objective function with the classical gradient ascend method.

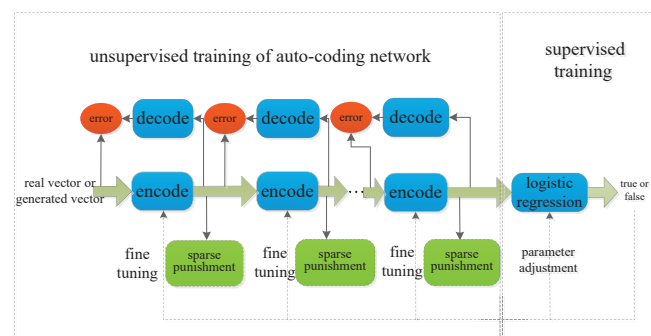


Figure 3. Auto-encoder neural network structure of the discriminator.

Fine-Tuning of Global Parameters

Considering that the label of the input data of the discriminator is known in a GAN, the classical back-propagation (BP) algorithm is employed to calculate the gradient of the objective function (7) with respect to the global parameters and fine-tune the global parameters. After the fine-tuning of the global parameters, the optimal discriminator can be obtained for a given generator. That is, the discriminator achieves its optimal ability of distinguishing the real ADOA vectors from the generated ones.

3.1.2. Training of the Generator

Considering that the real ADOA data possesses 2-dimensional characteristics, random 2-dimensional coordinates of the MT are chosen as the input of the generator. Correspondingly, the generator outputs the ‘fake’ ADOA vectors directly according to the geometric

relationship among the AP coordinates, the MT position and the ADOA vector in (1), rather than through a deep neural network, which is the most straightforward model in traditional GANs. Hence the AP coordinates are the generator's parameters to be optimized in our GAN.

The iterative topology-search process can be interpreted as follows. At each training iteration, a batch of random two-dimensional MT positions is sampled from the indoor coordinate region. Given the current AP-coordinate parameters, the explicit generator computes the corresponding generated ADOA vectors according to the geometric ADOA function. The discriminator then evaluates whether these generated ADOA vectors follow the same distribution as the measured ADOA vectors. The resulting adversarial loss is used to update the AP-coordinate parameters of the generator. Therefore, the topology search is not performed by adding or removing APs in a graphical procedure, but by continuously optimizing the AP-coordinate parameters in the explicit geometric generator. After convergence, the optimized AP-coordinate parameters are taken as the estimated AP topology.

Given the parameters and the input of the generator, that is, the coordinates of all the APs $\{\mathbf{a}_l\}_{l=1}^L$ and the random MT coordinate $\bar{\mathbf{p}}_n \in \mathbb{P}$ with $\mathbb{P}$ denoting the 2-dimensional indoor random coordinate set, the generated ADOA $\check{\theta}_{l_1, l_2}(\bar{\mathbf{p}}_n)$ for the AP pair $\{AP_{l_1}, AP_{l_2}\}$ at the random position $\bar{\mathbf{p}}_n$ satisfies

$$\cos \check{\theta}_{l_1, l_2}(\bar{\mathbf{p}}_n) = \frac{(\mathbf{a}_{l_1} - \bar{\mathbf{p}}_n) \bullet (\mathbf{a}_{l_2} - \bar{\mathbf{p}}_n)}{|\mathbf{a}_{l_1} - \bar{\mathbf{p}}_n| \cdot |\mathbf{a}_{l_2} - \bar{\mathbf{p}}_n|}. \quad (11)$$

Thus, the generated ADOA vector at this random MT position can be expressed as

$$\begin{aligned} \mathcal{G}_{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_L}(\bar{\mathbf{p}}_n) \\ = [\cos(\check{\theta}_{1,2}(\bar{\mathbf{p}}_n)), \cos(\check{\theta}_{2,3}(\bar{\mathbf{p}}_n)), \dots, \cos(\check{\theta}_{L-1,L}(\bar{\mathbf{p}}_n))] \end{aligned}$$

The Loss Function of the Generator

The generator of the proposed GAN aims to generate ADOA vectors with the same distribution characteristics as that of the real data. In other words, the generated data can not be differentiated from the real data by the discriminator. For this purpose, the generator is trained to minimize the following loss function with respect to the AP coordinates

$$\begin{aligned} \mathcal{L}(\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_L) = \\ \frac{1}{N} \sum_{n=1}^N \left\{ \log \left(1 - \mathcal{D}_{\mathbf{I}, \mathbf{I}^{(1)}, \mathbf{b}^{(1)}, \dots, \mathbf{I}^{(M)}, \mathbf{b}^{(M)}}(\mathcal{G}_{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_L}(\bar{\mathbf{p}}_n)) \right) \right\} \\ \text{s.t. } \mathbf{a}_1 = [0, 0], \mathbf{a}_2 = [0, 1] \end{aligned} \quad (12)$$

wherein the constraint condition fixes the coordinates of $\mathbf{a}_1$ and $\mathbf{a}_2$ in order to resolve the AP coordinates' ambiguity of rotation, translation and scaling. This loss function achieves its minimum as the discriminator outputs "1"s for each generated data $\mathcal{G}_{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_L}(\bar{\mathbf{p}}_n)$, which means that the discriminator classifies the 'fake' data as the real one.

Optimizing the Generator's Parameters

Based on the loss function (9), the gradient descent method is employed to iteratively optimize the generator's parameters. Although the generation function is the explicit geometric relationship in (8), the error between the actual output and the ideal output of discriminator depends on not only the generation function but also the discriminator structure, which involves hidden layers. Hence, the BP algorithm is still needed to calculate the gradient of the errors with respect to the generator parameters.

Denote the error square between the actual output and the ideal output of the discriminator corresponding to the input of the n -th generated vector, $\mathcal{G}_{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_L}(\bar{\mathbf{p}}_n)$, as

$$e_n = \left(1 - \mathcal{D}_{\bar{\mathbf{t}}, \mathbf{l}^{(1)}, \mathbf{b}^{(1)}, \dots, \mathbf{l}^{(M)}, \mathbf{b}^{(M)}}(\mathcal{G}_{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_L}(\bar{\mathbf{p}}_n))\right)^2$$

where “1” is the ideal output of the discriminator for inputting generated vectors. Denote the i -th element of the n -th generated vector as $\mathcal{G}_i$, i.e., $\mathcal{G}_i = \cos(\theta_{i,i+1}(\mathbf{p}_n))$.

Considering the BP-based gradient calculation is composed of T similar iterative steps, only the t -th step is exemplified as follows.

In the t -th iterative step, the coordinate of the l -th AP is updated as

$$\hat{\mathbf{a}}_l(t) = \hat{\mathbf{a}}_l(t-1) - \alpha \frac{\partial e_n}{\partial \mathbf{a}_l}, \quad l = 3, \dots, L \quad (13)$$

where α denotes the learning rate.

