2025 IEEE 36th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC): Track 4: Emerging Technologies, Applications and Platforms

2025 IEEE 36th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC) | 979-8-3503-6323-4/25/$31.00 ©2025 IEEE | DOI: 10.1109/PIMRC62392.2025.11274585

Adaptive Learning for Wireless Localization in Multiple Environments

Verónica Álvarez*, Santiago Mazuelas†‡, and Moe Z. Win§

*Wireless Information and Network Sciences Laboratory, Massachusetts Institute of Technology, Cambridge, MA, USA
†Basque Center for Applied Mathematics (BCAM), Bilbao, Spain
‡IKERBASQUE - Basque Foundation for Science
§Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA, USA

***Abstract*—Accurate wireless localization is crucial for multiple real-world applications such as modern autonomy, Internet-of-Things, and smart cities. Machine learning (ML) methods can significantly improve localization performance, especially through soft information (SI) techniques. However, existing ML methods are designed to determine positions within the same environment (network) where training data are collected and cannot exploit data from environments with assorted types. In addition, the usage of existing approaches across multiple environments would significantly increase privacy risks, as they require transferring large amounts of data to a central server. This paper presents a SI-based method for wireless localization that can leverage information from multiple environments and adapt to the specific characteristics of each environment. Differently from existing techniques, we propose methods that ensure privacy by only transfering summary statistics of the training data. The performance of the proposed method is evaluated using real-world datasets obtained in multiple localization environments. The results show that the proposed method significantly outperforms existing techniques and provides adaptation to assorted environments together with privacy protection of location data.**

***Index Terms*—Wireless localization, soft information, adaptive learning, federated learning.**

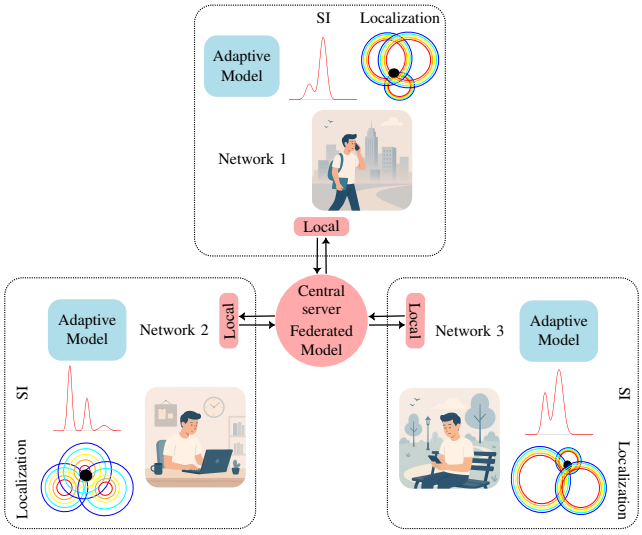


Fig. 1: The proposed localization techniques leverage information from multiple networks through a federated-based algorithm. Then, the methodology performs fine-tuning to adapt the parameters to the specific characteristics of each environment.

I. Introduction

ACCURATE wireless localization [1], [2] is crucial for multiple real-world applications such as modern autonomy [3], Internet-of-Things [4], and smart cities [5]. Recent localization techniques based on machine learning (ML) methods have achieved significant performance improvements by exploiting training data (e.g., measurement-distance pairs) [6], [7]. In particular, soft information (SI)-based localization can further improve performance by learning generative models that statistically characterize the relationship between measurements and distances [4], [8], [9]. However, the usage of training data for localization can increase privacy risks since localization information is often sensitive.

Existing ML-based localization techniques cannot leverage information from multiple environments (networks) with assorted typologies. These techniques learn a global predictive model by using all training data. For instance, existing ML-based localization techniques use training data to learn a global regression model utilizing support vector machines or Gaussian processes [6], [7], and to learn a global generative model for SI utilizing the expectation maximization (EM) algorithm [8], [9]. Existing ML-based techniques are designed to determine positions in the same environment where training data are collected, making them less effective when determining positions in new environments. For instance, existing methods trained in outdoor line-of-sight (LOS) environments often fail to perform well in indoor scenarios with severe non-line-of-sight (NLOS) conditions. In addition, if existing methods used data from multiple environments, they would need to transfer all the training data to a central server to learn a global model. Hence, such approaches can significantly increase privacy risks, as they would expose sensitive information.

This paper proposes localization techniques that effectively leverage data from multiple networks and adapt to the specific characteristics of each environment. In addition, the proposed methodology reduces privacy risks by transferring only summary statistics of the training data and preserving local generative models within each network. Specifically, the

The fundamental research described in this paper was supported, in part, by project PID2022-137063NBI00 funded by MCIN/AEI/10.13039/501100011033, by a postdoctoral grant from the Basque Government, by the National Science Foundation under Grant CNS-2148251, by the federal agency and industry partners in the RINGS program, and by the Robert R. Taylor Professorship. (*Corresponding author: Moe Z. Win.*)

979-8-3503-6323-4/25/$31.00 ©2025 IEEE

main contributions of the paper are as follows:

- we propose a methodology for SI-based localization that reduces privacy risks by transferring only summary statistics of training data and using a federated EM algorithm;
- we propose adaptive learning techniques that fine-tune global model parameters to capture the specific characteristics of each environment;
- we detail the efficient implementation of the steps for the proposed SI-based localization techniques based on federated and adaptive learning; and
- we numerically quantify the performance improvement provided by the proposed methods in comparison with existing techniques using real-world datasets.

