

Attentional Graph Neural Network Is All You Need for Robust Massive Network Localization

Yan, Wenzhong; Yin, Feng; Wang, Juntao; Leus, Geert; Zoubir, Abdelhak M.; Tian, Yang

DOI

[10.1109/JSTSP.2025.3590639](https://doi.org/10.1109/JSTSP.2025.3590639)

Licence

Dutch Copyright Act (Article 25fa)

Publication date

2025

Document Version

Final published version

Published in

IEEE Journal on Selected Topics in Signal Processing

Citation (APA)

Yan, W., Yin, F., Wang, J., Leus, G., Zoubir, A. M., & Tian, Y. (2025). Attentional Graph Neural Network Is All You Need for Robust Massive Network Localization. *IEEE Journal on Selected Topics in Signal Processing*, 19(7), 1493-1513. <https://doi.org/10.1109/JSTSP.2025.3590639>

Important note

To cite this publication, please use the final published version (if applicable). Please check the document version above.





Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights. We will remove access to the work immediately and investigate your claim.

Attentional Graph Neural Network is All You Need for Robust Massive Network Localization

Wenzhong Yan , *Student Member, IEEE*, Feng Yin , *Senior Member, IEEE*, Juntao Wang ,
Geert Leus, *Fellow, IEEE*, Abdelhak M. Zoubir , *Life Fellow, IEEE*, and Yang Tian

Abstract—In this paper, we design Graph Neural Networks (GNNs) with attention mechanisms to tackle an important yet challenging nonlinear regression problem: massive network localization. We first review our previous network localization method based on Graph Convolutional Network (GCN), which can exhibit state-of-the-art localization accuracy, even under severe Non-Line-of-Sight (NLOS) conditions, by carefully preselecting a constant threshold for determining adjacency. As an extension, we propose a specially designed Attentional GNN (AGNN) model to resolve the sensitive thresholding issue of the GCN-based method and enhance the underlying model capacity. The AGNN comprises an Adjacency Learning Module (ALM) and Multiple Graph Attention Layers (MGALs), employing distinct attention architectures to systematically address the demerits of the GCN-based method, rendering it more practical for real-world applications. Comprehensive analyses are conducted to explain the superior performance of these methods, including a theoretical analysis of the AGNN's dynamic attention property and computational complexity, along with a systematic discussion of their robust characteristic against NLOS measurements. Extensive experimental results demonstrate the effectiveness of the GCN-based and AGNN-based network localization methods. Notably, integrating attention mechanisms into the AGNN yields substantial improvements in localization accuracy, approaching the fundamental lower bound and showing approximately 37% to 53% reduction in localization error compared to the vanilla GCN-based method across various NLOS noise configurations. Both methods outperform all competing approaches by far in terms of localization accuracy, robustness, and computational time, especially for considerably large network sizes.

Received 15 June 2024; revised 16 December 2024 and 24 April 2025; accepted 6 July 2025. Date of publication 18 July 2025; date of current version 31 December 2025. This work was supported in part by the Shenzhen Science and Technology Program under Grant JCYJ20220530143806016 and Grant RCJC20210609104448114 and in part by the 1+1+1 CUHK-CUHKSZ-GDST Joint Collaboration Fund for Platform Development of Laboratory of Geospatial Intelligence. An earlier version of this paper was presented in part at the 2021 IEEE International Conference on Acoustics, Speech and Signal Processing [DOI: 1109/ICASSP39728.2021.9414520]. The guest editor coordinating the review of this article and approving it for publication was Prof. Aylin Yener. (*Corresponding author: Feng Yin.*)

Wenzhong Yan, Feng Yin, and Juntao Wang are with the School of Science & Engineering, The Chinese University of Hong Kong, Shenzhen 518172, China (e-mail: wenzhongyan@link.cuhk.edu.cn; yinfeng@cuhk.edu.cn; juntaowang@link.cuhk.edu.cn).

Geert Leus is with the Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, 2826, CD Delft, The Netherlands (e-mail: G.J.T.Leus@tudelft.nl).

Abdelhak M. Zoubir is with the Signal Processing Group, Technische Universität Darmstadt, 64283 Darmstadt, Germany (e-mail: zoubir@spg.tu-darmstadt.de).

Yang Tian is with the Huawei Technologies Company Ltd., Shanghai 201206, China (e-mail: yang.tian@huawei.com).

This article has supplementary downloadable material available at <https://doi.org/10.1109/JSTSP.2025.3590639>, provided by the authors.

Digital Object Identifier 10.1109/JSTSP.2025.3590639

Index Terms—Graph neural networks, attention mechanism, massive network localization, non-line-of-sight.

I. INTRODUCTION

GRAPH Neural Networks (GNNs) have recently achieved excellent results in a variety of graph-related learning tasks, such as node classification, link prediction, and graph classification [2], [3], [4], [5], [6]. By leveraging graph topology to aggregate neighbor information and enhance node representations [7], [8], GNNs offer a unique advantage in processing complex, interconnected datasets. Despite their effectiveness, GNNs have predominantly focused on classification tasks with discrete labels. However, nonlinear regression problems, which represent a substantial portion of practical applications in signal processing, remain less investigated.

In this paper, we delve into a classic yet challenging nonlinear regression problem: massive network localization. Over recent decades, a variety of canonical methods have been developed to tackle this issue, ranging from maximum likelihood [9], [10], [11], [12] and least-squares based [13] estimation methods to multi-dimensional scaling [14], mathematical programming [15], [16], [17], and Bayesian message passing approaches [18], [19]. Despite their diversity, a major challenge these methods face is the impact of Non-Line-of-Sight (NLOS) propagation, which can incur severe performance degradation of the localization accuracy due to severely biased distance estimation.

The prevalent strategy to combat NLOS effects is to perform NLOS identification for each link, and either discard or suppress the NLOS measurements [20]. However, this method requires large-scale offline calibration and incurs high operational cost. Alternatively, the NLOS effects can be dealt with from an algorithmic aspect. Based on the assumption that NLOS noise follows a certain probability distribution, maximum likelihood estimation based methods handling NLOS effects were developed in [21], [22]. Yet, such methods may suffer from performance deterioration due to model mismatch. Moreover, network localization has been formulated as a regularized optimization problem in which the NLOS-inducing sparsity of the ranging-bias parameters was exploited [23]. Unfortunately, all the above methods are computationally expensive for massive networks, which makes them less feasible for large-scale applications.

Most recently, we have introduced a novel Graph Convolutional Network (GCN)-based method for network localization

in [1], which significantly advances the field of massive network localization by offering excellent accuracy, robustness, and efficiency without requiring extensive offline calibration or NLOS identification. However, some challenges persist in the vanilla GCN-based method, particularly concerning its reliance on empirically predefined thresholds for establishing adjacency and its insufficient model expressiveness.

Recent studies have explored advanced GNN-based techniques to improve wireless localization. For instance, [24] proposed a graph-learning approach that incorporates temporal and directional mobility patterns into a time-varying graph structure, complemented by a mobility regularization term to improve performance. [25] developed a GNN-based localization method that utilizes Channel State Information (CSI) by modeling amplitude-phase relationships as graph inputs, enabling efficient position estimation via a GraphSAGE architecture [3]. These works illustrate the potential of domain-adapted graph-based methods to address challenges in wireless localization.

The advent of attention mechanisms has recently significantly impacted the modeling of structured data, including spatial and temporal data [4], [26], [27]. These mechanisms, by design, enable models to focus on the most relevant parts of the input data, enhancing their ability to learn complex patterns from the data. For example, [28] proposed a federated graph learning framework integrating GNNs with self-attention mechanisms for personalized Wi-Fi localization, effectively capturing spatial and client-specific dependencies. These advancements highlight the benefits of combining attention mechanisms with GNNs to further enhance network localization performance under diverse conditions.

Motivated by these factors, in this paper, we propose an Attentional GNN (AGNN) model, comprising two crucial modules: the Adjacency Learning Module (ALM) and Multiple Graph Attention Layers (MGALs), aimed at effectively mitigating the limitations inherent in the GCN-based network localization. Besides, we systematically analyze the superb properties of the proposed model. The main contributions can be summarized as follows.

- To the best of our knowledge, this is the first work to integrate GNNs and attention mechanisms for network localization. Together with our previous GCN-based method [1], we hope to pave a new path to solve massive network localization in our futuristic connected world.
- To remove the reliance on a predefined constant threshold for determining adjacency in the vanilla GCN-based method, we introduce an ALM that uses a tailored attention architecture to autonomously learn edge-specific thresholds. This module can characterize more reasonable and flexible node neighborhoods, particularly effective in filtering out NLOS-affected node pairs.
- To enhance the model expressiveness, we introduce MGALs, which leverage attention mechanisms to learn aggregation weights based on the features of each target node and its neighbors, instead of using constant weights in the GCN-based method. Moreover, differences in the distributions of the learned attention scores for Line-of-Sight (LOS) and NLOS links evidently show that our

TABLE I
SUMMARY OF NOTATIONS

| Notation | Definition |
|--|--|
| $\mathcal{S}, \mathcal{S} $ | set, cardinality of set |
| $\mathbf{x}, \mathbf{X}, (\cdot)^\top$ | vectors, matrices, vector/matrix transpose |
| $\mathbf{x}_i, \mathbf{x}_{[i,:]}, x_{ij}$ | column, row, element of matrix \mathbf{X} |
| $[N]$ | set of natural numbers $1, \dots, N$ |
| $[\cdot \parallel \cdot]$ | concatenation operation for two vectors |
| $ \mathbf{x} ^p$ | element-wise p -th power of the absolute value of \mathbf{x} |
| $\ \cdot\ , \ \cdot\ _F$ | Euclidean norm, Frobenius norm |

adopted attention mechanism is able to identify NLOS links, addressing a long-standing challenge in the field.

- To support the significant performance enhancement of the AGNN model, we perform a comprehensive analysis with the highlights on the dynamic attentional properties of our specially designed attention architectures, and a thorough complexity analysis for all competing models.
- To showcase the efficacy of the AGNN model, we conduct extensive experiments which revealed that AGNN most closely approaches the Cramér-Rao Bound (CRB) and exhibits the strongest robustness against NLOS noise among all tested methods. It significantly reduces localization error by approximately 37% to 53% compared to the GCN-based method under diverse noise conditions, while maintaining comparable computation times.

The remainder of this paper is organized as follows. The background of network localization and its problem formulation are introduced in Section II. In Section III, we briefly review the GCN framework for network localization and show its superb localization performance. For more general and practical scenarios, the AGNN model is further proposed in Section IV. In Section V, we conduct theoretical analyses of AGNN, followed by the numerical results in Section VI. Finally, we conclude the paper in Section VII. Notations used in this paper are summarized in Table I.

II. BACKGROUND AND PROBLEM FORMULATION

A. Localization Scenario

Fig. 1 illustrates a classic wireless network localization scenario. In the context of a massive wireless network, only a limited number of nodes, known as anchors, possess precise location information from satellite navigation systems. Conversely, the remaining nodes, referred to as agents, lack satellite navigation access and consequently need to be located. Equipped with omni-directional antennas, each node broadcasts signals containing information such as node ID, transmit power, transmit time, and network configurations.

This scenario is highly representative of many real-world massive wireless networks, such as 5G networks, IoT networks, vehicle networks, and metaverse [29], [30].

B. Problem Formulation

We restrict our study to a wireless network in a two-dimensional (2-D) space, as the extension to the 3-D case is straightforward. We let $\mathcal{S}_l = \{1, 2, \dots, N_l\}$ be the set of indices for the anchors (labeled nodes), whose positions $\mathbf{p}_i \in \mathbb{R}^2, \forall i \in$

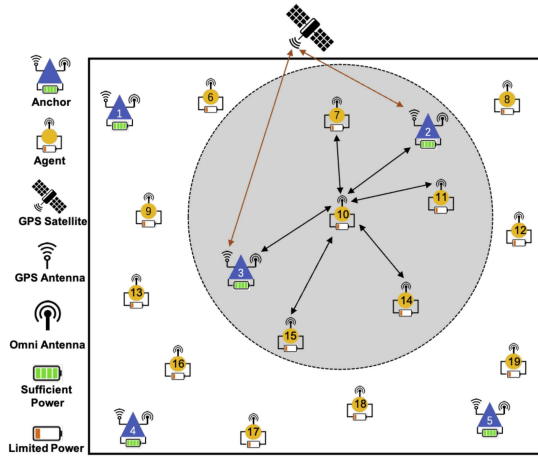


Fig. 1. Network localization scenario. The shaded area represents the region of interest for the 10-th node after threshold selection which will be introduced in Section III-A.

S_l are known and fixed, and $S_u = \{N_l + 1, N_l + 2, \dots, N\}$ be the set of indices for the agents (unlabeled nodes), whose positions are unknown.

In accordance with numerous established studies, e.g., [22], the measured distance between any two nodes i and j can be represented as follows:

$$x_{ij} = d_{ij} + n_{ij}, \quad (1)$$

where $d_{ij} := \|\mathbf{p}_i - \mathbf{p}_j\|$ is defined as the Euclidean distance between nodes (i, j) . The term n_{ij} accounts for an additive measurement error arising from line-of-sight (LOS) and non-line-of-sight (NLOS) propagation, which can be further detailed as follows:

$$n_{ij} = n_{ij}^L + b_{ij}n_{ij}^N. \quad (2)$$

Herein, the LOS noise, n_{ij}^L , follows a zero-mean Gaussian distribution, i.e., $n_{ij}^L \sim \mathcal{N}(0, \sigma^2)$, while the NLOS term, n_{ij}^N , is generated from a positive biased distribution; b_{ij} is sampled from the Bernoulli distribution $\mathcal{B}(p_B)$ with p_B being the NLOS occurrence probability [31]. Consequently, a measurement matrix, denoted by $\mathbf{X} \in \mathbb{R}^{N \times N}$, can be constructed by stacking the measured distances, i.e., the (i, j) -th entry of \mathbf{X} , x_{ij} . Notably, we consider a symmetric scenario where $x_{ij} = x_{ji}$ and $x_{ii} = 0$ for $i = 1, 2, \dots, N$.

The measured distances can be estimated from the Received Signal Strength (RSS), or the Time of Arrival (ToA) measurements, which can be acquired through diverse technologies, including cellular networks, wireless local area networks, ultra-wideband radio frequencies, as well as ultra and audible sound. In all cases, NLOS propagation can introduce either significant time delays in ToA measurements or substantial signal attenuation in RSS measurements [32], thereby impeding the accuracy of the eventual distance estimation.

Building on the aforementioned formulations, our primary objective is to precisely localize the agents within a massive wireless network while ensuring satisfactory computation time. To achieve this, we propose a novel GCN-based network localization method in the subsequent section.

III. NETWORK LOCALIZATION WITH CLASSIC GCN

In this section, we first introduce a GCN-based data-driven method for network localization that was originally proposed in [1]. Subsequently, we provide some key findings regarding the reasons behind the exceptional performance of this model. Lastly, we discuss some of its limitations that necessitate further improvements.

A. Graph Convolutional Network [1]

In this subsection, our focus lies on the formulation of the network localization problem utilizing classical GCNs. We formally define an undirected graph associated with a wireless network as $\mathcal{G} = (\mathcal{V}, \mathbf{A})$, where \mathcal{V} denotes the set of nodes $\{v_1, v_2, \dots, v_N\}$ in the network, including anchors and agents, and $\mathbf{A} \in \mathbb{R}^{N \times N}$ is a symmetric binary adjacency matrix. In classical GCNs, each element a_{ij} of \mathbf{A} represents the existence of the edge connecting nodes v_i and v_j , which can be inherently interpreted as a measure of similarity between nodes i and j . While, in the context of network localization, the concept of similarity is particularly relevant to proximity, essentially indicated by the Euclidean distance d_{ij} between nodes. To integrate this concept into the GCN-based method, we introduce an Euclidean distance threshold, denoted as T_h , to determine the existence of an edge between any two nodes, i.e., whether two nodes are proximal. Therefore, by applying this threshold, the adjacency matrix \mathbf{A} is constructed as follows:

$$a_{ij} = \begin{cases} 0, & \text{if } x_{ij} > T_h, \\ 1, & \text{otherwise.} \end{cases} \quad (3)$$

As further elaborated in Section III-B, this threshold plays a crucial role in localization performance. And we also need to note that \mathbf{A} includes self-connections, making it an augmented adjacency matrix. Based on the aforesaid definitions, we can construct a truncated measurement matrix $\hat{\mathbf{X}} = \mathbf{A} \odot \mathbf{X}$ which only contains those measured distances that are smaller than or equal to T_h .¹

With the well-defined graph, we propose our GCN-based method as follows. We first denote the node representation at the k -th layer as $\mathbf{H}^{(k)} \in \mathbb{R}^{N \times D_k}$, where D_k is the hidden dimension of the k -th layer. And the initial node representation is set to be the truncated measurement matrix, i.e., $\mathbf{H}^{(0)} = \hat{\mathbf{X}}$. In each layer, a *feature propagation* step is implemented first by the following update rule:

$$\bar{\mathbf{H}}^{(k)} \in \mathbb{R}^{N \times D_k} = \hat{\mathbf{A}} \mathbf{H}^{(k)}, \quad (4)$$

where $\hat{\mathbf{A}} \in \mathbb{R}^{N \times N} := \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}}$ signifies the normalized adjacency matrix [2], $\mathbf{D} \in \mathbb{R}^{N \times N} := \text{diag}(d_1, d_2, \dots, d_N)$ is the associated degree matrix of \mathbf{A} with $d_i = \sum_{j=1}^N a_{ij}$, and $\bar{\mathbf{H}}^{(k)}$ represents the propagated representation matrix in the k -th graph convolution layer. Conceptually, this step effectively smoothens the node representations within the local graph neighborhood, thereby promoting similar predictions for closely connected

¹In $\hat{\mathbf{X}}$, a zero means a null value, representing distances between corresponding nodes whose values exceed T_h .

nodes. Following the feature propagation step, the subsequent two stages of our GCN, namely *linear transformation* and *nonlinear activation*, are implemented as:

$$\mathbf{H}^{(k+1)} \in \mathbb{R}^{N \times D_k} = \varphi \left(\hat{\mathbf{H}}^{(k)} \mathbf{W}^{(k)} \right), \quad (5)$$

where $\mathbf{W}^{(k)} \in \mathbb{R}^{D_k \times D_{k+1}}$ is the trainable weight matrix in the k -th layer, and $\varphi(\cdot)$ is an element-wise nonlinear activation function such as $\text{ReLU}(\cdot) = \max(0, \cdot)$ [33].

