Distributed Cooperative Positioning in Mobile Wireless Networks: A GNN-Aided Joint Model- and Data-Driven Framework With High-Accuracy Closed-Form Message Representation

Yue Cao, Shaoshi Yang, *Senior Member, IEEE*, Zhiyong Feng, *Senior Member, IEEE*, Ping Zhang, *Fellow, IEEE*, and Sheng Chen, *Life Fellow, IEEE*

Abstract—Future mobile wireless networks will catalyze substantial demand for precise distributed cooperative positioning (DCP), especially when the global navigation satellite systems are unavailable. However, conventional message passing based DCP methods may suffer considerable performance degradation due to message approximation and sparsity/mobility of nodes. In this paper, we first present a high-accuracy parametric message approximation method, which achieves closed-form representations of all types of messages involved and reduces the computational complexity of message passing procedures. Using these representations, we propose a model- and data-driven hybrid inference approach, dubbed graph neural network enhanced spatio-temporal message passing (GNN-STMP), which fine-tunes parametric messages passed on factor graph and obtains more accurate *a posteriori* distribution of nodes' positions by exploiting GNN-generated messages. Furthermore, we develop a universal framework for the parametric message passing based DCP problem, by integrating GNN-STMP with the extend Kalman filter based node's state prediction and refinement. This framework significantly reduces the positioning ambiguity caused by insufficient spatial ranging measurements from neighbor nodes. Simulation results and analyses demonstrate that, compared with state-of-the-art methods, our proposed approaches achieve the best and near-best positioning accuracy when insufficient and sufficient spatial ranging measurements are available, respectively, while incurring modest computational complexity.

Index Terms—Distributed cooperative positioning, factor graph, graph neural network (GNN), mobile wireless networks, parametric message representation.

I. INTRODUCTION

ACCURATE location information plays a crucial role in many emerging applications of mobile wireless networks. For example, one of the major visions for future 6G mobile networks is to provide ubiquitous integrated communication and sensing services. Location information is typically provided by the global navigation satellite systems (GNSS). However, in harsh environments where GNSS is unavailable or denied, it remains a great challenge to obtain high-accuracy location information. In this context, distributed cooperative positioning (DCP) offers a means of improving the localization capability of agents (nodes with unknown positions) by solving an inference problem based on probabilistic graphical model [1], [2], [3], [4], [5], [6], [7]. Specifically, agent can benefit from cooperation with other agents (spatial cooperation) or cooperation with itself between successive time slots (temporal cooperation), upon exploiting inter-agent or intra-agent measurements, respectively. When provided with the true data generating process, the standard belief propagation (BP) [8] is capable of providing the correct posterior probabilities for the DCP problem in the singly connected graph [9], [10], [11], [12], [13] by locally marginalizing the joint *a posteriori* distribution. More recently, a generalized BP approach, dubbed sum-product algorithm over a wireless network (SPAWN) [1], has been extensively explored [2], [3], [6], [7], [14], [15], [16] due to its computational complexity advantages in distributed wireless network. Specifically, each agent broadcasts an approximation of the standard BP messages (real value functions of the associated variables[1]) to its neighbors in the SPAWN procedure, in contrast to the standard BP where each node transmits a different message to each of its neighbors.

In all message passing procedures, the message representation employed in transmission and computation on the probabilistic graphical model is the determining factor for the computational complexity and positioning performance

---

[1]These associated variables refer to the position states of the agents and the inter-agent measurements, which are used to compute the messages exchanged between nodes in the belief propagation process.

of DCP. The development of efficient and exact message representation methods in DCP has garnered significant attention [3], [5], [6], [7], [14], [15], [17], [18]. Conventional methods for message representation can be mainly divided into two types: sample-based methods [3], [14], [17], [18] and parametric methods [5], [6], [7], [14], [15]. The sample-based methods utilize thousands of weighted samples to represent each message, and it can achieve highly accurate message representations. However, these methods impose heavy computational complexity on agents [6], [14], [18]. The parametric methods represent each message as a set of parameters tailored to specific problem. In the two-dimensional scenario, the authors of [1], [14] considered the incoming message (message received by agent) as a two-dimensional circular distribution, and used six parameters to represent the messages. Unfortunately, this representation does not yield a closed-form representation for the *a posteriori* distribution of agent's position, and it requires the use of a numerical particle-based optimization method to approximate the *a posteriori* distribution. The lack of a closed-form solution increases the computational burden and makes real-time implementation more challenging, especially in distributed systems where efficiency is critical. Closed-form solutions allow for faster and more efficient message computations, which are essential for the practical deployment of DCP in large-scale dynamic networks. The authors of [6], [7], [15] considered the incoming messages as a one-dimensional Gaussian distribution. However, this assumption is not accurate and leads to positioning bias.

In some practical applications of mobile wireless networks, the agents are sparsely distributed and the number of neighbors for some agents may be insufficient, which can significantly degrade the positioning accuracy, resulting in increased positioning error over time. To achieve more accurate location estimation in this challenging scenario, several existing studies [7], [17], [19] advocated to utilize the extended Kalman filter (EKF) to estimate the velocities or positions of agents as the *a priori* information. However, they adopted different methods to deal with the nonlinearity involved. To elaborate a little further, the authors of [19] proposed a state-transition and observability constrained extended Kalman filter (STOC-EKF) scheme, which approximates the nonlinear system as a linear model around selected linearization points, thus the positioning accuracy is degraded. In [17], a DCP scheme that combines the sum-product algorithm (SPA) and EKF was proposed, which requires a large number of samples to approximate the nonlinear messages that pass on the graphical model, imposing a high computational complexity. Therefore, this scheme is impractical in either energy-constrained wireless networks or real-time mobile wireless networks. In our previous work [7], a low-complexity parametric message passing algorithm was proposed, which employs mean and variance based message representation, in contrast to the particle-based message approximation used in [17]. However, the algorithm in [7] assumes that the individual message components follow a one-dimensional Gaussian distribution, which is inconsistent with the ranging model, leading to a relatively poor confidence in the estimates [9], [10], [13].

In recent years, the potential of data-driven deep learning techniques for refining the accuracy of messages passing on graphical model has attracted intensive research interest [13], [20], [21], [22]. Graph neural network (GNN) [23], [24], [25], [26], [27], [28] is a connectionist model which is capable of capturing the dependence of graphs via message passing between the nodes of graphs. To combine the benefits of model-based inference and data-driven neural network for molecular prediction tasks in quantum chemistry, the authors of [20] proposed a general framework that abstracts the commonalities between neural networks for graph structure data, dubbed message passing neural network (MPNN). The authors of [13] extended GNN to factor graph (FG) and proposed a message passing approach called neural enhanced belief propagation (NEBP), which exploits the trained GNN model to refine the original BP message on the FG and applies the NEBP approach to error correction decoding tasks. In addition, the authors of [29] applied NEBP to the susceptible-infected-recovered epidemic model. However, the aforementioned approaches cannot be applied to distributed wireless positioning directly, since they require the global graph structure as *a priori*, which is not available for each agent in distributed wireless networks. Recently, the NEBP approach was applied to particle-based DCP [21] and particle-based network navigation [22], respectively. Unfortunately, both algorithms require thousands of weighted samples to represent each message and suffer from high computational complexity caused by massive particles and iterative message computations. Recent studies have explored the use of GNN at the signal level to address DCP. The authors of [27] applied GNNs to improve localization accuracy in large-scale networks under mixed non-line-of-sight (NLoS)/line-of-sight (LoS) conditions, while the authors of [28] introduced an attentional GNN to enhance robustness and flexibility. Most recently, the authors of [30] proposed a data-driven model that integrates the long short-term memory (LSTM) modules with MPNN. This model employs a message-passing-like mechanism to enable nodes to perform DCP with acceptable efficacy in loopy networks. However, the works [21], [22], [30] ignore both internal ranging measurements[2] based temporal information and state constraints.[3] As a result, the methods proposed in [21], [22], and [30] are inapplicable to scenarios where agents have insufficient neighbors, as they fail to provide accurate localization in such situations, leading to significant performance degradation in the entire network.

Against the above background, in this paper we propose a GNN-aided joint model- and data-driven framework with high-accuracy closed-form message representation for solving the DCP problem in mobile wireless networks. Our emphasis is on improving message representation accuracy and exploiting agents' state constraints. The contributions of this paper are summarized as follows.

---

[2]These messages correspond to the internal measurements of the distance an agent moves between time slots. Agents can measure this distance using pedometers.

[3]These constraints are applied during state estimation, based on the prediction and update steps of the EKF, to regulate state variables (e.g., position or velocity) and ensure accurate positioning, especially when agents have insufficient neighbors.

- We propose a high-precision parametric message representation method, which enables efficient computation of the SPAWN message passing procedure, upon using the Taylor polynomial (TP) to approximate the nonlinear terms of each SPAWN message passed on the associated FG. This allows us to derive the closed-form representations for each message and the *a posteriori* distribution of agent's position. The derived TP-SPAWN algorithm outperforms other parametric DCP algorithms or even particle-based algorithms, while incurring modest computational complexity, given that the number of neighbor nodes of the individual agents considered is sufficient.

- Building upon the above high-accuracy message representation method, we propose a model- and data-driven hybrid inference approach, dubbed GNN enhanced spatio-temporal message passing (GNN-STMP). This approach gleans the benefits of both the FG (model-driven) and the GNN (data-driven). To the best of our knowledge, it is the first time that a parametric message passing algorithm has been coherently integrated with GNN in the context of DCP. More fundamentally, our GNN-STMP is capable of obtaining closed-form expressions of the *a posteriori* distributions, thus dramatically reducing the computational complexity of *message multiplication* [14] involved in a wide range of applications. This has not been fulfilled by the existing GNN enhanced message passing algorithms [13], [21], [22]. Additionally, in the context of DCP, our GNN-STMP fine-tunes parametric messages passed on FG and leverages temporal information to obtain a more accurate *a posteriori* distribution of agents' positions by exploiting GNN-generated messages, thereby substantially improving the positioning accuracy of agents when they have sufficient neighboring nodes.

- We develop a universal framework, named EKF-GNN-DCP, for the parametric message passing based DCP, by integrating GNN-STMP with the EKF based agents' state prediction and refinement. This framework significantly reduces the positioning ambiguity caused by insufficient spatial ranging measurements from neighbor nodes. Therefore, it is valuable for improving the positioning performance in the challenging sparsely distributed mobile wireless networks.

The remainder of this paper is organized as follows. In Section II, we specify the system model and define the objective of the cooperative positioning problem. Section III presents the our proposed parametric message representation method and highlights the closed-form expressions for each message. Section IV and Section V develop the GNN enhanced message refinement approach and the EKF based *a posteriori* distribution refinement approach, respectively. In Section VI, we present the computational complexity analysis for various positioning algorithms considered. Thereafter, simulation results and discussions are provided in Section VII. In Section VIII, we draw the conclusions of this paper.

II. SYSTEM MODEL AND PROBLEM DEFINITION

We consider a distributed wireless network composed of N agents and A anchors. The transmission time of the network is slotted. Let $\mathbb{U}$ denote the set of agents, $\mathbb{A}_i^t$ and $\mathbb{U}_i^t$ denote the set of anchors and the set of particular agents, from which agent i receives signals at time slot t, respectively. Let $\boldsymbol{x}_i^t = \left[x_i^t, y_i^t\right]^{\mathrm{T}}$ and $\boldsymbol{v}_i^t$ be the position and velocity vectors of agent i at time slot t, respectively, where $(\cdot)^{\mathrm{T}}$ stands for the transpose operation. Then $\boldsymbol{s}_i^t = \left[\left(\boldsymbol{x}_i^t\right)^{\mathrm{T}}, \left(\boldsymbol{v}_i^t\right)^{\mathrm{T}}\right]^{\mathrm{T}}$ denotes the state vector of agent i at time slot t. At time slot t, agent i can receive spatial (external) ranging measurement from node j ($j \in \mathbb{A}_i^t \cup \mathbb{U}_i^t$), and temporal (internal) ranging measurement based on its hardware. The LoS[4] noise-contaminated spatial ranging measurement by agent i from node j at time slot t can be expressed as

$$z_{j \to i}^t = \|\boldsymbol{x}_i^t - \boldsymbol{x}_j^t\|_2 + e_{j \to i}, \tag{1}$$

where $\| \cdot \|_2$ represents the Euclidean norm, $e_{j \to i} \sim \mathcal{N}\left(0, \sigma_{j \to i}^2\right)$ is the Gaussian noise with zero-mean and variance $\sigma_{j \to i}^2$. The temporal ranging measurement of agent i from time slot $t-1$ to time slot t is given by

$$z_{i,\text{int}}^t = \|\boldsymbol{x}_i^t - \boldsymbol{x}_i^{t-1}\|_2 + e_{i,\text{int}}, \tag{2}$$

where $e_{i,\text{int}} \sim \mathcal{N}\left(0, \sigma_{i,\text{int}}^2\right)$. We denote all the noisy ranging measurements (both external and internal) obtained by agent i at time slot t as $\boldsymbol{z}_i^t$. In this work, we assume that the underlying communication protocols, such as medium-access control (MAC) and physical layer techniques, effectively manage packet collisions and channel randomness to ensure reliable data transmission. This assumption is reasonable given the widespread use and near-Shannon-limit performance of these protocols in modern wireless communication systems. As such, our framework focuses primarily on achieving precise DCP based on the ranging and positional information exchanged between nodes, assuming the communication layer provides sufficient reliability. In addition, the probability of correctly receiving messages from distant agents may decrease in practical applications.

