International Journal of Wireless Information Networks (2024) 31:61–72
https://doi.org/10.1007/s10776-023-00610-0

Federated Learning-Enabled Cooperative Localization in Multi-agent System

Fangwen Ye[1] · Ran Wang[1,2] · Sisui Tang[1] · Shihong Duan[1] · Cheng Xu[1,2]

Received: 04 June 2023 / Revised: 21 October 2023 / Accepted: 22 October 2023 / Published online: 30 November 2023

Abstract
Cooperative localization plays a significant role in various applications, such as emergency rescue and navigation path planning. The advent of swarm intelligence has opened doors to agent-based cooperative localization. However, sharing data between agents during the cooperative localization process can compromise privacy. One of the key challenges is to develop a cooperative localization model that safeguards the data privacy of agents. To tackle this issue, we initially adopt a state-space model to describe the movement of an agent for single-agent dynamic localization. This model effectively handles noise and improves accuracy. For the problem of multi-agent cooperative localization, we employ the federated learning framework coupled with the alternating direction multiplier method. Within this framework, the central node aggregates local models to create a global cooperative localization model, eliminating the need for data sharing and ensuring privacy protection. When compared to the centralized model, the federated model achieves satisfactory localization accuracy while demonstrating the robustness and generalization performance across different data distributions. Furthermore, when confronted with new scenarios, the federated model exhibits excellent transfer performance.

Keywords Cooperative localization · Federated learning · Multi-agent · Gaussian process · State-space model

1 Introduction

Recently, research on location-based services has gained popularity and is becoming increasingly integrated into our daily lives. Localization services are expanding, encompassing applications such as positioning navigation and route planning. Consequently, there is a growing demand for higher accuracy in localization services. The Global Positioning System (GPS) has been widely employed for outdoor localization and navigation, offering a cost-effective solution with high precision and real-time performance. However, in indoor scenarios, the signal strength of satellites is easily obstructed, leading to limitations in localization performance. In contrast, wireless localization technologies based on time difference of arrival (TDOA) and time of arrival (TOA) [1, 2] are widely adopted for indoor scenarios. These technologies typically require pre-set base stations and utilize methods such as ultra-wideband (UWB), infrared, ZigBee, and others [3]. Localization accuracy can be affected in the presence of non-line-of-sight (NLOS) interference, where the direct propagation path between nodes is obstructed [4]. On the other hand, localization technologies relying on inertial systems avoid NLOS interference by integrating information obtained from the system. However, the inertial system itself introduces deviations in the accelerometer and gyroscope, leading to cumulative system errors over time [5].

As mentioned earlier, achieving the desired localization accuracy based solely on an agent's observation data can be challenging. In the realm of cooperative localization networks, one of the main challenges is constructing a network where multiple agents work together to overcome perception limitations, fully utilize heterogeneous data, and enhance the overall performance of the localization system. In traditional cooperative localization, the target node obtains distance information from reference nodes and other target

✉ Ran Wang
wangran423@foxmail.com

✉ Cheng Xu
xucheng@ustb.edu.cn

[1] School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing, China

[2] Shunde Innovation School, University of Science and Technology Beijing, Foshan, China

nodes, and its position is determined by utilizing the aggregated distance information from these sources. Cooperative localization enables rapid and efficient localization of target nodes in scenarios with poor signal strength between some nodes [6]. However, conventional cooperative localization methods require sharing observed data between different agents. Notably, these sensor data inherently contain distinct characteristics of the respective agents or environmental features. Frequent transmission and communication may result in sensitive privacy leakage [7].

In recent years, Federated Learning (FL) [7], a distributed machine learning framework, has garnered significant attention as a model training method that prioritizes privacy protection. In the FL approach, participants do not need to share their data during training. Instead, they perform local iterations and then upload the updated information (e.g., gradients) of their local models to a central server [8–10]. The central server aggregates all the received gradients and perform back propagation to obtain the latest parameters. Then the server sends back the parameters to each participant for them to conduct local training. Moreover, in the settings of [11, 12], each client uses stochastic gradient descent (SGD) to optimize the current local model and then uploads the parameters to the server instead. The server performs the federated averaging algorithm (FedAvg) to obtain the latest parameters. This entire procedure ensures that data remains confined within local domains, and the transmission process only involves model parameters, effectively safeguarding privacy. By leveraging FL, localization methods based on received signal strength (RSS) fingerprints [12, 13] can achieve satisfactory localization accuracy while preserving the privacy of local data. Deep neural networks (DNNs) are commonly employed in these federated localization applications as the preferred choice for local models. To enhance the performance of deep models, appropriately increasing the number of hidden layers is necessary. However, as the number of network layers increases, the parameter size expands significantly, posing considerable communication challenges between the local model and the central server.

To address the aforementioned problems, we propose a multi-agent cooperative localization method based on FL. Our main contributions can be summarized as follows:

(1) We propose a Gaussian process-based method to address the challenge of dynamic localization in agent movement. This method handles the state-space by leveraging the temporal information in the dynamic network and effectively considering the uncertainty caused by noisy data during agent movement. As a result, it achieves dynamic tracking and localization.
(2) We introduce the alternating direction multiplier method (ADMM) to achieve the federated fusion of multiple local models with a small upload parameter size, effectively alleviating the communication burden. While ensuring the protection of local data privacy, our approach maintains satisfactory localization performance compared to centralized learning. Furthermore, it exhibits robustness to heterogeneous data, reflecting the generalization capabilities of the federated learning model.

The structure of this article is organized as follows. Section 2 briefly reviews the related work, and Sect. 3 describes the state-space model and the FL framework. Section 4 presents the details of our method, and the corresponding experiment is conducted in Sect. 5. Finally, we conclude the paper in Sect. 6.

2 Related Work

Existing localization methods can be broadly categorized into two groups: learning-based methods and optimization-based methods. Xie et al. [14] introduced an indoor localization method that combines random forest and deep learning. This approach leverages deep learning to train the channel propagation model during the offline phase and enables online determination of the agents' orientation. On the other hand, Yan et al. [15] proposed a localization method based on graph neural networks (GNNs), which effectively addresses computational challenges in large-scale networks. This method guarantees localization accuracy and model stability, enabling static localization of large-scale agents. However, in cooperative localization networks, GNNs often struggle to effectively utilize temporal information in dynamic networks due to the inherent dynamism of agents. Generally, sensor data in cooperative agent networks are affected by noise and exhibit some uncertainties. Regarding interpretability, deep learning methods are less proficient in handling uncertainties than optimization-based methods. The main optimization-based methods include great likelihood estimation [16, 17], Gaussian process (GP) [18, 19], Kalman filter [19, 20], Bayesian message passing [6, 21], and others.

In cooperative localization networks [6, 21–23], an agent relies not only on its own observation data but also gathers interaction information with other agents, leading to many compelling benefits, including the ability to extend position determination capabilities without ambiguity and enhance estimation accuracy performance. Sharma et al. [20] propose a cooperative approach for navigating Miniature Air Vehicles (MAVs) in scenarios where GPS signals are unavailable. Each MAV estimates the position, pose, and velocity of all MAVs within its sensor range, including itself. The fused measurements are then processed using an Extended Kalman Filter (EKF) to realize navigation state estimation. Jin et al

[21] proposed a Bayesian framework to handle the problem of RSS-based cooperative localization with unknown path loss exponent. This probabilistic inference is solved using message passing algorithm, which performs well in dense networks and low-to-medium noise scenarios. Wielandner et al. [24] present a factor-graph-based cooperative localization through RSS. This is accomplished by employing a parametric antenna pattern to model directivity and concurrently estimating the positions and orientations of the agents.