Notation: Random variables (RVs) are displayed in sans serif, upright fonts; their realizations in serif, italic fonts. Vectors and matrices are denoted by bold lowercase and uppercase letters, respectively. For example, a RV and its realization are denoted by x and x; a random vector and its realization are denoted by $\mathbf{x}$ and $\boldsymbol{x}$, respectively. The function $f_{\mathbf{x}}(\boldsymbol{x})$ denotes the probability density function (PDF) of a vector of continuous RVs $\mathbf{x}$; $f_{\mathbf{x}|\mathbf{z}}(\boldsymbol{x}\,|\,\boldsymbol{z})$ and, for brevity when possible, $f(\boldsymbol{x}\,|\,\boldsymbol{z})$ denote the PDF of $\mathbf{x}$ conditional on $\mathbf{y}=\boldsymbol{y}$; $\varphi(\boldsymbol{x}\,;\,\boldsymbol{\mu},\boldsymbol{\Sigma})$ denotes the PDF of a Gaussian RV $\mathbf{x}$ with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$; operator $\mathbb{E}\{\cdot\}$ denotes the expectation of the argument. For a vector $\boldsymbol{a}$, the transpose is denoted by $\boldsymbol{a}^{\mathrm{T}}$. The subscripts of f and $\mathbb{E}\{\cdot\}$ may be omitted for brevity when clear from the context.

II. Preliminaries

This section first formulates the problem of wireless localization in multiple networks and then recalls conventional approaches based on learning SIs.

A. Problem Formulation

Localization techniques estimate agents' positions given measurements related to the position of agents such as waveform samples, time-of-arrival (TOA), and received signal strength (RSS). Measurement vectors $\mathbf{y}\in\mathbb{R}^M$ are composed by features taken with respect to a single node or different networked nodes. For instance, measurements can be relative angles of a single node with respect to an anchor (e.g., angle-of-arrivals (AOAs)) or times between different nodes (e.g., round-trip-times (RTTs)). In the addressed setting, there are T local networks formed by $N_a^{(t)}$ agents and $N_b^{(t)}$ anchors for $t=1,2,\ldots,T$. Let $\{(\boldsymbol{y}_n^{(t)},d_n^{(t)})\}_{n=1}^{N^{(t)}}$ denote the $N^{(t)}$ measurement-distance pairs (training data) corresponding with the tth network for $t=1,2,\ldots T$ where $\mathbf{y}\in\mathbb{R}^M$ and $\mathrm{d}\in\mathbb{R}$ denote, respectively, the measurement vector and the distance between an unspecified pair of nodes in a network.

ML-based localization techniques exploit training data to obtain prediction functions. Conventional techniques commonly obtain prediction functions that assign distance estimates to measurements. Then, these distance estimates are used to determine the agents' positions [10], [11]. SI-based localization techniques utilize measurement-distance pairs to learn SI functions as described in the following.

B. Soft Information

The SI is any function proportional to the likelihood of distances and measurements [8], [4]. Specifically, for a measurement vector $\mathbf{y}=\boldsymbol{y}$, the SI is denoted as $\mathcal{L}_{\boldsymbol{y}}(d)$ and is proportional to the distribution $f_{\mathbf{y}|\mathrm{d}}(\boldsymbol{y}\,|\,d)$, that is $\mathcal{L}_{\boldsymbol{y}}(d)\propto f_{\mathbf{y}|\mathrm{d}}(\boldsymbol{y}\,|\,d)$. In the absence of prior knowledge, we have that $\mathcal{L}_{\boldsymbol{y}}(d)\propto f_{\mathbf{y},\mathrm{d}}(\boldsymbol{y},d)$ otherwise, we have that $\mathcal{L}_{\boldsymbol{y}}(d)\propto f_{\mathbf{y},\mathrm{d}}(\boldsymbol{y},d)/f_{\mathrm{d}}(d)$. Prior knowledge can be contextual data such as floor plans, signal propagation characteristics, and environmental constraints.

SI is obtained in two stages. At the offline stage, SI learning methods use training data to estimate the joint distribution of distances and measurements (i.e., generative model), denoted by $\widehat{f}_{\mathbf{y},\mathrm{d}}(\boldsymbol{y},d)$. During the online stage, for each new measurement vector $\boldsymbol{y}'\in\mathbb{R}^M$ learning methods estimate the SI $\mathcal{L}_{\boldsymbol{y}'}(d)$ as $\mathcal{L}_{\boldsymbol{y}'}(d)\propto\widehat{f}_{\mathbf{y},\mathrm{d}}(\boldsymbol{y}',d)$ using the generative model obtained at the offline stage.