Based on the aforementioned system model, our goal is to obtain the *a posteriori* distribution of the position concerning agent i ($i \in \mathbb{U}$) at any time slot t, given only these noisy measurements, i.e., to acquire $p\left(\boldsymbol{x}_i^t \mid \boldsymbol{z}_i^t\right)$.

III. HIGH-PRECISION PARAMETRIC MESSAGE REPRESENTATION

In the message passing procedure, the probabilistic position information of nodes is exchanged and computed through messages that flow on the probabilistic graphical model. The message representation employed for transmission and computation is the determining factor for the computational complexity and positioning performance of DCP. In this section, we propose a high-precision parametric message representation method which enables efficient computation of the message passing step.

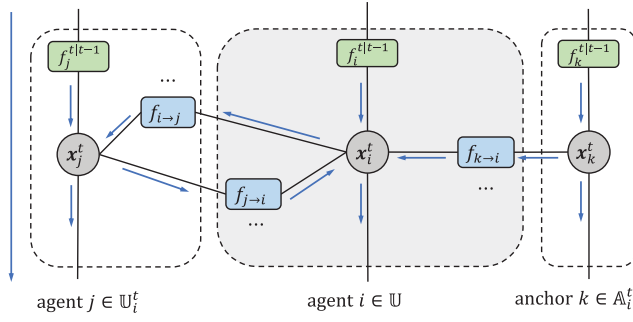[4]The NLoS/LoS mixed environment was considered in our previous work [5].

Fig. 1. FG of $p\left(\boldsymbol{x}_i^t \mid \boldsymbol{z}_i^t\right)$, where $f_{j\to i} = p\left(z_{j\to i}^t \mid \boldsymbol{x}_i^t, \boldsymbol{x}_j^t\right)$, and $f_i^{t|t-1} = p\left(\boldsymbol{x}_i^t \mid \boldsymbol{x}_i^{t-1}\right) p\left(z_{i,\text{int}}^t \mid \boldsymbol{x}_i^{t-1}, \boldsymbol{x}_i^t\right)$. Factors in the gray dashed box are local to agent i. The blue arrows represent the messages flow.

We first create an FG network by factorizing the *a posteriori* distribution of the position vector and mapping the spatial-domain and temporal-domain operations of nodes onto the FG. The factorization of $p\left(\boldsymbol{x}_i^t \mid \boldsymbol{z}_i^t\right)$ satisfies [31]

$$p\left(\boldsymbol{x}_i^t \mid \boldsymbol{z}_i^t\right) \propto p\left(\boldsymbol{x}_i^t\right) p\left(z_{i,\text{int}}^t | \boldsymbol{x}_i^t, \boldsymbol{x}_i^{t-1}\right) p\left(\boldsymbol{x}_i^t | \boldsymbol{x}_i^{t-1}\right) \\ \times \prod_{j\in\mathbb{A}_i^t\cup\mathbb{U}_i^t} p\left(z_{j\to i}^t | \boldsymbol{x}_i^t, \boldsymbol{x}_j^t\right), \quad (3)$$

and the corresponding FG has the structure illustrated in Fig. 1. For each factor, we create a vertex (drawn as a rectangle), and for each variable we create an edge (drawn as a line). When a variable appears in a factor, we connect the edge to the vertex. When a variable appears in more than two factors, an equality vertex, i.e., "=", is created. Then we exploit the iterative SPAWN to infer the *a posteriori* distribution of $\boldsymbol{x}_i^t$, i.e., $p\left(\boldsymbol{x}_i^t \mid \boldsymbol{z}_i^t\right)$, which satisfies

$$p\left(\boldsymbol{x}_i^t \mid \boldsymbol{z}_i^t\right) \approx b_{\ell_{\max}}\left(\boldsymbol{x}_i^t\right) \\ \propto \mu_{f_i^{t|t-1}\to\boldsymbol{x}_i^t} \prod_{j\in\mathbb{U}_i^t\cup\mathbb{A}_i^t} \mu_{\ell_{\max}, f_{j\to i}\to\boldsymbol{x}_i^t}, \quad (4)$$

where $b_{\ell_{\max}}\left(\boldsymbol{x}_i^t\right)$ denotes the belief of $\boldsymbol{x}_i^t$ at iteration $\ell_{\max}$, $\ell_{\max}$ is the maximum number of iterations, $\mu_{f_i^{t|t-1}\to\boldsymbol{x}_i^t}$ is the temporal message passed from factor $f_i^{t|t-1}$ to variable $\boldsymbol{x}_i^t$, and $\mu_{\ell_{\max}, f_{j\to i}\to\boldsymbol{x}_i^t}$ represents the spatial message passed from factor $f_{j\to i}$ to variable $\boldsymbol{x}_i^t$ at iteration $\ell_{\max}$. The temporal message $\mu_{f_i^{t|t-1}\to\boldsymbol{x}_i^t}$ satisfies

$$\mu_{f_i^{t|t-1}\to\boldsymbol{x}_i^t} = \int b_{\ell_{\max}}\left(\boldsymbol{x}_i^{t-1}\right) f_i^{t|t-1} \mathrm{d}\boldsymbol{x}_i^{t-1}, \quad (5)$$

where the belief of the position concerning agent i at time slot $t-1$ satisfies[5]

$$b_{\ell_{\max}}(\boldsymbol{x}_i^{t-1}) \propto \exp\left\{-\frac{1}{2}(\boldsymbol{x}_i^{t-1}-\hat{\boldsymbol{x}}_i^{t-1})^{\mathrm{T}}\boldsymbol{\Sigma}_{\boldsymbol{x}_i^{t-1}}^{-1}(\boldsymbol{x}_i^{t-1}-\hat{\boldsymbol{x}}_i^{t-1})\right\}, \quad (6)$$

where $\hat{\boldsymbol{x}}_i^{t-1} = \left[\hat{x}_i^{t-1}, \hat{y}_i^{t-1}\right]^{\mathrm{T}}$ denotes the mean vector of the position concerning agent i at time slot $t-1$, $\hat{x}_i^{t-1}$ and $\hat{y}_i^{t-1}$

are the estimates of x_i^{t-1} and y_i^{t-1}, respectively, and the covariance matrix of $\boldsymbol{x}_i^{t-1}$ satisfies

$$\boldsymbol{\Sigma}_{\boldsymbol{x}_i^{t-1}} = \begin{bmatrix} \sigma_{x_i^{t-1}}^2 & 0 \\ 0 & \sigma_{y_i^{t-1}}^2 \end{bmatrix} = \operatorname{diag}\left(\sigma_{x_i^{t-1}}^2, \sigma_{y_i^{t-1}}^2\right). \quad (7)$$

According to (2), which represents the temporal ranging measurement of agent i, the factor $f_i^{t|t-1}$ is given by

$$f_i^{t|t-1} = \frac{1}{\sqrt{2\pi\sigma_{i,\text{int}}^2}} \exp\left\{-\frac{\left(z_{i,\text{int}}^t - \left\|\boldsymbol{x}_i^t - \boldsymbol{x}_i^{t-1}\right\|_2\right)^2}{2\sigma_{i,\text{int}}^2}\right\}. \quad (8)$$

However, (5) involves integral and it is difficult to obtain the closed-form expression due to the nonlinear function[6]

$$d\left(\boldsymbol{x}_i^t, \boldsymbol{x}_i^{t-1}\right) \triangleq \left\|\boldsymbol{x}_i^t - \boldsymbol{x}_i^{t-1}\right\|_2. \quad (9)$$

We invoke the second-order TP[7] of $d\left(\boldsymbol{x}_i^t, \boldsymbol{x}_i^{t-1}\right)$ at points $\hat{\boldsymbol{x}}_i^{t-1} = \left[\hat{x}_i^{t-1}, \hat{y}_i^{t-1}\right]^{\mathrm{T}}$ and $\hat{\boldsymbol{x}}_i^t = \left[\hat{x}_i^t, \hat{y}_i^t\right]^{\mathrm{T}}$ to approximate it. Hence the temporal message is approximated by

$$\mu_{f_i^{t|t-1}\to\boldsymbol{x}_i^t} \propto \exp\left\{\boldsymbol{\alpha}_i^{\mathrm{T}}\boldsymbol{\psi}_i\right\}, \quad (10)$$

where

$$\boldsymbol{\alpha}_i = \left[\alpha_{i1}, \alpha_{i2}, \alpha_{i3}, \alpha_{i4}\right]^{\mathrm{T}}, \quad (11)$$
$$\boldsymbol{\psi}_i = \left[-(x_i^t)^2, -(y_i^t)^2, x_i^t, y_i^t\right]^{\mathrm{T}}, \quad (12)$$

with

$$\begin{cases} \alpha_{i1} = \dfrac{\|\boldsymbol{r}\|_2^3 - z_{i,\text{int}}r_2^2}{2\sigma_{i,\text{int}}^2\|\boldsymbol{r}\|_2^3}, \\ \alpha_{i2} = \dfrac{\|\boldsymbol{r}\|_2^3 - z_{i,\text{int}}r_1^2}{2\sigma_{i,\text{int}}^2\|\boldsymbol{r}\|_2^3}, \\ \alpha_{i3} = \dfrac{z_{i,\text{int}}r_1}{\sigma_{i,\text{int}}^2\|\boldsymbol{r}\|_2}, \\ \alpha_{i4} = \dfrac{z_{i,\text{int}}r_2}{\sigma_{i,\text{int}}^2\|\boldsymbol{r}\|_2}, \\ \boldsymbol{r} = [r_1, r_2]^{\mathrm{T}} = \left[\hat{x}_i^t - \hat{x}_i^{t-1}, \hat{y}_i^t - \hat{y}_i^{t-1}\right]^{\mathrm{T}}. \end{cases} \quad (13)$$

The spatial message received by agent i from anchor $k\in\mathbb{A}_i^t$ at time slot t and iteration ℓ, i.e., $\mu_{\ell, f_{k\to i}\to\boldsymbol{x}_i^t}$, satisfies

$$\mu_{\ell, f_{k\to i}\to\boldsymbol{x}_i^t} \propto \int f_{k\to i}b_{\ell-1}(\boldsymbol{x}_k^t)\mathrm{d}\boldsymbol{x}_k^t, \quad (14)$$

where $b_{\ell-1}(\boldsymbol{x}_k^t) = \delta\left(\boldsymbol{x}_k^t - \hat{\boldsymbol{x}}_k^t\right)$, $\delta(\cdot)$ is Dirac delta function.

According to (1), the factor $f_{k\to i}$, which represents the likelihood function of $z_{k\to i}^t$, satisfies

$$f_{k\to i} = \frac{1}{\sqrt{2\pi\sigma_{k\to i}^2}} \exp\left\{-\frac{\left(z_{k\to i}^t - \left\|\boldsymbol{x}_i^t - \boldsymbol{x}_k^t\right\|_2\right)^2}{2\sigma_{k\to i}^2}\right\}. \quad (15)$$

By substituting $b_{\ell-1}(\boldsymbol{x}_k^t)$ and (15) into (14), we obtain[8]

$$\mu_{\ell, f_{k\to i}\to\boldsymbol{x}_i^t} \propto \exp\left\{-\frac{\left(z_{k\to i}^t - d\left(\boldsymbol{x}_i^t, \hat{\boldsymbol{x}}_k^t\right)\right)^2}{2\sigma_{k\to i}^2}\right\}. \quad (16)$$

---

[5]Standard BP and SPAWN can operate on unnormalized beliefs [1], [13].

[6]This nonlinearity and the resulting intractability of the integral, are the fundamental reasons why approximation methods needs to be adopted.

[7]We note that using the second-order TP strikes a good balance between the computational complexity (see Table I and Table II) and the performance (see Fig. 5, etc.). Hence, without loss of generality, only the closed-form expression using the second-order TP is presented here.

[8]The positions of anchors are known constants during the iterations.