The gradient of the error square with respect to $\mathbf{a}_l$ is derived as

$$\begin{aligned} \frac{\partial e_n}{\partial \mathbf{a}_l} &= \sum_{i=1}^{L-1} \frac{\partial e_n}{\partial \mathcal{G}_i} \frac{\partial \mathcal{G}_i}{\partial \mathbf{a}_l} \\ &= \frac{\partial e_n}{\partial \mathcal{G}_{i=l-1}} \frac{\partial \mathcal{G}_{i=l-1}}{\partial \mathbf{a}_l} + \frac{\partial e_n}{\partial \mathcal{G}_{i=l}} \frac{\partial \mathcal{G}_{i=l}}{\partial \mathbf{a}_l} \end{aligned} \quad (14)$$

where

$$\frac{\partial e_n}{\partial \mathcal{G}_i} = \sum_{j=1}^{J^{(2)}} \frac{\partial e_n}{\partial \mathbf{z}_{n,j}^{(1)}} \frac{\partial \mathbf{z}_{n,j}^{(1)}}{\partial \mathcal{G}_i} = \sum_{j=1}^{J^{(2)}} (-\delta_{n,k}^{(1)}) (w_{ij}^{(1)})$$

where $\mathbf{z}_{n,j}^{(1)}$ represents the output (before activation function) of the j -th neuron of the first layer of discriminator, $w_{ij}^{(1)}$ represents the corresponding weight of the first layer of the discriminator, $\delta_{n,j}^{(1)}$ denotes the BP gradient at the j -th neuron in the first layer of the discriminator, $J^{(2)}$ represents the number of neuron of the second layer and

$$\begin{aligned} \frac{\partial \mathcal{G}_{i=l-1}}{\partial \mathbf{a}_l} &= \frac{\partial \frac{(\mathbf{a}_{l-1} - \bar{\mathbf{p}}_n) \bullet (\mathbf{a}_l - \bar{\mathbf{p}}_n)}{|\mathbf{a}_{l-1} - \bar{\mathbf{p}}_n| \cdot |\mathbf{a}_l - \bar{\mathbf{p}}_n|}}{\partial \mathbf{a}_l} \\ &= \frac{(\mathbf{a}_{l-1} - \bar{\mathbf{p}}_n)}{|\mathbf{a}_{l-1} - \bar{\mathbf{p}}_n| \cdot |\mathbf{a}_l - \bar{\mathbf{p}}_n|} - \frac{(\mathbf{a}_{l-1} - \bar{\mathbf{p}}_n) \bullet (\mathbf{a}_l - \bar{\mathbf{p}}_n)(\mathbf{a}_l - \bar{\mathbf{p}}_n)}{(|\mathbf{a}_{l-1} - \bar{\mathbf{p}}_n| \cdot |\mathbf{a}_l - \bar{\mathbf{p}}_n|)^3} \\ \frac{\partial \mathcal{G}_{i=l}}{\partial \mathbf{a}_l} &= \frac{\partial \frac{(\mathbf{a}_l - \bar{\mathbf{p}}_n) \bullet (\mathbf{a}_{l+1} - \bar{\mathbf{p}}_n)}{|\mathbf{a}_l - \bar{\mathbf{p}}_n| \cdot |\mathbf{a}_{l+1} - \bar{\mathbf{p}}_n|}}{\partial \mathbf{a}_l} \\ &= \frac{(\mathbf{a}_{l+1} - \bar{\mathbf{p}}_n)}{|\mathbf{a}_{l+1} - \bar{\mathbf{p}}_n| \cdot |\mathbf{a}_l - \bar{\mathbf{p}}_n|} - \frac{(\mathbf{a}_{l+1} - \bar{\mathbf{p}}_n) \bullet (\mathbf{a}_l - \bar{\mathbf{p}}_n)(\mathbf{a}_l - \bar{\mathbf{p}}_n)}{(|\mathbf{a}_{l+1} - \bar{\mathbf{p}}_n| \cdot |\mathbf{a}_l - \bar{\mathbf{p}}_n|)^3} \end{aligned} \quad (15)$$

Define $\delta_{n,j}^{(m)} = -\frac{\partial e_n}{\partial \mathbf{z}_{n,j}^{(m)}}$; then, the BP gradient, $\delta_{n,j}^{(1)}$, can be recursively obtained as

$$\begin{aligned} \delta_{n,j}^{(m)} &= \sum_{i=1}^{J^{(m+1)}} \delta_{n,i}^{(m+1)} w_{ji}^{(m+1)}, \\ j &= 1, \dots, J^{(m)}; m = 2, \dots, M-1 \\ \delta_{n,j}^{(M)} &= -2(1 - \mathbf{z}_{n,j}^{(M)}) \end{aligned} \quad (16)$$

Through iteratively training the discriminator and the generator, the proposed GAN system can generate ADOA vectors with the same distribution as the real vector finally.

Consequently, the parameters of the generator converge to the optimal solution; that is, the optimal estimation of the AP position coordinates is obtained.

The proposed framework does not first generate a large number of candidate virtual APs and then eliminate false ones. The APs in the generator correspond to the dominant paths observed in the ADOA measurements, including the LOS path from the physical AP and the dominant single-reflection NLOS paths represented by virtual APs. Therefore, under the assumed dominant-path model, the GAN estimates the coordinates of these APs/virtual APs by matching the measured ADOA distribution, and false virtual AP elimination is not a separate step in the current framework. If spurious paths are included due to incorrect path detection or association, they may introduce outliers in the ADOA dataset and degrade the topology estimation accuracy; robust path selection or pruning strategies can be incorporated in future work.

The pseudocode for the proposed GAN-based AP topology estimation method is given in Algorithm 1.

3.2. Localization of the MT and Construction of the Environment Map

Based on the estimated AP positions above, the MT position for each ADOA vector can be further obtained. From the geometric relationship between the MT's position and the APs' positions in (1), the estimation of the MT position for each ADOA vector can be formulated as the following minimization problem:

$$\begin{aligned} \hat{\mathbf{p}}_n = \arg \min_{\mathbf{p}_n \in \mathbb{P}} \sum_{\check{\theta}_{l_1, l_2}(\mathbf{p}_n) \in \Psi(\mathbf{p}_n)} \{ (\hat{\mathbf{a}}_{l_1} - \mathbf{p}_n) \bullet (\hat{\mathbf{a}}_{l_2} - \mathbf{p}_n) \\ - \cos \check{\theta}_{l_1, l_2}(\mathbf{p}_n) |\hat{\mathbf{a}}_{l_1} - \mathbf{p}_n| \cdot |\hat{\mathbf{a}}_{l_2} - \mathbf{p}_n| \}^2 \end{aligned} \quad (17)$$

Hence, each MT position can be obtained by searching discretely over possible indoor MT position-range $\mathbb{P}$.

Algorithm 1 GAN based AP topology estimation.

Input: Observed ADOA vectors $\{\tilde{\theta}(\mathbf{p}_n)\}_{n=1}^N$ at a large number of random and unknown positions

1: Initialization: $\hat{\mathbf{a}}_{3:L}^0, \hat{\mathbf{a}}_1 = [0 \ 0], \hat{\mathbf{a}}_2 = [1 \ 0], t = 0$

2: **for** number of training iterations **do**

3: **Training the discriminator:**

4: Sample minibatch of N observed ADOA vector samples from $\{\tilde{\theta}(\mathbf{p}_n)\}_{n=1}^N$

5: Train the auto-encoder neural network in the discriminator by descending its stochastic gradient with respect to the loss function $J(\omega^{(m)}, \mathbf{b}^{(m)}, \hat{\mathbf{b}}^{(m)})$ in (4)

$$\nabla_{\omega^{(m)}, \mathbf{b}^{(m)}} J(\omega^{(m)}, \mathbf{b}^{(m)}, \hat{\mathbf{b}}^{(m)})$$

6: Sample minibatch of N observed ADOA vector samples from $\{\tilde{\theta}(\mathbf{p}_n)\}_{n=1}^N$

7: Select M MT positions $\{\tilde{\mathbf{p}}_m\}_{m=1}^M$ randomly and generate ADOA vector samples Equation (8)

8: Update the discriminator by ascending its stochastic gradient with respect to the objective function $J'(\bar{\omega}, \omega^{(1)}, \mathbf{b}^{(1)}, \dots, \omega^{(M)}, \mathbf{b}^{(M)})$ in (7)

$$\nabla_{\omega^{(m)}, \mathbf{b}^{(m)}} J'(\bar{\omega}, \omega^{(1)}, \mathbf{b}^{(1)}, \dots, \omega^{(M)}, \mathbf{b}^{(M)})$$