Taking the output of the GCN as the estimated positions of all nodes, denoted by $\hat{\mathbf{P}} = [\hat{\mathbf{p}}_1, \hat{\mathbf{p}}_2, \dots, \hat{\mathbf{p}}_{N_l}, \hat{\mathbf{p}}_{N_l+1}, \dots, \hat{\mathbf{p}}_{N_r}]^\top \in \mathbb{R}^{N \times 2}$, we can optimize the GCN by the following optimization problem:

$$\begin{aligned} \min_{\mathbf{W}} \quad & \mathcal{L} := \|\mathbf{P}_l - \hat{\mathbf{P}}_l\|_F^2 \\ \text{s.t.} \quad & \hat{\mathbf{P}} = \text{GCN}_{\mathbf{W}}(\mathbf{A}, \hat{\mathbf{X}}). \end{aligned} \quad (6)$$

The objective function is the Mean Squared Error (MSE) between the true positions of anchors $\mathbf{P}_l = [\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_{N_l}]^\top$ and their respective estimations $\hat{\mathbf{P}}_l = [\hat{\mathbf{p}}_1, \hat{\mathbf{p}}_2, \dots, \hat{\mathbf{p}}_{N_l}]^\top$. Here, \mathbf{W} denotes all trainable matrices in the GCN model which can be optimized by gradient descent techniques, such as Stochastic Gradient Descent (SGD) [34] or Adam [35].

B. Key Findings of GCN

The numerical results, as detailed in Section VI, illustrate significant enhancements achieved by the GCN-based method compared to traditional benchmark methods. In the subsequent discussion, we identify key factors underpinning this method's outstanding performance, offering insights into its effectiveness for network localization and guiding future improvements. We pinpoint two major factors: 1) the threshold T_h and 2) the normalized adjacency matrix $\hat{\mathbf{A}}$.

1) *Effects of Threshold T_h* : An appropriate threshold serves two primary functions: a) truncating significant noise and b) preventing over-smoothing.

– *Noise Truncation*: Thresholding ensures that \mathbf{A} , the adjacency matrix, includes only edges a_{ij} where $n_{ij} \leq T_h - d_{ij}$, implying that $\hat{\mathbf{X}} = \mathbf{A} \odot \mathbf{X}$ retains distance measurements x_{ij} under the threshold T_h and accompanied by confined noise. For those measurements with the existence of NLOS noise, this condition can be further expressed as $n_{ij}^N \leq T_h - d_{ij} - n_{ij}^L$ with $n_{ij}^L \sim \mathcal{N}(0, \sigma^2)$. By applying an appropriate threshold, we not only constrain the distance but also truncate the noise, such that measurements are retained only if two nodes are adjacent and primarily affected by a small or moderate measurement noise.

– *Avoiding Over-smoothing*: In the extreme case with an excessively large threshold, rendering the graph fully connected with an all-ones adjacency matrix, all rows of the hidden representation matrix $\hat{\mathbf{H}}^{(k)}$ become indistinguishable. The loss of discriminative ability leads to the predicted positions of all nodes converging towards the same. This phenomenon is known as *over-smoothing*. Thus, selecting an appropriate threshold is crucial to prevent such detrimental effects.

To assess the threshold T_h 's impact on localization, we experimented across varying T_h values within three noise settings,

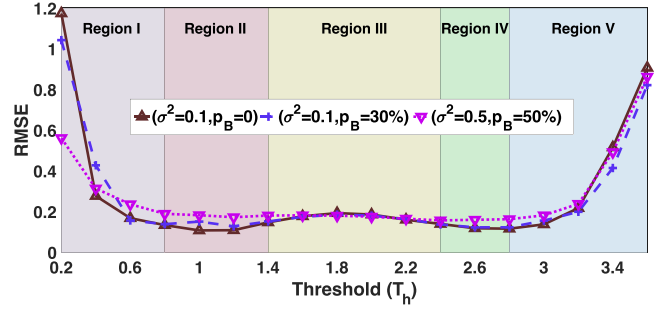


Fig. 2. The averaged loss (RMSE) v.s. threshold under different noise conditions in the GCN model.

illustrated in Fig. 2. The results delineate five performance regions corresponding to different threshold ranges. In Region I, initial performance issues stem from insufficient graph edges when T_h is set too low, leading to isolated nodes that hinder effective localization. Within Regions II to IV, a counterbalance is observed between the NLOS noise truncation effect (Region II) and the growing number of neighbors per node (Region IV). Region V exhibits a sharp increase in RMSE due to diminished noise filtering and severe over-smoothing issues. For detailed experimental settings and result descriptions, refer to our previous work [1].

2) *Effects of Normalized Adjacency Matrix $\hat{\mathbf{A}}$* : To comprehend the superior localization performance of GCN, we also analyze the effects of the normalized adjacency matrix, $\hat{\mathbf{A}}$, from both spatial and spectral perspectives.

– *Spatial Perspective. Aggregation and Combination*: To understand the spatial effect of $\hat{\mathbf{A}}$, we disassemble $\hat{\mathbf{A}}$, cf. Eq (4), into two components:

$$\bar{\mathbf{h}}_{[i,:]}^{(k)} = \underbrace{\frac{1}{d_i} \mathbf{h}_{[i,:]}^{(k)}}_{\text{Intrinsic information [i]}} + \underbrace{\sum_{j \in \mathcal{N}_i} \frac{a_{ij}}{\sqrt{d_i d_j}} \mathbf{h}_{[j,:]}^{(k)}}_{\text{Aggregated information [ii]}}. \quad (7)$$

This equation demonstrates the aggregation of neighboring features ((7)-[ii]) and their combination with a node's own features ((7)-[i]). As a result, in GCN, the propagated representation of each labeled node (i.e., anchor in the localization setting) is a weighted sum of its neighbors. Essentially, GCN leverages the attributes of both labeled and unlabeled nodes in its training process.

– *Spectral Perspective. Low-Pass Filtering*: From the spectral perspective, the feature propagation step in (7) can be rewritten as

$$\hat{\mathbf{A}} \mathbf{H}^{(k-1)} = (\mathbf{I} - \hat{\mathbf{L}}) \mathbf{H}^{(k-1)} \quad (8a)$$

$$= (\mathbf{I} - \mathbf{U} \hat{\mathbf{\Lambda}} \mathbf{U}^\top) \mathbf{H}^{(k-1)} \quad (8b)$$

$$= \mathbf{U} (\mathbf{I} - \hat{\mathbf{\Lambda}}) \mathbf{U}^\top \mathbf{H}^{(k-1)}, \quad (8c)$$

where $\hat{\mathbf{L}} := \mathbf{I} - \hat{\mathbf{A}}$ denotes the normalized Laplacian, \mathbf{U} and $\hat{\mathbf{\Lambda}}$ are the associated eigenvectors and eigenvalues. In terms of graph signal operations, $\tilde{\mathbf{H}}^{(k-1)} := \mathbf{U}^\top \mathbf{H}^{(k-1)}$ and $\mathbf{H}^{(k-1)} := \mathbf{U} \tilde{\mathbf{H}}^{(k-1)}$ correspond to the Graph Fourier Transform and Inverse Graph Fourier Transform, respectively. The vector \mathbf{u}_i

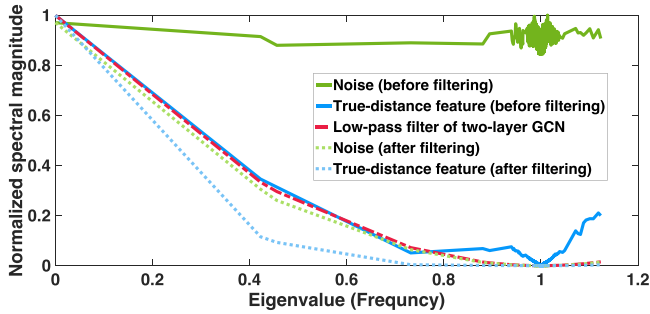


Fig. 3. Spectral components of the 10-th column vector (similar for other columns) of the noise, \mathbf{n}_{10} , and the true distance, \mathbf{d}_{10} , in dataset ($\sigma^2 = 0.1, p_B = 30\%$).

and the scalar $\hat{\lambda}_i$ represent the i -th graph Fourier basis and “frequency” component of the graph signal, respectively [36], [37]. According to (8c), each feature propagation operation acts as a spectral “low-pass” filter $g(\hat{\lambda}_i) := (1 - \hat{\lambda}_i)$. Consequently, a two-layer GCN can be approximated as $\varphi(\hat{\mathbf{A}}^2 \mathbf{X} \mathbf{W})$, which effectively applies the filter $(1 - \hat{\lambda}_i)^2$ [38]. This spectral filtering concept is essential for understanding how GCNs process and filter signal information across the network. Fig. 3 illustrates the “frequency” components of the noise (including LOS noise and NLOS bias) and the true distance matrix before and after filtering. Notably, in the unfiltered state, the majority of information within the true distance matrix is concentrated in the “low-frequency” band. Conversely, the noise displays both “low-frequency” and “high-frequency” components before filtering. Comparing the results before and after filtering, it becomes evident that the normalized adjacency matrix $\hat{\mathbf{A}}$, serving as a “low-pass” filter, effectively mitigates the “high-frequency” component of the LOS noise. From a spectral analysis standpoint, this “low-pass” filtering capability of $\hat{\mathbf{A}}$ justifies the improved performance of GCNs in localization tasks by filtering out the “high-frequency” noise component.

C. Limitations of Classic GCN

Despite the high suitability of the GCN-based method for network localization tasks demonstrated in the preceding discussion, the subsequent discussion reveals several key limitations of this method:

- *Optimal Threshold Challenge*: As shown in Fig. 2, the selection of threshold T_h plays a pivotal role in determining the localization performance. To achieve optimal results, T_h should be chosen within Region II or Region IV. However, as a hyperparameter, determining its optimal value for thresholding purpose is not straightforward, especially when applying the model to new scenarios with varying sizes and shapes. Furthermore, due to the distinct node positions and varying channel state information, each node may possess its unique optimal threshold, rather than a uniform threshold applicable to all nodes. Consequently, there arises a necessity to develop a more flexible model that is capable of autonomously learning the optimal threshold during the training process.

- *Limited Model Expressiveness*: The GCN-based model employs a constant aggregation weight (see (7)), determined

by adjacency and node degree, which constrains its capacity to adaptively aggregate and combine neighborhood information. From the spectral perspective, the GCN-based model represents a fixed graph convolutional filter, as delineated by the green dash-dotted line in Fig. 3. This fixed filtering characteristic (fixed adjacency matrix) compromises its versatility, necessitating the design of customized filters tailored to network noise characteristics at each graph convolutional layer.

In the following, we turn our focus to offering effective solutions to mitigate the aforementioned issues.

IV. NETWORK LOCALIZATION WITH AGNN

In this section, we first review the attention mechanism used in recent works and its challenges in network localization. Then, we design a fresh Attentional GNN (AGNN) model that can autonomously determine the optimal threshold and adaptively determine the aggregation weight of each node for various network localization tasks.

A. Attention Mechanism for Network Localization

Motivated by recent advancements in attention-based graph neural networks, such as Graph Attention network (GAT) [4] and refined version of GAT (GATv2) [39], we incorporate the attention mechanism into the network localization task, enabling the data-driven model to autonomously determine an optimal neighbor set for each node and acquire flexible aggregation weights for all pairs of nodes based on the so-called attention scores.

Graph attention architectures typically consist of a sequence of graph attention layers [4]. To elucidate, we start by explaining the basic structure of a single graph attention layer, which can be represented as:

$$e_{ij}^{(k)} = \text{Att} \left(\mathbf{h}_{[i,:]}^{(k)}, \mathbf{h}_{[j,:]}^{(k)} \right). \quad (9)$$

Herein, at the k -th layer, a shared attentional mechanism, denoted as $\text{Att}(\cdot, \cdot) : \mathbb{R}^{D_k} \times \mathbb{R}^{D_k} \rightarrow \mathbb{R}$, calculates an attention score $e_{ij}^{(k-1)}$ for each node pair i - j , reflecting the relative importance of node j 's features to those of node i . Essentially, (9) enables each node to determine the importance of all its neighbors through attention scores.

However, directly applying this mechanism to network localization tasks encounters significant hurdles. Specifically, in such tasks when a predefined threshold T_h is absent, i.e., the graph structure is not predetermined, it's common to assume a complete graph for deploying graph attention layers. As a result, the attention mechanism calculates scores for every node pair, leading to two main challenges:

- *High Computational Complexity*: As the number of nodes increases, the number of possible node pairs grows quadratically. Therefore, computing attention scores for all possible node pairs is computationally expensive. This issue can significantly slow down the training of the attention mechanism, especially when dealing with the localization of a massive network.

- *Over-smoothing Issues*: Similar to the analysis provided in Section III-B, in a complete graph, aggregating features from

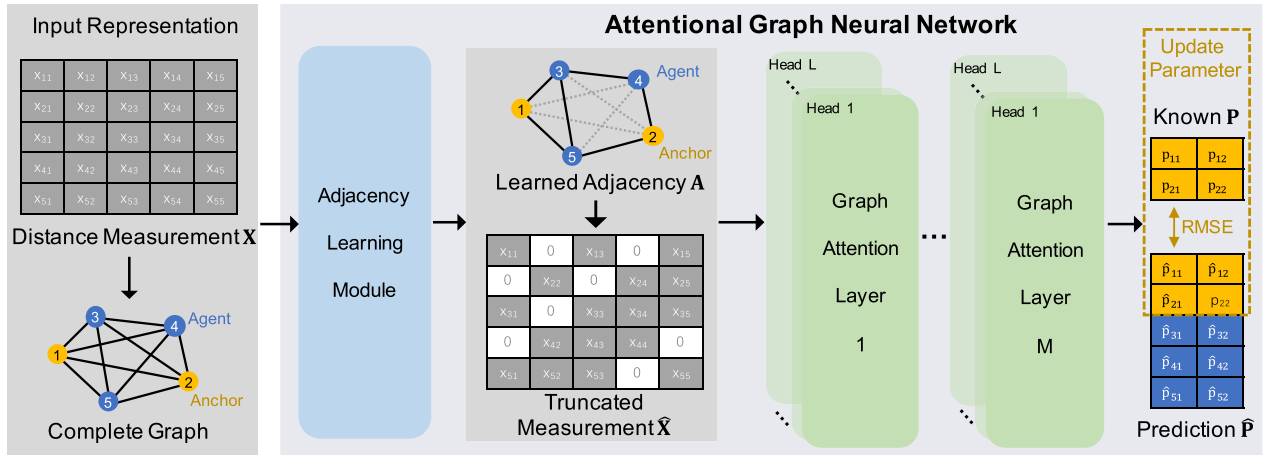


Fig. 4. Framework of AGNN.

all nodes causes over-smoothing. Even with varied attention scores for edges, this mechanism doesn't effectively mitigate over-smoothing, especially reducing the model's expressiveness exponentially as model depth increases [40].

To tackle these challenges, we introduce a novel AGNN model tailored for network localization tasks in the subsequent sections. This model is explicitly crafted to autonomously acquire graph structures during the training process, making it highly suitable for network localization tasks across diverse network scenarios.