The generative model $\widehat{f}_{\mathbf{y},\mathrm{d}}(\boldsymbol{y},d)$ can be approximated by a mixture of K Gaussian distributions [8] with parameters $\theta=\{\alpha_k,\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k\,:\,k=1,2,\ldots,K\}$, that is

$$\widehat{f}_{\mathbf{y},\mathrm{d}}(\boldsymbol{y},d\,;\,\theta)\;=\;\sum_{k=1}^{K}\alpha_k\,\varphi\big([\;\boldsymbol{y}^{\mathrm{T}}\;d\;]^{\mathrm{T}}\,;\,\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k\big)\qquad(1)$$

with weights $\alpha_k\geq 0$ for $k=1,2,\ldots,K$ and $\sum_{k=1}^{K}\alpha_k=1$. Vector $\boldsymbol{\mu}_k$ and matrix $\boldsymbol{\Sigma}_k$ denote mean and covariance parameters, respectively, of the kth Gaussian distribution for $k=1,2,\ldots,K$. In cases where the measurement vectors are high dimensional, SI methods can use a function that reduces the dimensionality of the measurements such as principal component analysis (PCA). In addition, measurement-distance pairs can be represented using normalization or "data sphering" functions [8].

Existing SI learning approaches transfer all training data to a central server to estimate the mixture of Gaussian distributions in (1) using the EM algorithm [8], [9]. Let $\{(\boldsymbol{y}_n,d_n)\}_{n=1}^{N}$ denote the set of training data from all the networks, that is $\{(\boldsymbol{y}_n,d_n)\}_{n=1}^{N}=\cup_{t=1}^{T}\{(\boldsymbol{y}_n^{(t)},d_n^{(t)})\}_{n=1}^{N^{(t)}}$. The EM algorithm aims to obtain weight, mean, and covariance parameters of the mixture of Gaussian distributions that minimize

$$\widehat{f}\;\in\;\underset{\tilde{f}}{\operatorname{argmin}}\;\frac{1}{N}\sum_{n=1}^{N}\log\,\tilde{f}_{\mathbf{y},\mathrm{d}}(\boldsymbol{y}_n,d_n)\qquad(2)$$

where $\tilde{f}$ is the density function for the training data.

The EM algorithm [12] is an iterative process executed on the central server. This algorithm estimates the model parameters θ that maximize the likelihood function defined in optimization problem (2). At each iteration l for $l=1,2,\ldots$, the EM algorithm obtains updated parameters $\theta(l)$ using the estimates from the previous iteration $\theta(l-1)$ and the measurement-distance pairs $\{(\boldsymbol{y}_n,d_n)\}_{n=1}^{N}$. Specifically,

at each iteration l, the EM algorithm obtains parameters $\alpha_k(l)$, $\boldsymbol{\mu}_k(l)$, and $\boldsymbol{\Sigma}_k(l)$ as

$$\gamma_{k,n}(l) = \frac{\alpha_k(l-1)\,\varphi(\boldsymbol{z}_n\,;\,\boldsymbol{\mu}_k(l-1),\boldsymbol{\Sigma}_k(l-1))}{\sum_{k=1}^{K}\alpha_k(l-1)\,\varphi(\boldsymbol{z}_n\,;\,\boldsymbol{\mu}_k(l-1),\boldsymbol{\Sigma}_k(l-1))} \tag{3}$$

$$\alpha_k(l) = \frac{1}{N}\sum_{n=1}^{N}\gamma_{k,n}(l) \tag{4}$$

$$\boldsymbol{\mu}_k(l) = \frac{\sum_{n=1}^{N}\gamma_{k,n}(l)\,\boldsymbol{z}_n}{\sum_{n=1}^{N}\gamma_{k,n}(l)} \tag{5}$$

$$\boldsymbol{\Sigma}_k(l) = \frac{\sum_{n=1}^{N}\gamma_{k,n}(l)\,(\boldsymbol{z}_n-\boldsymbol{\mu}_k(l-1))\,(\boldsymbol{z}_n-\boldsymbol{\mu}_k(l-1))^{\mathrm{T}}}{\sum_{n=1}^{N}\gamma_{k,n}(l)} \tag{6}$$

with $\boldsymbol{z}_n = [\,\boldsymbol{y}_n^{\mathrm{T}} \;\; d_n\,]^{\mathrm{T}}$ for $n = 1,2,\ldots,N$, and $k = 1,2,\ldots,K$. Weights $\alpha_k(l)$, means $\boldsymbol{\mu}_k(l)$, and covariances $\boldsymbol{\Sigma}_k(l)$ are the lth iterates for α_k, $\boldsymbol{\mu}_k$, and $\boldsymbol{\Sigma}_k$, respectively.