By substituting the second-order TP of $d(\boldsymbol{x}_i^t, \hat{\boldsymbol{x}}_k^t)$ at the first point $\hat{\boldsymbol{x}}_{i,\ell-1}^t = \left[\hat{x}_{i,\ell-1}^t, \hat{y}_{i,\ell-1}^t\right]^{\mathrm{T}}$ in (16), where $\hat{x}_{i,\ell-1}^t$ and $\hat{y}_{i,\ell-1}^t$ are the estimates of x_i^t and y_i^t at iteration $\ell-1$, respectively, the parametric representation of $\mu_{\ell, f_{k\to i} \to \boldsymbol{x}_i^t}$ is given by

$$\mu_{\ell, f_{k\to i} \to x_i^t} \propto \exp\left\{\boldsymbol{\beta}_{k,\ell}^{\mathrm{T}} \boldsymbol{\psi}_i\right\}, \tag{17}$$

where

$$\boldsymbol{\beta}_{k,\ell} = \left[\beta_{k1,\ell}, \beta_{k2,\ell}, \beta_{k3,\ell}, \beta_{k4,\ell}\right]^{\mathrm{T}}, \tag{18}$$

with

$$\begin{cases} \beta_{k1,l} = \dfrac{\|\boldsymbol{e}\|_2^3 - z_{k\to i}e_2^2}{2\sigma_{k\to i}^2\|\boldsymbol{e}\|_2^3}, \\[2mm] \beta_{k2,l} = \dfrac{\|\boldsymbol{e}\|_2^3 - z_{k\to i}e_1^2}{2\sigma_{k\to i}^2\|\boldsymbol{e}\|_2^3}, \\[2mm] \beta_{k3,l} = \dfrac{z_{k\to i}e_1\|\boldsymbol{e}\|_2^2 - z_{k\to i}e_2^2\hat{x}_k^t + z_{k\to i}e_1 e_2\hat{y}_k^t + \|\boldsymbol{e}\|_2^3\hat{x}_k^t}{\sigma_{k\to i}^2\|\boldsymbol{e}\|_2^3}, \\[2mm] \beta_{k4,l} = \dfrac{z_{k\to i}e_2\|\boldsymbol{e}\|_2^2 - z_{k\to i}e_1^2\hat{y}_k^t + z_{k\to i}e_1 e_2\hat{x}_k^t + \|\boldsymbol{e}\|_2^3\hat{y}_k^t}{\sigma_{k\to i}^2\|\boldsymbol{e}\|_2^3}, \\[2mm] \boldsymbol{e} = [e_1, e_2] = \left[\hat{x}_{i,\ell-1}^t - \hat{x}_k^t, \hat{y}_{i,\ell-1}^t - \hat{y}_k^t\right]^{\mathrm{T}}. \end{cases} \tag{19}$$

At time slot t and iteration ℓ, the spatial message received by agent i from agent $j \in \mathbb{U}_i^t$, i.e., $\mu_{\ell, f_{j\to i} \to x_i^t}$, is given by

$$\mu_{\ell, f_{j\to i} \to x_i^t} \propto \int \exp\left\{-\frac{\left(z_{j\to i}^t - d(\boldsymbol{x}_i^t, \boldsymbol{x}_j^t)\right)^2}{2\sigma_{j\to i}^2}\right\} b_{\ell-1}(\boldsymbol{x}_j^t) \mathrm{d}\boldsymbol{x}_j^t, \tag{20}$$

where

$$b_{\ell-1}(\boldsymbol{x}_j^t) \propto \exp\left\{-\frac{1}{2}(\boldsymbol{x}_j^t - \hat{\boldsymbol{x}}_{j,\ell-1}^t)^{\mathrm{T}} \boldsymbol{\Sigma}_{\boldsymbol{x}_j^t}^{-1}(\boldsymbol{x}_j^t - \hat{\boldsymbol{x}}_{j,\ell-1}^t)\right\}, \tag{21}$$

$$\hat{\boldsymbol{x}}_{j,\ell-1}^t = \left[\hat{x}_{j,\ell-1}^t, \hat{y}_{j,\ell-1}^t\right]^{\mathrm{T}}, \tag{22}$$

and

$$\boldsymbol{\Sigma}_{\boldsymbol{x}_j^t} = \mathrm{diag}\left(\sigma_{x_j^t,\ell-1}^2, \sigma_{y_j^t,\ell-1}^2\right), \tag{23}$$

in which $\sigma_{x_j^t,\ell-1}^2$ and $\sigma_{y_j^t,\ell-1}^2$ are the variances of x_j^t and y_j^t at iteration $l-1$, respectively, while $\hat{\boldsymbol{x}}_{j,\ell-1}^t$ is the estimate of $\boldsymbol{x}_j^t$ at iteration $\ell-1$. Again, we utilize the second-order TP of $d(\boldsymbol{x}_i^t, \hat{\boldsymbol{x}}_j^t)$ at points $\left[\hat{x}_{i,\ell-1}^t, \hat{y}_{i,\ell-1}^t\right]^{\mathrm{T}}$ and $\left[\hat{x}_{j,\ell-1}^t, \hat{y}_{j,\ell-1}^t\right]^{\mathrm{T}}$ to approximate it. Upon substituting (21) and this second-order TP into (20), the spatial message $\mu_{\ell, f_{j\to i} \to \boldsymbol{x}_i^t}$ is approximated by

$$\mu_{\ell, f_{j\to i} \to \boldsymbol{x}_i^t} \propto \exp\left\{\boldsymbol{\gamma}_{j,\ell}^{\mathrm{T}} \boldsymbol{\psi}_i\right\}, \tag{24}$$

where

$$\boldsymbol{\gamma}_{j,\ell} = \left[\gamma_{j1,\ell}, \gamma_{j2,\ell}, \gamma_{j3,\ell}, \gamma_{j4,\ell}\right]^{\mathrm{T}}, \tag{25}$$

with

$$\begin{cases} \gamma_{j1,\ell} = \dfrac{\|\boldsymbol{c}\|_2^3 - z_{j\to i}c_2^2}{2\sigma_{j\to i}^2\|\boldsymbol{c}\|_2^3}, \\[2mm] \gamma_{j2,\ell} = \dfrac{\|\boldsymbol{c}\|_2^3 - z_{j\to i}c_1^2}{2\sigma_{j\to i}^2\|\boldsymbol{c}\|_2^3}, \\[2mm] \gamma_{j3,\ell} = \dfrac{z_{j\to i}c_1}{\sigma_{j\to i}^2\|\boldsymbol{c}\|_2}, \\[2mm] \gamma_{j4,\ell} = \dfrac{z_{j\to i}c_2}{\sigma_{j\to i}^2\|\boldsymbol{c}\|_2}, \\[2mm] \boldsymbol{c} = [c_1, c_2] \\ \quad = \left[\hat{x}_{i,\ell-1}^t - \hat{x}_{j,\ell-1}^t, \hat{y}_{i,\ell-1}^t - \hat{y}_{j,\ell-1}^t\right]^{\mathrm{T}}. \end{cases} \tag{26}$$

Here $\hat{x}_{i,\ell-1}^t$ and $\hat{y}_{i,\ell-1}^t$ are the estimates of x_i^t and y_i^t at iteration $\ell-1$, respectively. Agent i can determine $\hat{x}_{i,\ell-1}^t$ and $\hat{y}_{i,\ell-1}^t$ by taking the mean value involved in its belief function at iteration $\ell-1$ [1], [14], [16].

When iteration $\ell_{\max}$ is reached, upon substituting (10), (17), and (24) into (4), the *a posteriori* distribution of the position of agent i at time slot t can be expressed as

$$p(\boldsymbol{x}_i^t|\boldsymbol{z}_i^t)$$
$$\propto \exp\left\{-\frac{1}{2}(\boldsymbol{x}_i^t - \mathrm{E}\{\boldsymbol{x}_i^t|\boldsymbol{z}_i^t\})^{\mathrm{T}} \boldsymbol{\Sigma}_{\boldsymbol{x}_i^t|\boldsymbol{z}_i^t}^{-1}(\boldsymbol{x}_i^t - \mathrm{E}\{\boldsymbol{x}_i^t|\boldsymbol{z}_i^t\})\right\}, \tag{27}$$

where

$$\mathrm{E}\{\boldsymbol{x}_i^t|\boldsymbol{z}_i^t\} = \left[\frac{\sum_{k\in\mathbb{A}_i^t}\beta_{k3,\ell_{\max}} + \sum_{j\in\mathbb{U}_i^t}\gamma_{j3,\ell_{\max}} + \alpha_{i3}}{2\left(\sum_{k\in\mathbb{A}_i^t}\beta_{k1,\ell_{\max}} + \sum_{j\in\mathbb{U}_i^t}\gamma_{j1,\ell_{\max}} + \alpha_{i1}\right)},\right.$$
$$\left.\frac{\sum_{k\in\mathbb{A}_i^t}\beta_{k4,\ell_{\max}} + \sum_{j\in\mathbb{U}_i^t}\gamma_{j4,\ell_{\max}} + \alpha_{i4}}{2\left(\sum_{k\in\mathbb{A}_i^t}\beta_{k2,\ell_{\max}} + \sum_{j\in\mathbb{U}_i^t}\gamma_{j2,\ell_{\max}} + \alpha_{i2}\right)}\right]^{\mathrm{T}}, \tag{28}$$

and

$$\boldsymbol{\Sigma}_{\boldsymbol{x}_i^t|\boldsymbol{z}_i^t} = \mathrm{diag}\left(\left(\sum_{k\in\mathbb{A}_i^t}\beta_{k1,\ell_{\max}} + \sum_{j\in\mathbb{U}_i^t}\gamma_{j1,\ell_{\max}} + \alpha_{i1}\right)^{-1},\right.$$
$$\left.\left(\sum_{k\in\mathbb{A}_i^t}\beta_{k2,\ell_{\max}} + \sum_{j\in\mathbb{U}_i^t}\gamma_{j2,\ell_{\max}} + \alpha_{i2}\right)^{-1}\right), \tag{29}$$

Algorithm 1 TP-SPAWN

Require: $p(\boldsymbol{x}_i^{t-1}|\boldsymbol{z}_i^{t-1})$, $\boldsymbol{z}_i^t$.
Ensure: $p(\boldsymbol{x}_i^t|\boldsymbol{z}_i^t)$.
    for agent $i \in \mathbb{U}$ **do**
        compute the temporal message according to (10).
        for iteration $\ell = 1$ to $\ell_{\max}$ **do**
            broadcast $b_{\ell-1}(\boldsymbol{x}_i^t)$.
            receive $b_{\ell-1}(\boldsymbol{x}_j^t)$.
            compute the corresponding spatial messages according to (17) and (24).
            update its belief $b_\ell(\boldsymbol{x}_i^t)$.
        end for
        compute the *a posteriori* distribution of its position according to (27).
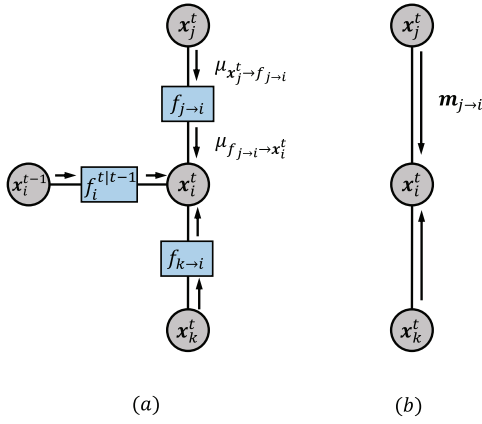   **end for**

Fig. 2. (a) Visual representation of the local FG of agent i at time slot t. (b) The equivalent representation of the FG in GNN (referred to as *GNN graph* in what follows). The temporal message based *a priori* distribution of agent i is treated as node attribute vector of agent i in the GNN graph.

The resulting TP-SPAWN is presented in Algorithm 1. At each iteration ℓ, distributed agent i only needs to broadcast its own position $\hat{x}_{i,\ell-1}^t, \hat{y}_{i,\ell-1}^t$ and the position variances $\sigma_{x_i^t,\ell-1}^2, \sigma_{y_i^t,\ell-1}^2$.

Note that SPAWN does not guarantee to obtain the correct *a posteriori* distribution, due to the fact that it applies approximate beliefs in order to reduce computational complexity for distributed implementation. In addition, due to the consideration of the covariance matrix of the cooperative agents' position, our TP-SPAWN may degrade the confidence level of the spatio-temporal messages [14], and may result in accumulation of positioning errors. In the next section, we consider how to improve the parametric SPAWN based spatial messages.

IV. GNN BASED MESSAGE REFINEMENT

Based on the MPNN framework of [20], this section presents a model- and data-driven hybrid inference approach, called GNN-STMP, to fine-tune the parametric message representations obtained in Section III, and obtains more accurate *a posteriori* distribution of agents' positions by exploiting GNN-generated messages. Fig. 2 illustrates the comparison of a typical local FG of agent i and the corresponding graph in GNN.