In addition to our proposed node-level attention-based approach, several other strategies have been explored to mitigate the over-smoothing problem. One such approach is the use of neighbor sampling [41], [42], which reduces the receptive field of each node, thereby preventing the model from overly depending on any single node and maintaining feature diversity across nodes. Another promising method is the incorporation of residual connections, which has been demonstrated to alleviate over-smoothing by preserving information flow across layers of the model [43]. While these methods have been proven effective in various contexts, we argue that the AGNN model introduced here, with its ability to learn graph structures dynamically during training, provides a tailored solution for network localization, where the graph structure is not predefined.

B. Overview of AGNN Model

We propose the AGNN model tailored for network localization tasks, which is composed of an Adjacency Learning Module (ALM) and Multiple Graph Attention Layers (MGALs), as illustrated in Fig. 4. The ALM leverages an attention mechanism to learn a distance-aware threshold. In other words, nodes can adaptively adjust the threshold based on the distance to their neighbors, enabling more flexible neighbor selection. Subsequently, the MGALs utilize another attention-based mechanism to acquire adaptable aggregation weights based on the adjacency matrix learned by ALM.

The ALM autonomously determines the neighbor set for each node based on the attention mechanism. This approach is able to learn an optimal adjacency matrix selectively, thereby significantly reducing computational complexity and effectively addressing the over-smoothing issue associated with the direct

utilization of the graph attention mechanism in a complete graph. Next, we delve into the details of the ALM.

C. Adjacency Learning Module (ALM)

To adaptively learn the optimal threshold during the training process, we propose an ALM, which incorporates an attention mechanism to learn a specific threshold T_{ij}^A for each node pair. Nevertheless, learning T_{ij}^A directly for each pair without prior graph structure knowledge still poses computational challenges. To mitigate this, ALM employs a two-step process: initially, a coarse-grained neighbor selection is conducted using a manually-set threshold, followed by a fine-grained neighbor refinement through an attention mechanism. Details of the two stages are as follows.

– *Stage I. Coarse-grained Neighbor Selection:* To address the challenge of computing attention scores without prior knowledge of the graph structure, we initially utilize a preselected threshold, T_h^0 , to perform a coarse-grained neighbor selection for each node, following the procedure in (3). This initial threshold can be manually set or optimized by a process detailed in Appendix A, thereby eliminating the reliance on manual selection. Notably, unlike the threshold in the GCN-based model which can significantly influence the localization accuracy, this initial threshold T_h^0 can be chosen from a broader range, as our objective here is to establish a coarse-grained neighbor set, \mathcal{N}_i^C , for each node i . Verification of this assertion can be found in Fig. 7 in Section VI.

– *Stage II. Fine-grained Neighbor Refinement:* To refine coarse-grained neighbor sets into fine-grained ones, we employ a distance-aware threshold matrix, which is calculated using attention scores. This matrix is employed to further refine the coarse-grained neighbor sets by comparison with the measured distance matrix.

Specifically, we incorporate the masked attention mechanism [27] into the coarse-grained graph structure. This means that we compute the attention scores e_{ij}^A only for nodes j within the coarse-grained neighbor set \mathcal{N}_i^C of node i . Our designed attention scores e_{ij}^A are calculated as follows:

$$e_{ij}^A = |\phi(\mathbf{x}_{[i,:]} \mathbf{W}_A) - \phi(\mathbf{x}_{[j,:]} \mathbf{W}_A)|^p \mathbf{v}_A, j \in \mathcal{N}_i^C. \quad (10)$$

Here, $\mathbf{x}_{[i,:]}$ represents the original measured distance row vector. The attention mechanism in this context is parametrized by an attention weight vector $\mathbf{v}_A \in \mathbb{R}^{F_A}$ and an attention weight matrix $\mathbf{W}_A \in \mathbb{R}^{N \times F_A}$, and applies a nonlinear mapping $\phi(\cdot)$, such as LeakyReLU. The reasons for using (10) to compute the attention scores, as opposed to a single-layer neural network used in GAT/GATv2, are as follows:

- Passing each node's feature through a linear transformation with \mathbf{W}_A followed by a nonlinear activation function enhances the expressive power of feature transformations;
- Taking the p -th power of the absolute difference between two transformed representations can be regarded as a learnable distance metric between their representations, which possesses the symmetry property that is not considered by GAT and GATv2 but is essential for network localization;
- The attention vector \mathbf{v}_A learns an effective dimensionality reduction mapping to project the learnable distance metric into a scalar, serving as the output attention score.

In our implementation, we set $p = 1$ as it provides the optimal tradeoff between threshold selection flexibility and denoising capability. The rationale and supporting empirical evidence are presented in Section VI-B.

To convert the attention scores into thresholds, which are comparable to the measured distances, we apply nonlinear scaling through:

$$T_{ij}^A = \max(\mathbf{x}_{[i,:]} \cdot \text{Sigmoid}(e_{ij}^A)), \quad (11)$$

where $\max(\mathbf{x}_{[i,:]})$ identifies the maximum value from the row vector $\mathbf{x}_{[i,:]}$. Using (11), we obtain a distance-aware threshold matrix denoted as $\mathbf{T}^A \in \mathbb{R}^{N \times N}$. With the distance-aware thresholds obtained within each coarse-grained neighbor set, we establish a criterion for selecting fine-grained neighbors by comparing the measured distance with the thresholds. Specifically, the adjacency can be represented as follows:

$$a_{ij} = \begin{cases} 1, & \text{if } x_{ij} < T_{ij}^A, \\ 0, & \text{otherwise.} \end{cases} \quad (12)$$

Nevertheless, (12), formulated as a step function, lacks differentiability concerning \mathbf{T}^A , which contains the parameters \mathbf{W}_A and \mathbf{v}_A that we intend to train using gradient-based optimization methods. Typically, conventional approaches often employ a Sigmoid function as an approximation for the step function. However, for our specific objective of achieving a sparse adjacency matrix by eliminating elements exceeding the threshold, this approach is inadequate as the Sigmoid function's transition at the threshold point is too smooth to effectively produce a sparse adjacency matrix. Consequently, it is essential to identify an approximation for (12) that is both differentiable concerning the trainable parameters and capable of filtering out values that exceed the threshold.

Based on the above requirements, we develop an appropriate approximated step function that addresses this issue. The approximated step function can be defined as follows:

$$\hat{\delta}(x) = \text{ReLU}(\tanh(\gamma x)), \quad (13)$$

where γ is a hyperparameter for adjusting the steepness of the hyperbolic tangent (tanh) function. Our devised approximated step function, employing ReLU and tanh, yields a notably sharper transition on the positive side and effectively truncates negative values to zero, demonstrating a more accurate approximation of the step function than the Sigmoid function. More importantly, its derivative can be obtained everywhere with respect to the trainable threshold. For more details, see Appendix F. Subsequently, the approximation of (12) can be expressed as:

$$a_{ij} = \text{ReLU}(-\tanh(\gamma(x_{ij} - T_{ij}^A))). \quad (14)$$

After obtaining the adjacency matrix \mathbf{A} via the attentional method, we construct the truncated measurement matrix $\hat{\mathbf{X}} = \mathbf{A} \odot \mathbf{X}$. Subsequently, they are effectively used as inputs to feed the graph attention layer introduced below.

D. Multiple Graph Attention Layers (MGALs)

We assume that the adjacency matrix \mathbf{A} has been learned via ALM. The fine-grained neighbor set of each node, denoted as \mathcal{N}_i^F for node i , is induced by \mathbf{A} . Subsequently, we introduce MGALs, a more generalized form of the graph attention mechanism inspired by GATv2 [39], to learn the aggregation weight and predict the positions of all nodes.

We begin by explicating the k -th individual graph attention layer, which functions as the foundational layer uniformly employed throughout the entire framework of MGALs. The node representation at the k -th layer is denoted as $\mathbf{H}^{(k)} \in \mathbb{R}^{N \times D_k}$, where D_k is the hidden dimension of the k -th layer. The initial node representations are set as the truncated measurement matrix: $\mathbf{H}^{(0)} = \hat{\mathbf{X}}$. Here, we employ a single-layer feed-forward neural network, parametrized by an attention weight vector $\mathbf{v}_{att}^{(k)} \in \mathbb{R}^{F_{att}^k}$ and an attention weight matrix $\mathbf{W}_{att}^{(k)} \in \mathbb{R}^{2D_{k+1} \times F_{att}^k}$, as the attention mechanism $\text{Att}(\cdot, \cdot)$. Then, the computation of the attention scores using (9) can be expressed as:

$$e_{ij}^{(k)} = \phi \left(\left[\hat{\mathbf{h}}_{[i,:]}^{(k)} \parallel \hat{\mathbf{h}}_{[j,:]}^{(k)} \right] \mathbf{W}_{att}^{(k)} \mathbf{v}_{att}^{(k)}, \quad j \in \mathcal{N}_i^F, \quad (15)$$

where

$$\hat{\mathbf{h}}_{[i,:]}^{(k)} = \mathbf{h}_{[i,:]}^{(k)} \mathbf{W}^{(k)}, \quad \hat{\mathbf{h}}_{[j,:]}^{(k)} = \mathbf{h}_{[j,:]}^{(k)} \mathbf{W}^{(k)}. \quad (16)$$

Here, $\mathbf{W}^{(k)} \in \mathbb{R}^{D_k \times D_{k+1}}$ denotes the weight matrix in the k -th graph attention layer.

Notably, we use two distinct learnable weight matrices, $\mathbf{W}^{(k)}$ and $\mathbf{W}_{att}^{(k)}$, with distinct functionalities in our approach. Concretely, $\mathbf{W}^{(k)}$ in (15) functions as a linear transformation, converting input representations into higher-level representations. In contrast, $\mathbf{W}_{att}^{(k)}$ collaborates with the weight vector $\mathbf{v}_{att}^{(k)}$ to establish an attentive rule, determining the correlation between any selected pair of nodes. Moreover, the proposed graph attention layer represents a more generalized form than GATv2 and can be reduced to GATv2 when $F_{att}^k = 2D_{k+1}$ and $\mathbf{W}_{att}^{(k)}$ assumes an identity matrix.

To make the attention scores comparable across different nodes, we normalize them across all choices of j using the

softmax function:

$$\alpha_{ij}^{(k)} = \frac{\exp\left(e_{ij}^{(k)}\right)}{\sum_{j' \in \mathcal{N}_i^F} \exp\left(e_{ij'}^{(k)}\right)}. \quad (17)$$

Once obtained, the normalized attention scores $\alpha_{ij}^{(k)}$ are used to update $\mathbf{h}_{[i,:]}^{(k+1)}$ with nonlinearity $\varphi(\cdot)$ as:

$$\mathbf{h}_{[i,:]}^{(k+1)} = \varphi\left(\sum_{j \in \mathcal{N}_i^F} \alpha_{ij}^{(k)} \hat{\mathbf{h}}_{[j,:]}^{(k)}\right). \quad (18)$$

Analogous to the feature updating mechanism in each graph convolutional layer of the GCN-based method, (18) can be decomposed into *feature propagation*, *linear transformation* and *nonlinear activation*. The primary distinction lies in the *feature propagation*: the GCN-based method employs a constant aggregation weight, whereas MAGL utilize learned aggregation weights through the attention mechanism.

In summary, we have proposed an ALM capable of autonomously learning optimal thresholds for individual edges through an attention mechanism. This ALM further leads to an adaptive adjacency matrix and the corresponding truncated measurement matrix. Subsequently, MGALs leverage these learned matrices and employ another attention mechanism to acquire adaptable aggregation weights, enhancing the expressiveness of the model. As a result, it effectively addresses the limitations of the classic GCN, as discussed in Section III-C.

Finally, we formulate the optimization problem for our proposed AGNN as follows:

$$\begin{aligned} \min_{\mathbf{W}_1, \mathbf{W}_2} \quad & \mathcal{L} = \|\mathbf{P}_l - \hat{\mathbf{P}}_l\|_F^2 \\ \text{s.t.} \quad & \hat{\mathbf{P}} = \text{MGAL}_{\mathbf{W}_2}(\mathbf{A}, \hat{\mathbf{X}}) \\ & \mathbf{A}, \hat{\mathbf{X}} = \text{ALM}_{\mathbf{W}_1}(\mathbf{X}, T_h^0). \end{aligned} \quad (19)$$

Here, \mathbf{W}_1 and \mathbf{W}_2 denote all trainable matrices in ALM and MGALs, respectively. Similarly to (6), this optimization problem can be addressed through the application of gradient descent techniques [34], [35].

V. PERFORMANCE ANALYSIS

In this section, we analyze our proposed AGNN model from the perspectives of two fundamental properties: dynamic attention property and complexity analysis. Through theoretical analysis, we aim to elucidate the reasons behind the superior performance of our AGNN in terms of model representational capacity and computational complexity.

A. Dynamic Attention Property

In general, attention serves as a mechanism to compute a distribution over a set of input key vectors when provided with an additional query vector. In the context of network localization, the query vector and input key vectors represent the feature

vectors of a target node and its neighbors,² respectively. The attention mechanism in network localization involves the process of learning how to allocate attention weights to all neighboring nodes concerning the target node. In this subsection, we have demonstrated that, in contrast to the classical GAT model [4] which is confined to computing static attention and thereby encounters significant limitations, our proposed graph attention mechanism can achieve dynamic attention [39].

Static attention [39], characterized by consistently assigning the highest weight to the same neighboring node for all target nodes, as visualized in Fig. 11(a). It exhibits limited flexibility and is less effective in network localization scenarios, as it uniformly prioritizes a single node, ignoring relative distance and NLOS conditions. A more reasonable form of attention for network localization is the so-called dynamic attention [39]. A dynamic attention function can assign the highest weight to the most relevant neighboring node based on each target node's surrounding physical constraints.

In the classical GAT model [4], a scoring function $e(\mathbf{h}_i, \mathbf{h}_j)$ is employed to compute a score for every edge (j, i) , which indicates the importance of neighbor j 's feature to the node i :

$$e(\mathbf{h}_{[i,:]}, \mathbf{h}_{[j,:]}) = \phi\left([\mathbf{h}_{[i,:]} \mathbf{W} \parallel \mathbf{h}_{[j,:]} \mathbf{W}] \mathbf{v}\right). \quad (20)$$

Theorem 1: A GAT layer computes only static attention, for any set of node representations $\mathcal{K} = \mathcal{Q} = \{\mathbf{h}_{[1,:]}, \dots, \mathbf{h}_{[N,:]}\}$. Specifically, for every function $e(\cdot, \cdot)$, there exists a "highest scoring" key $j_{m1} \in [N]$ such that for every query $i \in [N]$ and key $j \in [N]$, it holds that $e(\mathbf{h}_{[i,:]}, \mathbf{h}_{[j_{m1},:]}) \geq e(\mathbf{h}_{[i,:]}, \mathbf{h}_{[j,:]})$.

Proof: See Appendix B or refer to [39]. ■

Notably, based on the above proof, we can further conclude that in static attention, not only is the key with the highest weight fixed to j_{m1} , but the relative order of attention weights for all keys is also fixed, independent of the query. This phenomenon is also illustrated in Fig. 11(a). The primary issue with the scoring function given in (20) for the standard GAT is that the learned weights, \mathbf{W} and \mathbf{v} , are applied in succession, which allows them to be condensed into a single linear layer.

Compared with the standard GAT, each graph attention layer (15) in MGALs simply applies the attention weight matrix \mathbf{W}_{att} after the concatenation, and the attention weight vector \mathbf{v}_{att} after the nonlinearity $\phi(\cdot)$. Next, we show that our proposed graph attention layer addresses the limitation of static attention in GAT and possesses a significantly more expressive dynamic attention property.

Theorem 2: The graph attention function in (15), (16) computes dynamic attention for any set of node representations $\mathcal{K} = \mathcal{Q} = \{\mathbf{h}_{[1,:]}, \dots, \mathbf{h}_{[N,:]}\}$. Specifically, for any mapping $\varphi: [N] \rightarrow [N]$, there exists a function $e(\cdot, \cdot)$ such that for every query $i \in [N]$ and key $j_{\neq \varphi(i)} \in [N]$, the property $e(\mathbf{h}_{[i,:]}, \mathbf{h}_{[\varphi(i),:]}) \geq e(\mathbf{h}_{[i,:]}, \mathbf{h}_{[j_{\neq \varphi(i)},:]})$ holds.

Proof: See Appendix C. ■

The core idea in this proof centers around the principle that the graph attention function in (15), (16) can be interpreted as

²For clarity, we assume a complete graph (fully connected graph) in the subsequent discussions. Analogous results can be extended to the case of an incomplete graph, where the neighbor set of each node becomes a subset of the corresponding set in a complete graph.

a single-layer graph neural network. Consequently, it can be a universal approximator [44] of an appropriate function we define, thereby achieving dynamic attention.

In addition, within the ALM, we have devised a graph attention function, as given in (10), with the specific goal of implementing a distance-aware attention mechanism. In the sequel, we show that this function also possesses the dynamic attention property, a critical aspect contributing to its efficacy in refining neighbors at a fine-grained level.