9: **Training the generator:**

10: Select M MT positions $\{\tilde{\mathbf{p}}_m\}_{m=1}^M$ randomly and generate ADOA vector samples Equation (8)

11: Update the generator by descending its stochastic gradient with respect to the loss function $\mathcal{L}(\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_L)$ in (9)

$$\nabla_{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_L} \mathcal{L}(\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_L)$$

12: **end for**

Output: the AP position estimates

Finally, given by the estimated MT positions $\{\hat{\mathbf{p}}_n\}_{n=1}^N$ and the estimated AP positions $\{\hat{\mathbf{a}}_l\}_{l=1}^L$, the wall positions can be geometrically calculated as in [19], which are shown in Figure 4. For each estimated MT position $\{\hat{\mathbf{p}}_n\}$ and each position pair $\{\hat{\mathbf{a}}_1, \hat{\mathbf{a}}_l\}$ of the physical AP and a virtual AP, the reflection point on an indoor wall corresponding to a received NLOS path at $\{\hat{\mathbf{p}}_n\}$ can be estimated as the intersection between the segment $\overline{\hat{\mathbf{p}}_n \hat{\mathbf{a}}_l}$ and the line that bisects segment $\overline{\hat{\mathbf{a}}_1 \hat{\mathbf{a}}_l}$. Thus, the indoor walls, i.e., the environment map, can be constructed.

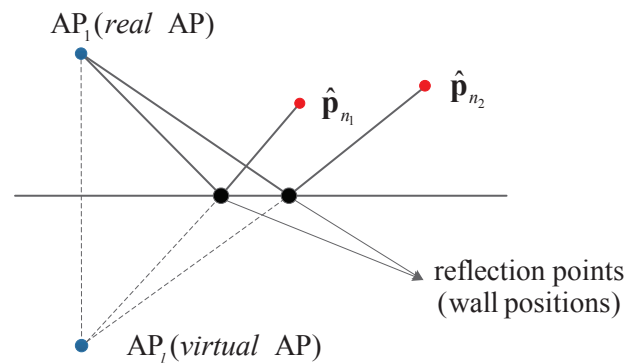


Figure 4. Illustration of the wall position estimation.

3.3. CRLB for the MT Localization

In this section, we derive the Fisher information matrix (FIM) and the Cramer–Rao lower bound (CRLB) of the estimation error of the MT position, and we compare the positioning accuracy of our algorithm with the CRLB of localization in Section 4.

From (1), the ADOA at position $\mathbf{p}_n$ for the AP pair $\{AP_{l_1}, AP_{l_2}\}$ can be derived as

$$\check{\theta}_{l_1, l_2}(\mathbf{p}_n) = \arccos\left\{\frac{(\mathbf{a}_{l_1} - \mathbf{p}_n) \bullet (\mathbf{a}_{l_2} - \mathbf{p}_n)}{|\mathbf{a}_{l_1} - \mathbf{p}_n| \bullet |\mathbf{a}_{l_2} - \mathbf{p}_n|}\right\}$$

Assume the random measurement error of the AOA follows zero-mean Gaussian distribution with variance of σ^2 , thus the probability density function of the measured ADOA vector $\boldsymbol{\psi}$ at random and unknown MT positions $\{\mathbf{p}_n\}_{n=1}^N$ is

$$\begin{aligned} \mathcal{P}(\boldsymbol{\psi}|\check{\boldsymbol{\theta}}) &= \frac{1}{(2\pi)^{\frac{L-1}{2}} (\det \mathbf{C})^{\frac{1}{2}}} \\ &\exp\left\{-\frac{1}{2}(\boldsymbol{\psi} - \check{\boldsymbol{\theta}})^T \mathbf{C}^{-1}(\boldsymbol{\psi} - \check{\boldsymbol{\theta}})\right\} \end{aligned} \quad (18)$$

where

$$\begin{aligned} \boldsymbol{\psi} &= [\boldsymbol{\psi}(\mathbf{p}_1)^T, \boldsymbol{\psi}(\mathbf{p}_2)^T, \dots, \boldsymbol{\psi}(\mathbf{p}_N)^T]^T \\ \boldsymbol{\psi}(\mathbf{p}_n) &= [\hat{\theta}_{1,2}(\mathbf{p}_n), \hat{\theta}_{2,3}(\mathbf{p}_n), \dots, \hat{\theta}_{L-1,L}(\mathbf{p}_n)]^T \\ \check{\boldsymbol{\theta}} &= [\check{\theta}(\mathbf{p}_1)^T, \check{\theta}(\mathbf{p}_2)^T, \dots, \check{\theta}(\mathbf{p}_N)^T]^T \\ \check{\theta}(\mathbf{p}_n)^T &= [\check{\theta}_{1,2}(\mathbf{p}_n), \dots, \check{\theta}_{L-1,L}(\mathbf{p}_n)]^T \end{aligned}$$

and $\mathbf{C}$ is the covariance matrix of the measured ADOA vector $\boldsymbol{\psi}$. Since the measured ADOA vectors at different positions are independent from each other, the covariance matrix $\mathbf{C}$ is a block diagonal matrix as

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}^{(1)} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{C}^{(2)} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{C}^{(N)} \end{bmatrix} \quad (19)$$

Moreover, it is not difficult to derive from $\check{\theta}_{l_1, l_2}(\mathbf{p}_n) = \theta_{l_1}(\mathbf{p}_n) - \theta_{l_2}(\mathbf{p}_n)$ that the sub-covariance matrix $\mathbf{C}^{(n)}$ satisfies

$$C_{l_1, l_2}^{(n)} = \begin{cases} 2\sigma^2, & \text{if } l_1 = l_2 \\ -\sigma^2, & \text{if } |l_1 - l_2| = 1 \\ 0, & \text{else} \end{cases} \quad (20)$$

Hence, the probability density function of the measured ADOA vector $\boldsymbol{\psi}$ can also be rewritten as

$$\mathcal{P}(\boldsymbol{\psi}|\check{\boldsymbol{\theta}}) = \frac{1}{(2\pi)^{\frac{L-1}{2}} (\det \mathbf{C})^{\frac{1}{2}}} \exp \left\{ -\frac{1}{2} \sum_{n=1}^N (\boldsymbol{\psi}(\mathbf{p}_n) - \check{\boldsymbol{\theta}}(\mathbf{p}_n))^T (\mathbf{C}^{(n)})^{-1} (\boldsymbol{\psi}(\mathbf{p}_n) - \check{\boldsymbol{\theta}}(\mathbf{p}_n)) \right\} \quad (21)$$

3.3.1. FIM of the ADOA Vector

The FIM of the ADOA vector $\check{\boldsymbol{\theta}}_n$ can be expressed as

$$\mathbf{J}_{\check{\boldsymbol{\theta}}} = \mathcal{E}_{\mathcal{P}(\boldsymbol{\psi}|\check{\boldsymbol{\theta}})} \left\{ -\frac{\partial^2 \ln \mathcal{P}(\boldsymbol{\psi}|\check{\boldsymbol{\theta}})}{\partial \check{\boldsymbol{\theta}} \partial \check{\boldsymbol{\theta}}^T} \right\}. \quad (22)$$