A. MPNN Procedure

MPNNs are a class of neural networks designed for processing graph-structured data. They operate by passing messages between nodes of a graph and updating the node states based on these messages, allowing for the incorporation of the structural information of the graph into the learning process. The MPNN procedure consists of a message passing phase and a readout phase [20].

In the message passing phase, agent i first aggregates the GNN-generated messages passed from its neighbor nodes on the corresponding graph of the locally FG by

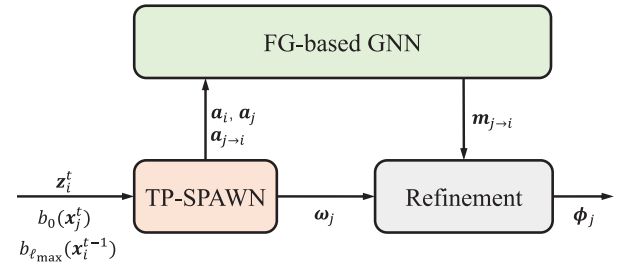$$m_i = \sum_{j \in \mathbb{A}_i^t \cup \mathbb{U}_i^t} m_{j \to i}, \tag{30}$$

Fig. 3. The graphical illustration of GNN-STMP.

where the GNN-generated messages from j to i satisfy

$$m_{j \to i} = o_1\left(h_j, h_i, a_{j \to i}\right). \tag{31}$$

Here $o_1(\cdot)$ is the message function, h_i is the node embedding vector that contains the abstract features, and $a_{j \to i}$ represents the edge attribute vector from j to i. Then agent i updates its node embedding vector by

$$\widetilde{h}_i = o_2\left(h_i, m_i\right), \tag{32}$$

where $o_2(\cdot)$ is the update function of agent i.

After several rounds of message passing and state updates, a readout function is applied to aggregate node states into a graph-level representation, which can be used for graph classification, node classification, or other tasks [20], [23], [24], [25].

B. GNN-STMP

The overall GNN-STMP structure, as shown in Fig. 3, is explained below. Using the TP-SPAWN module, we first obtain the closed-form representations of individual messages passed on the FG. Then, these high-accuracy message representations are treated as the inputs of the FG-based GNN module. Subsequently, the refinement module leverages the GNN-generated messages $m_{k \to i,\ell}$ and $m_{j \to i,\ell}$, to enhance the vector-form parametric message representations $\beta_{k,\ell}$ and $\gamma_{j,\ell}$, respectively, which are passed on the FG. As a result, the *a posteriori* probability of agent i's position is fine-tuned and its closed-form expression is obtained. The above procedure is repeated recursively for $\ell_{\max}$ iterations.

1) TP-SPAWN Module: With the inputs $\mathrm{E}\{x_i^{t-1}|z_i^{t-1}\}$ and $z_{i,\mathrm{int}}^t$, agent i first computes the parametric representation of its temporal message by using (13) before activating the iteration process. Building on the beliefs received from neighbors at iteration $\ell - 1$, i.e., $b_{\ell-1}(x_j^t)$ and $b_{\ell-1}(x_k^t)$, as well as the spatial ranging measurements, i.e., $z_{j \to i}^t$ and $z_{k \to i}^t$, agent i computes and outputs the parametric representations of each spatial message at iteration ℓ by using (19) and (26).

2) FG Based GNN Module: Corresponding to its local FG, agent i creates a distinct node embedding vector for each node on the GNN graph at iteration ℓ by

$$h_{i,\ell} = g_1\left(a_{i,\ell}\right), \tag{33}$$

where $a_{i,\ell} = \left[\hat{x}_{i,\ell-1}^t, \hat{y}_{i,\ell-1}^t, \sigma_{x_i^t,\ell-1}^2, \sigma_{y_i^t,\ell-1}^2\right]^{\mathrm{T}}$ denotes the node attribute vector. Meanwhile, the edge embedding vector

for each edge on the GNN graph, which corresponds to agent i's local FG, satisfies

$$\boldsymbol{h}_{j\to i,\ell} = g_2\left(\boldsymbol{a}_{j\to i,\ell}\right), \tag{34}$$

$$\boldsymbol{h}_{k\to i,\ell} = g_2\left(\boldsymbol{a}_{k\to i,\ell}\right), \tag{35}$$

where $\boldsymbol{a}_{j\to i,\ell}$ and $\boldsymbol{a}_{j\to i,\ell}$ are the edge attribute vectors of dimension 4 and are set as $\boldsymbol{a}_{j\to i,\ell} = \boldsymbol{\gamma}_{j,\ell}$ and $\boldsymbol{a}_{k\to i,\ell} = \boldsymbol{\beta}_{k,\ell}$, respectively. The GNN-generated messages sent from node $\boldsymbol{x}_j^t$ and node $\boldsymbol{x}_k^t$ to node $\boldsymbol{x}_i^t$ are given by

$$\boldsymbol{m}_{j\to i,\ell} = g_3\left(\boldsymbol{h}_{j\to i,\ell} \circ \boldsymbol{h}_{j,\ell}\right), \tag{36}$$

and

$$\boldsymbol{m}_{k\to i,\ell} = g_3\left(\boldsymbol{h}_{k\to i,\ell} \circ \boldsymbol{h}_{k,\ell}\right), \tag{37}$$

respectively, where $\circ$ denotes the element-wise multiplication. All of the mapping functions $g_1(\cdot)$, $g_2(\cdot)$ and $g_3(\cdot)$ are trainable GNNs.

Algorithm 2 GNN-STMP

Require: $p(\boldsymbol{x}_i^{t-1}|\boldsymbol{z}_i^{t-1})$, $\boldsymbol{z}_i^t$.
Ensure: $p(\boldsymbol{x}_i^t|\boldsymbol{z}_i^t)$.
    for agent $i \in \mathbb{U}$ **do**
        obtain the representation of its temporal message according to (10).
        for iteration $\ell = 1$ to $\ell_{\max}$ **do**
            broadcast $b_{\ell-1}(\boldsymbol{x}_i^t)$.
            receive $b_{\ell-1}(\boldsymbol{x}_j^t)$.
            obtain the representations of its spatial messages according to (17) and (24).
            compute the GNN-generated messages upon using (36) and (37).
            compute the refined spatial message upon using (40) and (41).
            update its belief $b_{\ell}(\boldsymbol{x}_i^t)$.
        end for
        compute the *a posteriori* distribution of its position upon using (42).
    end for

---

3) Refinement Module: Agent i exploits $\{\boldsymbol{m}_{j\to i,\ell}\}$ and $\{\boldsymbol{m}_{k\to i,\ell}\}$ to refine the spatial SPAWN message representations. The refined parametric spatial message representations are given by

$$\boldsymbol{\phi}_{j,\ell} = \left[\phi_{j1,\ell}, \phi_{j2,\ell}, \phi_{j3,\ell}, \phi_{j4,\ell}\right]^{\mathrm{T}}$$
$$= \boldsymbol{\gamma}_{j,\ell} g_4\left(\boldsymbol{m}_{j\to i,\ell}\right) + g_5\left(\boldsymbol{m}_{j\to i,\ell}\right), \tag{38}$$

$$\boldsymbol{\phi}_{k,\ell} = \left[\phi_{k1,\ell}, \phi_{k2,\ell}, \phi_{k3,\ell}, \phi_{k4,\ell}\right]^{\mathrm{T}}$$
$$= \boldsymbol{\beta}_{k,\ell} g_4\left(\boldsymbol{m}_{k\to i,\ell}\right) + g_5\left(\boldsymbol{m}_{k\to i,\ell}\right), \tag{39}$$

where $g_4(\cdot)$ and $g_5(\cdot)$ are GNNs with trainable parameters. Specifically, $g_4(\cdot)$ outputs a positive scalar and $g_5(\cdot)$ outputs a vector of dimension 4. The corresponding refined spatial messages satisfy

$$\zeta_{\ell, f_{j\to i}\to \boldsymbol{x}_i^t} \propto \exp\left\{\boldsymbol{\phi}_{j,\ell}^{\mathrm{T}}\boldsymbol{\psi}_i\right\}, \tag{40}$$

$$\zeta_{\ell, f_{k\to i}\to \boldsymbol{x}_i^t} \propto \exp\left\{\boldsymbol{\phi}_{k,\ell}^{\mathrm{T}}\boldsymbol{\psi}_i\right\}. \tag{41}$$

By substituting (10), (40) and (41) into (4), the *a posteriori* distribution of agent i's position at time slot t satisfies

$$p(\boldsymbol{x}_i^t|\boldsymbol{z}_i^t)$$
$$\propto \exp\left\{-\frac{1}{2}(\boldsymbol{x}_i^t - \mathrm{E}\{\boldsymbol{x}_i^t|\boldsymbol{z}_i^t\})^{\mathrm{T}}\boldsymbol{\Sigma}_{\boldsymbol{x}_i^t|\boldsymbol{z}_i^t}^{-1}(\boldsymbol{x}_i^t - \mathrm{E}\{\boldsymbol{x}_i^t|\boldsymbol{z}_i^t\})\right\}, \tag{42}$$

where

$$\mathrm{E}\{\boldsymbol{x}_i^t|\boldsymbol{z}_i^t\} = \left[\frac{\sum_{k\in\mathbb{A}_i^t}\phi_{k3,\ell_{\max}} + \sum_{j\in\mathbb{U}_i^t}\phi_{j3,\ell_{\max}} + \alpha_{i3}}{2\left(\sum_{k\in\mathbb{A}_i^t}\phi_{k1,\ell_{\max}} + \sum_{j\in\mathbb{U}_i^t}\phi_{j1,\ell_{\max}} + \alpha_{i1}\right)}, \right.$$
$$\left.\frac{\sum_{k\in\mathbb{A}_i^t}\phi_{k4,\ell_{\max}} + \sum_{j\in\mathbb{U}_i^t}\phi_{j4,\ell_{\max}} + \alpha_{i4}}{2\left(\sum_{k\in\mathbb{A}_i^t}\phi_{k2,\ell_{\max}} + \sum_{j\in\mathbb{U}_i^t}\phi_{j2,\ell_{\max}} + \alpha_{i2}\right)}\right]^{\mathrm{T}}, \tag{43}$$

and

$$\boldsymbol{\Sigma}_{\boldsymbol{x}_i^t|\boldsymbol{z}_i^t} = \mathrm{diag}\left(\left(\sum_{k\in\mathbb{A}_i^t}\phi_{k1,\ell_{\max}} + \sum_{j\in\mathbb{U}_i^t}\phi_{j1,\ell_{\max}} + \alpha_{i1}\right)^{-1}, \right.$$
$$\left.\left(\sum_{k\in\mathbb{A}_i^t}\phi_{k2,\ell_{\max}} + \sum_{j\in\mathbb{U}_i^t}\phi_{j2,\ell_{\max}} + \alpha_{i2}\right)^{-1}\right). \tag{44}$$

At any time slot t, each agent can determine the minimum mean squared error (MMSE) estimate of its own position by taking the mean value involved in $p(\boldsymbol{x}_i^t|\boldsymbol{z}_i^t)$. Our GNN-STMP is presented in Algorithm 2.

C. Loss Function

The training loss is computed from the estimated positions $\hat{\boldsymbol{x}}_i^t$ and the ground truth $\tilde{\boldsymbol{x}}_i^t$ according to:

$$\mathrm{Loss}(\Theta) = \mathcal{L}\left(\hat{\boldsymbol{x}}_i^t, \tilde{\boldsymbol{x}}_i^t\right) + \mathcal{R}, \tag{45}$$

where Θ represents the all the trainable parameters of the GNN. We apply the mean squared error loss for $\mathcal{L}(\cdot)$, and the regularization term $\mathcal{R}$ is chosen as

$$\mathcal{R} = \eta\frac{1}{|\mathbb{A}_i^t \cup \mathbb{U}_i^t|}\sum_{j\in\mathbb{A}_i^t\cup\mathbb{U}_i^t}\|g_4\left(\boldsymbol{m}_{j\to i}^t\right)\|_1, \tag{46}$$

which encourages the refined messages to be closer to the standard BP messages [13], helps avoid overfitting, and further enhances the model's generalization ability. In (46), η is the regularization parameter, $|\mathbb{A}_i^t \cup \mathbb{U}_i^t|$ denotes the cardinal number of set $\mathbb{A}_i^t \cup \mathbb{U}_i^t$, and $\|\cdot\|_1$ denotes the L_1-norm.