Theorem 3: The graph attention function in (10) computes dynamic attention for any set of node representations $\mathcal{K} = \mathcal{Q} = \{\mathbf{x}_{[1,:]}, \dots, \mathbf{x}_{[N,:]}\}$.

Proof: The expression for the attention coefficient in ALM, as defined in (10), can be reformulated as follows:

$$e_{ij}^A = \sum_{k=1}^{F_A} u_{ij}^k, \forall j \in \mathcal{N}_i^C, \quad (21)$$

where

$$u_{ij}^k = |\phi(\mathbf{x}_{[i,:]}[\mathbf{W}_A]_k) - \phi(\mathbf{x}_{[j,:]}[\mathbf{W}_A]_k)|^p \cdot [\mathbf{v}_A]_k. \quad (22)$$

Here, the notation $[\cdot]_k$ represents the k -th column vector or the k -th element of a matrix or vector.

Based on the decomposition of the summation in (21), we focus on the k -th term, denoted as u_{ij}^k . Let $c_j^k := \phi(\mathbf{x}_{[j,:]}[\mathbf{W}_A]_k)$, and then sort $c_j^k, \forall j \in \mathcal{N}_i^C$ in ascending order to obtain the corresponding indices of the node sorting, denoted as $\{\dot{j}_{min}, \dots, \dot{j}_{l_1}, \dot{j}, \dot{j}_{r_1}, \dots, \dot{j}_{max}\}$. Assuming the scalar $[\mathbf{v}_A]_k$ is positive, then for every $j \in \mathcal{N}_i^C$, we have:

$$u_{ij}^k \leq \max \left\{ |c_i^k - c_{\dot{j}_{max}}^k|^p, |c_i^k - c_{\dot{j}_{min}}^k|^p \right\}. \quad (23)$$

If the scalar $[\mathbf{v}_A]_k$ is negative, then for every $j \in \mathcal{N}_i^C$, we have:

$$u_{ij}^k \leq \max \left\{ |c_i^k - c_{\dot{j}_{l_1}}^k|^p, |c_i^k - c_{\dot{j}_{r_1}}^k|^p \right\}. \quad (24)$$

It is evident that the maximum value of u_{ij}^k is not fixed but rather depends on \mathcal{N}^C and $[\mathbf{v}_A]_k$. Thus, u_{ij}^k , as the attention coefficient in the summation of e_{ij}^A for the k -th term, exhibits a dynamic attention property.

Similar conclusions can be drawn for the remaining $F_A - 1$ terms, wherein the difference arises due to the varying orders of $c_j^k, \forall j \in \mathcal{N}_i^C$ determined by the specific values of $[\mathbf{W}_A]_k$ and $[\mathbf{v}_A]_k$, which further ensures the dynamic attention property. ■

This proof decomposes (10) into a summation form and focuses on the k -th term. It shows that the maximum value of this term is not fixed but depends on the set \mathcal{N}^C and the k -th element of the attention weight vector \mathbf{v}_A , denoted as $[\mathbf{v}_A]_k$. By sorting node representations and considering different cases based on the sign of $[\mathbf{v}_A]_k$, the dynamic attention property is established for each term in the summation, thereby ensuring the dynamic attention property for the entire function.

Overall, our proposed graph attention mechanisms in ALM and MGALs inherit the dynamic attention property, making them inherently more expressive than the standard GAT. In the context of network localization, this property ensures the optimal selection of threshold and aggregation weights for node pairs

based on their features, effectively addressing the limitations in Section III-C.

B. Complexity Analysis

From a computational perspective, our proposed AGNN exhibits high efficiency. The operations of the graph attention layer in MGALs can be parallelized across all edges, and the computations of output features can similarly be parallelized across all nodes. Notably, these models obviate the need for computationally expensive matrix operations, such as eigen-decompositions, which are typically mandatory in spectral-based methods [45], [46].

We provide the time complexity analysis for GCN-based localization models as follows:

Theorem 4: For the k -th graph convolutional layer, the time complexity can be summarized as $O(ND_{k-1}D_k + |E|D_{k-1})$, where $|E|$ represents the number of graph edges constructed based on a given threshold.

Proof: The complexity analysis of the k -th graph convolutional layer can be segmented into two components: (4) and (5).

First, we consider the operation described in (4), which entails a matrix product involving a sparse adjacency matrix and a dense representation matrix. Specifically, the i -th row of $\bar{\mathbf{H}}^{(k)}$ can be expressed as follows:

$$\bar{\mathbf{h}}_{[i,:]} = \hat{\mathbf{a}}_{[i,:]} \mathbf{H}^{(k-1)} = \sum_{j=1}^N \hat{a}_{ij} \mathbf{h}_{[j,:]}^{(k-1)} = \sum_{j \in \mathcal{N}_i} \hat{a}_{ij} \mathbf{h}_{[j,:]}^{(k-1)}. \quad (25)$$

Here, the computation of $\bar{\mathbf{h}}_{[i,:]}$ demands $O(|\mathcal{N}_i|D_{k-1})$ in terms of time complexity. Given that $2|E| = \sum_{i=1}^N |\mathcal{N}_i|$, the overall time complexity for (4) becomes $O(|E|D_{k-1})$.

Secondly, for the matrix multiplication between two dense matrices, as seen in (5), the well-known time complexity is $O(ND_{k-1}D_k)$.

In summary, the time complexity of the k -th graph convolutional layer can be succinctly expressed as $O(ND_{k-1}D_k + |E|D_{k-1})$. ■

Crucially, the time complexity for our proposed AGNN is presented as follows:

Theorem 5: The time complexity for ALM is $O(NNF_A + |E^C|F_A)$. For the k -th graph attention layer of MGALs, the time complexity can be summarized as $O(ND_kF_{att} + ND_{k-1}D_k + |E^F|F_{att} + |E^F|D_k)$, where $|E^C|$ and $|E^F|$ denote the numbers of edges in the coarse-grained and fine-grained neighbor sets, respectively.

Proof: See Appendix D. ■

If we make the assumptions that F_{att} , D_{k-1} , D_k , and N are approximately equal and that $|E|$, $|E^C|$, and $|E^F|$ are also roughly equivalent, then the time complexity of our proposed AGNN is comparable to that of the baseline GCN-based model. It's worth noting that when applying multi-head attention, the storage and parameter requirements increase by a factor of K , but the computations of individual heads are entirely independent and can be effectively parallelized.

TABLE II
THE SIMULATED LOCALIZATION SCENARIOS AND COMPUTER
IMPLEMENTATION DETAILS

| Scenarios | Size: 5m×5m # Nodes (# Anchors): 500 (20-160) | | | | |
|---------------|--|-------------|--------|---------------|---------|
| Configuration | 48 Inter Xeon E5-2650 2.2GHz CPUs 8 NVIDIA TITAN Xp 12GB GPUs | | | | |
| Model | Layers | Hidden size | Epochs | Learning rate | Dropout |
| | 2 | 2000 | 200 | 0.01 | 0.5 |
| Threshold | $T_h = 1.2$ for GCN and MLP $T_h^0 = 3.0$ for ALM $T_h = 0.6$ for benchmarks | | | | |

VI. NUMERICAL RESULTS

In this section, we evaluate the performance of several proposed GNN-based methods in terms of their localization accuracy, robustness against NLOS noise, and computational time. For benchmarks, we employ several classic statistical methods including the sparsity-inducing Semi-Definite Programming (SDP) method [23], the Expectation-Conditional Maximization (ECM) method [21], the Multi-Dimensional Scaling (MDS) method [14], the centralized Least-Squares (LS) method [13]. We further include the recently proposed SMILE model [47], which leverages low-rank matrix decomposition for network localization. To complement these comparisons, we evaluate our methods against well-established data-driven baselines, specifically MLP, GraphSAGE [3], GAT [4], and GATv2 [39]. Note that the purpose of incorporating the MLP-based method, which exclusively employs a *linear transformation* and *nonlinear activation* in each layer, into our comparison is to illustrate the performance enhancements attributed to the inclusion of graph structure in each GNN layer. Since the adjacency matrix is not directly available in network localization, all GNN-based baselines utilize the same threshold-based adjacency matrix construction strategy as the GCN-based methods.

Simulated localization scenarios and the computer implementation details are shown in Table II. Here, the measurement error, n_{ij} , is generated according to (2). The positive NLOS bias, n_{ij}^N , is generated from a uniform distribution³, $n_{ij}^N \sim \mathcal{U}[0, 10]$. The localization accuracy is measured in terms of the RMSE between estimated positions and true positions of agent nodes, $\mathcal{L}_P := \|\mathbf{P}_u - \hat{\mathbf{P}}_u\|_F$, where $\mathbf{P}_u = [\mathbf{p}_{N_l+1}, \mathbf{p}_{N_l+2}, \dots, \mathbf{p}_N]^\top$ and $\hat{\mathbf{P}}_u = [\hat{\mathbf{p}}_{N_l+1}, \hat{\mathbf{p}}_{N_l+2}, \dots, \hat{\mathbf{p}}_N]^\top$.

Next, we will evaluate the effectiveness of our proposed AGNN, focusing on two primary aspects: its overall superiority concerning accuracy, robustness, and efficiency; and the insights into how the ALM, MGALs, and dynamic attention property contribute to AGNN's exceptional performance.

A. Superiority of AGNN

– *Overall Performance Analysis:* Primarily, we conduct a comprehensive evaluation of the localization accuracy across diverse noise conditions and present the results in Table III. Our results reveal consistent superiority of the GNN-based methods (including GCN [1], GraphSage, GAT, GATv2, and

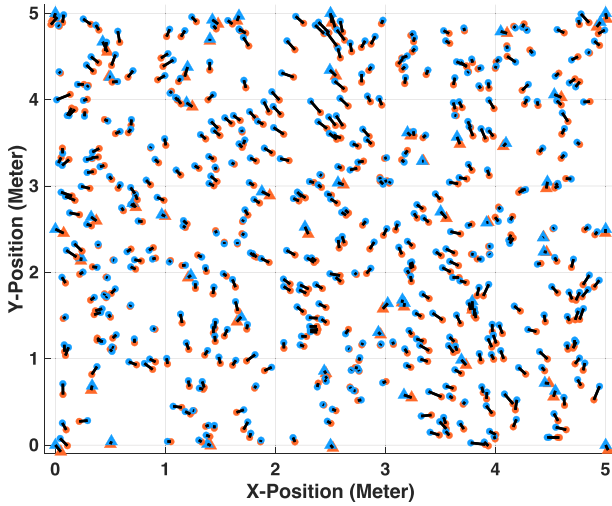
³Additional types of NLOS settings and corresponding results can be found in the Appendix E.

TABLE III
THE AVERAGED LOSS (RMSE) OF ALL METHODS UNDER DIFFERENT NOISE
CONDITIONS FOR $N_l = 50$

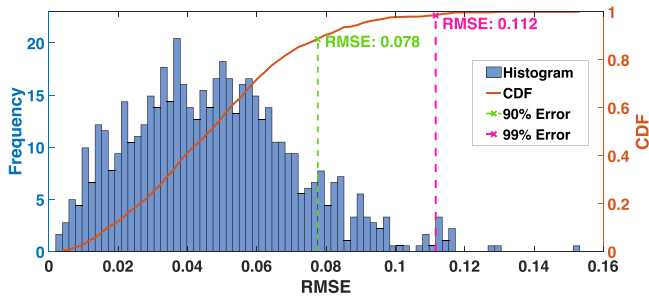
| Methods | Noise (σ^2, p_B) | | | | |
|-----------------------|---------------------------|---------------|---------------|---------------|---------------|
| | (0.04, 0%) | (0.10, 10%) | (0.25, 10%) | (0.25, 30%) | (0.50, 50%) |
| LS [13] | 0.2270 | 0.2675 | 0.3884 | 0.4187 | 0.7992 |
| MDS [14] | 0.2361 | 0.2394 | 0.2822 | 0.7593 | 1.1940 |
| ECM [21] | 0.1610 | 0.1857 | 0.3298 | 0.3824 | 0.8011 |
| SDP [23] | 0.1171 | 0.2599 | 0.4891 | 0.4641 | 0.9294 |
| SMILE [47] | 0.3235 | 0.3346 | 0.3797 | 0.5755 | 0.7467 |
| MLP | 0.1865 | 0.1769 | 0.2305 | 0.2623 | 0.3358 |
| GraphSage [3] | 0.1008 | 0.1066 | 0.1124 | 0.1276 | 0.1598 |
| GAT [4] | 0.1194 | 0.1178 | 0.1215 | 0.1423 | 0.1783 |
| GATv2 [39] | 0.0899 | 0.0942 | 0.0983 | 0.1107 | 0.1471 |
| GCN [1] | 0.1038 | 0.1128 | 0.1006 | 0.1302 | 0.1755 |
| GCN ₁₀₀₀₀ | 0.0771↓ | 0.0806↓ | 0.0932↓ | 0.0951↓ | 0.1217↓ |
| AGNN | 0.0486 | 0.0551 | 0.0638 | 0.0812 | 0.1015 |
| AGNN ₁₀₀₀₀ | 0.0378↓ | 0.0489↓ | 0.0585↓ | 0.0784↓ | 0.0960↓ |
| CRB | 0.0202 | 0.0344 | 0.0453 | 0.0635 | 0.0941 |

newly proposed AGNN) compared to all other benchmarks. Notably, their performance in high NLOS environments underscores the robustness and effectiveness of GNN-based localization approaches. Among the evaluated methods, AGNN achieves the best overall performance across all noise conditions, which highlights the effectiveness of attention mechanisms in learning both adjacency and propagation weights. For better visualization, the predicted positions obtained from the AGNN are depicted in Fig. 5(a) along with the true positions. Additionally, the histogram and Cumulative Distribution Function (CDF) of agents' RMSE for the corresponding scenario is provided in Fig. 5(b). To validate the effectiveness of our proposed GNN-based methods in massive networks, we test the localization performance of GCN and AGNN on networks with $N = 10000$, denoted by GCN₁₀₀₀₀ and AGNN₁₀₀₀₀, respectively. The results show that both GCN and AGNN perform better, corroborating the positive role played by agents as neighbors in graph aggregation and combination processes. Ultimately, the AGNN exhibits a minimal performance deviation in comparison to the Cramér-Rao Bound (CRB) [21], detailed in Appendix E. This observation attests to the AGNN's capability to approach the optimal theoretical performance limits.

– *Localization Accuracy Analysis:* Subsequently, we focus on the performance of our proposed GNN-based methods. We explore localization error by varying N_l from 20 to 180 with a stepsize of 20 under two distinct noise conditions, and present the results in Fig. 6. Three key observations can be drawn from these results. 1) The AGNN consistently achieves significantly lower RMSE for all N_l values compared to the GCN-based method. Notably, even under a 30% NLOS noise condition, AGNN consistently outperforms GCN in the scenarios influenced solely by LOS noise. This underscores the robustness of AGNN to NLOS noise, attributed to learned thresholds and aggregation weights through the attention mechanism. 2) As N_l increases, both AGNN and GCN progressively approach their respective performance limits, indicating the potential of GNN-based methods, employing semi-supervised learning, to



(a) Scatter plot of predicted and true positions. Herein, blue nodes represent the true positions, while orange nodes indicate the predicted positions. Large triangles denote anchor nodes, whereas small circles represent agent nodes. Additionally, black lines depict localization errors between the true positions and those predicted by AGNN.



(b) Histogram and CDF of agents' RMSE.

Fig. 5. Visualization of AGNN localization performance with $(\sigma^2 = 0.1, p_B = 0\%)$ and $N_I = 50$.

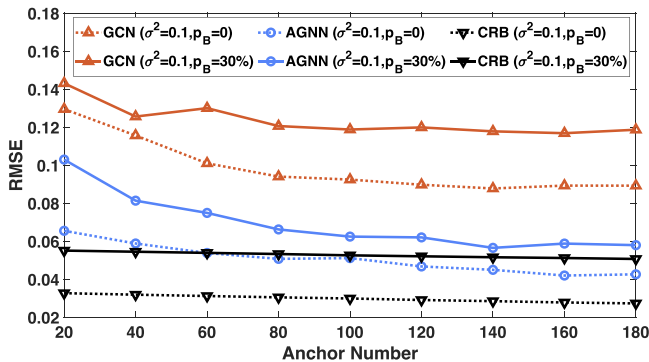


Fig. 6. The averaged loss (RMSE) v.s. the number of anchors under different noise conditions.

achieve accurate localization in massive networks with a limited number of anchors. 3) AGNN's performance limits exhibit a marginal gap compared to the CRB, particularly in NLOS noise conditions. This gap arises as the CRB represents the optimal scenario among unbiased methods, assuming complete

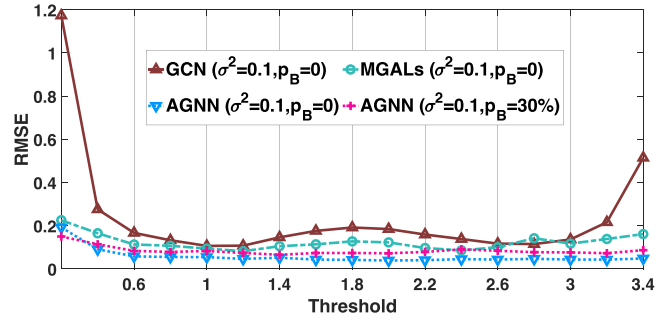


Fig. 7. The averaged loss (RMSE) v.s. the threshold (T_h or T_h^0) under different noise conditions.

knowledge of noise distributions and full utilization of node distance information. In contrast, AGNN, while mitigating NLOS noise during the truncation of the measurement matrix \mathbf{X} , excludes certain distance information, resulting in the observed gap. The reduction of the gap in NLOS scenarios, attributed to AGNN's noise truncation effect, provides additional validation for this observation. Consequently, these findings underscore AGNN's suitability for scenarios influenced by NLOS noise.