Due to the independence between the measured ADOA vectors at different positions, the FIM of the ADOA vector $\check{\boldsymbol{\theta}}$ is also a block diagonal matrix as

$$\mathbf{J}_{\check{\boldsymbol{\theta}}} = \begin{bmatrix} \mathbf{J}_{\check{\boldsymbol{\theta}}(\mathbf{p}_1)} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_{\check{\boldsymbol{\theta}}(\mathbf{p}_2)} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{J}_{\check{\boldsymbol{\theta}}(\mathbf{p}_N)} \end{bmatrix} \quad (23)$$

with

$$\begin{aligned} \mathbf{J}_{\check{\boldsymbol{\theta}}(\mathbf{p}_n)} &= \mathcal{E}_{\mathcal{P}(\boldsymbol{\psi}(\mathbf{p}_n)|\check{\boldsymbol{\theta}}(\mathbf{p}_n))} \left\{ -\frac{\partial^2 \ln \mathcal{P}(\boldsymbol{\psi}(\mathbf{p}_n)|\check{\boldsymbol{\theta}}(\mathbf{p}_n))}{\partial \check{\boldsymbol{\theta}}(\mathbf{p}_n) \partial \check{\boldsymbol{\theta}}(\mathbf{p}_n)^T} \right\} \\ &= -\frac{\partial^2 \left\{ -\frac{1}{2} (\boldsymbol{\psi}(\mathbf{p}_n) - \check{\boldsymbol{\theta}}(\mathbf{p}_n))^T (\mathbf{C}^{(n)})^{-1} (\boldsymbol{\psi}(\mathbf{p}_n) - \check{\boldsymbol{\theta}}(\mathbf{p}_n)) \right\}}{\partial \check{\boldsymbol{\theta}}(\mathbf{p}_n) \partial \check{\boldsymbol{\theta}}(\mathbf{p}_n)^T} \\ &= \frac{1}{2} \left((\mathbf{C}^{(n)})^{-1} + \{(\mathbf{C}^{(n)})^{-1}\}^T \right) = (\mathbf{C}^{(n)})^{-1} \end{aligned} \quad (24)$$

3.3.2. FIM of the MT Position and the AP Positions

Further, the FIM of the MT position and the AP positions is obtained through a transformation of variables from $\check{\boldsymbol{\theta}}$ to $\boldsymbol{\eta}$, where $\boldsymbol{\eta} = [\mathbf{a}_3^T, \dots, \mathbf{a}_L^T, \mathbf{p}_1^T, \dots, \mathbf{p}_N^T]^T$.

$$\mathbf{J}_{\boldsymbol{\eta}} = \mathbf{T} \mathbf{J}_{\check{\boldsymbol{\theta}}} \mathbf{T}^T \quad (25)$$

where the transformation matrix T is expressed as

$$T = \frac{\partial \check{\theta}^T}{\partial \eta} = \begin{bmatrix} \frac{\partial \check{\theta}(\mathbf{p}_1)^T}{\partial \eta(1)} & \cdots & \frac{\partial \check{\theta}(\mathbf{p}_N)^T}{\partial \eta(1)} \\ \frac{\partial \check{\theta}(\mathbf{p}_1)^T}{\partial \eta(2)} & \cdots & \frac{\partial \check{\theta}(\mathbf{p}_N)^T}{\partial \eta(2)} \\ \vdots & \cdots & \vdots \\ \frac{\partial \check{\theta}(\mathbf{p}_1)^T}{\partial \eta(L-2+N)} & \cdots & \frac{\partial \check{\theta}(\mathbf{p}_N)^T}{\partial \eta(L-2+N)} \end{bmatrix} \quad (26)$$

with

$$\frac{\partial \check{\theta}(\mathbf{p}_n)^T}{\partial \eta(m)} = \left[\frac{\partial \check{\theta}_{1,2}(\mathbf{p}_n)}{\partial \eta(m)} \quad \cdots \quad \frac{\partial \check{\theta}_{L-1,L}(\mathbf{p}_n)}{\partial \eta(m)} \right] \quad (27)$$

and

$$\frac{\partial \check{\theta}_{l,l+1}(\mathbf{p}_n)}{\partial \eta(m)} = \frac{\partial \check{\theta}_{l,l+1}(\mathbf{p}_n)}{\partial \mathbf{a}_{(m+2)}}, \quad m \leq L-2$$

$$= \begin{cases} -\frac{1}{\|\mathbf{p}_n - \mathbf{a}_l\|_2} [\sin(\theta_l(\mathbf{p}_n)), -\cos(\theta_l(\mathbf{p}_n))]^T, \\ \quad \text{if } (m+2) = l \\ \frac{1}{\|\mathbf{p}_n - \mathbf{a}_{(l+1)}\|_2} [\sin(\theta_{l+1}(\mathbf{p}_n)), -\cos(\theta_{l+1}(\mathbf{p}_n))]^T, \\ \quad \text{if } (m+2) = l+1 \\ 0, \quad \text{else} \end{cases} \quad (28)$$

$$\frac{\partial \check{\theta}_{l,l+1}(\mathbf{p}_n)}{\partial \eta(m)} = \frac{\partial \check{\theta}_{l,l+1}(\mathbf{p}_n)}{\partial \mathbf{p}_{(m-L+2)}}, \quad m > L-2$$

$$= \begin{cases} \frac{1}{\|\mathbf{p}_n - \mathbf{a}_l\|_2} [\sin(\theta_l(\mathbf{p}_n)), -\cos(\theta_l(\mathbf{p}_n))]^T - \\ \frac{1}{\|\mathbf{p}_n - \mathbf{a}_{(l+1)}\|_2} [\sin(\theta_{l+1}(\mathbf{p}_n)), -\cos(\theta_{l+1}(\mathbf{p}_n))]^T, \\ \quad \text{if } (m-L+2) = n \\ 0, \quad \text{else} \end{cases} \quad (29)$$

It can be derived by substituting (23) into (22) that the FIM J_η satisfies

$$J_\eta(2r-1:2r, 2s-1:2s) = \sum_{n=1}^N \frac{\partial \check{\theta}(\mathbf{p}_n)^T}{\partial \eta(r)} J_{\check{\theta}(\mathbf{p}_n)} \frac{\partial \check{\theta}(\mathbf{p}_n)}{\partial \eta(s)} \quad (30)$$

By integrating (25) and (26) into (27), the FIM J_η can be partitioned as

$$J_\eta = \left[\begin{array}{c|c} \mathbf{A}_{2(L-2) \times 2(L-2)} & \mathbf{B}_{2(L-2) \times (2N)} \\ \hline \mathbf{C}_{2N \times 2(L-2)} & \mathbf{D}_{2N \times 2N} \end{array} \right] \quad (31)$$