However, in traditional cooperative localization, the sensor data measured by the agent typically contains its own characteristics or environmental scene features, which introduces privacy concerns. With increasing emphasis on user privacy and data security, the traditional approach of using interaction information between agents in cooperative localization can no longer meet the contemporary requirements for privacy protection. To address this issue, federated learning (FL) [7], a distributed learning framework, eliminates the need to share raw data. Each participant computes local model update information (e.g., gradients) based on their local private data. In various FL scenarios, participants may send different types of information. Some send the gradients for the server-side update [8–10], while others perform local gradient descent and send updated local parameters to the server for aggregation [11, 12]. The updated global model is then distributed back to all participants for the next round of training. This iterative process continues until the model converges. Importantly, since the model update information usually contains less privacy-sensitive data, the original data remains confined within the local area, effectively preventing privacy data leakage.

FL has been applied in various fields, including medical image analysis [25], natural language processing [26], recommendation systems [27], and localization applications. For example, Ciftler et al. [13] utilized FL to implement privacy-preserving indoor localization. In this approach, users construct local models using beacon RSS measurements with location labels. The central server aggregates local updates to construct a global multi-layer perceptron (MLP), achieving accurate localization estimates while preserving privacy. Kong et al. [10] introduce a Federated Learning-based Vehicle Cooperative Positioning (FedVCP) system that utilizes the capabilities of the social Internet of Things (IoT) and Collaborative Edge Computing (CEC). It employs the MLP for regional positioning error prediction and aims to deliver accurate positioning corrections while preserving user privacy. Another work by Liu et al. [28] implemented federated localization for WiFi networks. Mobile devices construct local fingerprinting to train a DNN model, followed by a deep self-encoder to eliminate noise. The central server aggregates local weights to generate a generic model that combines high security and localization accuracy. In both cases, the updated information is closely related to the local model. However, the parameter size of the deep model poses a considerable burden on communication.

Liu et al. [29] proposed lifelong federated reinforcement learning (RL) for cloud robot navigation systems, where private models are fused into a global shared model in the cloud. The experiments demonstrate that FL can fuse prior knowledge and greatly improve the efficiency of robot navigation based on RL, enabling the robot to adapt rapidly to new environments. Furthermore, FL can be applied in RL agents to learn optimal control policies on IoT devices of the same type but with slightly different dynamic performances [30]. The experimental results show that this approach effectively facilitates the learning process of multiple non-identical IoT devices. Evidently, FL incorporates prior knowledge from participants during the local model aggregation process, improving the generalization performance of the global model and enabling quick adaptation to new scenarios.

To summarize, we propose a methodology that models the agent's state-space and utilizes Gaussian Process (GP) to handle uncertainty caused by noise, enabling the tracking of a single agent. Additionally, we achieve federated fusion of multiple agents using Alternating Direction Multiplier Method (ADMM) to reduce communication burden. Compared to centralized learning, the federated model maintains satisfactory localization accuracy while preserving data privacy. The experiments also demonstrate robustness and generalization performance when dealing with different data distributions.

3 Problem Definition

3.1 Dynamic Model

The agent's motion comprises a series of temporal information, making the state-space model well-suited for capturing its dynamics. In general, the state-space model is defined as:

$$\begin{aligned} x_{t+1} &= f(x_t, u_t) + q_t, \\ y_t &= g(x_t) + r_t, \end{aligned} \tag{1}$$

where $x_t \in \mathbb{R}^{D_x}$ represents the latent state, $y_t \in \mathbb{R}^{D_y}$ denotes the system output, $u_t \in \mathbb{R}^{D_u}$ represents the control input, $f : \mathbb{R}^{D_x} \times \mathbb{R}^{D_u} \to \mathbb{R}^{D_x}$ is the transition function, $g : \mathbb{R}^{D_x} \to \mathbb{R}^{D_y}$ is the measurement output function, and q_t and r_t represent the transition noise and measurement noise, respectively. Considering that noise has a greater impact during the agent's motion compared to statistical localization, we achieve dynamic localization by employing Gaussian Process (GP) regression to effectively handle the noise. Furthermore, GP has a smaller parameter size and simpler model structure compared to a neural network [31]. Similar to the choice of activation function in neural networks, the

kernel function in GP plays a crucial role in determining the model's expressiveness, such as periodicity and smoothness. In the following sections, we utilize the squared exponential (SE) kernel:

$$k(x,x') = \sigma_f^2 \exp\Big(-\frac{1}{2}(x-x')^T M^{-1}(x-x')\Big), \tag{2}$$

where $M = \text{diag}(l^2)$, with l representing the lengthscale, and σ_f^2 denoting the signal variance. The hyperparameters of the squared exponential (SE) kernel are expressed as $\theta_h = \{\sigma_f, l\}$. In general, the transition function f and the measurement output function g are modeled using Gaussian Process (GP) with a mean function $m(\cdot)$ and a kernel function $k(\cdot,\cdot)$.

The generative Gaussian process state-space model (GPSSM) can be described as follows:

$$\begin{aligned} f(x,u) &\sim \mathcal{GP}\big(m_f(x,u), k_f\big((x,u),(x',u')\big)\big), \\ g(x) &\sim \mathcal{GP}\big(m_g(x), k_g(x,x')\big), \\ x_{t+1}|\boldsymbol{f}_{t+1} &\sim \mathcal{N}(x_{t+1}|\boldsymbol{f}_{t+1}, \boldsymbol{Q}), \text{where } \boldsymbol{f}_{t+1} \triangleq f(x_t,u_t), \\ y_t|\boldsymbol{g}_t &\sim \mathcal{N}(y_t|\boldsymbol{g}_t, \boldsymbol{R}), \text{where } \boldsymbol{g}_t \triangleq g(x_t), \end{aligned} \tag{3}$$

with the parameter collections $\boldsymbol{\theta}_f$, $\boldsymbol{\theta}_g$, $\boldsymbol{Q}$, $\boldsymbol{R}$, where $\boldsymbol{\theta}_f$ and $\boldsymbol{\theta}_g$ are kernel parameters of the Gaussian process, and $\boldsymbol{Q}$ and $\boldsymbol{R}$ are the covariance matrices of the transition noise and measurement noise. A graphical model of a GPSSM is shown in Fig. 1.

3.2 Federated Learning

McMahan et al. [7] proposed the federated averaging algorithm (FedAvg) based on gradient descent, which is widely applicable to federated learning (FL) systems. In this algorithm, the central server possesses the initial FL model and aggregates the local model updates that do not involve privacy. The central server then sends the aggregated updated parameters to all participants, who perform local training based on these updated parameters and iteratively upload their results until the final model converges. The training architecture of the FL model is displayed in Fig. 2. FedAvg is particularly suitable for parameter aggregation in neural networks, while the alternating direction multiplier method (ADMM) is more appropriate for Gaussian process (GP) regression, providing an effective balance between computational and communication efficiency.

We consider the general GP model $y = f(x) + \epsilon$, where $x \in \mathbb{R}^d$, $y \in \mathbb{R}$, and ϵ is Gaussian noise with zero mean and variance σ_ϵ^2. In the existing distributed GP model based on the product-of-experts (PoE) [32], the log-marginal likelihood function associated with the overall dataset $\mathcal{D} = \{X, y\}$ can be approximated by the product of the log-marginal likelihoods corresponding to K non-overlapping and equally-sized subsets $\mathcal{D}_{(k)} = \{X_{(k)}, y_{(k)}\}$, as follows:

$$\log p(y|X;\theta) \approx \sum_{k=1}^{K} \log p(y_{(k)}|X_{(k)};\theta), \tag{4}$$

where each participant in federated learning (FL) owns the subset $\mathcal{D}_k$ and shares the global hyperparameters θ. The product-of-experts (PoE) method estimates the covariance matrix of the complete dataset by using a block-diagonal matrix with equivalent dimensions.