– *Robustness to Threshold Variation:* To comprehensively investigate the impact of the threshold, T_h , we conduct an experiment systematically exploring the RMSE of GNN-based models across a range of T_h , as depicted in Fig. 7. Here, “MGALs” denotes a model constructed solely using two graph attention layers from Section IV-D, and subject to a simple adjacency matrix determined by a manually-set threshold. For AGNN, we vary the initial threshold T_h^0 employed for the coarse-grained neighbor selection, see Section IV-C. Three key observations can be drawn from these results. 1) Both MGALs and AGNN constantly achieve lower RMSE values compared to the GCN-based method utilizing a constant aggregation weight. This underscores the effectiveness of the learned aggregation weights through the attention mechanism in Section IV-D. 2) Akin to MGALs, AGNN exhibits initially low localization error. As T_h increases, MGAL's localization error experiences fluctuations, while AGNN's error decreases rapidly and remains consistently at a low level. This indicates that for AGNN, the graph attention layer plays a crucial role in ensuring localization accuracy when the initial threshold T_h^0 is set to a small value. While T_h^0 is relatively large, AGNN resorts to the proposed ALM, which allows for further refinement of the coarse-grained neighbor sets, thereby enhancing the accuracy of localization. 3) AGNN demonstrates consistently low localization errors in two distinct noise environments across various values for T_h^0 , suggesting the versatility and robustness of the initial threshold choice under diverse noise conditions.

– *Model Generalization:* To demonstrate the generalization and transferability of our GNN-based methods, we evaluate the performance of GCN and AGNN on previously unseen datasets $\{\mathcal{D}^2, \mathcal{D}^3, \mathcal{D}^4\}$ using models trained on \mathcal{D}^1 , as summarized in Table IV. Datasets \mathcal{D}^1 and \mathcal{D}^2 share the same noise distribution and scene size but represent different realizations of the same setting. Both methods exhibit slight increases in localization error when tested on \mathcal{D}^2 , demonstrating strong robustness to

TABLE IV
PERFORMANCE OF GCN AND AGNN ON DATASETS WITH VARYING NOISE LEVELS, SCENE SIZES, AND NETWORK REALIZATIONS

| Train: \mathcal{D}^1 , Test: \mathcal{D}^1 | | Train: \mathcal{D}^1 , Test: $\{\mathcal{D}^2, \mathcal{D}^3, \mathcal{D}^4\}$ | | | |
|--|-----------------|--|-----------------|-----------------|--|
| Dataset | \mathcal{D}^1 | \mathcal{D}^2 | \mathcal{D}^3 | \mathcal{D}^4 | |
| Noise | (0.25, 10%) | (0.25, 10%) | (0.25, 10%) | (0.25, 30%) | |
| Size | 5m x 5m | 5m x 5m | 3m x 3m | 5m x 5m | |
| GCN | 0.1006 | 0.1273 | 0.2448 | 0.3689 | |
| AGNN | 0.0638 | 0.0676 | 0.0783 | 0.2181 | |

TABLE V
THE COMPUTATIONAL TIME (IN SECONDS) OF DIFFERENT METHODS WITH ($\sigma^2 = 0.1, p_B = 30\%$) AND $N_l = 50$

| N | GCN | AGNN | MLP | SMILE | LS | MDS | ECM | SDP |
|-------|------|-------|------|-------|-------|--------|-------|------|
| 500 | 3.24 | 5.23 | 2.33 | 5.78 | 32.47 | 22.64 | 82.85 | 1587 |
| 1000 | 5.82 | 14.66 | 3.94 | 8.48 | 89.92 | 100.74 | 353.4 | N.A. |
| 10000 | 707 | 1872 | 212 | 739 | N.A. | N.A. | N.A. | N.A. |

variations in network realizations. In the \mathcal{D}^3 scenario, where the scene size changes to $3\text{m} \times 3\text{m}$, GCN suffers a notable accuracy decline due to its reliance on fixed thresholds, which fail to adapt to the altered environment. In contrast, AGNN maintains stable performance with only a minor increase in error, as its attention mechanism dynamically recalculates an optimal threshold matrix \mathbf{T}^A , effectively adapting to the new scene size. Finally, in the more challenging \mathcal{D}^4 scenario, characterized by a significant increase in NLOS occurrence probability, both methods experience performance degradation. However, AGNN exhibits a smaller decline compared to GCN, highlighting its superior adaptability and robustness across diverse and noisy environments.

– *Computational Efficiency*: Beyond accuracy and robustness, real-world applications demand prompt responses. Table V presents the computational times for various methods. Notably, the proposed GNN-based methods exhibit significantly reduced training times compared to traditional model-driven statistical methods. Particularly, for a network with 500 nodes, SMILE, MLP, GCN, and AGNN only require a few seconds, whereas model-driven methods incur substantially higher computational costs, reaching up to 1587 s for SDP using commercial software. Furthermore, it's noteworthy that the computational time for the proposed GNN-based methods only exhibits a modest increase when doubling the number of nodes in the network. This scalability attribute highlights the efficiency of the proposed methods in large-scale scenarios. Especially, for massive scale networks, the proposed GNN-based methods exhibit a superior capability compared to traditional methods, exemplified by the case of $N = 10000$ where the GNN-based methods remain computationally affordable, while the traditional methods become impractical at such a scale. These results collectively suggest that the proposed GNN-based methods not only achieve high accuracy but also offer a prospective solution for large-scale network localization with efficient computational performance, making them suitable for real-world deployment.

TABLE VI
THE LOCALIZATION ERROR (RMSE) OF AGNN UNDER VARYING NOISE CONDITIONS FOR DIFFERENT VALUES OF p

| | Noise (σ^2, p_B) | | | | |
|-----------|---------------------------|---------------|---------------|---------------|---------------|
| | (0.04, 0%) | (0.10, 10%) | (0.25, 10%) | (0.25, 30%) | (0.50, 50%) |
| $p = 0.5$ | 0.0574 | 0.0642 | 0.0791 | 0.1021 | 0.1308 |
| $p = 1$ | 0.0486 | 0.0551 | 0.0638 | 0.0812 | 0.1015 |
| $p = 2$ | 0.0582 | 0.0657 | 0.0743 | 0.1103 | 0.1447 |
| $p = 3$ | 0.1171 | 0.0846 | 0.0936 | 0.1288 | 0.1585 |

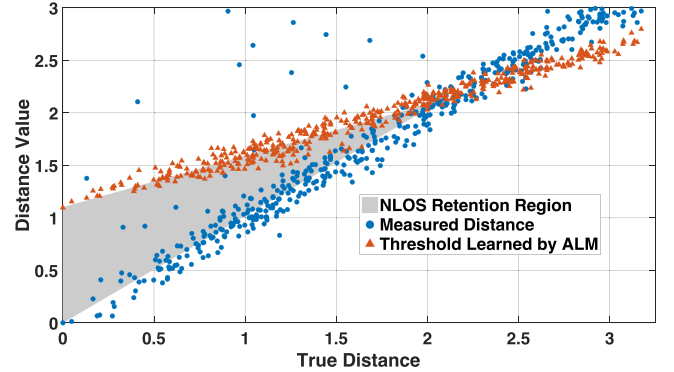


Fig. 8. The threshold learned by ALM versus the measured distance for the 10-th node in (0.25, 30%) dataset under $T_h^0 = 3.0$.

B. Insights Into Well-Performed AGNN

– *Selection of p in ALM*: To better understand the impact of different values of p in (10), we conduct experiments comparing the localization error of AGNN under various noise levels for different values of p . The results are presented in Table VI. As shown in the table, when $p = 1$, AGNN exhibits the lowest localization error across all noise conditions. Based on these results, we offer the following analysis. For a fixed \mathbf{v}_A and \mathbf{W}_A , increasing p results in a greater divergence of the attention scores e_{ij}^A , which in turn increases the variance of both the attention scores and the corresponding threshold values T_{ij}^A . A larger variance leads to a wider range of threshold values, which increases the likelihood of assigning larger thresholds to nodes affected by NLOS noise, thereby weakening the denoising ability. Conversely, decreasing p reduces the variance in the attention scores, but this may constrain the model's ability to select the optimal threshold for each node. Consequently, the choice of p represents a tradeoff between the flexibility of threshold selection and the denoising capability. As indicated by the results in Tab VI, the optimal balance between these factors is achieved when $p = 1$.

– *NLOS Truncation by ALM*: To delve deeper into the impact of the threshold matrix \mathbf{T}^A learned by the attention mechanism in our designed ALM, we conduct a scatter plot analysis, as depicted in Fig. 8. This plot compares the learned thresholds for the 10-th node, $t_{[10,:]}^A$, depicted by orange triangles, with the measured distances, $\mathbf{x}_{[10,:]}$, represented by blue circles. More examples of other nodes can be found in the Appendix G. The distribution of blue circles reveals that some points deviate significantly from the main trend due to the NLOS effect, which

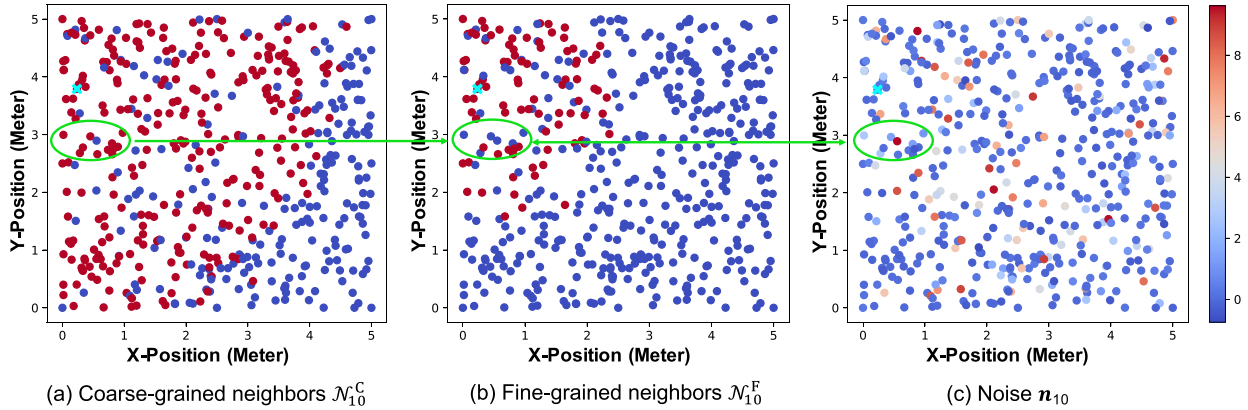


Fig. 9. The different types of scatter plots for the 10-th node, denoted by a cyan cross, in (0.25, 30%) dataset under $T_h^0 = 4.0$.

causes the measured distances to be much greater than the true distances. In contrast, the distribution of orange triangles leads to the following observations. 1) The trend of the orange triangles confirms that the attention mechanism effectively learns a distance-related threshold. 2) The filtered-out neighbors (blue circles above the thresholds) fall into two categories: those with short distances but significant NLOS noise (blue circles deviating from the main trend) and those with larger distances. 3) After applying the distance-related thresholds, the resulting fine-grained neighbor set consists of: a) Neighbors with short distances and LOS noise (blue circles along the main trend), contributing high-quality information for localization. b) Neighbors with short distances and relatively low NLOS noise (blue circles below the thresholds but slightly deviating from the main trend), represented by the gray region in Fig. 8. This set will be further refined by MGALs.

– *Visualization of ALM’s Effect*: We further visualize the neighbor selection process within ALM for the 10-th node using scatter plots and a heatmap. Fig. 9(a) and (b) illustrate the coarse-grained neighbor set \mathcal{N}_{10}^C and the fine-grained neighbor set \mathcal{N}_{10}^F , respectively, with the neighbors highlighted in red and the non-neighbors shown in blue. Fig. 9(c) presents a heatmap of additive noise \mathbf{n}_{10} , between the 10-th node and all the others, with varying colors indicating different noise levels. A comparison of Fig. 9(a) and (b) reveals that the neighbors in \mathcal{N}_{10}^C but excluded from \mathcal{N}_{10}^F primarily fall into two categories: those at relatively large distances from the 10-th node, and those close to it but suffering from relatively high noise levels. Notably, the cyan circle in Fig. 9(a) highlights close neighbors, some of which are initially included in \mathcal{N}_{10}^C (in red) but are excluded from \mathcal{N}_{10}^F (in blue) in Fig. 9(b). Examining the corresponding green region in Fig. 9(c), we observe that the excluded nodes exhibit higher noise levels (3.0–4.5 meters), confirming the effectiveness of ALM’s attention mechanism in refining \mathcal{N}_{10}^C by removing high-noise neighbors. This refinement reduces noise contamination, enhancing the quality of \mathcal{N}_{10}^F for further processing in MGALs.

– *NLOS Filtering by MGALs*: After truncating the measured distance using the learned threshold, the resulting truncated measurement matrix $\hat{\mathbf{X}}$ still retains a portion of NLOS noise, as depicted by the gray region in Fig. 8. To examine the effectiveness of MGALs’ attention mechanism in mitigating the

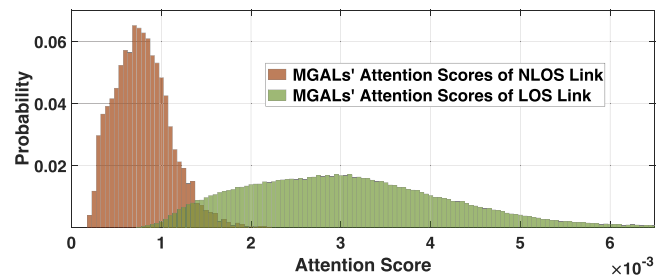


Fig. 10. Distributions of MGALs’ attention scores for LOS link and NLOS link in (0.25, 30%) dataset.

remaining NLOS noise, we present the distributions of attention scores for both LOS and NLOS links in Fig. 10. Primarily, it is evident that MGALs assign attention scores to NLOS and LOS links to distinct distributions, thus demonstrating the ability of the attention mechanism in MGALs to distinguish between NLOS and LOS links. Furthermore, the distribution of attention scores for NLOS links exhibits a smaller mean and variance compared to those of LOS links, indicating that NLOS links tend to receive lower attention scores, while LOS links exhibit relatively larger attention scores with a more flexible range of values. Consequently, in the feature propagation step (18), MGALs assign smaller aggregation weights to NLOS links while prioritizing LOS links, which further filter out NLOS noise to some degree.

– *Validation of Dynamic Attention Property*: As detailed in Section V-A, the dynamic attention property has been demonstrated for both the proposed ALM and MGALs methods. To validate this, we compute the attention scores between nodes with indices $\{1, \dots, 50\}$ using the well-trained parameters from GAT, ALM, and MGALs models. This process produces an attention score matrix denoted as $\mathbf{E} \in \mathbb{R}^{50 \times 50}$. For each row of the attention scores $e_{[i,:]}$, we individually rank the scores within each row in an ascending order, yielding three distinct patterns of attention score rankings corresponding to GAT, ALM, and MGALs, as illustrated in Fig. 11. In these heatmaps, a higher rank (darker red) corresponds to a higher attention score. Fig. 11(a) depicts the results for the standard GAT. Here, the attention scores in each row maintain a consistent rank,

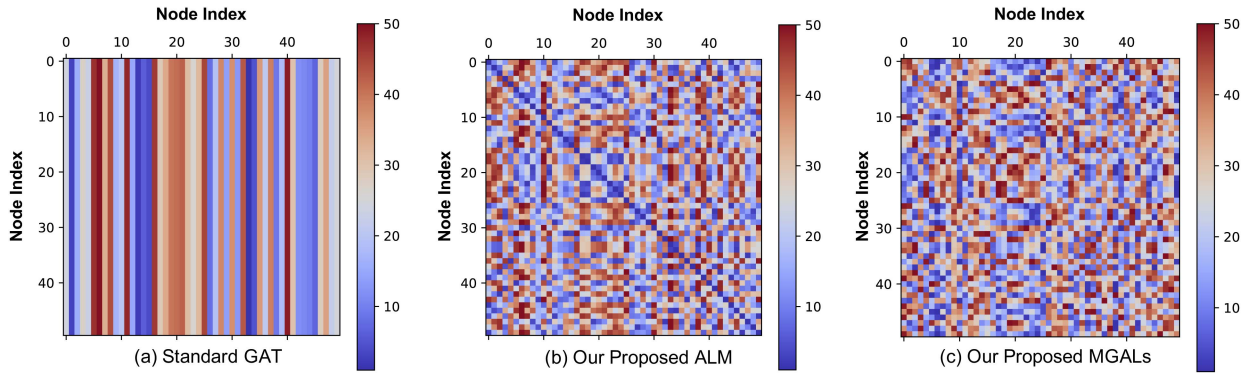


Fig. 11. Three heatmaps of attention score rankings obtained by different methods on the (0.25, 30%) dataset under T_h (or T_h^0) = 3.0.

reflecting a fixed order in assigning weights to the other nodes and indicating a static attention property. In contrast, Fig. 11(b) and (c) present the results for ALM and MGALs, respectively. In these modules, the attention scores exhibit varying ranks within each row, validating the dynamic attention property inherent to both ALM and MGALs.