Given the SI, the position estimate for the ith agent of the tth network is obtained by solving the optimization problem

$$\begin{aligned}\widehat{\boldsymbol{p}}_i^{(t)} &= \underset{\boldsymbol{p}_i^{(t)}}{\operatorname{argmax}}\; f(\{\boldsymbol{y}_{i,j}^{(t)}\}_{j=1}^{N_b^{(t)}}\,|\,\boldsymbol{p}_i^{(t)}) \\ &= \underset{\boldsymbol{p}_i^{(t)}}{\operatorname{argmax}}\prod_{j=1}^{N_b^{(t)}} f(\boldsymbol{y}_{i,j}^{(t)}\,|\,d_{i,j}^{(t)}) = \underset{\boldsymbol{p}_i^{(t)}}{\operatorname{argmax}}\prod_{j=1}^{N_b^{(t)}}\mathcal{L}_{\boldsymbol{y}_{i,j}^{(t)}}(d_{i,j}^{(t)})\end{aligned} \tag{7}$$

where $d_{i,j}^{(t)} \in \mathbb{R}$ and $\boldsymbol{y}_{i,j}^{(t)} \in \mathbb{R}^M$ denote the distance and measurement set between nodes i and j. The optimization problem in (7) can be solved by evaluating the SI over a grid of possible values for $\boldsymbol{p}_i^{(t)}$ and selecting the position estimate $\widehat{\boldsymbol{p}}_i^{(t)}$ corresponding to the maximum on that grid [4], [8].

Existing ML methods cannot effectively leverage information from multiple heterogeneous networks because the relationship between measurements and distances is network-dependent. For instance, in networks where most of the propagation is LOS, the measurements can have a more consistent relationship with actual distances. However, in networks affected by substantial NLOS propagation, the measurements can have a significantly more complex relationship with actual distances. A model trained on LOS data can perform poorly when applied to NLOS conditions, as it has not learned the appropriate relationship between measurements and distances specific to harsh propagation conditions. In the following sections, we describe techniques to estimate generative models $\widehat{f}_{\mathbf{y},\mathrm{d}}(\boldsymbol{y},d)$ that can adapt to multiple networks with assorted types and also reduce privacy risks.

III. Methodology of Adaptive Learning

This section first presents the new learning methodology for localization and then describes how the proposed methods obtain summary statistics of the data, capture the specific characteristics of each environment, and reduce privacy risks.

Figure 2 describes the federated and the adaptive stages of the proposed learning methodology. The federated stage leverages information from multiple networks by learning a global model from local training data and reduces privacy risks by transferring only summary statistics of the data. The adaptive stage adapts to network-specific characteristics by fine-tuning the global model. Such adaptive stage also reduces privacy risks by maintaining a specific local model for each network that is not shared with other nodes.

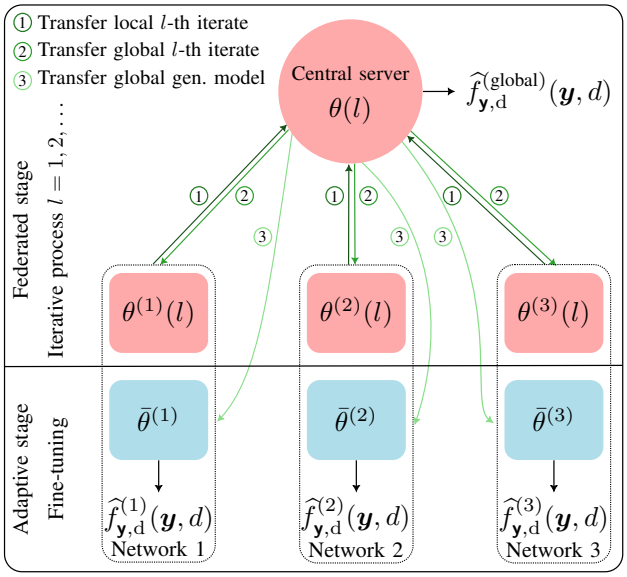


Fig. 2: Diagram of the proposed methodology.

The federated stage is performed through an iterative process characterized by multiple communication rounds (iterations) between the local networks and the central server. In each iteration l, each network computes summary statistics of its data by estimating the parameters $\theta^{(t)}(l)$ of a Gaussian mixture model for $t = 1,2,\ldots,T$. These statistics are transferred to the central server, where they are aggregated to obtain the global model $\theta(l)$. The updated global parameters are then transferred back to the networks. The global model learned by this federated process effectively exploits information from all the networks without requiring direct access to local data.

The adaptive stage is performed locally within each network after completing the federated stage. Each network fine-tunes the parameters of the global model to obtain local parameters $\bar{\theta}^{(t)}$ for $t = 1,2,\ldots,T$ that better capture the specific characteristics of the environment. Finally, the proposed methodology obtains a generative model $\widehat{f}_{\mathbf{y},\mathrm{d}}^{(t)}(\boldsymbol{y},d)$ as in (1) for each tth network using fine-tuned parameters $\bar{\theta}^{(t)}$.

Algorithm 1 details the efficient implementation of the federated and adaptive stages of the proposed methodology, which are detailed in the following.