V. THE EKF-GNN-DCP FRAMEWORK

The positioning capability of mobile wireless networks can be significantly degraded by agents lacking sufficient spatial ranging measurements, a phenomenon that is particularly acute in networks with sparsely distributed agents. This section presents a universal framework, dubbed EKF-GNN-DCP, to address this issue. The proposed EKF-GNN-DCP integrates GNN-STMP with the EKF based agents' state prediction and refinement, which ensures that agents can obtain reliable positional estimates in the challenging sparsely distributed mobile wireless networks, through both enhancing the accuracy of
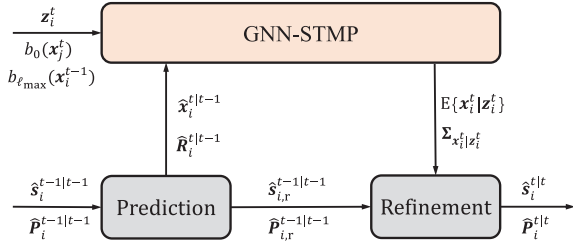
Fig. 4. The schematic diagram of EKF-GNN-DCP, where $\hat{s}_{i,\mathrm{r}}^{t-1|t-1}$ denotes the rest parameters in $\hat{s}_i^{t-1|t-1}$ other than $\hat{x}_i^{t-1|t-1}$, and $\hat{P}_{i,\mathrm{r}}^{t-1|t-1}$ denotes the rest parameters in $\hat{P}_i^{t-1|t-1}$ other than $\hat{R}_i^{t-1|t-1}$.

message representation and leveraging the state constraints of agents. Our EKF-GNN-DCP consists of three stages: 1) prediction, 2) GNN-STMP, and 3) refinement. The schematic diagram of EKF-GNN-DCP is illustrated in Fig. 4.

A. Prediction

In this step, agent i first provides a coarse prediction of its state at time slot t. Let s_i^t be the state of agent i at time slot t. We consider a common state transition model as

$$s_i^t = F s_i^{t-1} + w_i^t, \tag{47}$$

where F denotes the state transition matrix given by

$$F = \begin{bmatrix} I_2 & \Delta T I_2 \\ 0_2 & I_2 \end{bmatrix}, \tag{48}$$

in which I_2 and 0_2 represent the 2×2 identity matrix and zero matrix, respectively, and ΔT is the duration of a single time slot, while w_i^t represents the state transition noise that is modeled by the 4-dimensional Gaussian vector with zero mean and covariance matrix Q_i^t.

Hence, given the estimated mean and covariance of the state for again i at time $t-1$, $\hat{s}_i^{t-1|t-1}$ and $\hat{P}_i^{t-1|t-1}$, the predicted mean and the covariance of the state for agent i are given by

$$\hat{s}_i^{t|t-1} = F \hat{s}_i^{t-1|t-1}, \tag{49}$$

$$\hat{P}_i^{t|t-1} = F \hat{P}_i^{t-1|t-1} F^\mathrm{T} + Q_i^t, \tag{50}$$

respectively. $\hat{s}_i^{t|t-1}$ and $\hat{P}_i^{t|t-1}$ constitute the mean and covariance of the *a priori* distribution for the next step.

B. GNN-STMP

In this step, agent i utilizes GNN-STMP to fuse the nonlinear soft information.

First, based on the predicted mean and covariance, $\hat{s}_i^{t|t-1}$ and $\hat{P}_i^{t|t-1}$, the *a priori* distribution of agent i's position at time slot t is given by $x_i^t \sim \mathcal{N}(\hat{x}_i^{t|t-1}, \hat{R}_i^{t|t-1})$, where $\hat{x}_i^{t|t-1} = \left[\hat{x}_i^{t|t-1}, \hat{y}_i^{t|t-1}\right]^\mathrm{T}$ denotes the position components[9] in $\hat{s}_i^{t|t-1}$, and $\hat{R}_i^{t|t-1} = \mathrm{diag}\left((\hat{\sigma}_{i,x}^{t|t-1})^2, (\hat{\sigma}_{i,y}^{t|t-1})^2\right)$, with $(\hat{\sigma}_{i,x}^{t|t-1})^2$ and $(\hat{\sigma}_{i,y}^{t|t-1})^2$ being the $(1,1)$th and $(2,2)$th elements of $\hat{P}_i^{t|t-1}$, respectively.

[9]The velocity components are not involved in the message passing procedure.

Then we run the iterative GNN-STMP procedure based on the FG of $p\left(x_i^t \mid z_i^t\right)$, obtaining parametric representations of each kind of FG messages and fine-tuning them by exploiting the GNN-generated messages. Next, we obtain the enhanced mean vector $\mathrm{E}\{x_i^t|z_i^t\}$ and covariance matrix $\Sigma_{x_i^t|z_i^t}$.

C. Refinement

In this step, agent i uses the update step of EKF to refine its *a posteriori* distribution. Consider the observation model for agent i given by

$$\mathrm{E}\{x_i^t|z_i^t\} = H s_i^t + \nu_i^t, \tag{51}$$

where $H = \begin{bmatrix} I_2 & 0_2 \end{bmatrix}$ is the observation matrix, and ν_i^t is a Gaussian observation noise with zero mean. The measurement residual Δm_i^t and its covariance matrix C_i^t are given by

$$\Delta m_i^t = \mathrm{E}\{x_i^t|z_i^t\} - H \hat{s}_i^{t|t-1}, \tag{52}$$

$$C_i^t = H \hat{P}_i^{t|t-1} H^\mathrm{T} + \Sigma_{x_i^t|z_i^t}, \tag{53}$$

respectively. Thus the near-optimal Kalman gain is given by

$$K_i^t = \hat{P}_i^{t|t-1} H^\mathrm{T} \left(C_i^t\right)^{-1}, \tag{54}$$

and the marginal state distribution is refined by weighting its mean and covariance with the measurement residual:

$$\hat{s}_i^{t|t} = \hat{s}_i^{t|t-1} + K_i^t \Delta m_i^t, \tag{55}$$

$$\hat{P}_i^{t|t} = \hat{P}_i^{t|t-1} - K_i^t C_i^t \left(K_i^t\right)^\mathrm{T}. \tag{56}$$

Algorithm 3 EKF-GNN-DCP

Require: $\hat{s}_i^{t-1|t-1}, \hat{P}_i^{t-1|t-1}, z_i^t$.
Ensure: $\hat{s}_i^{t|t}, \hat{P}_i^{t|t}$.
    for agent $i = 1$ to N **do**
        compute the predict distribution according to (49) and (50).
        obtain the representation of its temporal message according to (10).
        for iteration $\ell = 1$ to $\ell_{\max}$ **do**
            broadcast $b_{\ell-1}(x_i^t)$.
            receive $b_{\ell-1}(x_j^t)$ and $b_{\ell-1}(x_k^t)$, and obtain the representation of the corresponding spatial messages according to (17) and (24).
            compute the GNN-generated messages upon using (36) and (37).
            compute the refined spatial messages upon using (40) and (41).
            update its belief $b_\ell(x_i^t)$.
        end for
        compute the *a posteriori* distribution upon using (42).
        refine the *a posteriori* state upon using (55) and (56).
    end for

In conclusion, the resulting EKF-GNN-DCP algorithm is summarized in Algorithm 3. Furthermore, the iterative nature of the BP ensures that our framework is robust to occasional message loss or communication errors. Since BP relies on iterative updates, if a message is lost during one iteration,

TABLE I

ANALYSIS OF COMPUTATIONAL COMPLEXITY

Genre	Computational Complexity	Representative Algorithms
Parametric SPAWN	$\mathcal{O}(N_{\text{rel}})$	SPA-TE [15], EKF-STDF [7], TP-SPAWN, GNN-STMP, EKF-GNN-DCP
Particle-based SPAWN	$\mathcal{O}(N_s^2 N_{\text{rel}})$	SPA-EKF [17], NEBP [21], STOC-EKF [19], GSPAWN [14]
Particle-based BP	$\mathcal{O}(N_s^2 N_{\text{rel}}^2)$	particle-based standard BP [32], [33]

TABLE II

COMPARISON OF BASIC ARITHMETIC OPERATIONS ACROSS DIFFERENT TAYLOR APPROXIMATION STRATEGIES WHEN APPROXIMATING THE NONLINEAR TERM D ONCE, FOR A SINGLE MESSAGE, A SINGLE NEIGHBOR AND A SINGLE ITERATION

Strategy	Addition	Subtraction	Multiplication	Division	Square Root
First-order	1	2	4	2	1
Second-order	5	8	14	6	1
Third-order	12	19	26	10	1

subsequent iterations will incorporate new information from neighboring nodes to mitigate this. This feature helps maintain the overall accuracy of DCP despite communication uncertainties.

VI. COMPUTATIONAL COMPLEXITY ANALYSIS

We first analyze the computational complexity of the main classes of message passing based DCP algorithms available so far, i.e., the parametric SPAWN, the particle-based SPAWN, and the particle-based BP, in Table I. We use N_s and N_{rel} to denote the number of particles and neighbors, respectively. Given that this paper focuses on distributed schemes, only the computational complexity of a single agent in a single iteration is considered. The computational complexity of the aforementioned message passing schemes primarily stems from *message representation* and *message multiplication* [14]. For parametric SPAWN, message representation requires only a limited set of parameters, and message multiplication typically yields closed-form expressions of beliefs. In contrast, the particle-based SPAWN methods have to represent messages with a large number of particles and their associated weights, entailing extensive computations during both message representation and message multiplication. The particle-based BP methods need to represent and compute different messages for individual neighbors, which results in higher computational complexity than the particle-based SPAWN methods. Notably, our proposed TP-SPAWN, which utilizes vectors for message representation, allows its corresponding GNN enhanced scheme, i.e., GNN-STMP, to maintain high computational efficiency in message multiplication and obtain closed-form expression for the *a posteriori* distribution of agents' positions. This has not been fulfilled by the existing GNN enhanced message passing algorithms [13], [21], [22]. Furthermore, if the existing representative parametric methods [6], [7], [14], [15] could be integrated with GNN, they would still rely on particle-based approaches to compute the *a posteriori* distributions in message multiplication, which results in the lack of closed-form expressions.

Next, we consider the number of basic arithmetic operations required when using different orders of the TP strategy. Given that this study focuses on a distributed scheme, we only account for the process of approximating the nonlinear term

d during a single iteration of information processing by an individual agent. Table II lists the number of basic arithmetic operations required for each TP strategy. We observe that the third-order TP significantly increases the number of subtractions, multiplications, and divisions compared with the first- and second-order TPs. It is important to note that in DCP tasks, each agent must use a TP to update its position estimate for every message at each time step and iteration. Specifically, the third-order TP requires a total of 68 basic operations per iteration, whereas the second-order TP requires only 34 operations, effectively halving the computational load.

VII. SIMULATION RESULTS AND DISCUSSIONS

In this section, we provide numerical results to evaluate our proposed DCP algorithms in mobile wireless networks, by using several representative state-of-the-art DCP schemes as benchmarks. In general, the experiments conducted involve two different mobile network configurations. In the first setting, it is rare for any particular agent to lack a sufficient number of neighbors in a long period, thus the occurrence of loopy network topologies is scarce. Conversely, in the sparse node distribution scenario, agents may often experience a shortage of neighbor nodes during their movements. We also investigate the impact of agents' mobility on the positioning performance, and analyzed.

To verify the effectiveness of our proposed high-precision message representation method, GNN-based message refinement, and EKF-GNN-DCP framework, we selected several representative baseline algorithms. These algorithms encompass both parametric and particle-based DCP methods, which are applicable to different network conditions. Firstly, Particle-based SPAWN [1] and Particle-based BP [8] are classical particle-based DCP methods that perform well in scenarios with sufficient neighbors. Among them, Particle-based BP offers higher positioning accuracy compared to Particle-based SPAWN, albeit at the cost of higher computational complexity and communication overhead. Both methods perform excellently when agents have sufficient neighbors, particularly with the use of thousands of particles. However, neither method provides strategies for addressing scenarios with insufficient neighbors. Secondly, GSPAWN [32] and SPA-TE [15] represent advancements in parametric methods. These methods also

TABLE III
DETAILS OF THE GNN STRUCTURES

$g_1(\cdot)$		$g_2(\cdot)$		$g_3(\cdot)$		$g_4(\cdot)$		$g_5(\cdot)$	
Input	4×1	Input	4×1	Input	16×1	Input	16×1	Input	16×1
Linear+ReLU	64×1	Linear+ReLU	64×1	Linear+ReLU	32×1	Linear+ReLU	16×1	Linear+ReLU	16×1
Linear+ReLU	32×1	Linear+ReLU	32×1	Linear+ReLU	32×1	Linear	1×1	Linear	4×1
Linear+ReLU	16×1	Linear+ReLU	16×1	Linear+ReLU	16×1				

perform well when agents have sufficient neighbors, with the benefit of relatively low computational complexity. However, like the particle-based methods, they lack solutions for scenarios with insufficient neighbors. Thirdly, LSTM-MPNN [30] and NEBP [21] are deep learning-based DCP methods proposed in recent years. Notably, NEBP enhances the accuracy of message passing using GNN and is currently the best-performing algorithm in sufficient-neighbor scenarios, except for our proposed methods. However, these methods also do not offer solutions for scenarios with insufficient neighbors. Lastly, to validate the effectiveness of our EKF-GNN-DCP framework in scenarios where agents have insufficient neighbors, we selected STOC-EKF [19], SPA-EKF [17], and EKF-STDF [7] as baselines. These methods integrate EKF techniques to effectively predict and refine agent states. In particular, SPA-EKF and EKF-STDF represent particle-based and parametric approaches, respectively.