To better align with practical situations, local hyperparameters θ_k can be introduced for each local dataset $\mathcal{D}_{(k)} = \{X_{(k)}, y_{(k)}\}$. Therefore, the local loss function during the distributed optimization of GP hyperparameters can be written as:

$$\mathcal{L}_k(\theta_k) = y_{(k)}^T C_k(\theta_k)^{-1} y_{(k)} + \log|C_k(\theta_k)|, \tag{5}$$

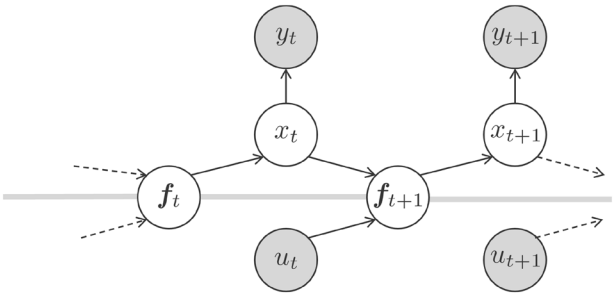

Fig. 1 Graphic model of GPSSM. The gray nodes denote the control input and the measurement output while the white nodes show the latent variables

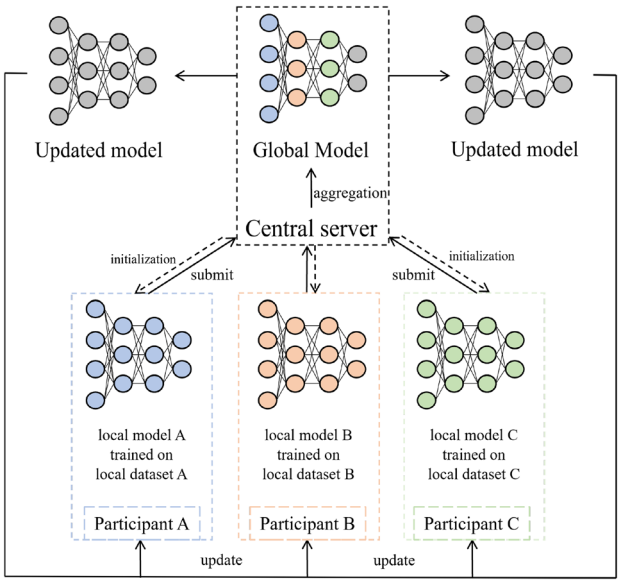

Fig. 2 An example of the FL training architecture

where $C_k(\theta_k) = K(X_{(k)}, X_{(k)}; \theta_k) + \sigma_e^2 I$, $K(\cdot,\cdot)$ is the kernel matrix. The global loss is given by

$$\mathcal{L}_g(\theta) = \sum_{k=1}^{K} \mathcal{L}_k(\theta_k). \tag{6}$$

FL training is performed to find the optimal parameters that minimize the global loss function: $\theta_g^* = \arg\min_{\theta} \mathcal{L}_g(\theta)$.

4 Cooperative Localization based on Federated Learning

In this section, we first describe the GPSSM-based algorithm for a single agent. Then, we introduce the alternating direction multiplier method to aggregate multiple local models based on federated learning, enabling cooperative localization.

4.1 Variational Inference for Localization of a Single Agent

Commonly, to address the unobserved latent variable of the transition function f and reduce computational costs, we introduce M inducing inputs $\boldsymbol{z} = [z_1, z_2, ..., z_M]$ with corresponding auxiliary inducing outputs $\boldsymbol{v} = [v_1, v_2, ..., v_M]$. Consequently, the GP distribution is estimated by $p(f^*|x^*, \boldsymbol{f}, X) \approx p(f^*|x^*, \boldsymbol{v}, \boldsymbol{z})$ with the sparse GP prior $p(\boldsymbol{v})$. The latent state sequence and the output sequence are denoted by $y_{1:T} = [y_1, y_2, ..., y_T]$ and $x_{1:T} = [x_1, x_2, ..., x_T]$, respectively. Then, the output of the transition function can be represented by $\boldsymbol{f}_{t+1} = f(\hat{x}_t)$, where $\hat{x}_t = (x_t, u_t)$. Furthermore, independent GP priors are applied for each latent state dimension d given auxiliary points $\boldsymbol{z}_d$ and $\boldsymbol{v}_d$, using mean-field approximations. This results in $p(\boldsymbol{v}) \approx \prod_{d=1}^{D_x} \mathcal{N}(\boldsymbol{v}_d|0, K(\boldsymbol{z}, \boldsymbol{z}))$. The joint distribution of GPSSM in the dynamics sequence is given by:

$$\begin{aligned} p(y_{1:T}, x_{1:T}, \boldsymbol{f}_{2:T}, \boldsymbol{v}) &= \prod_{t=1}^{T} p(y_t|x_t) \times \\ &\Big[\prod_{t=2}^{T} p(x_t|\boldsymbol{f}_t) p(\boldsymbol{f}_t|\hat{x}_{t-1}, \boldsymbol{v})\Big] p(x_1) p(\boldsymbol{v}), \end{aligned} \tag{7}$$

where $p(\boldsymbol{f}_t|\hat{x}_{t-1}, \boldsymbol{v}) = \prod_{d=1}^{D_x} p(\boldsymbol{f}_{t,d}|\hat{x}_{t-1}, \boldsymbol{v}_d)$.

To simplify the implementation, we model the output function g as $g(x_t) = C\, x_t$, where $C = [I, 0] \in \mathbb{R}^{D_y \times D_x}$ is used to select the first D_y elements of the latent state. Thus, the output model is defined as:

$$p(y_t|x_t) = \mathcal{N}(y_t|g(x_t), \mathrm{diag}(\sigma_{y,1}^2, \sigma_{y,2}^2, ..., \sigma_{y,D_y}^2)). \tag{8}$$

The transition process is modeled as:

$$p(x_t|\boldsymbol{f}_t) = \mathcal{N}(x_t|\boldsymbol{f}_t, \mathrm{diag}(\sigma_{x,1}^2, \sigma_{x,2}^2, ..., \sigma_{x,D_x}^2)), \tag{9}$$

where the unknown distribution of the initial latent state $p(x_1)$ is typically estimated.

Due to the inherent nonlinearity of the GP dynamics model in the latent state, computing the log-likelihood or posterior based on the joint distribution in Eq. (7) is often challenging. However, the Evidence Lower Bound (ELBO) of the log-marginal likelihood $\log p(y_{1:T})$ can provide an estimate of the posterior distribution. Specifically, a variational distribution $q(x_{1:T}, \boldsymbol{f}_{2:T}, \boldsymbol{v})$ is introduced to approximate the true posterior $p(x_{1:T}, \boldsymbol{f}_{2:T}, \boldsymbol{v}|y_{1:T})$. By applying Bayes' theorem, the log-marginal likelihood $\log p(y_{1:T})$ can be expressed as:

$$\begin{aligned} \log p(y_{1:T}) = &\log \frac{p(y_{1:T}, x_{1:T}, \boldsymbol{f}_{2:T}, \boldsymbol{v})}{q(x_{1:T}, \boldsymbol{f}_{2:T}, \boldsymbol{v})} \\ &- \log \frac{p(x_{1:T}, \boldsymbol{f}_{2:T}, \boldsymbol{v}|y_{1:T})}{q(x_{1:T}, \boldsymbol{f}_{2:T}, \boldsymbol{v})}. \end{aligned} \tag{10}$$

By taking the expectation with respect to the distribution $q(x_{1:T}, \boldsymbol{f}_{2:T}, \boldsymbol{v})$ on both sides of Eq. (10), the first term is transformed into the Evidence Lower Bound (ELBO), defined as:

$$\mathbb{E}_{q(x_{1:T}, \boldsymbol{f}_{2:T}, \boldsymbol{v})}\left[\log \frac{p(y_{1:T}, x_{1:T}, \boldsymbol{f}_{2:T}, \boldsymbol{v})}{q(x_{1:T}, \boldsymbol{f}_{2:T}, \boldsymbol{v})}\right] \triangleq \mathrm{ELBO}. \tag{11}$$

The second term in Eq. (12) represents the KL-divergence between the variational distribution $q(x_{1:T}, \boldsymbol{f}_{2:T}, \boldsymbol{v})$ and the true posterior $p(x_{1:T}, \boldsymbol{f}_{2:T}, \boldsymbol{v}|y_{1:T})$, which can be expressed as:

$$\begin{aligned} &- \mathbb{E}_{q(x_{1:T}, \boldsymbol{f}_{2:T}, \boldsymbol{v})}\left[\log \frac{p(x_{1:T}, \boldsymbol{f}_{2:T}, \boldsymbol{v}|y_{1:T})}{q(x_{1:T}, \boldsymbol{f}_{2:T}, \boldsymbol{v})}\right] \\ =&\mathrm{KL}\big(q(x_{1:T}, \boldsymbol{f}_{2:T}, \boldsymbol{v})||p(x_{1:T}, \boldsymbol{f}_{2:T}, \boldsymbol{v}|y_{1:T})\big). \end{aligned} \tag{12}$$

Approximating the true posterior $p(x_{1:T}, \boldsymbol{f}_{2:T}, \boldsymbol{v}|y_{1:T})$ with the variational distribution $q(x_{1:T}, \boldsymbol{f}_{2:T}, \boldsymbol{v})$ is equivalent to minimizing Eq. (12). This process can also be interpreted as maximizing Eq. (11).