VII. CONCLUSION

This paper investigated the robust cooperative localization of massive wireless networks in mixed LOS/NLOS environments. To address this challenging problem, we leveraged cutting-edge GNNs augmented with tailored attention mechanisms. Specifically, we proposed a novel AGNN model capable of autonomously learning the underlying graph structure of the network and the aggregation weight of each node. Consequently, our AGNN model outperforms our previous vanilla GCN-based method that employed predefined graph structures. Further theoretical analyses demonstrated that the proposed AGNN model exhibits dynamic attention properties and affordable computational complexity. Numerical results underscored the superiority of the class of GNN-based network localization methods in terms of localization accuracy, robustness, and computational efficiency, rendering them highly suitable for future large-scale networks in complex settings. The AGNN model, in particular, is explored by detailed dissection experiments to demonstrate significant advancements, establishing its potential for massive network applications in challenging environments.

APPENDIX

A. The Optimization Method for T_h^0 Selection

Originally, ALM employs a manually predetermined threshold, denoted as T_h^0 , uniformly applied across all nodes to establish a coarse-grained neighbor set. To address the issue related to the predetermined threshold, we introduce an optimization method capable of autonomously learning an individual threshold for each node. More precisely, the optimization method transforms the uniform threshold into a trainable vector (a scalar or a matrix case is analogous), with each element representing the threshold value for a specific node. This vector is randomly initialized and it can be updated and optimized through the

backpropagation process. The specific approach is outlined as follows.

The trainable threshold vector is represented as $\mathbf{t}^v \in \mathbb{R}^N$. Since \mathbf{t}^v serves as a trainable vector, it is imperative to ensure that each element within this vector is bounded within the interval $[0, l_{\max}]$, where l_{\max} corresponds to the maximum distance over which a node in the network can receive signals. To enforce this constraint on \mathbf{t}^v during the training process, we apply the following rescaling operation:

$$\hat{\mathbf{t}}^v = l_{\max} \cdot \text{Sigmoid}(\mathbf{t}^v). \quad (26)$$

Subsequently, it is natural to construct adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$ as similar to (3) that

$$a_{ij} = \begin{cases} 0, & \text{if } x_{ij} > \hat{t}_i^v, \\ 1, & \text{otherwise,} \end{cases} \quad (27)$$

and further construct the truncated measurement matrix as $\hat{\mathbf{X}} = \mathbf{A} \odot \mathbf{X}$.

Based on the approximated step function which is elucidated in (13), (27) and relative expression of \hat{x}_{ij} can be expressed as follows:

$$a_{ij} = \text{ReLU}(-\tanh(\gamma(x_{ij} - \hat{t}_i^v))) \quad (28)$$

$$\hat{x}_{ij} = x_{ij} \cdot \text{ReLU}(-\tanh(\gamma(x_{ij} - \hat{t}_i^v))). \quad (29)$$

Once the learnable adjacency matrix and truncated measurement matrix are obtained via the optimization method, they can be effectively employed as inputs for MGALs.

The optimization problem for this procedure can be formulated as follows:

$$\begin{aligned} \arg \min_{\mathbf{W}, \mathbf{t}^v} \quad & \mathcal{L} = \|\mathbf{P}_l - \hat{\mathbf{P}}_l\|_F^2 \\ \text{s.t.} \quad & \hat{\mathbf{P}} = \text{MGALs}_{\mathbf{W}}(\mathbf{A}, \hat{\mathbf{X}}) \\ & a_{ij} = \text{ReLU}(-\tanh(\gamma(x_{ij} - \hat{t}_i^v))) \\ & \hat{x}_{ij} = x_{ij} \cdot \text{ReLU}(-\tanh(\gamma(x_{ij} - \hat{t}_i^v))) \\ & \hat{\mathbf{t}}^v = l_{\max} \cdot \text{Sigmoid}(\mathbf{t}^v). \end{aligned} \quad (30)$$

Herein, \mathbf{W} represents all trainable matrices in MGALs. Upon obtaining the optimized initial threshold vector, and its associated coarse-grained neighbor set, this set is employed in the

fine-grained neighbor refinement stage within the ALM. Subsequently, the application of MGALs ensues, ultimately leading to the attainment of the final localization.

B. Proof of Theorem 1

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph modeled by a GAT layer with some \mathbf{v} and \mathbf{W} values, and having node representations $\{\mathbf{h}_{[i,:]}, \dots, \mathbf{h}_{[N,:]}\}$. The learned parameter \mathbf{v} can be written as a concatenation $\mathbf{v} = [\mathbf{v}_1 \parallel \mathbf{v}_2] \in \mathbb{R}^{2D_k}$ such that $\mathbf{v}_1, \mathbf{v}_2 \in \mathbb{R}^{D_k}$, and GAT can be re-written as:

$$\alpha_{ij} = e(\mathbf{h}_{[i,:]}, \mathbf{h}_{[j,:]}) = \phi(\mathbf{h}_{[i,:]} \mathbf{W} \mathbf{v}_1 + \mathbf{h}_{[j,:]} \mathbf{W} \mathbf{v}_2). \quad (31)$$

Since \mathcal{V} is finite, there exists a node $j_{\max} \in \mathcal{V}$ such that $\mathbf{h}_{[j_{\max},:]} \mathbf{W} \mathbf{v}_2$ is maximal among all nodes $j \in \mathcal{V}$. Due to the monotonicity of $\phi(\cdot)$ and softmax, for every query node $i \in \mathcal{V}$, the node j_{\max} also leads to the maximal value of its attention distribution $\{\alpha_{ij} \mid j \in \mathcal{V}\}$. Thus, it computes only static attention.

C. Proof of Theorem 2

Let's consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. This graph comprises transformed node representations, denoted as $\{\mathbf{h}_{[i,:]} \mathbf{W}, \dots, \mathbf{h}_{[N,:]} \mathbf{W}\}$. Suppose $\varphi: [N] \rightarrow [N]$ is any node mapping.

We introduce a function $g: \mathbb{R}^{2D_k} \rightarrow \mathbb{R}$, defined as follows:

$$g(\mathbf{x}) = \begin{cases} 1 & \text{when } \mathbf{x} = [\mathbf{h}_{[i,:]} \mathbf{W} \parallel \mathbf{h}_{[\varphi(i),:]} \mathbf{W}] \\ 0 & \text{otherwise} \end{cases} \quad (32)$$

Further, we define a continuous function $\tilde{g}: \mathbb{R}^{2D_k} \rightarrow \mathbb{R}$, which matches g at precisely N^2 specific inputs: $\tilde{g}(\mathbf{x}) = g(\mathbf{x})$, when $\mathbf{x} = [\mathbf{h}_{[i,:]} \mathbf{W} \parallel \mathbf{h}_{[j,:]} \mathbf{W}]$, $\forall i, j \in [N]$. For all other inputs $\mathbf{x} \in \mathbb{R}^{2D_k}$, $\tilde{g}(\mathbf{x})$ assumes any values that preserve its continuity.

Note that \tilde{g} is formulated for ease of proof. As our goal is for the graph attention layer's scoring function e in (15), (16) to approximate the mapping φ at a finite set of points, we need e to approximate g at specific points. Although g is discontinuous, \tilde{g} is continuous, facilitating the use of the universal approximation theorem. By approximating \tilde{g} , e effectively approximates g at our specified points. We only require \tilde{g} to match g at N^2 specific points $[\mathbf{h}_{[i,:]} \mathbf{W} \parallel \mathbf{h}_{[j,:]} \mathbf{W}] : i, j \in [N]$. Beyond these points, the \tilde{g} 's values are flexible, as long as its continuity is maintained.

Therefore, for every node $i \in \mathcal{V}$ and every $j_{\neq \varphi(i)} \in \mathcal{V}$, we have:

$$\underbrace{\tilde{g}([\mathbf{h}_{[i,:]} \mathbf{W} \parallel \mathbf{h}_{[j,:]} \mathbf{W}])}_{=0} < \underbrace{\tilde{g}([\mathbf{h}_{[i,:]} \mathbf{W} \parallel \mathbf{h}_{[\varphi(i),:]} \mathbf{W}])}_{=1}. \quad (33)$$

If we concatenate the input vectors and define the graph attention layer's scoring function e based on the concatenated vector $[\mathbf{h}_{[i,:]} \mathbf{W} \parallel \mathbf{h}_{[j,:]} \mathbf{W}]$, then according to the universal approximation theorem, e can approximate \tilde{g} for any compact subset of \mathbb{R}^{2D_k} .

Hence, for any sufficiently small ϵ (where $0 < \epsilon < \frac{1}{2}$), there exist parameters \mathbf{W}_{att} and \mathbf{v}_{att} such that for every node $i \in \mathcal{V}$

and every $j_{\neq \varphi(i)} \in \mathcal{V}$:

$$\underbrace{e(\mathbf{h}_{[i,:]} \mathbf{W}, \mathbf{h}_{[j,:]} \mathbf{W})}_{<0+\epsilon} < \underbrace{e(\mathbf{h}_{[i,:]} \mathbf{W}, \mathbf{h}_{[\varphi(i),:]} \mathbf{W})}_{1-\epsilon}. \quad (34)$$

Owing to the increasing monotonicity of the softmax function, this implies:

$$\alpha_{i,j} < \alpha_{i,\varphi(i)}. \quad (35)$$

This shows that our proposed graph attention layer possesses dynamic attention, as it assigns the highest attention score to any neighbor node depending on the concatenated feature vector.

D. Proof of Theorem 5

The proof of Theorem 5 comprises two distinct parts, pertaining to the time complexity analysis of ALM and the k -th graph attention layer within the MGALs.

The primary computational complexity within ALM arises from the calculation of attention coefficients, as articulated in (10). Firstly, we compute $\mathbf{x}_{[i,:]} \mathbf{W}_A$ for every $i \in [N]$, which necessitates $O(NNF_A)$. Subsequently, for each edge (i, j) , we compute $|\phi(\mathbf{x}_{[i,:]} \mathbf{W}_A) - \phi(\mathbf{x}_{[j,:]} \mathbf{W}_A)|^p$ using the pre-computed $\mathbf{x}_{[i,:]} \mathbf{W}_A$ and $\mathbf{x}_{[j,:]} \mathbf{W}_A$, incurring a time complexity of $O(|E^C|F_A)$. Finally, computing the results of the linear layer \mathbf{v}_A adds an additional $O(|E^C|F_A)$. In summary, the time complexity of ALM amounts to $O(NNF_A + |E^C|F_A)$.

The computational complexity of the k -th graph attention layer within MGALs can be broken down into several components. Initially, we compute (16) for every $i \in [N]$, demanding $O(ND_{k-1}D_k)$. Subsequently, we calculate $\hat{\mathbf{h}}_{[i,:]}^{(k-1)} \mathbf{W}_{att}^{(k-1)}$ for every $i \in [N]$ using the pre-obtained $\hat{\mathbf{h}}_{[i,:]}^{(k-1)}$, which incurs a time complexity of $O(ND_k F_{att})$. Further, computing the results of the linear layer \mathbf{v}_{att}^{k-1} takes $O(|E^F|F_A)$. Finally, we compute (18) for every $i \in [N]$, requiring $O(|E^F|D_k)$. Consequently, the time complexity for the k -th graph attention layer within MGALs is given by $O(ND_k F_{att} + ND_{k-1}D_k + |E^F|F_{att} + |E^F|D_k)$.

By systematically analyzing each component, we have established the respective time complexities for ALM and the k -th graph attention layer within MGALs as outlined above.

E. Comprehensive Analysis of CRB

1) *The General Derivation of CRB*: We first provide a detailed derivation of the CRB under various NLOS conditions. When establishing the performance bound, we assume precise knowledge of the actual measurement error distribution, denoted as $p_n(n)$. The entries corresponding to unknown positions of agents, represented by $\mathbf{P}_u := [\mathbf{p}^x, \mathbf{p}^y] \in \mathbb{R}^{(N-N_i) \times 2}$, are reorganized into a vector form $\mathbf{p}_u^v := [\mathbf{p}^x \parallel \mathbf{p}^y] \in \mathbb{R}^{2(N-N_i)}$ for ease of analysis.

According to [21], the Fisher information matrix (FIM) of \mathbf{p}_u^v given the true $p_n(n)$ is readily obtained

$$\mathbf{F}(\mathbf{p}_u^v) = \begin{pmatrix} \mathbf{F}_{xx} & \mathbf{F}_{xy} \\ \mathbf{F}_{xy}^T & \mathbf{F}_{yy} \end{pmatrix} \quad (36)$$

where \mathbf{F}_{xx} , \mathbf{F}_{xy} and \mathbf{F}_{yy} are all square matrices of dimension $(N - N_l) \times (N - N_l)$. The matrix elements are defined as:

$$[\mathbf{F}_{mn}]_{i,i'} = \begin{cases} \mathcal{I}_n \cdot \sum_{\forall j \in \mathcal{N}(i)} \frac{(p_i^m - p_j^m)(p_i^n - p_j^n)}{\|[\mathbf{P}_u]_{[i,:]} - [\mathbf{P}_u]_{[j,:]} \|^2}, & i = i' \\ -\mathcal{I}_n \cdot \delta_{i,i'} \cdot \frac{(p_i^m - p_{i'}^m)(p_i^n - p_{i'}^n)}{\|[\mathbf{P}_u]_{[i,:]} - [\mathbf{P}_u]_{[i',:]} \|^2}, & i \neq i' \end{cases} \quad (37)$$

for $m, n \in \{x, y\}$. Here, $\delta_{i,i'}$ is Kronecker's delta defined by

$$\delta_{i,i'} = \begin{cases} 1, & \text{if } i' \in \mathcal{N}(i) \\ 0, & \text{if } i' \notin \mathcal{N}(i) \end{cases}. \quad (38)$$

Additionally, the intrinsic accuracy scalar factor is given by:

$$\mathcal{I}_n = \int \frac{[\nabla_n p_n(n)]^2}{p_n(n)} dn. \quad (39)$$

The scalar factor \mathcal{I}_n is often approximated via Monte Carlo integration as

$$\mathcal{I}_n \approx \frac{1}{N_M} \sum_{n=1}^{N_M} \frac{[\nabla_n p_n(n^{(k)})]^2}{p_n^2(n^{(k)})} \quad (40)$$

where $n^{(k)}$, $k = 1, 2, \dots, N_M$ are i.i.d. samples generated from $p_n(n)$. Finally, the CRB is given by $\text{CRB}(\mathbf{p}_u^v) := \mathbf{F}(\mathbf{p}_u^v)^{-1}$.

In the simulations, the localization accuracy is assessed in terms of the overall localization RMSE. The lower bound of RMSE is given by

$$\overline{\text{CRB}}(\mathbf{p}_u^v) := \sqrt{\frac{1}{N - N_l} \text{tr}[\text{CRB}(\mathbf{p}_u^v)]}. \quad (41)$$

The subsequent focus is on deriving the PDF of noise p_n and its derivative under distinct noise conditions. This will be detailed in Section E2 and E3 for uniformly distributed NLOS and Rayleigh distributed NLOS, respectively.