A. Mobile Networks Where Agents Have Sufficient Neighbors

1) Training Dataset and Training Settings: We consider a wireless network composed of 30 agents and 5 static anchors in a $[0, 300]$ m$\times[0, 300]$ m area. Agents are uniformly scattered across a $[30, 270]$ m$\times[30, 270]$ m plane. The ranging radius is set to 90 m. The variance of the ranging noise is 0.5 m^2. The maximum number of iterations $\ell_{\max}$ is set to 20. Each time slot is assumed to be one second. At time slot t, each agent moves a distance $d_i^t \sim \mathcal{N}(2, 1)$ in a direction $\theta_i^t \sim \mathcal{U}(0, 2\pi)$,[10] where $\mathcal{U}(0, 2\pi)$ denotes the uniform distribution between 0 and 2π. We collect 600 realizations of the agents' trajectories for 20 time slots as the training data. All the GNNs have input layer, one or two hidden layers and output layer, while the activation functions of $g_1(\cdot)$, $g_2(\cdot)$, $g_3(\cdot)$, $g_4(\cdot)$ and $g_5(\cdot)$ are mainly rectified linear units (ReLUs). The regularization parameter η is set to 0.5. Adam optimizer [34] is employed to train GNN with batch size of 1. The learning rate declines from 10^{-3} to 10^{-5} with 10^{-1} decay rate for every 10 epochs. The details of GNN structures are shown in Table III.

2) Experimental Results and Discussions: To verify the generalization of our models, we conduct evaluation experiments with a different scale wireless network. Specifically, the network covers an area of $[0, 900]$ m$\times[0, 900]$ m with 13 static anchors, while 50 agents are uniformly scattered across an area of $[100, 800]$ m$\times[100, 800]$ m. The ranging radius to perform external measurements is set to 180 m. The variance of the ranging noise is 3 m^2. At time slot t, each agent moves a distance $d_i^t \sim \mathcal{N}(v, 1)$ in a direction $\theta_i^t \sim \mathcal{U}(0, 2\pi)$. To maintain a constant number of agents in the network, we replace a new

[10]Typically agents do not know in which direction they move, but they do know the distance they travel by internal ranging measurements.
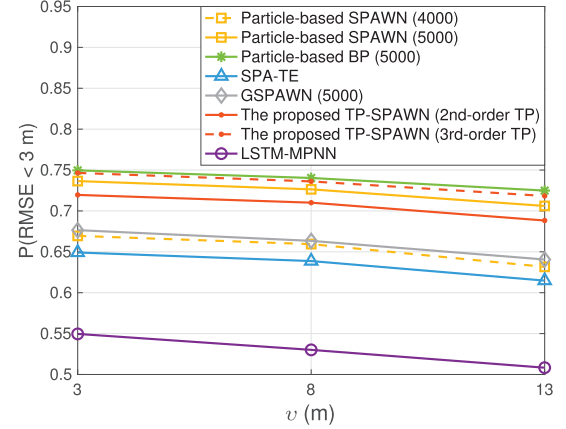

Fig. 5. The probability that the average positioning error (i.e., RMSE) of agents is less than 3m under different values of v, namely the distance moved by agents per time slot.

agent whenever an existing agent has left the considered area. The positioning performance is characterized by the root mean squared error (RMSE) and the average is obtained by Monte-Carlo simulations.

In the first experiment, we aim to validate the message representation accuracy of the proposed TP-SPAWN algorithm. Since TP-SPAWN is a foundational algorithm designed for scenarios where agents have sufficient neighbors and does not involve any accuracy refinement operations, we have selected the most representative baseline algorithms, all of which also do not include any accuracy refinement mechanisms. Specifically, we compare the second-order (labelled as "The proposed TP-SPAWN (2nd-order TP)") and third-order TP-SPAWN (labelled as "The proposed TP-SPAWN (3rd-order TP)") against particle-based SPAWN [1], particle-based standard BP [32], [33], SPA-TE [15], GSPAWN [14], and LSTM-MPNN [30], under different values of the distance moved by agents per time slot (denoted by parameter v, ranging from 3m to 13m), with $\ell_{\max}$ set to 20. Fig. 5 presents the RMSE performance comparison between the proposed TP-SPAWN and these baseline algorithms. We provide temporal ranging measurement to all the algorithms. The curves labeled "particle-based SPAWN (5000)", "particle-based BP (5000)" and "GSPAWN (5000)" denote the performance of particle-based SPAWN, particle-based BP and GSPAWN with 5000 particles, respectively. We have the following observations. Firstly, standard BP with 5000 particles outperforms the others. Unlike standard BP where each node sends an exact message to every neighbor, in SPAWN, each agent broadcasts an approximate standard BP message to its neighbors to reduce computational complexity. Therefore, SPAWN offers lower computational complexity than BP, at the cost of slightly
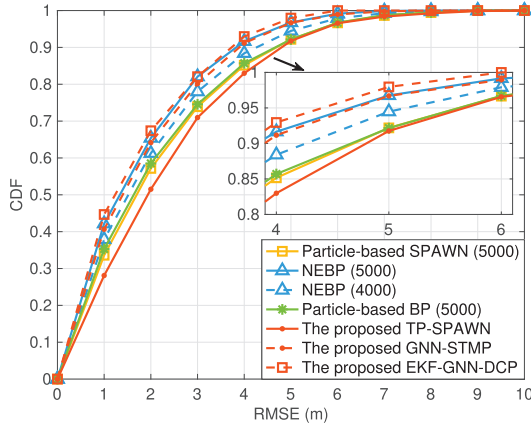
Fig. 6. CDFs of the positioning error (i.e., RMSE) for our TP-SPAWN, GNN-STMP and EKF-GNN-DCP, as well as for particle-based SPAWN, particle-based BP, and NEBP, given the number of agents $N = 60$.

Fig. 7. The ρ value of our TP-SPAWN and GNN-STMP, as well as that of the particle-based SPAWN and NEBP, under different numbers of agents.

inferior performance. Secondly, the positioning performance of both second-order and third-order TP-SPAWN is superior to those of SPAWN, SPA-TE, and GSPAWN, while slightly inferior to BP with 5000 particles. This observation indicates that our TP-SPAWN is a competitive algorithm in DCP. Considering that the computational complexity of parameter-based approaches is orders of magnitude lower than that of particle-based approaches, our TP-SPAWN has a significant advantage in practical DCP applications by offering dramatically lower computational complexity than BP with 5000 particles, only at the cost of slight performance loss. Thirdly, while the positioning performance of the third-order TP is better than that of the second-order TP, the improvement is marginal. Considering the distributed nature of the system, where each agent typically has limited computational resources, and the real-time requirements of the positioning task, the additional computational burden introduced by the third-order TP is significant. Therefore, using the second-order TP strikes a good balance between computational efficiency and approximation accuracy, ensuring timely and effective updates during the positioning process.

We next evaluate the effectiveness of the proposed GNN-STMP and EKF-GNN-DCP to improve on our TP-SPAWN when agents have sufficient neighbors. Given that Particle-based SPAWN and Particle-based BP are the two best-performing methods without accuracy refinement, and NEBP is the only advanced method with GNN-based refinement, we selected these three algorithms as the baseline methods. In this experiment, the number of agents is set to $N = 60$, $\ell_{max} = 20$ and $\upsilon = 3$. Fig. 6 shows the cumulative distribution functions (CDFs) of the position error, which is defined as the average RMSE between the estimated position and the true position, for our TP-SPAWN, GNN-STMP, and EKF-GNN-DCP,[11] as well as particle-based SPAWN, particle-based BP and NEBP [21]. The curve labeled "NEBP (5000)" denotes the performance of NEBP with 5000 particles. It can be seen that our TP-SPAWN is slightly inferior to BP and SPAWN

with 5000 particles, which again confirms the results of Fig. 5. Observe that NEBP with 5000 particles and our GNN-STMP outperform the others. Specifically, for NEBP and our GNN-STMP, 80% of agents achieve the position error smaller than 3 m, while this is increased to more than 3.5 m for the other three methods. Our GNN-STMP has an added advantage of imposing significantly lower computational complexity than NEBP. The results of Fig. 6 also clearly verify the effectiveness of the proposed GNN based message refinement to improve on our proposed parametric SPAWN. In addition, we observe that when there are sufficient neighboring nodes, the improvement in positioning accuracy from EKF-based state constraints is minimal, while the improvement from GNN-based message refinement is more significant. The result is consistent with intuition, as the EKF-based state prediction and refinement are designed to help the agent with positioning when there are insufficient neighboring nodes.

Also, the overall accuracy-consistency-confidence of various algorithms considered is investigated by using the normalized estimation error squared (NEES) metric [21]:

$$\text{NEES} = (\boldsymbol{x}_i^t - \text{E}\{\boldsymbol{x}_i^t | \boldsymbol{z}_i^t\})^{\text{T}} \boldsymbol{\Sigma}_{\boldsymbol{x}_i^t | \boldsymbol{z}_i^t}^{-1} (\boldsymbol{x}_i^t - \text{E}\{\boldsymbol{x}_i^t | \boldsymbol{z}_i^t\}). \quad (57)$$

For an effective estimator, the value of average NEES (ANEES) should approach the dimension of the estimate vector, i.e., the degree of freedom. For the sake of convenience, we define

$$\rho = \|\text{ANEES} - 2\|_1, \quad (58)$$

where 2 is the degree of freedom of $\boldsymbol{x}_i^t$. It is imperative to note that for a given estimator a smaller ρ value signifies enhanced consistency[12] and proper confidence level that is neither overly confident nor overly conservative.[13] However, we note that

---

[11]Unless otherwise specified, the curves labeled "The proposed TP-SPAWN", "The proposed GNN-STMP", and "The proposed EKF-GNN-DCP" in all the figures of this paper represent results obtained using the second-order TP approximations.
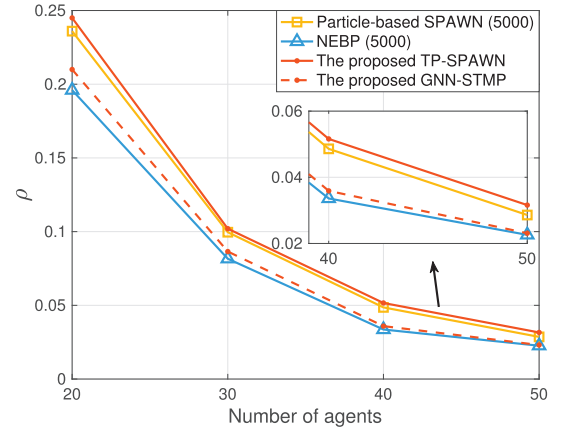
[12]An estimator is considered to be consistent if its estimation error matches its own reported uncertainty (i.e., error covariance). The ideal ANEES value should be close to its theoretical value or degrees of freedom, which indicates that the estimated uncertainty matches the actual observed error, i.e., the estimator is not only accurate, but also reliable in its own assessment of accuracy.

[13]If the ANEES value is much lower than the theoretical value, it may indicate that the estimator is overly conservative, i.e., its reported uncertainty is larger than the actual observed error; if the ANEES value is much higher than the theoretical value, it may indicate that the estimator is overly confident, i.e., its reported uncertainty is smaller than the actual observed error.
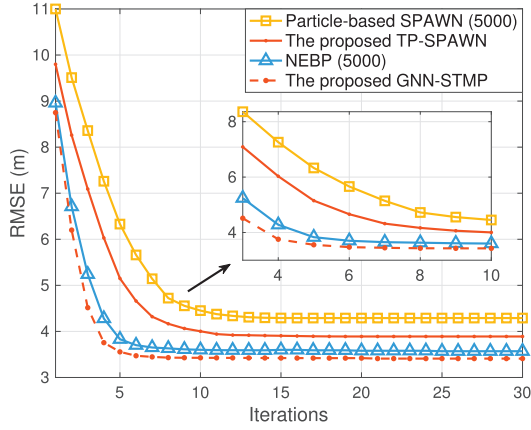
Fig. 8. Convergence of RMSE for the proposed GNN-STMP and TP-SPAWN against SPAWN and NEBP in terms of iterations.