A. Federated Stage

The proposed methodology estimates a global Gaussian mixture model using a federated EM algorithm [13] that leverages training data from multiple networks. The federated EM algorithm is an iterative process performed over multiple rounds of communication between the networks and the central server. Such algorithm estimates the model parameters θ that maximize the likelihood function defined in optimization problem (2). At each iteration l for $l = 1,2,\ldots$, each tth

Algorithm 1 Methodology of Adaptive Learning

Input: Training data $\{(\boldsymbol{y}_n^{(t)}, d_n^{(t)})\}_{n=1}^{N^{(t)}}$ for $t = 1, 2, \ldots, T$ networks, number of mixtures' components K

Output: Generative model $\widehat{f}_{\mathbf{y},\mathrm{d}}^{(t)}(\boldsymbol{y}, d)$ for $t = 1, 2, \ldots, T$ networks, global generative model $\widehat{f}_{\mathbf{y},\mathrm{d}}^{(\mathrm{global})}(\boldsymbol{y}, d)$

Federated step

for $l = 1, 2, \ldots$ **do**
  for $t = 1, 2, \ldots, T$ **do**
    for $k = 1, 2, \ldots, K$ **do**
      Obtain local parameters $\alpha_k^{(t)}(l), \boldsymbol{\mu}_k^{(t)}(l)$, and $\boldsymbol{\Sigma}_k^{(t)}(l)$ using (4)-(6)
    Transfer local parameters $\theta^{(t)}(l)$ to the central server
  for $k = 1, 2, \ldots, K$ **do**
    Obtain global parameters $\alpha_k(l), \boldsymbol{\mu}_k(l)$, and $\boldsymbol{\Sigma}_k(l)$ using (8)-(10)
  Transfer global parameters $\theta(l)$ to each network
Approximate global model $\widehat{f}_{\mathbf{y},\mathrm{d}}^{(\mathrm{global})}(\boldsymbol{y}, d)$ as in (1) using θ

Adaptive step

for each network t **do**
  for $k = 1, 2, \ldots, K$ **do**
    Initialize local parameters as $\bar{\alpha}_k^{(t)}(0) = \alpha_k, \bar{\boldsymbol{\mu}}_k^{(t)}(0) = \boldsymbol{\mu}_k$, and $\bar{\boldsymbol{\Sigma}}_k^{(t)}(0) = \boldsymbol{\Sigma}_k$
  for $l = 1, 2, \ldots$ **do**
    for $k = 1, 2, \ldots, K$ **do**
      Fine-tune local parameters $\bar{\alpha}_k^{(t)}(l), \bar{\boldsymbol{\mu}}_k^{(t)}(l)$, and $\bar{\boldsymbol{\Sigma}}_k^{(t)}(l)$ using (4)-(6)
  Approximate model $\widehat{f}_{\mathbf{y},\mathrm{d}}^{(t)}(\boldsymbol{y}, d)$ as in (1) using $\bar{\theta}^{(t)}$

network obtains local parameters $\theta^{(t)}(l)$ using the global estimates from the previous iteration $\theta(l-1)$ at the central server and the measurement-distance pairs from the local tth network $\{(\boldsymbol{y}_n^{(t)}, d_n^{(t)})\}_{n=1}^{N^{(t)}}$. Specifically, at each iteration l, each network obtains parameters $\alpha_k^{(t)}(l), \boldsymbol{\mu}_k^{(t)}(l)$, and $\boldsymbol{\Sigma}_k^{(t)}(l)$ using equations (4)-(6) for $k = 1, 2, \ldots, K$. Then, each tth network transfers the summary statistics $\alpha_k^{(t)}(l), \boldsymbol{\mu}_k^{(t)}(l)$, and $\boldsymbol{\Sigma}_k^{(t)}(l)$ to the central server (see Algorithm 1).

The central server aggregates local parameters received from the T networks to obtain updated global parameters. At each iteration l for $l = 1, 2, \ldots$, the central server obtains global parameters $\theta(l)$ using local estimates $\theta^{(t)}(l)$ for $t = 1, 2, \ldots, T$. Specifically, at each iteration l, the central server obtains parameters $\alpha_k(l), \boldsymbol{\mu}_k(l)$, and $\boldsymbol{\Sigma}_k(l)$ as

$$\alpha_k(l) = \frac{1}{T} \sum_{t=1}^{T} \alpha_k^{(t)}(l) \tag{8}$$

$$\boldsymbol{\mu}_k(l) = \frac{\sum_{t=1}^{T} N^{(t)}\, \alpha_k^{(t)}(l)\, \boldsymbol{\mu}_k^{(t)}(l)}{\sum_{t=1}^{T} N^{(t)}\, \alpha_k^{(t)}(l)} \tag{9}$$

$$\boldsymbol{\Sigma}_k(l) = \frac{\sum_{t=1}^{T} N^{(t)}\, \alpha_k^{(t)}(l)\, \boldsymbol{\Sigma}_k^{(t)}(l)}{\sum_{t=1}^{T} N^{(t)}\, \alpha_k^{(t)}(l)} \tag{10}$$

for $k = 1, 2, \ldots, K$. These updated parameters are transferred back to each network for the next iteration until they converge. As a result, the method learns a global model $\widehat{f}_{\mathbf{y},\mathrm{d}}^{(\mathrm{global})}(\boldsymbol{y}, d)$ that effectively leverages data from multiple networks. The proposed methodology for SI-based localization utilize the federated EM algorithm to estimate the global parameters by aggregating information from all the networks without transferring their local training data.