Now, let us consider the factorization of $q(x_{1:T}, \boldsymbol{f}_{2:T}, \boldsymbol{v})$. We utilize an explicit representation [33] of the variational distribution, which allows for independent GP predictions given explicit inducing points. Following the mean-field theory, each latent state dimension d with diagonal variance Σ_d and the inducing output distribution are denoted as $q(\boldsymbol{v}) = \prod_{d=1}^{D_x} \mathcal{N}(\boldsymbol{v}_d|\mu_d, \Sigma_d)$. By marginalizing out the inducing outputs, we obtain the posterior distribution of GP prediction for each dimension of the latent state x_{t+1}, with mean and variance given by:

$$\mu_d(\hat{x}_t) = m(\hat{x}_t) + \alpha(\hat{x}_t)(\mu_d - m(\boldsymbol{z})), \tag{13}$$

$$\sigma_d^2(\hat{x}_t,\hat{x}_t) = k(\hat{x}_t,\hat{x}_t) - \alpha(\hat{x}_t)(K(z,z)-\Sigma_d)\alpha(\hat{x}_t)^T \tag{14}$$

with $\alpha(\hat{x}_t) = k(\hat{x}_t,z)K(z,z)^{-1}$. Furthermore, variational distribution can be approximated by factorization [34]:

$$q(x_{1:T}\boldsymbol{f}_{2:T},\boldsymbol{v}) = q(x_1)\prod_{t=2}^{T} p(x_t|\boldsymbol{f}_t)\times \Big[\prod_{t=2}^{T}\prod_{d=1}^{D_x} p(\boldsymbol{f}_{t,d}|\hat{x}_{t-1},\boldsymbol{v}_d)q(\boldsymbol{v}_d)\Big], \tag{15}$$

with $q(x_1) = \mathcal{N}(x_1|\mu_{x_1},\Sigma_{x_1})$, $q(\boldsymbol{v}_d) = \mathcal{N}(\boldsymbol{v}_d|\mu_d,\Sigma_d)$.

The parameters of the GPSSM model include the variational distribution parameters for the initial state and inducing pseudo points, the noise parameters for the state transition and observed output, and the kernel function parameters. We can represent these parameters as $\theta_{GPSSM} = \{\mu_{x_1},\Sigma_{x_1};\mu_{1:D_x},\Sigma_{1:D_x},z;\sigma^2_{x,1:D_x},\sigma^2_{y,1:D_y};\theta_h\}$. It is worth noting that in the GPSSM, the number of parameters only grows based on the latent dimensionality, regardless of the time series.

By substituting Eqs. (7) and (15) into (11), we obtain:

$$\text{ELBO} = \sum_{t=1}^{T}\mathbb{E}_{q(x_t)}[\log p(y_t|x_t)] - \sum_{d=1}^{D_x}\text{KL}\big(q(\boldsymbol{v}_d)||p(\boldsymbol{v}_d;z)\big) - \text{KL}\big(q(x_1)||p(x_1)\big). \tag{16}$$

The first term represents the expectation of the log-likelihood of the observed output y_t based on the variational distribution $q(x_t)$ of the latent states and the observed output model. It quantifies how well the latent state model explains the observed output of the system. The second term involves the regularization of the pseudo output distribution to penalize the deviation between the variational distribution and the GP prior. It helps to reduce the discrepancy between the two distributions. The last term corresponds to the regularization of the initial state distribution.

During the maximization of ELBO described in Eq. (16) with respect to θ_{GPSSM}, the computation of the last two terms is relatively straightforward. However, the first term requires taking the expectation with respect to the latent state distribution $q(x)$. Since the variational approximation Eq. (15) involves the nonlinear dynamics of the latent state, obtaining an analytical form for $q(x)$ is challenging. To overcome this issue, the Markov structure of the latent states and sparse GP can be utilized to approximate the expectation through sampling. Specifically, given the distribution of the previous state $q(x_{t-1})$ and an explicit representation of the inducing points, the marginal distribution $q(x_t)$ is conditionally independent of past time steps. Therefore, samples $\tilde{x}_t \sim q(x_t)$ can be recursively generated by sampling from the sparse GP posterior in Eqs. (13) and (14) for $t = 1, 2, ..., T$. Each dimension of the latent state is given by

$$\tilde{x}_{t+1,d} = \mu_d(\hat{x}_t) + \epsilon\sqrt{\sigma_d^2(\hat{x}_t,\hat{x}_t)+\sigma_{x,d}^2}, \tag{17}$$

where $\hat{x}_t = (\tilde{x}_t, u_t)$, $\tilde{x}_1 \sim q(x_1)$, and $\epsilon \sim \mathcal{N}(0,1)$ is the resampling parameters used for back-propagation of the gradient. Afterward, an unbiased estimator of the first term in ELBO (16) is given by

$$\mathbb{E}_{q(x_t)}[\log p(y_t|x_t)] \approx \frac{1}{N}\sum_{i=1}^{N}\log p(y_t|\tilde{x}_t^{(i)}), \tag{18}$$

where N is the number of state samples.

The optimization of parameters is accomplished through back-propagation during the maximization of ELBO (Eq. 16). The complete algorithm for single-agent dynamic localization is outlined in Algorithm 1. Once the model has been optimized by maximizing ELBO (Eq. 16), location predictions can be obtained using a new input sequence $u_{1:T}$ and an initial latent state x_1.

Algorithm 1 Gaussian process state-space model (GPSSM)

1. **Input**: $u_{1:T} \leftarrow$ Control Input; $y_{1:T} \leftarrow$ Ground Truth
2. **Output**: $\theta^*_{GPSSM} \leftarrow$ Optimal Model Parameters
3. Initialization Parameters: $\theta_{GPSSM} = \{\mu_{x_1},\Sigma_{x_1};\mu_{1:D_x},\Sigma_{1:D_x},z;\sigma^2_{x,1:D_x},\sigma^2_{y,1:D_y};\theta_h\}$
4. **FOR** $\eta = 1:$ max iteration
5. **FOR** $t = 1:T$
6. $\tilde{x}_t \sim q(x_t) \leftarrow$ state samples
7. $\mathbb{E}_{q(x_t)}[\log p(y_t|x_t)] \approx \frac{1}{N}\sum_{i=1}^{N}\log p(y_t|\tilde{x}_t^{(i)})$
8. **FOR** $d = 1:D_x$
9. $\tilde{x}_{t+1,d} = \mu_d(\hat{x}_t) + \epsilon\sqrt{\sigma_d^2(\hat{x}_t,\hat{x}_t)+\sigma_{x,d}^2} \leftarrow$ state transition
10. **ENDFOR**
11. **ENDFOR**
12. calculate $\sum_{d=1}^{D_x}\text{KL}\big(q(\boldsymbol{v}_d)||p(\boldsymbol{v}_d;z)\big)$
13. calculate $\text{KL}\big(q(x_1)||p(x_1)\big)$
14. max ELBO (16) based on gradient
15. **ENDFOR**

4.2 Multi-agent Cooperative Localization

In the state-space model (Eq. 1), the latent state x provides a better reflection of the agent's motion. Therefore, inferring the latent state x is more meaningful than the observed output y, and realistic transitions of the latent state are desired. The model parameters of the GPSSM used for dynamic localization of a single agent can be categorized as inducing points, noise parameters during state transition

and observation output, and hyperparameters in the GP kernel. In this study, the inducing points and noise parameters remain as local parameters for each agent without fusion. Only the kernel parameters are optimized through federated fusion. The overall process is illustrated in Fig. 3.