2) *Uniformly Distributed NLOS Case:* For scenarios where the noise is a combination of additive Gaussian distribution (LOS) and Uniform distribution (NLOS), with an occurrence probability of p_B , the noise expression is given by:

$$n = \begin{cases} n^L + n^N, & \text{w.p. } p_B, \\ n^L, & \text{w.p. } 1 - p_B, \end{cases} \quad (42)$$

where n^L and n^N are random variables with Gaussian distribution and Uniform distribution, respectively. The PDFs of n^L and n^N are shown as follows.

$$p_{n^L}(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}, \quad -\infty < x < \infty, \quad (43)$$

$$p_{n^N}(x) = \frac{1}{b-a}, \quad a < x < b. \quad (44)$$

To derive the PDF of noise n , we first need to derive the PDF of $n_1 = n^L + n^N$. The probability distribution of the sum of two or more independent random variables is the convolution of their distributions. Thus, we have

$$p_{n_1}(x) = \int_{-\infty}^{\infty} p_{n^N}(u) p_{n^L}(x-u) du \quad (45a)$$

$$= \frac{1}{b-a} \int_a^b \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-u)^2}{2\sigma^2}} du \quad (45b)$$

$$= \frac{1}{b-a} \frac{1}{2\sqrt{2}\sigma} \frac{2}{\sqrt{\pi}} \int_a^b e^{-\frac{(x-u)^2}{2\sigma^2}} du \\ \times \left(\text{Let } t = \frac{x-u}{\sqrt{2}\sigma}, \text{ then } u = x - \sqrt{2}\sigma t \right) \quad (45c)$$

$$= \frac{1}{b-a} \frac{1}{2\sqrt{2}\sigma} \frac{2}{\sqrt{\pi}} \int_a^b e^{-t^2} d(x - \sqrt{2}\sigma t) \quad (45d)$$

$$= \frac{1}{b-a} \frac{1}{2\sqrt{2}\sigma} \frac{2}{\sqrt{\pi}} (-\sqrt{2}\sigma) \int_{\frac{x-b}{\sqrt{2}\sigma}}^{\frac{x-a}{\sqrt{2}\sigma}} e^{-t^2} dt \quad (45e)$$

$$= -\frac{1}{2(b-a)} \frac{2}{\sqrt{\pi}} \left(\int_0^{\frac{x-b}{\sqrt{2}\sigma}} e^{-t^2} dt - \int_0^{\frac{x-a}{\sqrt{2}\sigma}} e^{-t^2} dt \right)$$

$$\times \left(\text{We know } \text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \right) \quad (45f)$$

$$= -\frac{1}{2(b-a)} \left(\text{erf}\left(\frac{x-b}{\sqrt{2}\sigma}\right) - \text{erf}\left(\frac{x-a}{\sqrt{2}\sigma}\right) \right) \quad (45g)$$

$$= \frac{1}{2(b-a)} \left(\text{erf}\left(\frac{b-x}{\sqrt{2}\sigma}\right) - \text{erf}\left(\frac{a-x}{\sqrt{2}\sigma}\right) \right). \quad (45h)$$

Now, since we know the PDF of n_1 , the PDF of n can be derived as:

$$p_n(x) = p_{n_1}(x) p_B + p_{n^L}(x) (1 - p_B) \quad (46a)$$

$$= \frac{p_B}{2(b-a)} \left(\text{erf}\left(\frac{b-x}{\sqrt{2}\sigma}\right) - \text{erf}\left(\frac{a-x}{\sqrt{2}\sigma}\right) \right) \\ + \frac{1-p_B}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}. \quad (46b)$$

Then, the derivation of $p_n(x)$ is given by:

$$\nabla_x p_n(x) = \frac{p_B}{2(b-a)} \left(-\frac{1}{\sqrt{2}\sigma} \right) \left(\frac{2}{\sqrt{\pi}} \right) \left(e^{-\left(\frac{b-x}{\sqrt{2}\sigma}\right)^2} \right. \\ \left. - e^{-\left(\frac{a-x}{\sqrt{2}\sigma}\right)^2} \right) + \frac{1-p_B}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \left(-\frac{x}{\sigma^2} \right). \quad (47)$$

3) *Rayleigh Distributed NLOS Case:* In the case where noise consists of additive Gaussian distribution (LOS) and Rayleigh distribution (NLOS) with occurrence probability p_B , the PDFs of n^L and n^N are shown as follows.

$$p_{n^L}(x) = \frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{x^2}{2\sigma_1^2}}, \quad -\infty < x < \infty, \quad (48)$$

$$p_{n^N}(x) = \frac{x}{\sigma_2^2} e^{-\frac{x^2}{2\sigma_2^2}}, \quad 0 < x < \infty. \quad (49)$$

TABLE VII
 THE AVERAGED LOSS (RMSE) OF CRB AND GNN-BASED METHODS UNDER DIFFERENT NOISE CONDITIONS FOR $N_l=50$

| methods \ Noise (σ_1, σ_2, p_B) | (0.1, 0.5, 10%) | (0.1, 1, 10%) | (0.25, 1, 30%) | (0.25, 3, 30%) | (0.25, 5, 50%) |
|---|-----------------|---------------|----------------|----------------|----------------|
| GCN | 0.1085 | 0.1111 | 0.1126 | 0.1319 | 0.1599 |
| AGNN | 0.0514 | 0.0561 | 0.0764 | 0.0805 | 0.1031 |
| CRB | 0.0351 | 0.0423 | 0.0671 | 0.0714 | 0.0986 |

 TABLE VIII
 THE LOCALIZATION ERROR (RMSE) OF AGNN UNDER VARYING NOISE CONDITIONS FOR DIFFERENT APPROXIMATED STEP FUNCTIONS

| | Noise (σ^2, p_B) | | | | |
|------------------------------------|---------------------------|---------------|---------------|---------------|---------------|
| | (0.04, 0%) | (0.10, 10%) | (0.25, 10%) | (0.25, 30%) | (0.50, 50%) |
| ReLU(tanh(γx)) | 0.0486 | 0.0551 | 0.0638 | 0.0812 | 0.1015 |
| ReLU(Sigmoid(γx) - 0.5) | 0.0501 | 0.0596 | 0.0672 | 0.0905 | 0.1128 |
| ReLU(Softsign(γx)) | 0.0494 | 0.0566 | 0.0651 | 0.0825 | 0.1039 |
| ReLU(Softplus(γx) - ln2) | 0.0562 | 0.0676 | 0.0802 | 0.1004 | 0.1285 |

The PDF of the combined noise $n_1 = n^L + n^N$ is derived through convolution, yielding:

$$p_{n_1}(x) = \frac{\sigma_2 x}{(\sigma_1^2 + \sigma_2^2)} e^{-\frac{x^2}{2(\sigma_1^2 + \sigma_2^2)}} \phi\left(\frac{\sigma_2 x}{\sigma_1 \sqrt{(\sigma_1^2 + \sigma_2^2)}}\right) + \frac{\sigma_1}{\sqrt{2\pi}(\sigma_1^2 + \sigma_2^2)} e^{-\frac{x^2}{2\sigma_1^2}}, \quad (50)$$

where $\phi(x)$ is the cumulative distribution function of the standard normal random variable. Thus the PDF of n is then expressed as:

$$p_v(x) = p_{n_1}(x)p_B + p_{n^L}(x)(1 - p_B) \quad (51a)$$

$$= p_B \left(\frac{\sigma_2 x}{(\sigma_1^2 + \sigma_2^2)} e^{-\frac{x^2}{2(\sigma_1^2 + \sigma_2^2)}} \phi\left(\frac{\sigma_2 x}{\sigma_1 \sqrt{(\sigma_1^2 + \sigma_2^2)}}\right) + \frac{\sigma_1}{\sqrt{2\pi}(\sigma_1^2 + \sigma_2^2)} e^{-\frac{x^2}{2\sigma_1^2}} \right) + (1 - p_B) \left(\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \right). \quad (51b)$$

Then, the derivation of $p_v(x)$ is (52) shown at the bottom of the next page.

4) *Experimental Results for Rayleigh Distributed NLOS*: In this subsection, we explore the scenario where NLOS conditions follow a Rayleigh distribution denoted as $\mathcal{R}(\sigma_2)$. The RMSE results of CRB and GNN-based methods (including GCN [1] and newly proposed AGNN) are shown in Table 7. The results indicate that, under Rayleigh-distributed NLOS conditions, AGNN consistently outperforms GNN in terms of localization accuracy. Furthermore, AGNN demonstrates proximity to the CRB across various levels of Rayleigh noise, highlighting its effectiveness and robustness in diverse noise environments.

F. The Approximated Step Function

An effective approximated step function must satisfy two essential properties: 1) differentiability with respect to the trainable parameters, and 2) the capability to filter out values beyond

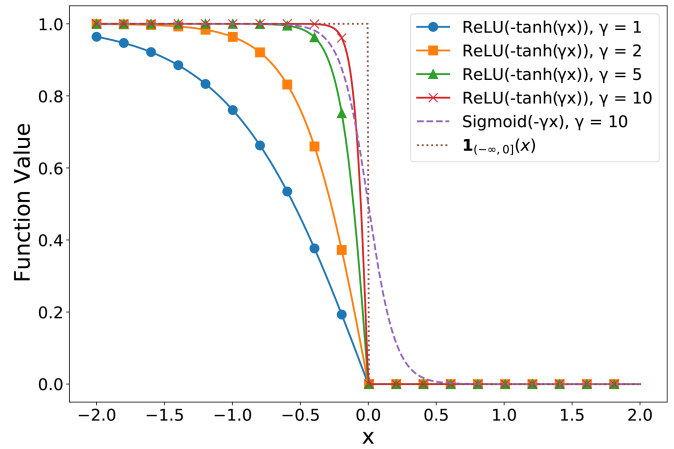


Fig. 12 Comparison between sigmoid function, step function, and the approximated step functions.

a threshold. The filtering property plays a key role in excluding coarse-grained neighbors, which results in a fine-grained neighbor set for further processing.

As shown in Fig. 12, the approximated step function, employing ReLU and tanh, yields a notably sharper transition on one side and effectively truncates values to zero on the other side, consequently achieving a more accurate approximation of the step function than the sigmoid function. Moreover, its derivative can be obtained everywhere (In the standard definition of ReLU, the derivative at 0 is typically taken to be 0.), which makes it feasible to compute gradients with respect to the trainable threshold.

In general, any function that satisfies the aforementioned two criteria can be adopted in the ALM framework. The ReLU function is globally differentiable and inherently performs truncation, making it an essential component of the step function approximation. The tanh(\cdot) function, in this context, may be replaced with other smooth activation functions, such as a shifted Sigmoid, i.e., ReLU(Sigmoid(γx) - 0.5). To further validate this choice, we compare the performance of several alternatives,

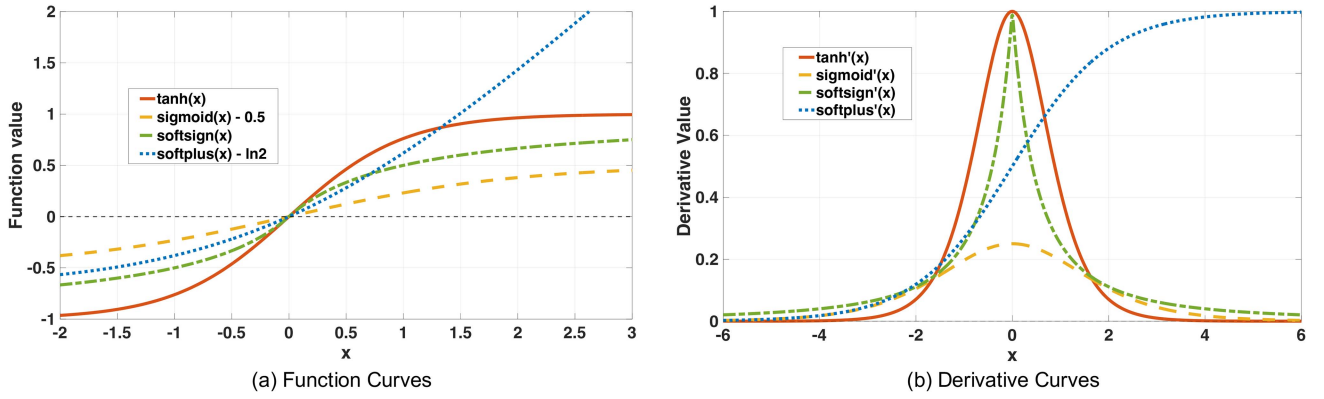
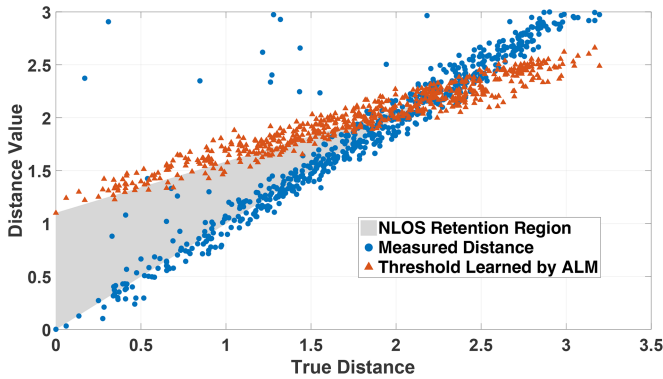
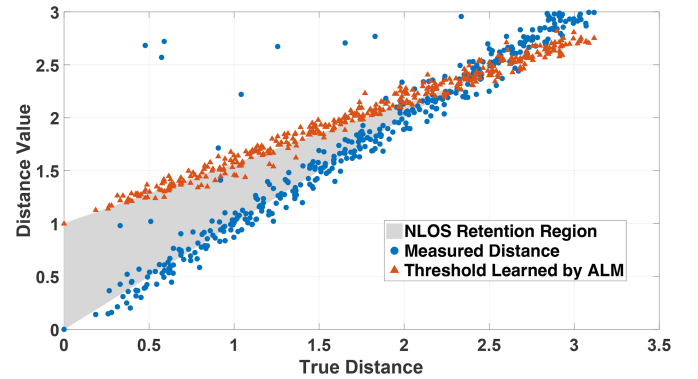


Fig. 13. Comparison between different activation functions.

Fig. 14 The threshold learned by ALM versus the measured distance for the 100-th node in (0.25, 30%) dataset with $T_h^0 = 3.0$.Fig. 15 The threshold learned by ALM versus the measured distance for the 200-th node in (0.25, 30%) dataset with $T_h^0 = 3.0$.

including the shifted Sigmoid, Softsign [48], and shifted Softplus [49] functions. Table VIII summarizes the RMSE of AGNN achieved by each approximation under various noise conditions. Meanwhile, Fig. 13 compares the function and derivative curves of different activation functions.

Notably, the combination of ReLU and Tanh consistently achieves the lowest RMSE across all settings, while maintaining simplicity in its functional form. This superior performance may be attributed to the properties of tanh, which exhibits a broader and more stable gradient range. In contrast, the sigmoid function suffers from vanishing gradients for inputs

with large magnitude, thereby potentially impeding convergence and necessitating additional training epochs. The Softplus function exhibits the weakest performance, possibly due to its relatively shallow slope near the truncation point ($x = 0$), resulting in insufficient training. The Softsign function, commonly used as an alternative to tanh, shows comparable performance, probably due to its similar functional shape and gradient behavior.

G. Additional Experimental Results

To further illustrate the generality of our experimental findings, we present additional scatter plots depicting the

$$\begin{aligned}
 \nabla_x p_v(x) = p_B & \left[\frac{\sigma_2}{\sigma_1^2 + \sigma_2^2} e^{-\frac{x^2}{2(\sigma_1^2 + \sigma_2^2)}} \phi\left(\frac{\sigma_2 x}{\sigma_1 \sqrt{\sigma_1^2 + \sigma_2^2}}\right) + \frac{\sigma_2 x}{\sigma_1^2 + \sigma_2^2} e^{-\frac{x^2}{2(\sigma_1^2 + \sigma_2^2)}} \left(-\frac{x}{\sigma_1^2 + \sigma_2^2}\right) \phi\left(\frac{\sigma_2 x}{\sigma_1 \sqrt{\sigma_1^2 + \sigma_2^2}}\right) \right. \\
 & \left. + \frac{\sigma_2 x}{\sigma_1^2 + \sigma_2^2} e^{-\frac{x^2}{2(\sigma_1^2 + \sigma_2^2)}} \frac{e^{-\left(\frac{\sigma_2 x}{\sigma_1 \sqrt{\sigma_1^2 + \sigma_2^2}}\right)^2 / 2}}{\sqrt{2\pi}} \frac{\sigma_2}{\sigma_1 \sqrt{\sigma_1^2 + \sigma_2^2}} + \frac{\sigma_1}{\sqrt{2\pi}(\sigma_1^2 + \sigma_2^2)} e^{-\frac{x^2}{2\sigma_1^2}} \left(-\frac{x}{\sigma_1^2}\right) \right] \\
 & + (1 - p_B) \left[\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \left(-\frac{x}{\sigma^2}\right) \right]
 \end{aligned} \tag{52}$$

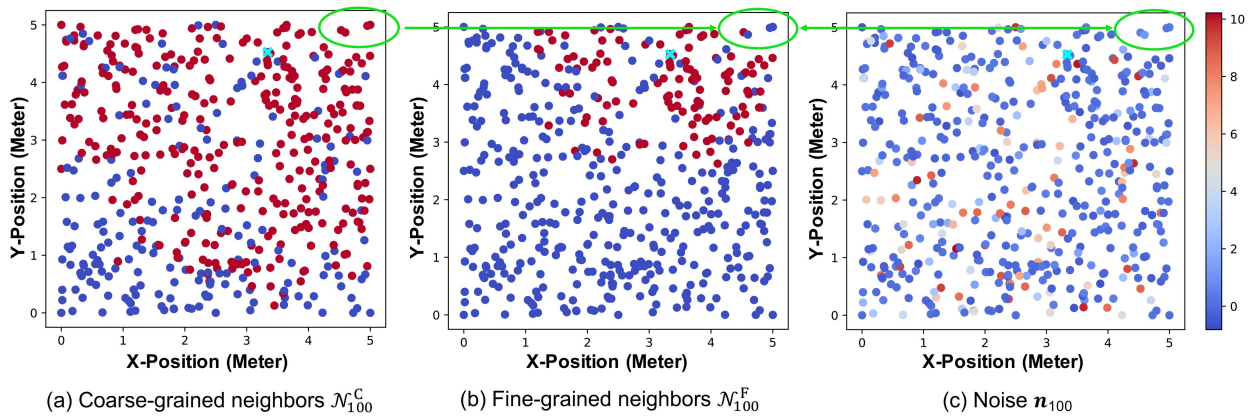


Fig. 16 The different types of scatter plots for the 100-th node, denoted by a cyan cross, in (0.25, 30%) dataset with $T_h^0 = 4.0$.