with

$$\begin{aligned}
 & \mathbf{A}(2r-1:2r, 2s-1:2s) \\
 &= \sum_{n=1}^N \frac{\partial \check{\boldsymbol{\theta}}(\mathbf{p}_n)^T}{\partial \boldsymbol{\eta}(r)} J_{\check{\boldsymbol{\theta}}(\mathbf{p}_n)} \frac{\partial \check{\boldsymbol{\theta}}(\mathbf{p}_n)}{\partial \boldsymbol{\eta}(s)} = \sum_{n=1}^N \frac{\partial \check{\boldsymbol{\theta}}(\mathbf{p}_n)^T}{\partial \mathbf{a}_{r+2}} J_{\check{\boldsymbol{\theta}}(\mathbf{p}_n)} \frac{\partial \check{\boldsymbol{\theta}}(\mathbf{p}_n)}{\partial \mathbf{a}_{s+2}} \\
 & \quad r = 1, 2, \dots, L-2; s = 1, 2, \dots, L-2 \\
 & \mathbf{B}(2r-1:2r, 2s-1:2s) \\
 &= \frac{\partial \check{\boldsymbol{\theta}}(\mathbf{p}_s)^T}{\partial \boldsymbol{\eta}(r)} J_{\check{\boldsymbol{\theta}}(\mathbf{p}_s)} \frac{\partial \check{\boldsymbol{\theta}}(\mathbf{p}_s)}{\partial \boldsymbol{\eta}(s+L-2)} = \frac{\partial \check{\boldsymbol{\theta}}(\mathbf{p}_s)^T}{\partial \mathbf{a}_{r+2}} J_{\check{\boldsymbol{\theta}}(\mathbf{p}_s)} \frac{\partial \check{\boldsymbol{\theta}}(\mathbf{p}_s)}{\partial \mathbf{p}_s} \\
 & \quad r = 1, 2, \dots, L-2; s = 1, 2, \dots, N \\
 & \mathbf{C}(2r-1:2r, 2s-1:2s) \\
 &= \frac{\partial \check{\boldsymbol{\theta}}(\mathbf{p}_r)^T}{\partial \boldsymbol{\eta}(r+L-2)} J_{\check{\boldsymbol{\theta}}(r)} \frac{\partial \check{\boldsymbol{\theta}}(\mathbf{p}_r)}{\partial \boldsymbol{\eta}(s)} = \frac{\partial \check{\boldsymbol{\theta}}(\mathbf{p}_r)^T}{\partial \mathbf{p}_r} J_{\check{\boldsymbol{\theta}}(\mathbf{p}_r)} \frac{\partial \check{\boldsymbol{\theta}}(\mathbf{p}_r)}{\partial \mathbf{a}_{s+2}} \\
 & \quad r = 1, 2, \dots, N; s = 1, 2, \dots, L-2 \\
 & \mathbf{D}(2r-1:2r, 2s-1:2s) \\
 &= \delta(r-s) \frac{\partial \check{\boldsymbol{\theta}}(\mathbf{p}_r)^T}{\partial \boldsymbol{\eta}(r+L-2)} J_{\check{\boldsymbol{\theta}}(\mathbf{p}_r)} \frac{\partial \check{\boldsymbol{\theta}}(\mathbf{p}_s)}{\partial \boldsymbol{\eta}(s+L-2)} \\
 &= \delta(r-s) \frac{\partial \check{\boldsymbol{\theta}}(\mathbf{p}_r)^T}{\partial \mathbf{p}_r} J_{\check{\boldsymbol{\theta}}(\mathbf{p}_r)} \frac{\partial \check{\boldsymbol{\theta}}(\mathbf{p}_s)}{\partial \mathbf{p}_s} \\
 & \quad r = 1, 2, \dots, N; s = 1, 2, \dots, N
 \end{aligned} \tag{32}$$

It is worth mentioning that the submatrix $\mathbf{D}$ is a block diagonal matrix and each block is a 2×2 matrix.

According to the inversion property of partitioned matrices, the inverse of the FIM $J_{\boldsymbol{\eta}}$ can be expressed as

$$J_{\boldsymbol{\eta}}^{-1} = \left[\begin{array}{c|c} (\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1} & -(\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1}\mathbf{B}\mathbf{D}^{-1} \\ \hline -\mathbf{D}^{-1}\mathbf{C}(\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1} & \mathbf{D}^{-1} + \mathbf{D}^{-1}\mathbf{C}(\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1}\mathbf{B}\mathbf{D}^{-1} \end{array} \right] \tag{33}$$

Considering that the submatrix $\mathbf{D}$ is a block diagonal matrix, its inverse $\mathbf{D}^{-1}$ is also a block diagonal matrix and each block is the inverse of the corresponding block in the submatrix $\mathbf{D}$. Moreover, the dimension of $(\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1}$ is $2(L-2) \times 2(L-2)$. As a result, the inversion of the FIM $J_{\boldsymbol{\eta}}$ does not involve high computation complexity from (30).

3.3.3. Bounds on Errors of the MT Position and the AP Positions

The mean squared error (MSE) of $\boldsymbol{\eta} = [\mathbf{a}_3, \dots, \mathbf{a}_L, \mathbf{p}_1, \dots, \mathbf{p}_N]^T$ is bounded as [27]

$$\mathcal{E}_{\mathcal{P}(\psi(\mathbf{p}_n)|\check{\boldsymbol{\theta}}_n)} \left[(\hat{\boldsymbol{\eta}} - \boldsymbol{\eta})(\boldsymbol{\eta} - \boldsymbol{\eta})^T \right] \geq J_{\boldsymbol{\eta}}^{-1}$$

Therefore, the position error bounds (PEBs) of the AP and the MT are respectively obtained as

$$\begin{aligned}
 PEB_{\mathbf{a}_l} &= \sqrt{\text{tr}\{[J_{\boldsymbol{\eta}}^{-1}]_{(2l-5):(2l-4), (2l-5):(2l-4)}\}}, \\
 & \quad l = 3, 4, \dots, L
 \end{aligned} \tag{34}$$

$$\begin{aligned}
 PEB_{\mathbf{p}_n} &= \sqrt{\text{tr}\{[J_{\boldsymbol{\eta}}^{-1}]_{2n'-1:2n', 2n'-1:2n'}\}}, \\
 & \quad n' = (L-2+n), n = 1, 2, \dots, N
 \end{aligned} \tag{35}$$

4. Simulation Results

In order to evaluate the performance of the proposed algorithm, we have conducted computer simulations in a relatively simple $10\text{ m} \times 8\text{ m}$ (length $\times$ width) 2-dimension indoor WLAN environment, in which there is only one physical AP located at $(0.5, 4.5, 2.5)$ and four reflective walls. The 2-dimension layout of the indoor WLAN environment is shown in Figure 5. The classical ray-tracing propagation model [10] is adopted in the simulations to generate the multipath AOAs at random indoor positions. Although more recent ray-tracing tools, such as Sionna RT and MATLAB 2022a ray-tracing functions with STL indoor models [28], can generate more realistic channel responses by considering material properties, blockage, diffuse scattering, and antenna patterns, the simulations in this work mainly focus on the multipath geometric relationship among the physical/virtual APs, the MT positions, and the ADOA vectors. And the AOA measurement error is modeled as a zero-mean Gaussian random error [9] to represent the practical uncertainty caused by beam training, channel estimation, and AOA extraction. Therefore, no specific antenna-array configuration or fixed number of antenna elements is explicitly adopted in the simulation. In the simulations, the mmWave propagation model is simplified according to its severe path loss and quasi-optical propagation characteristics in air [10]. Specifically, only the LOS component and the first-order specular reflections from the side walls are retained, while higher-order reflected paths are neglected. Accordingly, the four virtual APs are generated as the mirror images of the physical AP with respect to the four corresponding reflecting walls.

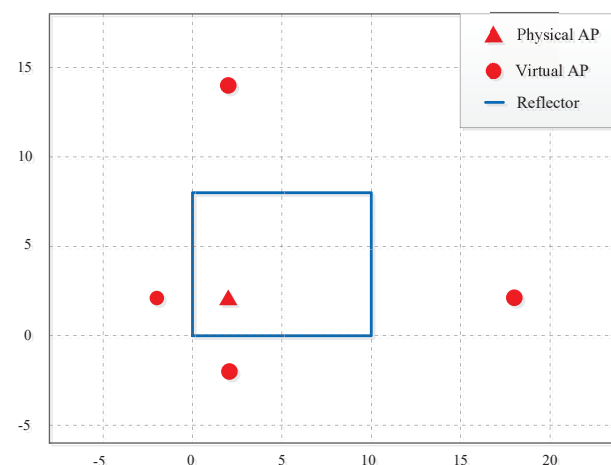


Figure 5. The 2-dimension layout of the indoor mmW WLAN environment.

Similar to the proposed algorithm, the JADE algorithm in [9] can also estimate the AP positions and the MT positions jointly by employing the geometric relationship between the APs and the MT based on the multipath ADOA measurements in indoor environments with only one AP deployed. Hence, we compare their localization and mapping performance in the simulations. Although many SLAM or JLAM methods have been proposed in the literature [20,29,30], they usually require additional information beyond what is directly available from the mmWave communication system, such as known floorplans, known reflector positions, synchronization information, AOD measurements, array orientation, or other prior environmental knowledge. In contrast, the proposed GAN-based JLAM method works under a zero-initial-information setting using only the collected ADOA measurements, and therefore the JADE algorithm is selected as the main baseline because it follows comparable information assumptions and can jointly estimate the AP topology and MT positions without prior floorplan information. In addition, the Cramer–Rao lower bound of localization is simulated as the ideal performance benchmark for mmW

positioning. In the proposed GAN system, the discriminator is modeled as a three-layer auto-encoder neural network with 100 neurons in each layer.