Given the global model parameters θ, the proposed methodology transfers θ to the networks and adapts to the specific environment-characteristics as described in the following.

B. Adaptive Stage

The proposed methodology fine-tunes the global parameters by using measurement-distance pairs of each network. The fine-tuning algorithm is an iterative process performed in each network. Such algorithm estimates final model parameters $\bar{\theta}^{(t)}$ for each tth network with $t = 1, 2, \ldots, T$ using global parameters θ obtained in the federated step and measurement-distance pairs from the tth network $\{(\boldsymbol{y}_n^{(t)}, d_n^{(t)})\}_{n=1}^{N^{(t)}}$. Specifically, at each iteration l for $l = 1, 2, \ldots$, each tth network for $t = 1, 2, \ldots, T$ obtains parameters $\bar{\alpha}^{(t)}(l), \bar{\boldsymbol{\mu}}^{(t)}(l)$, and $\bar{\boldsymbol{\Sigma}}^{(t)}(l)$ as in (4)-(6) with $\bar{\alpha}_k^{(t)}(0) = \alpha_k, \bar{\boldsymbol{\mu}}_k^{(t)}(0) = \boldsymbol{\mu}_k$, and $\bar{\boldsymbol{\Sigma}}_k^{(t)}(0) = \boldsymbol{\Sigma}_k$ for $t = 1, 2, \ldots, T$ and $k = 1, 2, \ldots, K$. The adaptive stage uses as input the global parameters obtained during the federated stage and refines them to produce network-specific parameters (see Algorithm 1). As a result, the method learns generative models $\widehat{f}_{\mathbf{y},\mathrm{d}}^{(t)}(\boldsymbol{y}, d)$ for each tth network with $t = 1, 2, \ldots, T$ that effectively capture the network-specific characteristics.

Local fine-tuning allows the proposed methods to capture the specific characteristics of each environment using specific parameters for each network, which is crucial in heterogeneous environments that result in training data with varying statistical behavior. In addition, local fine-tuning allows the proposed methods to further enhance user privacy, since the adaptation is performed entirely within each network and the local models are not communicated to other networks. This approach ensures privacy and avoids potential legal or ethical concerns associated with data transferring in scenarios where data includes sensitive location or behavioral information.

C. Data Privacy

The proposed methodology enhances privacy by transferring only summary statistics between each network and the central server. In addition, differently to existing ML methods for localization the proposed techniques transfer only a limited number of bits per iteration.

Algorithm 1 specifies the information transferred during the federated and the adaptive stages of the proposed methodology. If each scalar is represented using B bits, the communication cost per iteration is $BTK(1 + (M+1) + (M+1)^2)$ bits, where T is the number of networks, K is the number of Gaussian distributions, and M is the dimension of the measurement vector. The three terms inside the parentheses correspond respectively to weights, means, and covariances.

In contrast, existing ML techniques would require to transfer all the training data from the networks to the central server. This would result in a communication cost of

TABLE I: RMSE in meters of the proposed methods and existing techniques using 3 real-world datasets.

Rounds	Methods	House	Office	Industrial
	Least squares (Centralized)	0.45 ± 0.01	1.12 ± 0.03	0.62 ± 0.01
	NNs (Centralized)	0.42 ± 0.04	1.09 ± 0.12	0.56 ± 0.04
	EM SI (Centralized)	**0.31** ± 0.01	**1.07** ± 0.07	**0.54** ± 0.03
	FedAvg with NNs (Federated)	0.61 ± 0.08	**1.07** ± 0.42	0.75 ± 0.06
10	EM federated SI (Federated)	**0.34** ± 0.02	1.09 ± 0.12	0.56 ± 0.03
	EM adaptive SI (Federated)	**0.34** ± 0.02	**1.07** ± 0.52	**0.52** ± 0.05
	FedAvg with NNs (Federated)	0.42 ± 0.03	1.03 ± 0.44	0.61 ± 0.05
50	EM federated SI (Federated)	0.33 ± 0.02	1.09 ± 0.12	0.55 ± 0.03
	EM adaptive SI (Federated)	**0.33** ± 0.04	**1.01** ± 0.22	**0.52** ± 0.05
	FedAvg with NNs (Federated)	0.39 ± 0.04	1.04 ± 0.45	0.59 ± 0.05
100	EM federated SI (Federated)	0.33 ± 0.03	1.09 ± 0.12	0.55 ± 0.03
	EM adaptive SI (Federated)	**0.32** ± 0.02	**1.03** ± 0.10	**0.52** ± 0.07
	FedAvg with NNs (Federated)	0.38 ± 0.05	1.03 ± 0.45	0.58 ± 0.05
200	EM federated SI (Federated)	0.33 ± 0.03	1.09 ± 0.12	0.55 ± 0.03
	EM adaptive SI (Federated)	**0.32** ± 0.01	**1.02** ± 0.12	**0.51** ± 0.01

$B(\sum_{t=1}^{T} N^{(t)})(M+1)$ bits, where $N^{(t)}$ denotes the number of training samples corresponding with the tth network. Unlike the proposed methods, the communication costs in centralized methods increase linearly with the number of training data, which can be expensive in large-scale networks.