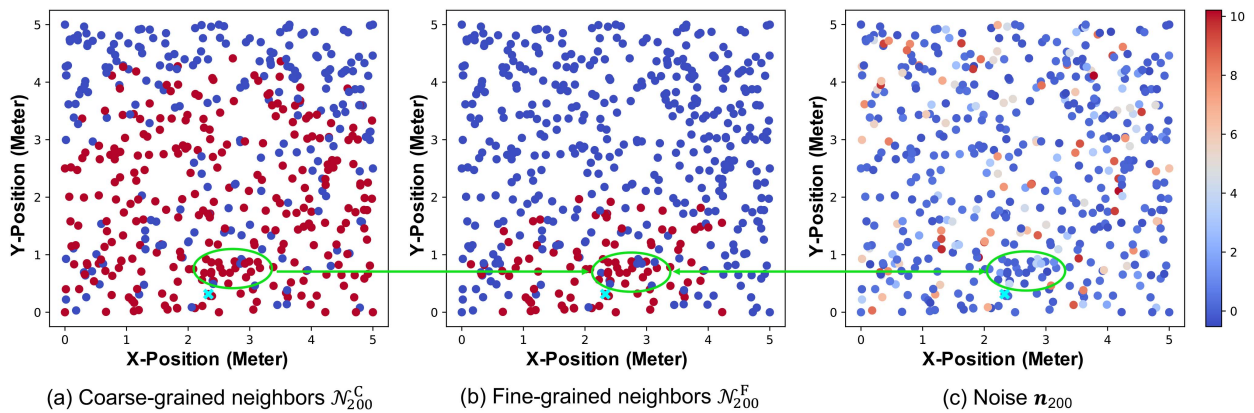


Fig. 17 The different types of scatter plots for the 200-th node, denoted by a cyan cross, in (0.25, 30%) dataset with $T_h^0 = 4.0$.

relationship between true distances and learned thresholds for the 100-th and the 200-th nodes, as depicted in Figs. 14 and 15, respectively. Furthermore, scatter plots (Figs. 16 and 17) are provided to demonstrate the coarse-grained neighbors (\mathcal{N}_{100}^C and \mathcal{N}_{200}^C), fine-grained neighbors (\mathcal{N}_{100}^F and \mathcal{N}_{200}^F), and the noise (\mathbf{n}_{100} and \mathbf{n}_{200}) associated with the 100-th and the 200-th nodes.

REFERENCES

- [1] W. Yan, D. Jin, Z. Lin, and F. Yin, "Graph neural network for large-scale network localization," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2021, pp. 5250–5254.
- [2] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Representations*, Toulon, France, 2017.
- [3] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1024–1034.
- [4] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," in *Proc. Int. Conf. Learn. Representations*, 2017.
- [5] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks," in *Proc. Int. Conf. Learn. Representations*, 2019.
- [6] E. Isufi, F. Gama, and A. Ribeiro, "EdgeNets: Edge varying graph neural networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 11, pp. 7457–7473, Nov. 2022.
- [7] A. Ortega, P. Frossard, J. Kovačević, J. M. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proc. IEEE*, vol. 106, no. 5, pp. 808–828, May 2018.
- [8] F. Gama, A. G. Marques, G. Leus, and A. Ribeiro, "Convolutional neural network architectures for signals supported on graphs," *IEEE Trans. Signal Process.*, vol. 67, no. 4, pp. 1034–1049, Feb. 2019.
- [9] N. Patwari, A. O. Hero, M. Perkins, N. S. Correal, and R. J. O’Dea, "Relative location estimation in wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 51, no. 8, pp. 2137–2148, Aug. 2003.
- [10] F. Yin, C. Fritsche, F. Gustafsson, and A. M. Zoubir, "EM-and JMAP-ML based joint estimation algorithms for robust wireless geolocation in mixed LOS/NLOS environments," *IEEE Trans. Signal Process.*, vol. 62, no. 1, pp. 168–182, Jan. 2014.
- [11] A. Simonetto and G. Leus, "Distributed maximum likelihood sensor network localization," *IEEE Trans. Signal Process.*, vol. 62, no. 6, pp. 1424–1437, Mar. 2014.
- [12] F. Yin, C. Fritsche, F. Gustafsson, and A. M. Zoubir, "TOA-Based robust wireless geolocation and Cramér-Rao lower bound analysis in harsh LOS/NLOS environments," *IEEE Trans. Signal Process.*, vol. 61, no. 9, pp. 2243–2255, May 2013.
- [13] H. Wymeersch, J. Lien, and M. Z. Win, "Cooperative localization in wireless networks," *Proc. IEEE*, vol. 97, no. 2, pp. 427–450, Feb. 2009.
- [14] J. Costa, N. Patwari, and A. Hero, "Distributed weighted-multidimensional scaling for node localization in sensor networks," *ACM Trans. Sensor Netw.*, vol. 2, no. 1, pp. 39–64, 2006.
- [15] P. Biswas, T. Lian, T. Wang, and Y. Ye, "Semidefinite programming based algorithms for sensor network localization," *ACM Trans. Sensor Netw.*, vol. 2, no. 2, pp. 188–220, May 2006.
- [16] P. Tseng, "Second-order cone programming relaxation of sensor network localization," *SIAM J. Optim.*, vol. 18, no. 1, pp. 156–185, 2007.
- [17] Y. Xiong, N. Wu, Y. Shen, and M. Z. Win, "Cooperative localization in massive networks," *IEEE Trans. Inf. Theory*, vol. 68, no. 2, pp. 1237–1258, Feb. 2022.
- [18] A. Ihler, J. W. Fisher, R. L. Moses, and A. S. Willksy, "Nonparametric belief propagation for self-localization of sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 4, pp. 809–819, Apr. 2005.

- [19] D. Jin, F. Yin, C. Fritsche, F. Gustafsson, and A. M. Zoubir, "Bayesian cooperative localization using received signal strength with unknown path loss exponent: Message passing approaches," *IEEE Trans. Signal Process.*, vol. 68, pp. 1120–1135, 2020.
- [20] S. Marano, W. M. Gifford, H. Wymeersch, and M. Z. Win, "NLOS identification and mitigation for localization based on UWB experimental data," *IEEE J. Sel. Areas Commun.*, vol. 28, no. 7, pp. 1026–1035, Sep. 2010.
- [21] F. Yin, C. Fritsche, D. Jin, F. Gustafsson, and A. M. Zoubir, "Cooperative localization in WSNs using Gaussian mixture modeling: Distributed ECM algorithms," *IEEE Trans. Signal Process.*, vol. 63, no. 6, pp. 1448–1463, Mar. 2015.
- [22] H. Chen, G. Wang, Z. Wang, H. C. So, and H. V. Poor, "Non-line-of-sight node localization based on semi-definite programming in wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 11, no. 1, pp. 108–116, Jan. 2012.
- [23] D. Jin, F. Yin, A. M. Zoubir, and H. C. So, "Exploiting sparsity of ranging biases for NLOS mitigation," *IEEE Trans. Signal Process.*, vol. 69, pp. 3782–3795, 2021.
- [24] K. Han, S. M. Yu, S.-L. Kim, and S.-W. Ko, "Mobility-induced graph learning for WiFi positioning," *IEEE J. Sel. Areas Commun.*, vol. 42, no. 9, pp. 2487–2502, Sep. 2024.
- [25] X. Tang and J. Yan, "A CSI amplitude-phase information based graph construction for GNN localization," in *Proc. 2023 Int. Conf. Comput., Inf. Telecommun. Syst.*, 2023, pp. 1–7.
- [26] K. C. Dzmitry Bahdanau and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. Int. Conf. Learn. Representations*, 2015.
- [27] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017.
- [28] Z. Wu, X. Wu, and Y. Long, "Multi-level federated graph learning and self-attention based personalized Wi-Fi indoor fingerprint localization," *IEEE Commun. Lett.*, vol. 26, no. 8, pp. 1794–1798, Aug. 2022.
- [29] F. Yin et al., "FedLoc: Federated learning framework for data-driven cooperative localization and location data processing," *IEEE Open J. Signal Process.*, vol. 1, pp. 187–215, 2020.
- [30] H. Ning et al., "A survey on the metaverse: The state-of-the-art, technologies, applications, and challenges," *IEEE Internet Things J.*, vol. 10, no. 16, pp. 14671–14688, Aug. 2023.
- [31] Y. Qi, H. Kobayashi, and H. Suda, "Analysis of wireless geolocation in a non-line-of-sight environment," *IEEE Trans. Wireless Commun.*, vol. 5, no. 3, pp. 672–681, Mar. 2006.
- [32] R. Zekavat and R. M. Buehrer, *Handbook of Position Location: Theory, Practice and Advances*, vol. 27. Hoboken, NJ, USA: Wiley, 2011.
- [33] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," in *Proc. Int. Conf. Learn. Representations*, 2018.
- [34] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proc. Proc. 19th Int. Conf. Comput. Statist.*, 2010, pp. 177–186.
- [35] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2014.
- [36] A. Sandryhaila and J. M. Moura, "Discrete signal processing on graphs: Frequency analysis," *IEEE Trans. Signal Process.*, vol. 62, no. 12, pp. 3042–3054, Jun. 2014.
- [37] F. Gama, E. Isufi, G. Leus, and A. Ribeiro, "Graphs, convolutions, and neural networks: From graph filters to graph neural networks," *IEEE Signal Process. Mag.*, vol. 37, no. 6, pp. 128–138, Nov. 2020.
- [38] F. Wu, T. Zhang, A. H. D. Souza Jr, C. Fifty, T. Yu, and K. Q. Weinberger, "Simplifying graph convolutional networks," in *Proc. Int. Conf. Learn. Representations*, 2019.
- [39] S. Brody, A. Uri, and Y. Eran, "How attentive are graph attention networks," in *Proc. Int. Conf. Learn. Representations*, 2022.
- [40] X. Wu, A. Ajorlou, Z. Wu, and A. Jadbabaie, "Demystifying oversmoothing in attention-based graph neural networks," *Adv. Neural Inf. Process. Syst.*, vol. 36, pp. 35084–35106, 2023.
- [41] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph convolutional neural networks for web-scale recommender systems," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 974–983.
- [42] J. Chen, T. Ma, and C. Xiao, "FastGCN: Fast learning with graph convolutional networks via importance sampling," in *Proc. Int. Conf. Learn. Representations*, 2018.
- [43] M. Chen, Z. Wei, Z. Huang, B. Ding, and Y. Li, "Simple and deep graph convolutional networks," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1725–1735.
- [44] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Netw.*, vol. 4, no. 2, pp. 251–257, 1991.
- [45] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and deep locally connected networks on graphs," in *Proc. Int. Conf. Learn. Representations*, 2014.
- [46] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 3844–3852.
- [47] L. Clark, S. Mohanty, and B. Krishnamachari, "SMILE: Robust network localization via sparse and low-rank matrix decomposition," in *Proc. 22nd Int. Conf. Inf. Process. Sensor Netw.*, 2023, pp. 322–323.
- [48] G. Lin and W. Shen, "Research on convolutional neural network based on improved Relu piecewise activation function," *Procedia Comput. Sci.*, vol. 131, pp. 977–984, 2018.
- [49] H. Zheng, Z. Yang, W. Liu, J. Liang, and Y. Li, "Improving deep neural networks using softplus units," in *Proc. IEEE 2015 Int. Joint Conf. Neural Netw.*, 2015, pp. 1–4.



Wenzhong Yan (Student Member, IEEE) received the B.Eng. degree in communication engineering and the M.Eng. degree in communication and information systems from Northwestern Polytechnical University, Xi'an, China, in 2016 and 2019, respectively. He is currently working toward the Ph.D. degree in computer and information engineering with The Chinese University of Hong Kong, Shenzhen, China. His research interests include graph learning, wireless localization, and spatio-temporal data modeling.



Feng Yin (Senior Member, IEEE) received the B.Sc. degree from Shanghai Jiao Tong University, Shanghai, China, and the M.Sc. and Ph.D. degrees from Technische Universitaet Darmstadt, Germany. From 2014 to 2016, he was a Postdoc Researcher with Ericsson Research, Linköping, Sweden. He is a tenured Associate Professor of the School of Science and Engineering and partially affiliated with the School of Artificial Intelligence with The Chinese University of Hong Kong, Shenzhen. He has authored or coauthored more than 100 top-tier journal and conference papers, and 20 patents/standards. His research interests include statistical signal processing, Bayesian learning and optimization, and sensory data fusion. He was a recipient of the Chinese Government Award for Outstanding Self-Financed Students Abroad in 2013 and the Marie Curie Young Fellowship from the European Union in 2014. He was an Associate Editor for the EURASIP Signal Processing Journal. He is currently an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING. He is a member of the IEEE Machine Learning and Signal Processing (MLSP) technical committee and a member of the IEEE SPS scholarship committee.



Juntao Wang received the B.Sc. degree in statistics from The Chinese University of Hong Kong, Shenzhen, China, in 2022, where he is currently working toward the Ph.D. degree in computer and information engineering with the School of Science and Engineering. His research interests include deep learning, optimization, signal processing, and wireless communication.



Geert Leus (Fellow, IEEE) received the M.Sc. and Ph.D. degree in electrical engineering from the KU Leuven, Leuven, Belgium, in June 1996 and May 2000, respectively. He is currently a Full Professor with the Faculty of Electrical Engineering, Mathematics and Computer Science of the Delft University of Technology, The Netherlands. He was the recipient of the 2021 EURASIP Individual Technical Achievement Award, 2024 IEEE Signal Processing Society Best Paper Award, 2005 IEEE Signal Processing Society Best Paper Award, and 2002 IEEE

Signal Processing Society Young Author Best Paper Award. He is a Fellow of EURASIP. He was a Member-at-Large of the Board of Governors of the IEEE Signal Processing Society, the Chair of the IEEE Signal Processing for Communications and Networking Technical Committee, the Chair of the EURASIP Technical Area Committee on Signal Processing for Multisensor Systems, the Editor in Chief of the *EURASIP Journal on Advances in Signal Processing* and the Editor in Chief of *EURASIP Signal Processing*.



Yang Tian received the M.Eng. degree in communication engineering from Zhejiang University, Hangzhou, China, in 2020. Since she has been working on Beyond5G wireless system research, development, and standardization. She is currently with Huawei's Wireless Research Team in Shanghai, China. Her research interests include the fields of signal processing based on AI/ML technologies, applications of advanced AI in wireless systems, and their standardization in 5G/6G.



Abdelhak M Zoubir (Life Fellow, IEEE) has been a Professor with the Darmstadt University of Technology, Darmstadt, Germany, since 2003 and Head of Signal Processing Group. He has authored or coauthored more than 500 journal and conference papers on these areas, for which he received several Best Paper Awards. His research interests include statistical methods for signal processing with emphasis on bootstrap techniques, robust detection and estimation, and array processing applied to telecommunications, radar, sonar, car engine monitoring and biomedicine.

Prof. Zoubir was the General Chair, Technical Program Chair, and Member of the Organization Committees of numerous international conferences and workshops, most notably he was Technical Program Co-Chair of the largest and flagship conference on signal processing, The International Conference on Acoustics, Speech, and Signal Processing, Florence, Italy, in 2014. He was on numerous editorial boards, as the Editor-In-Chief of *IEEE Signal Processing Magazine* from 2012 to 2014. He was elected as Chair from 2010 to 2011 of IEEE SPS Technical Committee Signal Processing Theory and Methods and Member from 2007 to 2012 of IEEE SPS Technical Committee Sensor Array and Multi-channel Signal Processing. He was on the Board of Governors of IEEE SPS as Member-at-Large from 2015 to 2017 and Member of the Board of Directors of the European Association for Signal Processing from 2009 to 2016, and President from 2017 to 2018. He has been inducted to the German National Academy of Science and Engineering in 2024. He is a fellow of EURASIP and Distinguished Lecturer (Class 2010-2011).