Once the local model has been trained, the state sequence prediction $\bar{x}_{1:T}$ is obtained from the local training data. To protect data privacy, only a portion of the model parameters are transmitted to the server during the fusion process based on the prediction series, without aggregating the original data. The fused kernel parameters are expected to generalize over multiple local data. Based on the prediction of latent states, we consider the objective transition model:

$$\bar{x}_{t+1} = f(\bar{x}_t^*) + q_t, \tag{19}$$

where $\bar{x}_t^* = (\bar{x}_t, u_t)$ and $t = 1, 2, ..., T-1$. To simplify the calculation, we assume that each state dimension is mutually independent according to the mean-field theory, and each dimension experiences the same Gaussian noise disturbance $\epsilon \sim \mathcal{N}(0, \sigma^2)$. Subsequently, the negative log-likelihood function for the prior distribution of the output in Eq. (19) is obtained as:

$$\begin{aligned}\mathcal{L}(\theta_h) &= \sum_{d=1}^{D_x} \bar{x}_{2:T,d}^T C^{-1}(\theta_h)\bar{x}_{2:T,d} + \log\det|C(\theta_h)| \\ &= \mathrm{tr}\left(\bar{x}_{2:T}^T C^{-1}(\theta_h)\bar{x}_{2:T}\right) + \log\det|C(\theta_h)|,\end{aligned} \tag{20}$$

where $C(\theta_h) = K(\bar{x}_{1:T-1}^*, \bar{x}_{1:T-1}^*; \theta_h) + \sigma^2 I$, $\mathrm{tr}(\cdot)$ denotes the trace of a matrix, and $|\cdot|$ represents the determinant of a matrix.

Equation (20) can be considered as a local loss in the distributed parameter fusion process. An optimization scheme for Gaussian processes based on the classical Alternating Direction Method of Multipliers (ADMM) [35] enables participants to train local models independently, with significantly reduced communication overhead to achieve global consensus. By introducing a set of hyperparameters $\{\theta_{h,1}, \theta_{h,2}, ..., \theta_{h,K}\}$ and the global hyperparameter $\mathcal{Z}$, Eq. (20) is transformed into a nonconvex consensus problem:

$$\begin{aligned}&\min \sum_{k=1}^{K} \mathcal{L}_k(\theta_{h,k}) \\ &\text{s.t.} \quad \theta_{h,k} - \mathcal{Z} = 0, \quad \forall k = 1, 2, ..., K.\end{aligned} \tag{21}$$

Generally, $\mathcal{L}_k(\theta_{h,k})$ is nonconvex with respect to the local hyperparameter $\theta_{h,k}$. Removing the constraint leads to the augmented Lagrangian function of Eq. (21):

$$\begin{aligned}&\mathcal{L}_g\left(\{\theta_h\}, \mathcal{Z}, \{\beta\}\right) = \\ &\sum_{k=1}^{K}\left(\mathcal{L}_k(\theta_{h,k}) + \beta_k^T(\theta_{h,k} - \mathcal{Z}) + \frac{\rho_k}{2}\|\theta_{h,k} - \mathcal{Z}\|_2^2\right),\end{aligned} \tag{22}$$

where β_k is the dual variable and ρ_k represents a predetermined regularization parameter. The $(t+1)$-th iteration of solving Eq. (22) can be expressed as:

$$\begin{aligned}\mathcal{Z}^{t+1} &= \frac{1}{K}\sum_{k=1}^{K}\left(\theta_{h,k}^t + \frac{1}{\rho_k}\beta_k^t\right), \\ \theta_{h,k}^{t+1} &= \arg\min_{\theta_{h,k}}\Big(\mathcal{L}_k(\theta_{h,k}) + (\beta_k^t)^T(\theta_{h,k} - \mathcal{Z}^{t+1}) \\ &\quad + \frac{\rho_k}{2}\|\theta_{h,k} - \mathcal{Z}^{t+1}\|_2^2\Big), \\ \beta_k^{t+1} &= \beta_k^t + \rho_k\left(\theta_{h,k}^{t+1} - \mathcal{Z}^{t+1}\right).\end{aligned} \tag{23}$$

After a certain number of iterations or if the difference between the global hyperparameters of two consecutive iterations becomes sufficiently small, i.e., $\|\mathcal{Z}^t - \mathcal{Z}^{t-1}\|_2^2 \le \epsilon$, the global hyperparameters $\mathcal{Z}$ are considered as the final federated fused parameters. Finally, participants substitute the fused parameters for the original kernel parameters in the

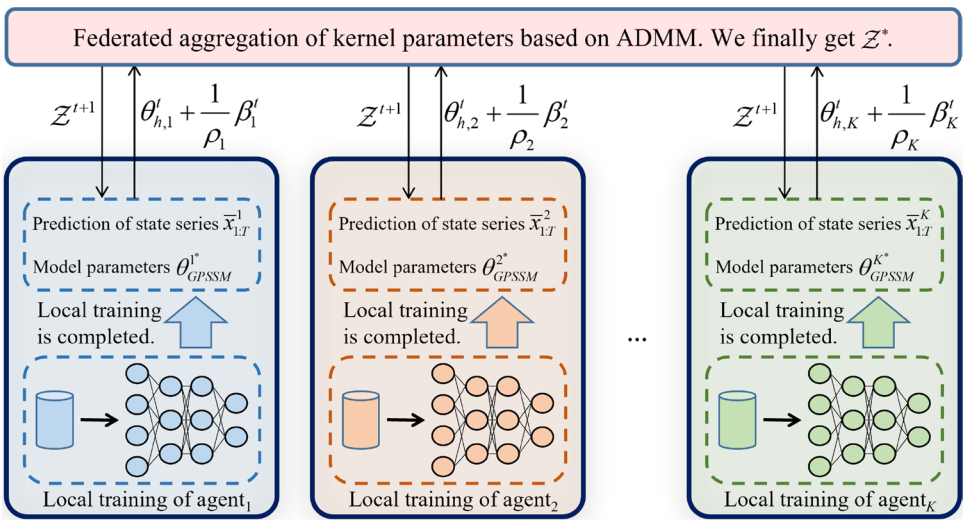

Fig. 3 Federated aggregation of multiple models

local models, obtaining the federated model. The federated aggregation of multiple models is shown in Algorithm 2.