The proposed localization techniques only transfer TK summary statistics of the data from the local networks to the central server. In contrast, existing techniques would require to transfer all training data $\{(\boldsymbol{y}_n^{(t)}, d_n^{(t)})\}_{n=1}^{N^{(t)}}$, which can expose sensitive localization information. In addition, the proposed methods enable each network to learn a local generative model that remains exclusively within the network. Therefore, the presented approach improves privacy and provides better adaptability to heterogeneous environments.

IV. Numerical Results

This section first describes the datasets used for the experimentation and the experimental setup, and then compares the performance of the proposed method with respect to that of existing techniques.

Three real-world datasets are selected for numerical experimentation [14]: 'House', 'Office', and 'Industrial' that are composed by $15,810$, $14,694$, and $15,624$ samples, respectively. The measurements were made from 8 anchors and are composed by RTT, RSS, and channel impulse response (CIR) data. Each dataset represents a different environment (network).

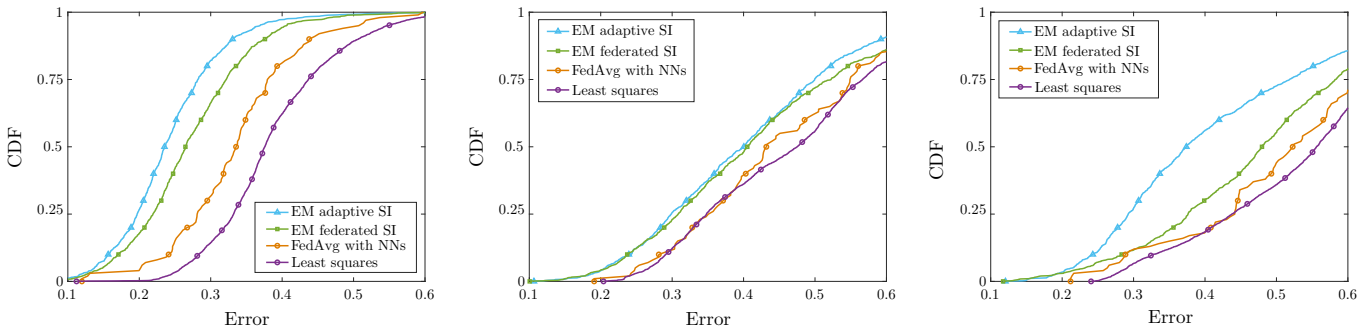
The performance of the proposed method is compared with the baseline of least squares and a federated learning based on the Federated Averaging (FedAvg) algorithm [15]. Least squares method uses distance estimates obtained directly from RTTs [7]. The federated learning baseline estimates distances using neural networks (NNs) trained under the FedAvg framework. Specifically, each network trains a NN consisting of two fully connected layers with 32 neurons each and ReLU activation functions. The models are trained locally for 100 epochs on normalized network training data. After local training, the models are transferred to the central server, where they are aggregated using FedAvg. This process is performed iteratively over several rounds of communication between the networks and the central server.

The proposed adaptive SI-based techniques obtain position estimates directly from measurements solving (7) using generative models obtained as described in Algorithm 1. The federated stage is initialized with all parameters set to zero. Each network obtains a Gaussian mixture model with $K = 5$ components, which are transferred to the central server during the federated stage. The parameters are then fine-tuned using 20 iterations in the adaptive stage. In addition, the numerical results of the proposed method and FedAvg-based method are obtained by preprocessing the data through "data sphering" and normalization, respectively, and reducing the dimensionality of the measurements using PCA, as described in [8].

We evaluate the performance of the localization techniques using the root mean square error (RMSE), given by $\text{RMSE} = \sqrt{\mathbb{E}\{||\widehat{\mathbf{p}} - \mathbf{p}||^2\}}$ where $\widehat{\mathbf{p}}$ denotes the position estimate. These numerical results are obtained computing the RMSE over 10 random instantiations for each number of communication rounds (iterations) between the networks and the central server and using 50 samples per anchor. The samples in each network are randomly splitted in 10% of the samples for test and the rest of the samples for training.

In order to more clearly assess the performance improvement due to the adaptive stage, we also compare the results with a version of the proposed methodology that does not perform the adaptive stage (EM federated SI). This version estimates positions using the global generative model obtained in the federated stage $\widehat{f}_{\mathbf{y},\mathrm{d}}^{(\mathrm{global})}(\boldsymbol{y}, d)$.