The present simulation considers one physical AP and four virtual APs generated by single-bounce reflections, which is a typical virtual-AP-based mmWave SLAM scenario and is consistent with the comparison setting of the JADE algorithm. When multiple physical APs and/or more virtual APs are adopted, the ADOA dataset will contain more spatial angle observations and more independent geometric constraints. Therefore, the learned ADOA manifold becomes richer, and the AP topology and MT position estimation ambiguity can be further reduced. As a result, the cumulative distribution curve of estimation errors is expected to shift toward smaller error values compared with the single-physical-AP case. However, multiple physical APs and/or more virtual APs also increase the ADOA vector dimension, the number of AP topology parameters, the neural network training overhead, and the complexity of multipath data association.

4.1. Computational Complexity Comparison

The computational cost of the proposed GAN-based JLAM method mainly comes from the auto-encoder-based discriminator training and the geometric generator update. For one discriminator update, the forward and backward propagation complexity is approximately $O\left(\sum_{r=1}^R d_{r-1}d_r\right)$, where N is the number of measured ADOA vectors and $d_{r-1} \times d_r$ is the weight matrix size of the r -th neural layer. Since the generator is constrained by the explicit ADOA geometric model, one generator update mainly requires the calculation of generated ADOA vectors and their gradients with respect to AP coordinates, with complexity approximately $zO(NL)$.

Therefore, the total complexity of the proposed adversarial training can be expressed as $O\left(T_{\text{GAN}}\left[K_D N \sum_{r=1}^R d_{r-1}d_r + K_G NL\right]\right)$ where T_{GAN} is the number of adversarial iterations, and K_D and K_G are the numbers of discriminator and generator updates in each iteration. In comparison, the JADE algorithm mainly relies on iterative least-squares estimations, and its complexity can be written as $O(L^2(NT_{\text{JADE}} + G))$. The proposed method therefore introduces additional neural network training overhead compared with the least-squares iterations in JADE. However, the proposed generator is an explicit geometric generator rather than a fully trainable neural network, which reduces the number of trainable parameters. Due to the limited revision time, we did not add a new execution-time simulation, but the above complexity comparison clarifies the additional overhead and the computational difference between the proposed method and JADE.

4.2. Software Implementation

The proposed GAN-based JLAM algorithm was implemented using Python 3.11 numerical routines. In contrast to a conventional GAN with a black-box neural network generator, the generator in the proposed framework is constructed as an explicit geometric ADOA mapping function. The trainable parameters of the generator are the coordinates of the candidate physical and virtual APs, and the generated samples are calculated directly from the geometric relationship among the AP coordinates, the random MT coordinates, and the ADOA measurements. The discriminator is implemented as a deep auto-encoder-based neural network followed by a logistic regression layer. In the simulations, the discriminator is modeled as a three-layer auto-encoder neural network with 100 neurons in each layer.

The complete implementation is currently maintained as internal research code and is not publicly released at this stage, because it contains unpublished extensions and project-specific modules that are not directly related to the core algorithm presented in this paper. Nevertheless, all essential algorithmic components required for reproducing the

reported results are described in the manuscript, including the explicit generator function, the discriminator structure, the loss functions, the training procedure, the initialization setting, and the simulation parameters.

4.3. Comparison of Localization Performance

To show the localization performance of the proposed algorithm, the JADE algorithm and the Cramer–Rao lower bound, their cumulative probability distribution curves of errors in positioning the MT are presented in Figure 6. In this comparison, $N = 2000$ ADOA vectors are collected at random indoor MT positions, and the standard deviation of the AOA measurement noise is set to 2° . The JADE method addresses the otherwise NP-hard joint AP-and-MT estimation problem by converting it into an iterative least-squares framework, where the AP coordinates and the MT positions are alternately updated through two LS estimation steps. Although this strategy reduces the optimization difficulty, it relies on a relaxation of the original geometric constraints linking the AP positions, MT positions, and ADOA observations. As a result, some useful geometric information contained in the measured ADOA data may not be fully preserved. On the other hand, the proposed algorithm uses a large number of ADOA vector measurements and the geometric relationship to train the generative adversarial network without any relaxation of the geometric constraint, thus can extract more information from the measured ADOA data than the JADE algorithm. This explains why the proposed algorithm performs better than the JADE algorithm in Figure 6.

Furthermore, Figure 6 indicates that the proposed GAN-based JLAM method has not yet reached the ideal Cramer–Rao lower bound, and a visible performance gap remains between their cumulative error distributions. This is because the proposed GAN-based algorithm is subject to insufficient ADOA vector measurements and the non-optimal selection of the number of neural layers and neurons in the discriminator. Moreover, the final MT positioning stage is performed over a discretized two-dimensional spatial grid, which also introduces a quantization error floor. Even if the adversarial network estimates the physical and virtual AP topology accurately, the estimated MT position is restricted to one of the predefined grid points. Therefore, the discretization step size limits the minimum achievable positioning error. It should be noted that the Cramer–Rao lower bound is not a practical localization algorithm but an ideal theoretical lower bound derived under the assumed noise model and unbiased-estimator conditions. Therefore, practical methods such as the proposed GAN-based JLAM method and JADE are expected to perform worse than the Cramer–Rao lower bound due to finite ADOA samples, adversarial training approximation, AP topology estimation errors, discretized position search, and measurement noise.

Figure 7 presents the average MT localization error of the proposed GAN-based JLAM method, the JADE algorithm, and the Cramer–Rao lower bound under different ADOA noise levels. In this set of simulations, the number of measured ADOA vectors collected from random indoor positions is fixed at $N = 2000$. As shown in Figure 7, the positioning accuracy of both the proposed method and JADE improves as the standard deviation of the ADOA measurement noise decreases, since more accurate angular observations provide more reliable geometric constraints and reduce the averaged influence of measurement errors. Moreover, the proposed method consistently achieves a much smaller localization error than JADE. This improvement is mainly attributed to the fact that the proposed GAN-based framework can exploit the geometric information embedded in the ADOA dataset more effectively and can maintain better robustness under noisy measurements. By contrast, JADE relies on LS-based estimation steps, whose performance is more easily affected by measurement noise.

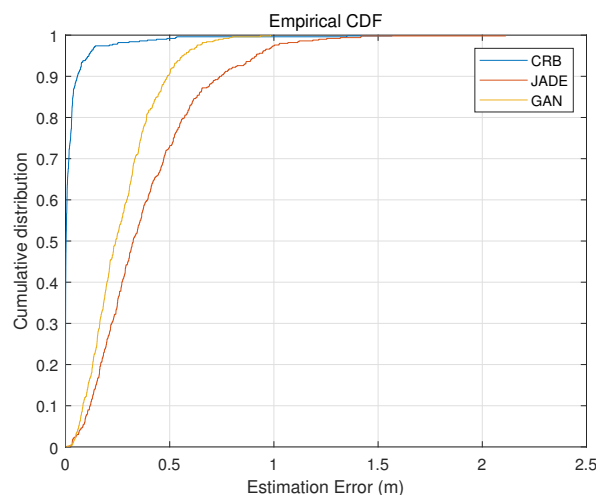


Figure 6. The cumulative probability distribution curves of localization error.