Algorithm 2 Parameter aggregation for multi-agent cooperative localization

1. **Input**: for $k = 1, 2, ..., K$:
   $\bar{x}_{k,1:T} \leftarrow$ Prediction of state series
   $\theta_{h,k} \leftarrow$ Kernel parameters for local model
2. **Output**: $\mathcal{Z}^* \leftarrow$ The fused kernel parameters
3. Initialization Parameters: ρ_k, β_k for $k = 1, 2, ..., K$
   $\epsilon \leftarrow$ Difference threshold of two successive iterations
4. **FOR** $t = 1$: max iteration
5. $\mathcal{Z}^t = \frac{1}{K}\sum_{k=1}^{K}\left(\theta_{h,k} + \frac{1}{\rho_k}\beta_k\right)\triangleleft$ for central server
6. **IF** $\|\mathcal{Z}^t - \mathcal{Z}^{t-1}\|_2^2 \leq \epsilon$ **THEN BREAK**
7. **ELSE** $\mathcal{Z}^t$ is sent to local models
8. **FOR** $k = 1 : K\triangleleft$ for local participants
9. $\theta_{h,k} \leftarrow \arg\min_{\theta_{h,k}}\left(\mathcal{L}_k(\theta_{h,k}) + (\beta_k)^T(\theta_{h,k} - \mathcal{Z}^t) + \frac{\rho_k}{2}\|\theta_{h,k} - \mathcal{Z}^t\|_2^2\right)$
10. $\beta_k \leftarrow \beta_k + \rho_k\left(\theta_{h,k} - \mathcal{Z}^t\right)$
11. upload $\theta_{h,k}, \beta_k$
12. **ENDFOR**
13. **ENDFOR**
14. $\mathcal{Z}^* \leftarrow \mathcal{Z}^t$

5 Numerical Simulation and Analysis

5.1 Experimental Environment and Local Model Settings

During the simulation stage, the experiment is conducted in the Gazebo environment. The agent used is the Turtlebot3 Burger, which navigates a $4 \times 5.5\text{m}^2$ area with coordinates $[-1, -1]$ and $[3, 4.5]$ representing the lower left and upper right corners of the area, respectively. Landmarks are positioned at [0, 0], [2, 0], [2, 3.5], and [0, 3.5] within the area. UWB (Ultra-Wideband) technology is utilized to measure the distance and angle between the agent and the four landmarks, while the IMU (Inertial Measurement Unit) provides data on angular velocity and linear acceleration. The UWB and IMU data are sampled at a frequency of 100Hz, which serves as the system input u in the state-space model (1) for inferring the agent's position at the next time step. Each agent undergoes two repetitions of motion trajectories, moving from the bottom left corner to the top right corner for a duration of 12 seconds. These trajectories are used as local training data.

Based on the simulation settings, the dimensions of the latent state x, system input u, and system output y are set to $D_x = 2$, $D_u = 14$, and $D_y = 2$, respectively. The system input u consists of 8-dimensional UWB readings and 6-dimensional IMU readings, while the system output y represents the position prediction of the trajectory. During the training process, incorporating the complete trajectory data into the model at each iteration would result in high computational complexity. This could hinder the convergence of model parameters to the optimal values and negatively impact the performance of the final federated model. To alleviate the computational burden, randomly selected batches of data are used for each training iteration. The hyperparameters for single-agent localization are presented in Table 1.

Table 1 Hyperparameter settings of single model

Hyperparameter	Value
Iteration	1000
Sampling times	20
Number of pseudo points	100
Learning rate	0.005
Batch size	5
Batch length	500

Table 2 Initialization of some parameters

Parameter	Value (unit)
Variance of initial latent state Σ_{x_1}	$[0.01^2] * D_x\ (m^2)$
Variance of inducing outputs $\Sigma_{1:D_x}$	$[0.01^2] * D_x\ (m^2)$
Noise of state transition $\sigma^2_{x,1:D_x}$	$[0.002^2] * D_x\ (m^2)$
Noise of measurement output $\sigma^2_{y,1:D_y}$	$[0.05^2] * D_y\ (m^2)$
Lengthscale l	$[2] * (D_x + D_u)$
Signal variance σ_f^2	0.5^2

Besides, the initial values of θ_{GPSSM} play a crucial role in determining both the training efficiency and performance. The mean value of the initial state μ_{x_1} varies depending on the selected batch in each iteration. In our experiment, we set μ_{x_1} as the starting point of the selected batch. This means that the model will predict the trajectory based on the initial position. Since there is a difference in μ_{x_1} for each iteration, we choose to fix the variance of the initial latent state Σ_{x_1}. Consequently, the third term in Eq. (16) can be omitted. Additionally, for each dimension, the inducing inputs are sampled from a uniform distribution $\mathcal{U} \sim (-2, 2)$, while the mean of the inducing outputs μ_d for each dimension is sampled from a Gaussian distribution $\mathcal{N}(0, 0.05^2)$. The initialization of other parameters is provided in Table 2. It should be noted that, according to the mean-field theory, the variance and noise parameters are considered to be independent across dimensions without considering covariance. After training

the local model for a single agent, the predicted latent state series is obtained for subsequent federated aggregation.

5.2 Results and Analysis

In this paper, we evaluate the performance of a multi-agent cooperative localization model based on federated learning in the following three aspects: (1) analyzing the impact of different numbers of participants in both i.i.d. and non-i.i.d. scenarios; (2) comparing the results with centralized learning as a benchmark to verify the robustness of the federated model trained in non-i.i.d. scenarios; (3) examining the transfer performance of the federated model trained in non-i.i.d. scenarios.

Since the value of ELBO does not intuitively reflect the precision and accuracy of the predicted trajectory, we introduce the root mean square error (RMSE) as a quantitative evaluation index. RMSE is defined as:

$$RMSE = \sqrt{\frac{1}{T}\sum_{t=1}^{T}\left[(x_t - \hat{x}_t)^2 + (y_t - \hat{y}_t)^2\right]} \tag{24}$$

where (x_t, y_t) is the predicted 2-D position at time step t, and $(\hat{x}_t, \hat{y}_t)$ is the corresponding ground truth.

In the experiments involving i.i.d. scenarios, the training data of all the agents are subjected to zero-mean Gaussian noise interference, as shown in Table 3. On the other hand, in the non-i.i.d. experiments, the data of different agents are exposed to distinct zero-mean Gaussian noise interference. The noise configuration of the test set is different from the noise settings of all the agents, with variance sampled from various intervals for different sensors, as presented in Table 4.

When conducting kernel parameter aggregation for multi-agent cooperative localization according to Algorithm 2, the dual variables β_k and the regularization parameters ρ_k are initialized to $\beta_k = 0$ and $\rho_k = 500$ $(k = 1, 2, ..., K)$.

The effect of varying numbers of agents involved on the final performance in both i.i.d. scenarios and non-i.i.d. scenarios is shown in Fig. 4. The effect of varying numbers of agents involved on the final performance in both i.i.d. scenarios and non-i.i.d. scenarios is shown in Fig. 4. In the i.i.d. settings, an increasing number of participants provides more homogeneous data for the federated model, resulting in a decrease in overall RMSE. The median RMSE values are 0.1474m, 0.1412m,0.1343m, and 0.1368m for the number of agents of 4, 8, 12, and 16. While in the non-i.i.d. scenarios, the median RMSE values are 0.1605m, 0.1640m,0.1599m, and 0.1614m, respectively. Unlike in the i.i.d. scenario, the overall error does not exhibit a decreasing trend and remains almost constant. Nevertheless, the federated model exhibits robust generalization as the number of agents increases. The performance of different agents applying federated kernel parameters remains stable when confronted with data from novel scenarios, which is shown in Fig. 4 as a concentrated distribution of blue boxes.

Table 3 Noise Settings in i.i.d. Scenarios

Sensor data	Noise settings
Angular velocity in IMU	2e-4 *rad/s* (stddev)
Linear acceleration in IMU	1.7e-2 m/s^2 (stddev)
Distance in UWB	0.05 m^2 (dev)

Table 4 Noise Settings in non-i.i.d. Scenarios

Sensor data	Noise sampling interval
Angular velocity in IMU	[1e-5, 2e-2] *rad/s* (stddev)
Linear acceleration in IMU	[1.7e-4, 1.7e-1] m/s^2 (stddev)
Distance in UWB	[0.04, 0.15] m^2 (dev)

In actual scenarios, the data collected by different agents often exhibit heterogeneity. Thus, this study aims to enhance the adaptability, robustness, and generalization of the federated model across diverse scenarios by leveraging the federated learning architecture, while ensuring the privacy of data. Figure 5 presents a comparison between the trajectory predictions obtained through centralized learning and federated learning on the identical test dataset. In Fig. 5, all trajectories are smoothed with a sliding window and RMSE is recalculated. Centralized learning, similar the traditional machine learning, requires the aggregation of heterogeneous training data from all agents [13]. The centralized model achieves a performance