Fig. 9. RMSE performance of an agent with insufficient number of neighbors, when using various DCP schemes.

when ANEES approaches the degree of freedom, i.e., ρ approaches zero, it does not indicate a high accuracy, because smaller error is usually indicated by lower values of ANEES. Fig. 7 illustrates the value of ρ for our TP-SPAWN and GNN-STMP as well as the particle-based SPAWN and NEBP, under different numbers of agents ($20 \sim 50$), $\ell_{\max} = 20$ and $\upsilon = 3$. We have the following observations. Firstly, the consistency and confidence of all the above algorithms improve with the increase of the number of agents. This observation aligns with our expectations, since agents lacking a sufficient number of neighbors in the mobile network can adversely affect the consistency and confidence of the position estimate. Secondly, NEBP and our GNN-STMP outperform the particle-based SPAWN and our TP-SPAWN. This observation demonstrates that our GNN-STMP achieves a commendable consistency and confidence level between the estimation errors and the uncertainty of the estimates. The aforementioned advantages can be primarily attributed to the robust learning capabilities of the GNN, the internal temporal information and the high accuracy second-order TP based approximation.

Subsequently, we evaluate the convergence performance of our proposed GNN-STMP, where the convergence is characterized by the RMSE values as a function of the number of iterations. Fig. 8 shows the RMSE convergence for the proposed GNN-STMP and TP-SPAWN against SPAWN and NEBP in terms of iterations. The curves "SPAWN (4000)" and "NEBP (4000)" denote SPAWN and NEBP with 4000 sample points, respectively. We observe that the algorithms utilizing GNN for message correction on factor graphs, i.e., the proposed GNN-STMP and NEBP, demonstrate a rapid decline in RMSE within the first 6 iterations, after which the error stabilizes. In contrast, the degraded versions of these algorithms, TP-SPAWN and SPAWN, converge after 10 iterations and show minimal performance improvement beyond 20 iterations. This indicates that incorporating GNN can accelerate the convergence of belief propagation-based algorithms, albeit at the cost of increased computational complexity per iteration. It is worth noting that the additional complexity introduced by GNN correction in NEBP is significantly higher than in GNN-STMP. This is because, in NEBP, the weight of each sampling point needs to be relearned, whereas in GNN-STMP,
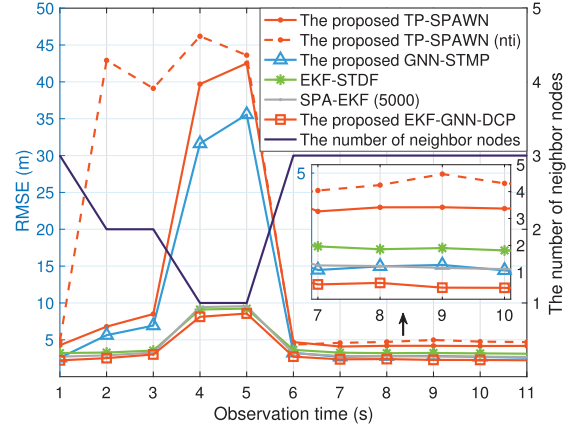
only a few parameter values require learning. Furthermore, the performance of GNN-STMP surpasses that of the NEBP algorithm with 4000 particles, further demonstrating the high approximation accuracy of the second-order TP and the effectiveness of using GNN for parametric message correction.

B. Mobile Networks With Sparsely Distributed Agents

In the scenario of mobile networks with sparsely distributed agents, we first evaluate the effectiveness of our proposed EKF-GNN-DCP framework and the impact of temporal information. We use SPA-EKF with 5000 particles and EKF-STDF as benchmarks in this experiment due to their effectiveness in scenarios with insufficient neighbors, as SPA-EKF and EKF-STDF are the representative particle-based and parametric algorithms, respectively. The number of agents is set to $N = 30$, $\ell_{\max} = 20$, and each agent moves a distance $d_i^t \sim \mathcal{N}(3, 1)$ in a direction $\theta_i^t \sim \mathcal{U}(0, 2\pi)$. More specifically, we consider a single agent that has insufficient spatial ranging measurements, i.e., insufficient number of neighbors, to gain insight. Fig. 9 illustrates the RMSE performance of such an agent achieved by our TP-SPAWN, GNN-STMP and EKF-GNN-DCP as well as EKF-STDF [7] and SPA-EKF with 5000 particles [17]. The curve labeled "TP-SPAWN (nti)" denotes the performance of our TP-SPAWN without temporal information, while temporal information are provided for the other cases. The black line in Fig. 9 indicates the number of neighbors that the agent has during the observation time slots. More specifically, the agent has 3 neighbors at time slot 1, 2 neighbors at time slot 2, 1 neighbor at time slots 3 and 4, and 3 neighbors at the subsequent time slots. Observe that the performance of our TP-SPAWN, TP-SPAWN (nti) and GNN-STMP degrade considerably when the agent does not have sufficient spatial ranging measurements. In particular, the performance gap between TP-SPAWN (nti) and TP-SPAWN is quite large when the number of neighbors is 2 or 1. This indicates that the internal ranging measurement based temporal information can be utilized as the useful supplement of external ranging measurement based spatial information for agents to improve performance. In addition, the performance of our GNN-STMP is considerably superior to that of our TP-
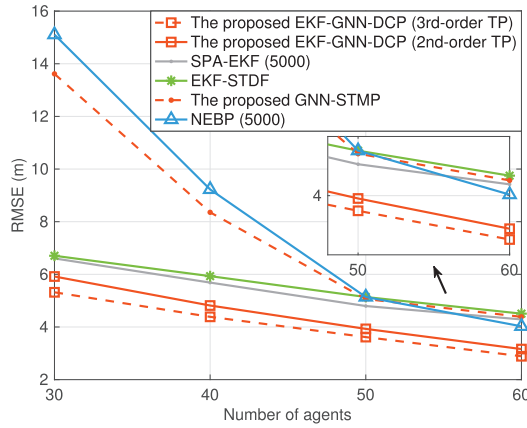
Fig. 10. RMSE positioning performance as the function of the number of agents, when using various DCP schemes.
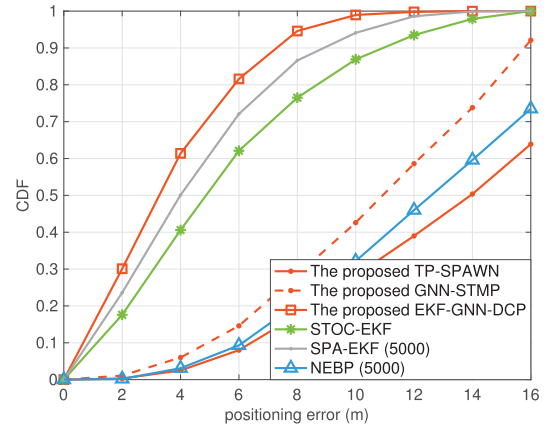

Fig. 11. CDFs of the positioning error (i.e., RMSE) for our TP-SPAWN, GNN-STMP and EKF-GNN-DCP, as well as for STOC-EKF, SPA-EKF and NEBP in the scenario of mobile networks with sparsely distributed agents, given the number of agents $N = 30$.

SPAWN. This verifies that the data-driven GNN is capable of refining the accuracy of SPAWN messages, and consequently improves the positioning performance. However, using only GNN based messages refinement does not make up for lack of neighbors, and the performance of our GNN-STMP is still quite poor in the case of just one neighbor. By contrast, the performance of our EKF-GNN-DCP, EKF-STDF and SPA-EKF are much more robust to the deficiency of spatial ranging measurements. This is because the prediction operation of EKF is useful to improve the accuracy of the agent position estimation, and the high-precision prior values used in message passing procedure can reduce the ambiguity of the agent position estimation. Furthermore, the proposed EKF-GNN-DCP outperforms EKF-STDF. This is because EKF-STDF considers the SPAWN messages as Gaussian distributions, which reduces the accuracy of the message representation inevitably. By contrast, our EKF-GNN-DCP exploits a more accurate message representation method and apply GNN-generated messages to further refine the spatial SPAWN message representations.

Then we investigate the impact of the number of agents on positioning performance in scenario where the agent moves faster. Consider an area of $[0, 2000]\,\text{m} \times [0, 2000]\,\text{m}$. Agents are uniformly scattered across an area of $[200, 1800]\,\text{m} \times [200, 1800]\,\text{m}$. The ranging radius to perform external measurements is 400 m, and the variance of the ranging noise is 6 m^2. At time slot t, each agent moves a distance $d_i^t \sim \mathcal{N}(25, 5)$ in a direction $\theta_i^t \sim \mathcal{U}(0, 2\pi)$, and $\ell_{\max}$ is set to 20. Fig. 10 depicts the RMSE performance as the functions of the number of agents, when using our GNN-STMP and EKF-GNN-DCP as well as EKF-STDF, SPA-EKF with 5000 particles and NEBP with 5000 particles. The curve "The proposed EKF-GNN-DCP (2nd-order TP)" and "The proposed EKF-GNN-DCP (3rd-order TP)" denote our EKF-GNN-DCP with second-order TP and third-order TP, respectively. By comparing the numerical results under different values of the number of agents, e.g., under $N = 30$ and $N = 60$, we conclude that the positioning performance of mobile networks is substantially compromised by agents lacking adequate spatial ranging measurements. In addition, our EKF-GNN-DCP consistently achieves the best performance, and SPA-EKF has the second best performance except when the number

of agents is larger than 50, further demonstrating the role of EKF-based state constraints when agents have insufficient neighbors. When the number of agents is larger than 50, SPA-EKF becomes slightly inferior to NEBP and our GNN-STMP, which indicates that, as the number of neighbors increases, the message refinement based on GNN outperforms the EKF-based state constraints. Also, our GNN-STMP outperforms NEBP when the number of agents is smaller than 50, while for more than 50 agents, the former becomes slightly inferior to the latter, further validating the role of temporal information in GNN-STMP for positioning when agents lack sufficient neighbors. In addition, it can be observed that the improvement with the third-order TP over the second-order TP in EKF-GNN-DCP is minimal. Considering approximation accuracy, the computational complexity in Table II, and numerical results for distributed agents, employing a second-order TP strategy is adequate for achieving satisfactory positioning accuracy in DCP tasks.

Next we assess the performance of our approaches against several representative algorithms. Fig. 11 depicts the CDFs of the position error for our TP-SPAWN, GNN-STMP and EKF-GNN-DCP with second-order TP as well as STOC-EKF [19], SPA-EKF [17] and NEBP [21], where the number of particles for SPA-EKF and NEBP are 5000. Fig. 11 presents the simulation results with the experimental parameters being consistent with those used in Fig. 10. Given that EKF-STDF performs worse than SPA-EKF with 5000 particles, we have omitted the numerical results of EKF-STDF to keep Fig. 11 concise. Observe that our EKF-GNN-DCP achieves the best performance. Specifically, 80% of agents in our EKF-GNN-DCP achieve the position error smaller than 6m, compared to 7 m and 8.8 m in STOC-EKF and SPA-EKF, respectively. This superior performance of our EKF-GNN-DCP is attributed to its exploitation of the internal measurements based temporal information and GNN based messages refinement. Also, STOC-EKF approximates the nonlinear system model as a linear one around the selected linearization points, which degrades the positioning accuracy inevitably. In this sparsely scenario, the performance of our TP-SPAWN and GNN-STMP

as well as NEBP are inadequate. However, our GNN-STMP outperforms NEBP considerably, owning to its ability of effectively exploiting temporal information. Specifically, 60% of agents in our GNN-STMP achieve the position error smaller than 12 m, compared to 14 m and 15.4 m in NEBP and our TP-SPAWN, respectively.

C. Discussions on Future Work

A potential extension of this work may involve incorporating a probabilistic communication model, where the reliability of signal transmission is affected by both the distance and the channel fading between agents. By modeling message loss or degradation as a function of variables that characterize the channels, a more realistic representation of communication dynamics in DCP systems can be offered to help improve our proposed method. Additionally, our current framework does not explicitly address communication errors arising from time slot misalignment or channel contention. Therefore, it is necessary to explore robust communication strategies, such as dynamic time slot management or enhanced message redundancy techniques, to further improve the resilience of our DCP method in mobile wireless networks.

VIII. CONCLUSION

Conventional DCP approaches suffer from inadequate accuracy of FG based messages and excessively high computational complexity in mobile wireless networks. To address these problems, we have first presented a high-precision parametric message representation method, which employs the TP to approximate the nonlinear spatio-temporal messages passed on the FG. We have derived closed-form representations for all types of messages involved and the *a posteriori* distribution of agents' positions based on second-order TP. Building upon this message representation method, we have developed a model- and data-driven hybrid inference approach, i.e., GNN-STMP, which is capable of fine-tuning parametric messages passed on FG and obtaining more accurate closed-form expression for the *a posteriori* distribution of agents' positions. Furthermore, we have developed a universal framework, named EKF-GNN-DCP, for the parametric message passing based DCP, by integrating GNN-STMP with the EKF based agents' state prediction and refinement. This framework significantly reduces the positioning ambiguity caused by insufficient spatial ranging measurements from neighbor nodes. Simulation results and analysis have revealed that: 1) using the second-order TP approximation is a preferred choice, as it strikes an appealing balance between the computational complexity and the approximation accuracy; 2) thanks to the proposed high-accuracy parametric message representations, our TP-SPAWN outperforms other parametric DCP algorithms or even particle-based algorithms with 4000 particles, while incurring modest computational complexity, given that the number of neighbor nodes of the individual agents is sufficient; 3) the refinement relying on GNN-generated messages significantly improves the message representation accuracy on FG, hence our GNN-STMP substantially improves the positioning accuracy of agents when they have sufficient neighbor nodes; and

4) the proposed EKF-GNN-DCP framework achieves the best positioning performance in the challenging sparsely distributed mobile wireless networks among all the DCP algorithms considered.