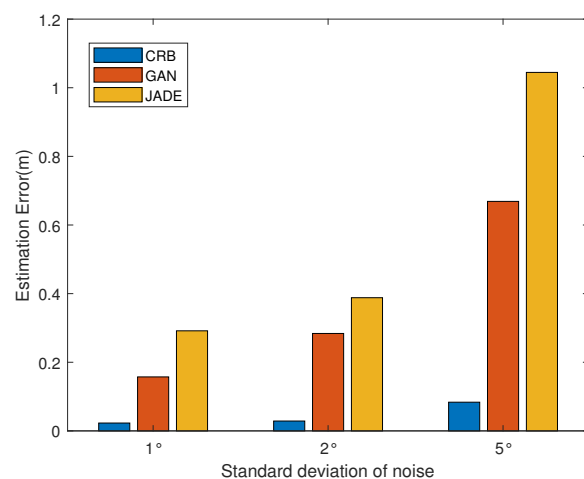


Figure 7. Localization performance versus AOA measurement noise.

Figure 8 shows the localization performance of the proposed algorithm and the JADE algorithm against the number of ADOA vector measurements. The standard deviation of Gaussian noise is set to 2° . It can be seen that the localization error of both the proposed algorithm and the JADE algorithm decreases with the increase in the number of samples due to the averaging effects. In addition, the proposed algorithm performs better than the JADE algorithm mainly due to the information loss caused by the relaxation of the geometric constraint in the JADE algorithm.

4.4. Comparison of Mapping Performance

We now test the room boundary estimation capabilities, i.e., the mapping performance of the proposed method as explained in Section 4.4. Figure 9 and Figure 10 demonstrate the mapping performances of the proposed algorithm and the JADE algorithm versus the ADOA measurement noise and the number of ADOA vector measurements, respectively. The number of measured ADOA vectors at random indoor positions is set to 2000 in Figure 9 and the standard deviation of Gaussian noise is set to 2° in Figure 10. Similarly, Figures 9 and 10 show that both the proposed algorithm and the JADE algorithm achieve better mapping performance with the standard deviation of the ADOA measurement noise decreasing or with the increase in the number of samples. Moreover, it is obvious that the proposed algorithm estimates the room boundary more accurately than the JADE algorithm in all cases in Figures 9 and 10, because the proposed method requires no relaxation of the

geometric constraint and thus achieves higher utilization of information contained in the ADOA measurement data. This further confirms the merit of the proposed GAN-based SLAM algorithm.

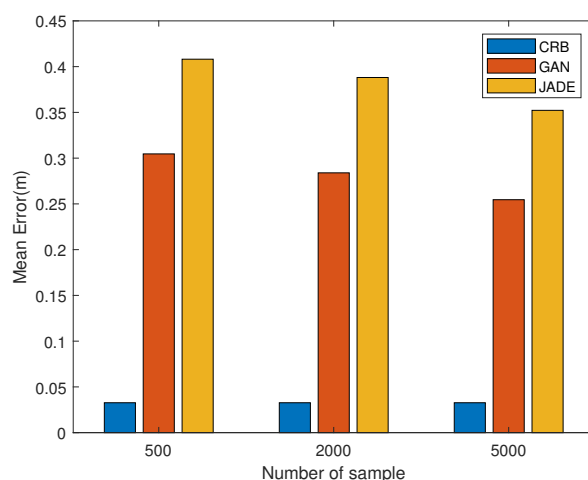


Figure 8. Localization performance versus the number of measured ADOA vectors.

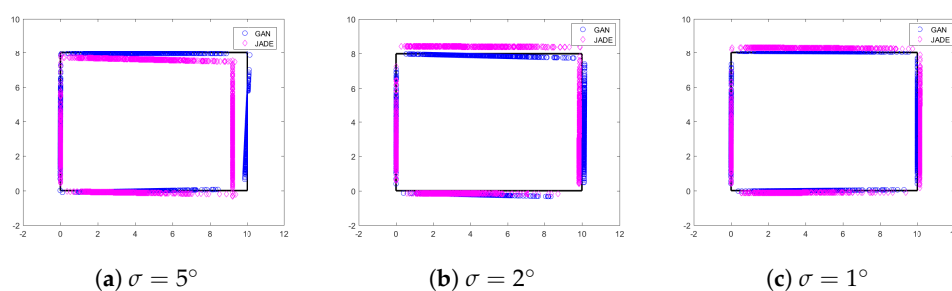


Figure 9. The mapping performance for different standard deviations of ADOA errors: $N = 2000$.

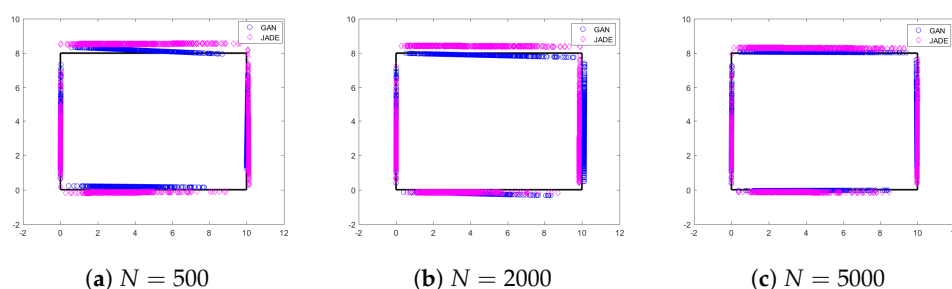


Figure 10. The mapping performance for different numbers of ADOA vectors: $\sigma = 2^\circ$.

5. Conclusions

In this paper, we proposed a novel generative adversarial network (GAN)-based joint localization and mapping (JLAM) method for millimeter-wave (mmWave) communication systems. The proposed method exploits the two-dimensional distribution characteristics of high-dimensional angle difference of arrival (ADOA) vectors collected at a large number of random and unknown mobile terminal (MT) positions. Different from conventional GAN structures, the generator in the proposed framework is constructed as an explicit geometric ADOA function, whose trainable parameters are the physical and virtual access point (AP) coordinates. Therefore, the proposed method can learn the distribution of unlabeled ADOA measurements and transform them into the AP geometric topology. Based on the

estimated AP topology, the MT positions can be further estimated and the indoor map can be constructed without requiring any a priori knowledge of the indoor environment.

The simulation results quantitatively confirm the effectiveness of the proposed method. Under the representative setting with $N = 2000$ measured ADOA vectors and AOA measurement noise standard deviation $\sigma = 2^\circ$, the proposed method achieves an average localization error of about 0.25 m, while the JADE algorithm achieves about 0.60 m, corresponding to an error reduction of approximately 58%. The mapping results also show that the proposed method estimates the room boundary more accurately than JADE under different noise levels and different numbers of ADOA measurements. These results demonstrate that the proposed GAN-based JLAM method can efficiently exploit unlabeled ADOA measurements and achieve improved localization and mapping performance for mmWave communication systems.

Supplementary Materials: The following supporting information can be downloaded at <https://www.mdpi.com/article/10.3390/s26134319/s1>.

Author Contributions: Z.Z. designed the simulation scheme and wrote the paper. Z.C. proposed the GAN-based SLAM algorithm, gave suggestions on the proposed algorithm and helped to improve the quality of the final manuscript. L.C. helped to perform the simulation tests and analyzed the results. All authors have read and agreed to the published version of the manuscript.

Funding: The work was supported by the Project of Shaanxi Provincial Key Research and Development Program (Program No.2025CG-GJHX-08).