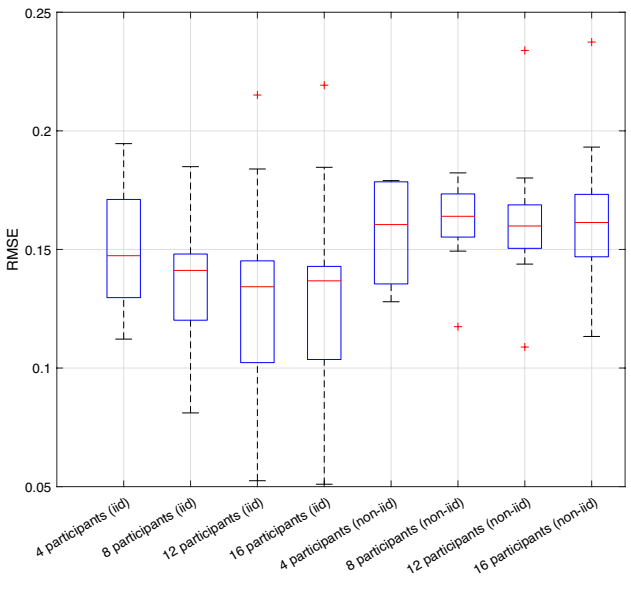

Fig. 4 Effect of varying number of agents in iid and non-iid scenarios

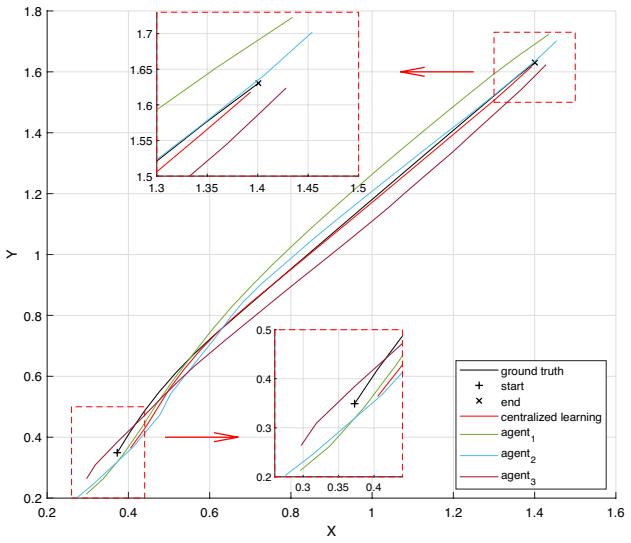

Fig. 5 Performance comparison between centralized learning and federated learning in non-iid scenarios

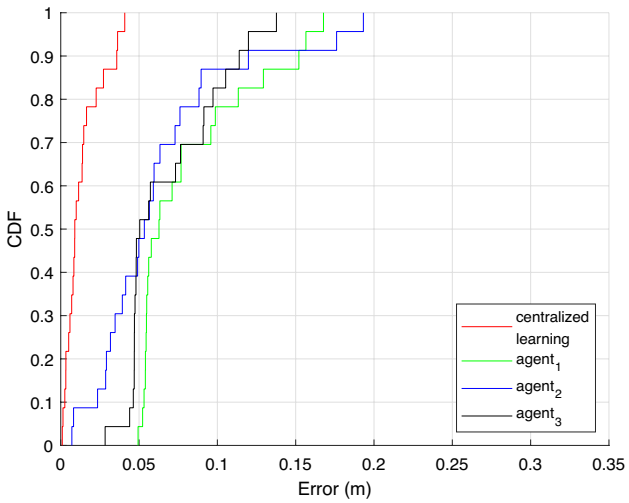

Fig. 6 CDF comparison between centralized learning and federated learning in non-iid scenarios

of RMSE=0.0175m on the test dataset, yet this approach undoubtedly exposes data privacy during the data transmission. In contrast, with the help of the federated learning framework, the RMSEs were 0.0888m, 0.0779m, and 0.0742m for three agents employing the federated kernel parameters constructed by the 16 agents jointly. Although the federated model has a slight disparity with the centralized model in position prediction accuracy, it avoids privacy leakage caused by data sharing and better secures local data. Furthermore, the utilization of federated kernel parameters by different agents yields satisfactory accuracy in position prediction, which indicates the robustness of the federated learning model. Figure 6 displays the cumulative error distribution, revealing that when employing the federated kernel parameters, different agents achieve approximately 80% of points with errors below 0.1m.

Ultimately, Fig. 7 illustrates the transfer performance of the federated model in novel scenarios. The blue line represents the loss on the validation set during regular training, which exhibits high fluctuations and poor convergence in the early stages. In contrast, the federated model, constructed on heterogeneous data, demonstrates remarkable robustness when confronted with novel scenes. The agents apply the federated learning model directly to new environments, resulting in rapid and stable loss convergence that consistently surpasses regular training methods.

6 Conclusion

This paper proposes a novel cooperative localization method based on federated learning. A Gaussian process state-space model dynamically localized the single agent, effectively handling noise. To avoid data privacy leakage in cooperative localization, we realize an aggregation of multi-agent models based on a federated learning framework to form a cooperative model. Compared with centralized learning, the federated model achieves satisfactory position accuracy while ensuring the privacy of the original data. Moreover, it demonstrates robustness and generalization performance when dealing with heterogeneous data. Furthermore, the federated model is adept at transfer learning, enabling rapid adaptation to new scenarios. In future work, we will focus on improving the quality of training data and enriching the diversity of agent motions to further enhance the performance of the federated model.

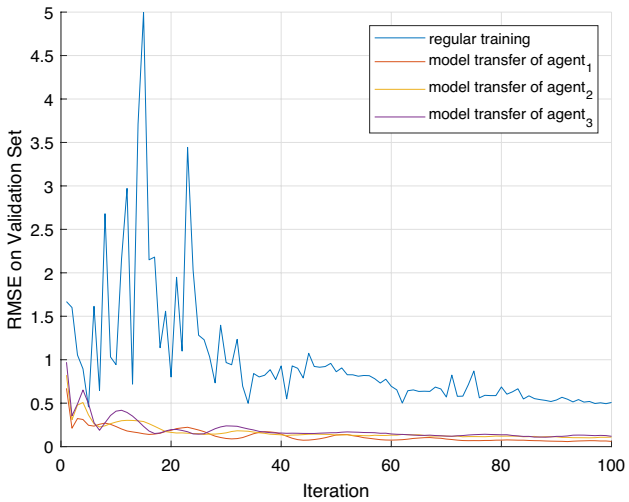

Fig. 7 The performance of model transfer in federated learning

Acknowledgements This work is supported in part by the National Natural Science Foundation of China under Grant 62101029, and in part by the China Scholarship Council Award under Grant 202006465043.

Data Availability All related simulation data will be made available upon reasonable request from the corresponding authors (R. Wang and C. Xu) for academic use and within the limitations of the provided informed consent by the corresponding authors upon acceptance.