REFERENCES

[1] H. Wymeersch, J. Lien, and M. Z. Win, "Cooperative localization in wireless networks," *Proc. IEEE*, vol. 97, no. 2, pp. 427–450, Feb. 2009.

[2] M. Z. Win et al., "Network localization and navigation via cooperation," *IEEE Commun. Mag.*, vol. 49, no. 5, pp. 56–62, May 2011.

[3] T. Lv, H. Gao, X. Li, S. Yang, and L. Hanzo, "Space-time hierarchical-graph based cooperative localization in wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 64, no. 2, pp. 322–334, Jan. 2016.

[4] 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.

[5] Y. Cao, S. Yang, and Z. Feng, "Geo-spatio-temporal information based 3D cooperative positioning in LOS/NLOS mixed environments," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Rio de Janeiro, Brazil, Dec. 2022, pp. 5637–5642.

[6] Y. Cao, S. Yang, Z. Feng, L. Wang, and L. Hanzo, "Distributed spatio-temporal information based cooperative 3D positioning in GNSS-denied environments," *IEEE Trans. Veh. Technol.*, vol. 72, no. 1, pp. 1285–1290, Jan. 2023.

[7] Y. Cao, S. Yang, X. Ma, and Z. Feng, "Cooperative positioning for sparsely distributed high-mobility wireless networks with EKF based spatio-temporal data fusion," *IEEE Commun. Lett.*, vol. 27, no. 9, pp. 2343–2347, Sep. 2023.

[8] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Understanding belief propagation and its generalizations," in *Exploring Artificial Intelligence in the New Millennium*, G. Lakemeyer and B. Nebel, Eds., San Diego, CA, USA: Morgan Kaufmann, 2003, pp. 239–270.

[9] Y. Weiss and W. T. Freeman, "Correctness of belief propagation in Gaussian graphical models of arbitrary topology," in *Proc. 12th Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 13, Denver, CO, USA, Oct. 2001, pp. 2173–2200.

[10] K. Murphy, Y. Weiss, and M. I. Jordan, "Loopy belief propagation for approximate inference: An empirical study," in *Proc. 15th Conf. Uncertainty Artif. Intell. (UAI)*, Stockholm, Sweden, Jul. 1999, pp. 467–475.

[11] M. Leng, W. P. Tay, T. Q. S. Quek, and H. Shin, "Distributed local linear parameter estimation using Gaussian SPAWN," *IEEE Trans. Signal Process.*, vol. 63, no. 1, pp. 244–257, Jan. 2015.

[12] S. Rangan, P. Schniter, A. K. Fletcher, and S. Sarkar, "On the convergence of approximate message passing with arbitrary matrices," *IEEE Trans. Inf. Theory*, vol. 65, no. 9, pp. 5339–5351, Sep. 2019.

[13] 54 V. G. Satorras and M. Welling, "Neural enhanced belief propagation on factor graphs," in *Proc. Int. Conf. Artif. Intell. Statist.*, San Diego, CA, USA, 2021, pp. 685–693.

[14] J. Lien, U. J. Ferner, W. Srichavengsup, H. Wymeersch, and M. Z. Win, "A comparison of parametric and sample-based message representation in cooperative localization," *Int. J. Navigat. Observ.*, vol. 2012, pp. 1–10, Sep. 2012.

[15] N. Wu, B. Li, H. Wang, C. Xing, and J. Kuang, "Distributed cooperative localization based on Gaussian message passing on factor graph in wireless networks," *Sci. China Inf. Sci.*, vol. 58, no. 4, pp. 1–15, Apr. 2015.

[16] B. Li and N. Wu, "Convergence analysis of Gaussian SPAWN under high-order graphical models," *IEEE Signal Process. Lett.*, vol. 27, pp. 1725–1729, 2020.

[17] C. Fan, L. Li, M. Zhao, and X. Xu, "A dynamic constrained cooperative localization algorithm," in *Proc. 25th Int. Conf. Telecommun. (ICT)*, Saint-Malo, France, Jun. 2018, pp. 88–92.

[18] S. Wang and X. Jiang, "Three-dimensional cooperative positioning in vehicular ad-hoc networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 2, pp. 937–950, Feb. 2021.

[19] E. Kevin and H. Guoquan, "State-transition and observability constrained EKF for multi-robot cooperative localization," in *Proc. 34th Chin. Control Conf. (CCC)*, Hangzhou, China, Jul. 2015, pp. 7404–7410.

[20] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *Proc. 34th ICML*, vol. 70, Sydney, NSW, Australia, 2017, pp. 1263–1272.

[21] M. Liang and F. Meyer, "Neural enhanced belief propagation for cooperative localization," in *Proc. IEEE Stat. Signal Process. Workshop (SSP)*, Jul. 2021, pp. 326–330.

[22] M. Liang and F. Meyer, "Neural enhanced belief propagation for data association in multiobject tracking," in *Proc. 25th Int. Conf. Inf. Fusion (FUSION)*, Linköping, Sweden, Jul. 2022, pp. 1–7.

[23] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, New Orleans, LA, USA, May 2019, pp. 1–17.

[24] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *Proc. Eur. Semantic Web Conf.*, 2018, pp. 593–607.

[25] M. Wang, X. Li, X. Zhang, and Y. Zhang, "Hierarchical graph attention network with pseudo-metapath for skeleton-based action recognition," *Neurocomputing*, vol. 501, pp. 822–833, Aug. 2022.

[26] M. Wang, X. Li, S. Chen, X. Zhang, L. Ma, and Y. Zhang, "Learning representations by contrastive spatio-temporal clustering for skeleton-based action recognition," *IEEE Trans. Multimedia*, vol. 26, pp. 3207–3220, 2024.

[27] 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. (ICASSP)*, Toronto, ON, Canada, Jun. 2021, pp. 5250–5254.

[28] W. Yan, J. Wang, F. Yin, Y. Tian, and A. M. Zoubir, "Attentional graph neural networks for robust massive network localization," 2023, *arXiv:2311.16856*.

[29] F. Gao, J. Zhang, and Y. Zhang, "Neural enhanced dynamic message passing," in *Proc. Int. Conf. Artif. Intell. Statist. (AISTATS)*, Mar. 2022, pp. 10471–10482.

[30] B. Camajori Tedeschini, M. Brambilla, and M. Nicoli, "Message passing neural network versus message passing algorithm for cooperative positioning," *IEEE Trans. Cognit. Commun. Netw.*, vol. 9, no. 6, pp. 1666–1676, Dec. 2023.

[31] A. Papoulis and S. U. Pillai, *Probability, Random Variables and Stochastic Processes*. New York, NY, USA: McGraw-Hill, Dec. 2001.

[32] E. B. Sudderth, A. T. Ihler, M. Isard, W. T. Freeman, and A. S. Willsky, "Nonparametric belief propagation," *Commun. ACM*, vol. 53, no. 10, pp. 95–103, 2010.

[33] F. Meyer, O. Hlinka, and F. Hlawatsch, "Sigma point belief propagation," *IEEE Signal Process. Lett.*, vol. 21, no. 2, pp. 145–149, Feb. 2014.

[34] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent.*, San Diego, CA, USA, 2015, pp. 1–15.

Yue Cao received the B.Eng. degree in radio and television engineering from Nanjing University of Posts and Telecommunications, Nanjing, China, in 2015, and the M.Eng. degree in information and communication engineering from Beijing Institute of Graphic Communication, Beijing, China, in 2019. He is currently pursuing the Ph.D. degree with the School of Information and Communication Engineering, Beijing University of Posts and Telecommunications (BUPT), Beijing, China. His research interests include distributed cooperative positioning and machine learning for positioning.

Shaoshi Yang (Senior Member, IEEE) received the B.Eng. degree in information engineering from Beijing University of Posts and Telecommunications (BUPT), China, in 2006, and the Ph.D. degree in electronics and electrical engineering from the University of Southampton, U.K., in 2013. From 2008 to 2009, he was a Researcher with Intel Labs China. From 2013 to 2016, he was a Research Fellow with the School of Electronics and Computer Science, University of Southampton. From 2016 to 2018, he was a Principal Engineer with Huawei Technologies Company Ltd., where he made significant contributions to the products, solutions, and standardization of 5G, the wideband IoT, and cloud gaming/VR. He was a Guest Researcher with the Isaac Newton Institute for Mathematical Sciences, University of Cambridge. He is currently a Professor with BUPT, a member with the Key Laboratory of Universal Wireless Communications, Ministry of Education, and the Deputy Director of the Key Laboratory of Mathematics and Information Networks, Ministry of Education. His research interests include 5G/5G-A/6G, massive MIMO, mobile ad hoc networks, distributed artificial intelligence, and cloud gaming/VR. He is a Standing Committee Member of the CCF Technical Committee on Distributed Computing and Systems. He received the Dean's Award for Early Career Research Excellence from the University of Southampton in 2015, the Huawei President Award for Wireless Innovations in 2018, the IEEE TCGCC Best Journal Paper Award in 2019, the IEEE Communications Society Best Survey Paper Award in 2020, the Xiaomi Young Scholars Award in 2023, the CAI Invention and Entrepreneurship Award in 2023, the CIUR Industry-University-Research Cooperation and Innovation Award in 2023, and the First Prize of Beijing Municipal Science and Technology Advancement Award in 2023. He is an Editor of IEEE Transactions on Communications, IEEE Transactions on Vehicular Technology, and *Signal Processing* (Elsevier). He was also an Editor of IEEE Systems Journal and IEEE Wireless Communications Letters. For more details on his research progress, please refer to https://shaoshiyang.weebly.com/.

Zhiyong Feng (Senior Member, IEEE) received the B.Eng., M.Eng., and Ph.D. degrees from Beijing University of Posts and Telecommunications (BUPT), Beijing, China. She is currently a Professor with the School of Information and Communication Engineering, BUPT, and the Director of the Key Laboratory of Universal Wireless Communications, Ministry of Education, China. Her research interests include wireless network architecture design and radio resource management in 5G/6G, spectrum sensing and dynamic spectrum management in cognitive wireless networks, and universal signal detection and identification. She is the Vice Chair of the Information and Communication Test Committee of Chinese Institute of Communications (CIC), the Associate Editor-in-Chief of *China Communications*, and a Technological Advisor of the International Forum on NGMN (Next Generation Mobile Networks).

Ping Zhang (Fellow, IEEE) is currently a Professor with the School of Information and Communication Engineering, Beijing University of Posts and Telecommunications. He was the Chief Scientist of the National Basic Research Program (973 Program), an Expert with the Information Technology Division, National High-Tech Research and Development Program (863 Program), and a member with the Consultant Committee on International Cooperation, National Natural Science Foundation of China. He is currently the Director of State Key Laboratory of Networking and Switching Technology and a member with the IMT-2020 (5G) Experts Panel and the Experts Panel for China's 6G Development. He is an Academician with Chinese Academy of Engineering (CAE). His research interests include wireless communications.

Sheng Chen (Life Fellow, IEEE) received the B.Eng. degree in control engineering from East China Petroleum Institute, Dongying, China, in 1982, the Ph.D. degree in control engineering from the City, University of London, in 1986, and the Doctor of Sciences (D.Sc.) degree from the University of Southampton, Southampton, U.K., in 2005. From 1986 to 1999, he held research and academic appointments at The University of Sheffield, U.K., The University of Edinburgh, U.K., and the University of Portsmouth, U.K. Since 1999, he has been with the School of Electronics and Computer Science, University of Southampton, where he is currently a Professor of intelligent systems and signal processing. He has published over 700 research articles. He has more than 21,000 Web of Science citations with an H-index of 63 and more than 40,000 Google Scholar citations with an H-index of 85. His research interests include adaptive signal processing, wireless communications, modeling and identification of nonlinear systems, neural network and machine learning, evolutionary computation methods, and optimization. He is also a fellow of the Royal Academy of Engineering, U.K.; Asia–Pacific Artificial Intelligence Association; and IET. He was one of the original ISI Highly Cited Researchers in Engineering in March 2004.