Table I shows the RMSE of the proposed method and existing techniques using the 3 real-world datasets. This table



(a) House dataset (b) Office dataset (c) Industrial dataset

Fig. 3: CDFs of prediction errors provide detailed performance comparison between the proposed method, a FedAvg-based method, and least squares using 3 real-world datasets.

shows that the proposed EM adaptive SI method consistently achieves the highest accuracy across all datasets and for different numbers of communication rounds. Table I shows that federated methods achieve accuracy comparable to that of centralized approaches. However, existing techniques such as FedAvg-based method require significantly more communication rounds than the proposed methods to reach levels of accuracy near those of the centralized implementation. In addition, Table I shows that the proposed adaptive approach provides further performance improvements across all the networks by enabling local models to adapt to the specific characteristics of each environment.

Figure 3 provides more detailed comparisons of the proposed methods, FedAvg method, and least squares. This figure shows the empirical cumulative distribution functions (CDFs) of prediction errors using the 3 real-world datasets. Figure 3 shows that low errors occur with high probability for the proposed method. In 'House' dataset, the prediction error of EM adaptive SI method is less than 0.3 meters with probability 0.8 while the FedAvg-based method reaches errors that are approximately 33% higher at the same probability level.

The proposed adaptive SI-based method leverages information from multiple localization networks and significantly reduces the privacy risks. Experiments confirm that the proposed method better captures network-specific characteristics and adapts to heterogeneous environments than existing methods.

V. Conclusion

The paper proposes an adaptive methodology based on SI learning that enhances wireless localization in multiple environments. The proposed methodology effectively leverages information from multiple networks by learning a global model from local training data and reduces privacy risks by transferring only summary statistics of the data. In addition, the proposed methods adapt to network-specific characteristics by fine-tuning the global model and further reduces privacy risks by maintaining a specific local model for each network. The numerical results assess the performance improvement of the proposed localization techniques using real-world datasets. These results also show that the proposed method achieves accurate results using a reduced number of communication rounds. Overall, the proposed adaptive approach can be especially effective for handling the scalability and privacy demands of next-generation wireless localization systems.

References

[1] M. Z. Win, A. Conti, S. Mazuelas, Y. Shen, W. M. Gifford, D. Dardari, and M. Chiani, "Network localization and navigation via cooperation," *IEEE Commun. Mag.*, vol. 49, no. 5, pp. 56–62, May 2011.

[2] M. Z. Win, Y. Shen, and W. Dai, "A theoretical foundation of network localization and navigation," *Proc. IEEE*, vol. 106, no. 7, pp. 1136–1165, Jul. 2018, special issue on *Foundations and Trends in Localization Technologies*.

[3] D. Wu, D. Chatzigeorgiou, K. Youcef-Toumi, and R. Ben-Mansour, "Node localization in robotic sensor networks for pipeline inspection," *IEEE Trans. Ind. Informat.*, vol. 12, no. 2, pp. 809–819, 2015.

[4] A. Conti, S. Mazuelas, S. Bartoletti, W. C. Lindsey, and M. Z. Win, "Soft information for localization-of-things," *Proc. IEEE*, vol. 107, no. 11, pp. 2240–2264, Nov. 2019.

[5] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for smart cities," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 22–32, Feb. 2014.

[6] H. Wymeersch, S. Maranò, W. M. Gifford, and M. Z. Win, "A machine learning approach to ranging error mitigation for UWB localization," *IEEE Trans. Commun.*, vol. 60, no. 6, pp. 1719–1728, Jun. 2012.

[7] S. Maranò, W. M. Gifford, H. Wymeersch, and M. Z. Win, "NLOS identification and mitigation for localization based on UWB experimental data," *IEEE J. Sel. Areas Commun.*, vol. 28, no. 7, pp. 1026–1035, Sep. 2010.

[8] S. Mazuelas, A. Conti, J. C. Allen, and M. Z. Win, "Soft range information for network localization," *IEEE Trans. Signal Process.*, vol. 66, no. 12, pp. 3155–3168, Jun. 2018.

[9] F. Morselli, S. M. Razavi, M. Z. Win, and A. Conti, "Soft information-based localization for 5G networks and beyond," *IEEE Wireless Commun.*, vol. 22, no. 12, pp. 9923–9938, 2023.

[10] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero, R. L. Moses, and N. S. Correal, "Locating the nodes: Cooperative localization in wireless sensor networks," *IEEE Signal Process. Mag.*, vol. 22, no. 4, pp. 54–69, Jul. 2005.

[11] J. Hightower and G. Borriello, "Location systems for ubiquitous computing," *Computer*, vol. 34, no. 8, pp. 57–66, Aug. 2001.

[12] R. A. Rednerf and H. F. Walker, "Mixture densities, maximum likelihood and the EM algorithm," *SIAM Review*, vol. 26, no. 2, pp. 195–239, Apr. 1984.

[13] Y. Wu, S. Zhang, W. Yu, Y. Liu, Q. Gu, D. Zhou, H. Chen, and W. Cheng, "Personalized federated learning under mixture of distributions," in *Proc. Int. Conf. Mach. Learning*, 2023, pp. 37 860–37 879.

[14] K. Bregar and M. Mohorčič, "Improving indoor localization using convolutional neural networks on computationally restricted devices," *IEEE Access*, vol. 6, pp. 17 429–17 441, 2018.

[15] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Art. Int. Stat.*, 2017, pp. 1273–1282.