References

1. X. Li and F. Cao, Location based TOA algorithm for UWB wireless body area networks, in: *2014 IEEE 12th International Conference on Dependable, Autonomic and Secure Computing*. pp. 507–511, IEEE, 2014.
2. A. Chehri, P. Fortier and P.-M. Tardif, On the TOA estimation for UWB ranging in complex confined area, *International Symposium on Signals, Systems and Electronics, IEEE*, Vol. 2007, pp. 533–536, 2007.
3. H. Liu, H. Darabi, P. Banerjee and J. Liu, Survey of wireless indoor positioning techniques and systems, *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, Vol. 37, No. 6, pp. 1067–1080, 2007.
4. C. Zhang, W. Wang, C. Xu, X. Sun, J. Guo, Z. Tu and M. Long, A TOA-based optimization positioning algorithm for non-line-of-sight errors, *Journal of Nanjing University of Posts and Telecommunications (Natural Science Edition)*, Vol. 42, pp. 56–63, 2022.
5. J. Guo, J. Du and D. Xu, Navigation and positioning system applied in underground driverless vehicle based on IMU, in: *2018 International Conference on Robots & Intelligent System (ICRIS)*. pp. 13–16, IEEE, 2018.
6. H. Wymeersch, J. Lien and M. Z. Win, Cooperative localization in wireless networks, *Proceedings of the IEEE*, Vol. 97, No. 2, pp. 427–450, 2009.
7. B. McMahan, E. Moore, D. Ramage, S. Hampson and B.A. Arcas, Communication-efficient learning of deep networks from decentralized data, in: *Artificial Intelligence and Statistics*. pp. 1273–1282, PMLR, 2017.
8. Y. Zhao, J. Zhao, M. Yang, T. Wang, N. Wang, L. Lyu, D. Niyato and K.-Y. Lam, Local differential privacy-based federated learning for internet of things, *IEEE Internet of Things Journal*, Vol. 8, No. 11, pp. 8836–8853, 2020.
9. G. Xu, H. Li, S. Liu, K. Yang and X. Lin, Verifynet: Secure and verifiable federated learning, *IEEE Transactions on Information Forensics and Security*, Vol. 15, pp. 911–926, 2019.
10. X. Kong, H. Gao, G. Shen, G. Duan and S. K. Das, Fedvcp: A federated-learning-based cooperative positioning scheme for social internet of vehicles, *IEEE Transactions on Computational Social Systems*, Vol. 9, No. 1, pp. 197–206, 2021.
11. X. Cheng, C. Ma, J. Li, H. Song, F. Shu and J. Wang, Federated learning-based localization with heterogeneous fingerprint database, *IEEE Wireless Communications Letters*, Vol. 11, No. 7, pp. 1364–1368, 2022.
12. W. Li, C. Zhang and Y. Tanaka, Pseudo label-driven federated learning-based decentralized indoor localization via mobile crowdsourcing, *IEEE Sensors Journal*, Vol. 20, No. 19, pp. 11556–11565, 2020.
13. B. S. Ciftler, A. Albaseer, N. Lasla and M. Abdallah, Federated learning for RSS fingerprint-based localization: A privacy-preserving crowdsourcing method, in: *International Wireless Communications and Mobile Computing (IWCMC)*, Vol. 2020, pp. 2112–2117, IEEE, 2020.
14. H. Xie and H. Yang, An indoor location method combining random forest with deep learning, *Journal of Shanghai Maritime University*, Vol. 41, pp. 117–121, 2020.
15. W. Yan, D. Jin, Z. Lin and F. Yin, Graph neural network for large-scale network localization, in: *ICASSP 2021–2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. pp. 5250–5254, IEEE, 2021.
16. N. Patwari, A. O. Hero, M. Perkins, N. S. Correal and R. J. O'dea, Relative location estimation in wireless sensor networks, *IEEE Transactions on signal processing*, Vol. 51, No. 8, pp. 2137–2148, 2003.
17. A. Simonetto and G. Leus, Distributed maximum likelihood sensor network localization, *IEEE Transactions on Signal Processing*, Vol. 62, No. 6, pp. 1424–1437, 2014.
18. J. Yoo, K. H. Johansson and H. J. Kim, Indoor localization without a prior map by trajectory learning from crowdsourced measurements, *IEEE Transactions on Instrumentation and Measurement*, Vol. 66, No. 11, pp. 2825–2835, 2017.
19. M. Brossard and S. Bonnabel, Learning wheel odometry and IMU errors for localization, in: *2019 International Conference on Robotics and Automation (ICRA)*. pp. 291–297, IEEE, 2019.
20. R. Sharma and C. Taylor, Cooperative navigation of MAVS in GPS denied areas, in: *2008 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*. pp. 481–486, IEEE, 2008.
21. 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 Transactions on Signal Processing*, Vol. 68, pp. 1120–1135, 2020.
22. R. M. Buehrer, H. Wymeersch and R. M. Vaghefi, Collaborative sensor network localization: Algorithms and practical issues, *Proceedings of the IEEE*, Vol. 106, No. 6, pp. 1089–1114, 2018.
23. S. Li, W. Ni, C. K. Sung and M. Hedley, Recent advances on cooperative wireless localization and their application in inhomogeneous propagation environments, *Computer Networks*, Vol. 142, pp. 253–271, 2018.
24. L. Wielandner, E. Leitinger and K. Witrisal, RSS-based cooperative localization and orientation estimation exploiting antenna directivity, *IEEE Access*, Vol. 9, pp. 53046–53060, 2021.
25. L. Huang, A. L. Shea, H. Qian, A. Masurkar, H. Deng and D. Liu, Patient clustering improves efficiency of federated machine learning to predict mortality and hospital stay time using distributed electronic medical records, *Journal of Biomedical Informatics*, Vol. 99, 103291, 2019.
26. M. Chen, R. Mathews, T. Ouyang, and F. Beaufays, Federated learning of out-of-vocabulary words. arXiv preprint arXiv:1903.10635, 2019.
27. C. Wu, F. Wu, Y. Cao, Y. Huang and X. Xie, Fedgnn: Federated graph neural network for privacy-preserving recommendation. arXiv preprint arXiv:2102.04925, 2021.
28. Y. Liu, H. Li, J. Xiao and H. Jin, Floc: Fingerprint-based indoor localization system under a federated learning updating framework, in: *2019 15th International Conference on Mobile Ad-Hoc and Sensor Networks (MSN)*. pp. 113–118, IEEE, 2019.
29. B. Liu, L. Wang and M. Liu, Lifelong federated reinforcement learning: A learning architecture for navigation in cloud robotic systems, *IEEE Robotics and Automation Letters*, Vol. 4, No. 4, pp. 4555–4562, 2019.
30. H.-K. Lim, J.-B. Kim, J.-S. Heo and Y.-H. Han, Federated reinforcement learning for training control policies on multiple IoT devices, *Sensors*, Vol. 20, No. 5, pp. 1359, 2020.
31. Z. He, G. Liu, X. Zhao and M. Wang, Overview of gaussian process regression, *Control and Decision*, Vol. 28, pp. 1121–1129, 2013.

32. M. Deisenroth and J. W. Ng, Distributed gaussian processes, in: *International Conference on Machine Learning*. pp. 1481–1490, PMLR, 2015.
33. J. Hensman, N. Fusi and N. D. Lawrence, Gaussian processes for big data. arXiv preprint arXiv:1309.6835, 2013.
34. A. Doerr, C. Daniel, M. Schiegg, N.-T. Duy, S. Schaal, M. Toussaint and T. Sebastian, Probabilistic recurrent state-space models, in: *International Conference on Machine Learning*. pp. 1280–1289, PMLR, 2018.
35. S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, et al., Distributed optimization and statistical learning via the alternating direction method of multipliers, *Foundations and Trends® in Machine Learning*, Vol. 3, No. 1, pp. 1–122, 2011.

Fangwen Ye is currently working toward the Master degree at University of Science and Technology Beijing. His research interests include patter recognition and internet of things.

Ran Wang is currently working toward the Doctor degree at University of Science and Technology Beijing. Her research interests include distributed security, swarm intelligence and internet of things.

Sisui Tang is currently working toward the Master degree at University of Science and Technology Beijing. Her research interests include blockchain, swarm robots and internet of things.

Shihong Duan received Ph.D. degree in computer science from University of Science and Technology Beijing (USTB). She is an associate professor with the School of Computer and Communication Engineering, USTB. Her research interests includes wireless indoor positioning, human gesture recognition and motion capture.

Cheng Xu received the B.E., M.S. and Ph.D. degree from the University of Science and Technology Beijing (USTB), China in 2012, 2015 and 2019 respectively. He is currently working as an associate professor in the Data and Cyber-Physical System Lab (DCPS) at University of Science and Technology Beijing. He is supported by the Post-doctoral Innovative Talent Support Program from Chinese government in 2019. He is an associate editor of International Journal of Wireless Information Networks. His research interests now include swarm intelligence, multi-robots network, wireless localization and internet of things. He is a member of the IEEE.