Data Availability Statement: The original data presented in the study are openly available in the Supplementary Materials: Simulation Results for GAN-based JLAM.zip.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

SLAM	Simultaneous Localization and Mapping
GAN	Generative Adversarial Networks
mmW	Millimeter Wave
AP	Access Point
ADOA	Angle Difference of Arrival
MT	Mobile Terminal
MPCs	Multipath Components
JADE	Joint Anchor and Device location Estimation
LS	Least-Squares
JLAM	Joint Localization and Mapping
CLAM	Communication-driven Localization and Mapping
NLOS	Non-Line of Sight
2D/3D	2-dimension/3-dimension

References

1. Zafari, F.; Gkelias, A.; Leung, K.K. A Survey of Indoor Localization Systems and Technologies. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 2568–2599.
2. Fan, J.; Gao, J.; Li, R. A High-Resolution TOA Estimation Algorithm Based on Coherent Subspace for Dual Band Signals. *IEEE Trans. Wirel. Commun.* **2025**, *24*, 4321–4335.
3. Zhang, Y.; Wang, R.; Liu, E.; Li, B.; Ge, H. Asynchronous Time of Arrival Based 5G Localization Methods and Optimal Geometry Analysis. *IEEE Trans. Wirel. Commun.* **2024**, *23*, 14567–14580.
4. Li, Y.; Chen, S.; Liu, H. NLoS Mitigation for Practical UWB Localization. In Proceedings of the 20th IEEE International Conference on Ubiquitous Intelligence and Computing (UIC 2023), Portsmouth, UK, 28–31 August 2023; pp. 45–52.

5. Wang, Y.; Cheng, H.; Meng, M.Q.-H. A Learning Based Sequence to Sequence WiFi Fingerprinting Framework for Accurate Pedestrian Indoor Localization Using Unconstrained RSSI. *IEEE Internet Things J.* **2025**, *12*, 18234–18248.
6. Zheng, Z.; Huang, B.; Lu, Z. MLP-mmWP: High-Precision Millimeter Wave Positioning Based on MLP-Mixer Neural Networks. *Sensors* **2023**, *23*, 3864.
7. Tedeschini, B.C.; Nicoli, M. Cooperative Deep-Learning Positioning in mmWave 5G-Advanced Networks. *IEEE J. Sel. Areas Commun.* **2023**, *41*, 3799–3815.
8. Yassin, A.; Nasser, Y.; Al-Dubai, A.Y.; Awad, M. MOSAIC: Simultaneous Localization and Environment Mapping Using mmWave Without A-Priori Knowledge. *IEEE Access* **2018**, *6*, 68932–68947.
9. Palacios, J.; Casari, P.; Widmer, J. JADE: Zero-knowledge device localization and environment mapping for millimeter wave systems. In Proceedings of the IEEE INFOCOM 2017-IEEE Conference on Computer Communications, Atlanta, GA, USA, 1–4 May 2017; pp. 1–9.
10. Maltsev, A.; Maslennikov, R.; Sevastyanov, A. Experimental Investigations of 60 GHz WLAN Systems in Office Environment. *IEEE J. Sel. Areas Commun.* **2009**, *27*, 1488–1499.
11. Xing, Y.; Kanhere, O.; Ju, S.; Rappaport, T.S. Indoor Wireless Channel Properties at Millimeter Wave and Sub-Terahertz Frequencies. In Proceedings of the 2019 IEEE Global Communications Conference (GLOBECOM), Waikoloa, HI, USA, 9–13 December 2018; pp. 1–6.
12. Rappaport, T.S.; Xing, Y.; MacCartney, G.R.; Molisch, A.F.; Mellios, E.; Zhang, J. Overview of millimeter wave communications for fifth-generation (5G) wireless networks—With a focus on propagation models. *IEEE Trans. Antennas Propag.* **2017**, *65*, 6213–6230.
13. Ali, A.; González-Prelcic, N.; Heath, R.W.; Ghosh, A. Position locationing for millimeter wave systems. *IEEE Access* **2018**, *6*, 44618–44630.
14. Huang, Y.; Wang, W.; Zhang, D.; Wang, J.; Chen, C.; Cao, Y.; Wang, W. Enhancing WiFi CSI Fingerprinting: A Deep Auxiliary Learning Approach. *IEEE Internet Things J.* **2025**, *12*, 52217–52230.
15. Hao, Z.; Yan, H.; Dang, X.; Ma, Z.; Jin, P.; Ke, W. Millimeter-wave radar localization using indoor multipath effect. *Sensors* **2022**, *22*, 5671.
16. Chen, F.; Au, W.; Valaee, S.; Tan, Z. Received-Signal-Strength-Based Indoor Positioning Using Compressive Sensing. *IEEE Trans. Mob. Comput.* **2012**, *11*, 1983–1993.
17. Wang, X.; Gao, L.; Mao, S.; Pandey, S. CSI-based fingerprinting for indoor localization: A deep learning approach. *IEEE Trans. Veh. Technol.* **2017**, *66*, 763–776.
18. Yassin, A.; Nasser, Y.; Awad, M. Simultaneous Context Inference and Mapping using mm-Wave for Indoor Scenarios. In Proceedings of the 2017 IEEE International Conference on Communications (ICC), Paris, France, 21–25, May 2017; pp. 1–6.
19. Palacios, J.; Bielsa, G.; Casari, P.; Widmer, J. Communication-Driven Localization and Mapping for Millimeter Wave Networks. In Proceedings of the IEEE INFOCOM 2018-IEEE Conference on Computer Communications, Honolulu, HI, USA, 16–19 April 2018; pp. 2402–2410.
20. Yang, J.; Wen, C.-K.; Xu, J.; Que, H.; Wei, H.; Jin, S. Angle-Based SLAM on 5G mmWave Systems: Design, Implementation, and Measurement. *IEEE Internet Things J.* **2023**, *10*, 17755–17771.
21. Kim, H.; Wymeersch, H.; Garcia, N.; Seco-Granados, G.; Kim, S. 5G mmwave vehicular tracking. In Proceedings of the 2018 52nd Asilomar Conference on Signals, Systems, and Computers (ACSSC), Pacific Grove, CA, USA, 28–31 October 2018; pp. 541–547.
22. Leitinger, E.; Meyer, F.; Hlawatsch, F.; Witrals, K.; Tufvesson, F.; Win, M.Z. A belief propagation algorithm for multipath-based slam. *IEEE Trans. Wirel. Commun.* **2019**, *18*, 5613–5629.
23. Chen, L.; Chen, Z.; Ji, Z. Expectation–Maximization-Based Simultaneous Localization and Mapping for Millimeter-Wave Communication Systems. *Sensors* **2022**, *22*, 6941:1–6941:17.
24. Njima, W.; Bazzi, A.; Chafii, M. DNN-Based Indoor Localization Under Limited Dataset Using GANs and Semi-Supervised Learning. *IEEE Access* **2022**, *10*, 69896–69909.
25. Creswell, A.; White, T.; Dumoulin, V. Generative Adversarial Networks: An Overview. *IEEE Signal Process. Mag.* **2018**, *35*, 53–65.
26. Krichen, M. Generative Adversarial Networks. In Proceedings of the 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT), Delhi, India, 6–8 July 2023; pp. 1–7.
27. Kay, S.M. *Fundamentals of Statistical Signal Processing: Estimation Theory*; Prentice-Hall: New York, NY, USA, 2010.
28. Hoydis, J.; Ait Aoudia, F.; Cammerer, S.; Nimier-David, M.; Binder, N.; Marcus, G.; Keller, A. Sionna RT: Differentiable ray tracing for radio propagation modeling. *arXiv* **2023**, arXiv:2303.11103.

29. Kim, H.; Granström, K.; Gao, L.; Battistelli, G.; Kim, S.; Wymeersch, H. 5G mmWave Cooperative Positioning and Mapping Using Multi-Model PHD Filter and Map Fusion. *IEEE Trans. Wirel. Commun.* **2021**, *20*, 1185–1200.
30. Jiang, W.; Wei, Z.; Feng, Z.; Zhang, H. Joint Beam Management and SLAM for mmWave Communication Systems. *IEEE Trans. Veh. Technol.* **2023**, *72*, 17892–17905.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.